Efficient Gradient Methods for Distributed Saddle Problems

Ruichen Luo
IST Austria
Klosterneuburg, Austria
rluo@ist.ac.at

Anton Rodomanov
CISPA Helmholtz Center
Saarbrücken, Germany
anton.rodomanov@cispa.de

Sebastian U. Stich
CISPA Helmholtz Center
Saarbrücken, Germany
stich@cispa.de

Abstract

The distributed setting for Saddle Problems (SPs) has recently emerged as a framework for modern applications in machine learning and multiagent systems. Despite its relevance, the theoretical foundations of this setting have not yet been thoroughly established. In this paper, we advance this research direction by formalizing the distributed setup for SPs and providing rigorous definitions of communication and oracle costs. Further, we prove lower bounds for any distributed gradient-span algorithm, which reveals the gap from existing methods and this theoretical limit. To this end, we provide a Decoupled Method built upon a novel multi-stage reduction that reduces the SP into a sequence of decoupled minimization tasks of residual norms. Our algorithm matches the communication lower bound, thus settling the communication complexity within the gradient-span algorithms. Moreover, it yields the first strict improvement over the long-standing oracle cost of the Extragradient method for general SPs. Finally, we study the extension of distributed SP into Variational Inequality Problem (VIP), which generalizes two-player zero-sum games to multiplayer general-sum games. We show that our Decoupled Method achieves a new state-of-the-art communication complexity for this broader class.

1 Introduction

Motivation. Saddle problems (SPs) and their generalizations, variational inequality problems (VIPs), are of fundamental importance to optimization and game theory. These problems have a wide array of modern applications, including the training of GANs [Goodfellow et al., 2014], robust optimization [Ben-Tal and Nemirovski, 2002], and equilibrium computation in game theory and multiagent systems [von Neumann and Morgenstern, 1947, Rosen, 1965, Hu et al., 1998].

The growing scale of modern problems—driven by applications in machine learning, complex game dynamics, and multiagent protocols—renders reliance on a single central processor increasingly impractical. Beyond scalability, a more crucial factor is that many applications are inherently *distributed*: agents are often geographically dispersed, driven by their own individual interests, and bound by privacy constraints that prohibit the sharing of raw data or utilities. Consequently, distributed computation has become an essential regime for these problems. This perspective underlies a growing body of work in large-scale learning, game-theoretic models, and multiagent systems [McMahan et al., 2017, Zhang et al., 2024, Conitzer and Sandholm, 2004, Nisan and Segal, 2006, Hart and Mansour, 2010, Yoon et al., 2025].

In this work, we consider a natural setup where the decision variables and oracles of the SPs or VIPs are partitioned among distributed agents. For instance, in a classic saddle problem $\min_{\mathbf{x}} \max_{\mathbf{y}} F(\mathbf{x}, \mathbf{y})$, we consider one agent controls the minimizing variable $\mathbf{x}$ while another controls the maximizing variable $\mathbf{y}$. This partition naturally models, for instance, the strategic autonomy of players in game theory, the interaction protocol in multiagent systems, and the physical

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

separation of the generator and discriminator in GANs [Conitzer and Sandholm, 2004, Goodfellow et al., 2014]. Since the decision variables are coupled within their utilities, these agents must coordinate to reach a mutual equilibrium. To do so, they form a communication network that allows them to exchange certain information, such as their current decision variables. Thus, this provides a natural distributed setup where the decision variables are separated among the different agents.

While distributed optimization is well-established for finite-sum minimization and federated learning [Schmidt et al., 2017, McMahan et al., 2017], the literature on SPs has primarily focused on extending these data-distributed paradigms [Deng and Mahdavi, 2021, Beznosikov et al., 2025]. In contrast, the study of distributed variables and oracles, which are essential to the multi-agent systems, remains a relatively new topic.

Although a few recent works have touched upon this direction [Zhang et al., 2024, Zindari et al., 2025, Yoon et al., 2025, Yoon and Loizou, 2025], they predominantly focus on algorithms tailored to specific, favorable scenarios. Consequently, a fundamental gap persists: the lack of a theoretical framework for the general distributed settings of SPs and VIPs. Existing discussions regarding performance often remain at a vague conceptual level, lacking a rigorous formalization of the distributed environment itself. Specifically, there are no standardized definitions for communication and oracle costs in this context. Without such a foundation, it can be difficult to determine the performance limits (lower bounds) or to formally compare the efficiency of different protocols.

To enable a rigorous analysis, it is essential to establish metrics that reflect the constraints of distributed multiagent systems, where network latency and bandwidth often dwarf local processing time. In this regime, the primary bottleneck is the communication cost (exchange rounds), while the computational cost (local gradient queries) is a secondary objective. Viewed through this lens, the Extragradient (EG) method [Tseng, 1995, Nemirovski, 2004] serves as the "gold standard" baseline, though the challenges associated with it differ by metric. Regarding computational complexity, consistently improving upon EG for general monotone problems has remained an elusive goal despite over two decades of research. Regarding communication complexity—a metric that has recently come into focus with the rise of distributed systems—EG similarly defines the current state-of-the-art. Surpassing this baseline in the general setting represents a new but critical open problem.

This leads to the following research questions:

- **Formalization and Limits:** How can we rigorously formalize the communication and oracle costs for distributed SPs?
- **Communication Efficiency:** Can we design an algorithm that surpasses the state-of-the-art communication bounds for distributed SPs and VIPs?
- **Oracle Efficiency:** Is it possible to consistently improve upon the long-standing oracle complexity of the EG method for general SPs?

Contributions. We answer the aforementioned questions in the affirmative, which advances the current theory of distributed SPs and VIPs.

- In Section 2, we formalize the distributed saddle-point problem, gradient-span algorithms, and their communication and oracle costs. We review EG and other methods, casting them as gradient-span algorithms to analyze their costs.
- In Section 3, by drawing connection to classic convex minimization, we establish the lower bounds for both communication and oracle costs for distributed gradient-span algorithms.
- In Section 4, we start with a template DM-SP algorithm with a simple, one-loop communication protocol, which improves the state-of-the-art communication cost and matches the communication lower bound we constructed earlier in Section 3.
- Then, continuing in Section 4, after making the novel multi-stage reduction, we equip the template method with a concrete implementation, thereby consistently improving the (long-standing) oracle cost of EG for general SPs.
- Finally, in Section 5, we extend the results to multi-agent settings. We propose DM-VIP and improve the state-of-the-art communication cost for the class of distributed VIPs.

Table 1: Summary of algorithms and complexity results for distributed SPs.

Method $\mathscr{M}$	**Communication Cost**	**Better Oracle?**[a]	**Multi-Agent**[b]
EG[c]	$\mathcal{O}\Big(\theta\frac{L_{xy}D_xD_y}{\epsilon}+\frac{L_xD_x^2+L_yD_y^2}{\epsilon}\Big)$	No	Yes
DGDA[c]	$\mathcal{O}\big(\log\frac{1}{\epsilon}\big)$ (only applied to weakly coupled problems)	No	Yes
Cat-EG[c]	$\mathcal{O}\Big(\Big(\frac{L_{\max}\hat{D}_x\hat{D}_y}{\epsilon}+\sqrt{\frac{L_x\hat{D}_x^2+L_y\hat{D}_y^2}{\epsilon}}\Big)\log^2\big(\frac{1}{\epsilon}\big)\Big)$	Maybe	No
Cat-Cat-DAGDA[c]	$\mathcal{O}\Big(\frac{L_{xy}\hat{D}_x\hat{D}_y}{\epsilon}\log^3\big(\frac{1}{\epsilon}\big)\Big)$	Maybe	No
Lower Bound (Thm. 1)	$\Omega\Big(\frac{L_{xy}D_xD_y}{\epsilon}\Big)$	–	–
DM-SP (Thm. 2)	$\mathcal{O}\Big(\theta\frac{L_{xy}D_xD_y}{\epsilon}\Big)$	Yes	Yes

[a] Indicates whether the method's theoretical oracle cost outperforms the Extragradient (EG) baseline. "Maybe" means better than EG under certain conditioning and worse under other conditioning.
[b] Indicates whether the method supports multi-agent extensions. [c] These methods are for the non-composite subclass $\mathscr{P}_{\mathrm{SP}}^{\circ}$.

Notations. Let $[n] \triangleq \{1,\dots,n\}$, for any positive integer n. For any finite-dimensional real vector space $\mathcal{E}$, we denote its Euclidean norm by $\|\cdot\|_{\mathcal{E}}$ and its dual norm by $\|\cdot\|_{\mathcal{E}^*}$. Specifically, we equip the space $\mathcal{E}_x = \mathbb{R}^{n_x}$ with the norm $\|\mathbf{x}\|_x = \langle \mathbf{P}_x\mathbf{x},\mathbf{x}\rangle^{1/2}$, where $\mathbf{P}_x\colon \mathcal{E}_x \to \mathcal{E}_x^*$ is a self-adjoint positive definite operator and the dual pairing $\langle \phi_x,\mathbf{x}\rangle$ denotes $\phi_x(\mathbf{x})$. We denote its corresponding dual norm by $\|\cdot\|_{x^*}$. We assume analogous geometries for $\mathcal{E}_y = \mathbb{R}^{n_y}$, $\mathcal{E}_i = \mathbb{R}^{n_i}$ ($i \in [K]$), and $\mathcal{E}_w = \mathbb{R}^{n_w}$, associated with their respective operators $\mathbf{P}_y$, $\mathbf{P}_i$, and $\mathbf{P}_w$. For a function $\psi\colon \mathcal{E} \to \mathbb{R}\cup\{+\infty\}$, let $\operatorname{dom}\psi$ denote its effective domain and $\partial\psi(\mathbf{z})$ its subdifferential at $\mathbf{z} \in \operatorname{dom}\psi$. Finally, for any set of vectors S, let $\operatorname{span} S$ denote its linear span.

2 Saddle problems with distributed oracles

In the context of saddle problems, we consider two separate finite-dimensional real vector spaces, $\mathcal{E}_x = (\mathbb{R}^{n_x},\|\cdot\|_x)$ and $\mathcal{E}_y = (\mathbb{R}^{n_y},\|\cdot\|_y)$. We are interested in solving composite Saddle Problems (SPs) of the following form:

$$\min_{\mathbf{x}\in\operatorname{dom}\psi_x}\max_{\mathbf{y}\in\operatorname{dom}\psi_y}\big[F(\mathbf{x},\mathbf{y}) \triangleq f(\mathbf{x},\mathbf{y})+\psi_x(\mathbf{x})-\psi_y(\mathbf{y})\big], \tag{1}$$

where $\psi_x\colon \mathcal{E}_x \to \mathbb{R}\cup\{+\infty\}$ and $\psi_y\colon \mathcal{E}_y \to \mathbb{R}\cup\{+\infty\}$ represent relatively simple local components (such as regularizers or indicator functions for constrained sets), and $f(\cdot,\cdot)$ is a real-valued coupling function defined on an open set containing the domain $Q \triangleq \operatorname{dom}\psi_x \times \operatorname{dom}\psi_y$. To simplify the notation, we denote the joint variable by $\mathbf{z} \triangleq (\mathbf{x},\mathbf{y}) \in Q$.

2.1 Problem class

Function family. We consider the function instances satisfying the following assumptions:

(A1) For any fixed $\mathbf{y} \in \operatorname{dom}\psi_y$, the function $f(\cdot,\mathbf{y})$ is convex; and for any fixed $\mathbf{x} \in \operatorname{dom}\psi_x$, the function $f(\mathbf{x},\cdot)$ is concave. The functions ψ_x and ψ_y are proper, closed, and convex.

(A2) Let $D_x, D_y > 0$ be distance parameters, and let $\mathbf{z}^0 = (\mathbf{x}^0,\mathbf{y}^0) \in Q$ be given initial points. Relative to this initialization, Problem (1) has at least one saddle point $(\mathbf{x}^*,\mathbf{y}^*) \in Q$ that lies within the bounded initial sets $\mathbf{x}^* \in \mathcal{B}_x$ and $\mathbf{y}^* \in \mathcal{B}_y$, where

$$\mathcal{B}_x \triangleq \big\{\mathbf{x}\in\mathcal{E}_x \;\big|\; \|\mathbf{x}^0-\mathbf{x}\|_x \le D_x\big\} \qquad \text{and} \qquad \mathcal{B}_y \triangleq \big\{\mathbf{y}\in\mathcal{E}_y \;\big|\; \|\mathbf{y}^0-\mathbf{y}\|_y \le D_y\big\}.$$

(A3) The function $f(\cdot,\cdot)$ is continuously differentiable over Q. Moreover, its gradients are Lipschitz continuous. That is, with Lipschitz parameters $L_x, L_{xy}, L_y > 0$, for all $\mathbf{x},\mathbf{x}' \in \operatorname{dom}\psi_x$ and $\mathbf{y},\mathbf{y}' \in \operatorname{dom}\psi_y$, we have:

$$\|\nabla_x f(\mathbf{x}',\mathbf{y}')-\nabla_x f(\mathbf{x},\mathbf{y})\|_{x^*} \le L_x\|\mathbf{x}'-\mathbf{x}\|_x + L_{xy}\|\mathbf{y}'-\mathbf{y}\|_y,$$
$$\|\nabla_y f(\mathbf{x}',\mathbf{y}')-\nabla_y f(\mathbf{x},\mathbf{y})\|_{y^*} \le L_{xy}\|\mathbf{x}'-\mathbf{x}\|_x + L_y\|\mathbf{y}'-\mathbf{y}\|_y.$$

With these assumptions in place, we can formally define the function family of interest. Let $\mathscr{F}$ denote the *function family* consisting of all instances $\mathcal{F} = (f, \psi_x, \psi_y, \mathbf{z}^0)$ that satisfy (A1) to (A3) for a fixed set of parameters $(L_x, L_{xy}, L_y, D_x, D_y)$. To facilitate our discussion, we refer to the terms $L_x D_x^2$ and $L_y D_y^2$ as the *diagonal conditioning*, and the term $L_{xy} D_x D_y$ as the *cross-coupled conditioning*.

Distributed oracle model. We consider a distributed architecture where the variables $\mathbf{x}$ and $\mathbf{y}$ are maintained and updated by two distinct entities: Agent x and Agent y, respectively. First-order information is acquired through *(deterministic) partial gradient oracles*, denoted abstractly by $\mathscr{O}_x$ and $\mathscr{O}_y$. Specifically, for a given function instance $\mathcal{F}$ from the function family $\mathscr{F}$ with coupling function f, and for any input point $\mathbf{z} \in Q$:

- Agent x queries the oracle $\mathscr{O}_x$, which returns $\mathscr{O}_x^f(\mathbf{z}) = \nabla_x f(\mathbf{z})$.
- Agent y queries the oracle $\mathscr{O}_y$, which returns $\mathscr{O}_y^f(\mathbf{z}) = -\nabla_y f(\mathbf{z})$.

These oracles are strictly decoupled. They can be queried independently by their respective agents, differing in both query frequency and the input points evaluated. Most importantly, the agents are fully distributed, meaning an agent cannot query the counterpart's oracle. Moreover, each agent can evaluate the proximal mapping of its own component, ψ_x for Agent x and ψ_y for Agent y, and the domain Q is known to both agents.

To provide a concrete example of this abstract setting, consider the objective $f(\mathbf{z}) = g(\mathbf{z}) + f_x(\mathbf{x}) - f_y(\mathbf{y})$, where $g(\mathbf{z})$ is a coupled global utility, while $f_x(\mathbf{x})$ and $f_y(\mathbf{y})$ are private utilities accessible only to Agents x and y, respectively. In this scenario, the partial gradient oracles take the form:

$$\mathscr{O}_x^f(\mathbf{z}) = \nabla_x g(\mathbf{z}) + \nabla f_x(\mathbf{x}) \qquad \mathscr{O}_y^f(\mathbf{z}) = -\nabla_y g(\mathbf{z}) + \nabla f_y(\mathbf{y}),$$

for all $\mathbf{z} \in Q$. Due to the distributed setting, Agent x is entirely blind to the private utility f_y and can only execute $\mathscr{O}_x$, and vice versa.

Accuracy measure. To evaluate the quality of a candidate solution $(\bar{\mathbf{x}}, \bar{\mathbf{y}}) \in Q$, we rely on the restricted duality gap. Over the bounded domain $\mathcal{B} \triangleq \mathcal{B}_x \times \mathcal{B}_y$, the duality gap is defined as:

$$\Delta(\bar{\mathbf{x}}, \bar{\mathbf{y}}) \triangleq \max_{\mathbf{z} \in \mathcal{B} \cap Q} \big[F(\bar{\mathbf{x}}, \mathbf{y}) - F(\mathbf{x}, \bar{\mathbf{y}})\big].$$

We say that a pair $(\bar{\mathbf{x}}, \bar{\mathbf{y}}) \in Q$ is an *ϵ-saddle point* of Problem (1) if $\Delta(\bar{\mathbf{x}}, \bar{\mathbf{y}}) \leq \epsilon$. Our goal is to design an algorithm capable of finding such an ϵ-saddle point for any given $\epsilon > 0$.

We remark that for the classic problem of constrained optimization with bounded domains, one can enclose the constrained sets in the balls $\mathcal{B}_x$ and $\mathcal{B}_y$ with sufficiently large radius (e.g., the diameter of the constrained sets), then the restricted saddle problem in form (1) coincides with the original one.

Problem class. Finally, we formally define the overall *problem class*, denoted by $\mathscr{P}_{\text{SP}}(\mathscr{F}, \mathscr{O}_x, \mathscr{O}_y, \epsilon)$, or for short $\mathscr{P}_{\text{SP}}$. A specific problem instance $P \in \mathscr{P}_{\text{SP}}$ is constructed by drawing a function instance $\mathcal{F}$ from the function family $\mathscr{F}$, equipping it with the specific distributed oracles $(\mathscr{O}_x^f, \mathscr{O}_y^f)$, and specifying a target accuracy $\epsilon > 0$. Solving the instance P requires an algorithm to output an ϵ-saddle point of $\mathcal{F}$ utilizing the distributed oracles.

2.2 Distributed gradient-span algorithms

Standard gradient-based methods are commonly analyzed under the assumption that the algorithm generates iterates within the linear span of all historically evaluated gradients [Nesterov, 2004]. However, classic span conditions implicitly assume centralized, real-time access to the full problem state. This falls short for the problem class $\mathscr{P}_{\text{SP}}$. Because Agent x and Agent y are strictly decoupled and communication between them is typically the primary system bottleneck, they cannot continuously synchronize their variables.

Instead, practical algorithms must proceed in discrete communication rounds. Within a given round, an agent performs multiple local computational steps in isolation. For instance, Agent x generates new iterates for $\mathbf{x}$ by taking linear combinations of its previous iterates and its locally accumulated partial gradients $(\nabla_x f)$, followed by a proximal step of ψ_x. Crucially, to evaluate the partial gradient of the coupling function, Agent x also needs to know the value of $\mathbf{y}$. Cut off from real-time updates,

it must rely on a local, delayed approximation (denoted $\hat{\mathbf{y}}$) constructed entirely from the historical messages Agent y sent before the current round started. Agent y operates symmetrically.

To rigorously analyze algorithms under these strict communication constraints, we formalize the class of *distributed gradient-span algorithms*. This mathematical framework explicitly separates communication rounds from local oracle queries, enforcing that each agent constructs its updates using only locally accessible history and delayed remote messages.

Algorithm trajectories and histories. Suppose an algorithm $\mathscr{M}$ proceeds in T rounds. In each round $t \in \{0, \dots, T-1\}$, Agent x and Agent y generate local trajectories of lengths τ_x^t and τ_y^t, denoted respectively by:

$$\hat{Z}_x^t = \left\{\mathbf{z}_x^{t,l} = (\mathbf{x}^{t,l}, \hat{\mathbf{y}}^{t,l})\right\}_{l=0}^{\tau_x^t - 1} \quad \text{and} \quad \hat{Z}_y^t = \left\{\mathbf{z}_y^{t,l} = (\hat{\mathbf{x}}^{t,l}, \mathbf{y}^{t,l})\right\}_{l=0}^{\tau_y^t - 1}.$$

To track the information available to the agents, let $Z_x^{t-1} = \bigcup_{i=0}^{t-1} \hat{Z}_x^i$ and $Z_y^{t-1} = \bigcup_{i=0}^{t-1} \hat{Z}_y^i$ denote the accumulated histories from all prior rounds (where $Z_x^{-1} = Z_y^{-1} = \varnothing$). Within round t, the histories up to local step l are denoted by $Z_x^{t,l} = Z_x^{t-1} \cup \{\mathbf{z}_x^{t,i}\}_{i=0}^{l-1}$ and $Z_y^{t,l} = Z_y^{t-1} \cup \{\mathbf{z}_y^{t,i}\}_{i=0}^{l-1}$. Finally, the accumulated histories up to and including round t are given by Z_x^t and Z_y^t, and the complete trajectories across all T rounds are denoted by $\bar{Z}_x = Z_x^{T-1}$ and $\bar{Z}_y = Z_y^{T-1}$. For any finite set $Z \subseteq Q$, we denote

$$\mathcal{S}_x(Z) \triangleq \operatorname{span}\left\{\mathbf{x} - \mathbf{x}^0,\ \mathbf{P}_x^{-1}\nabla_x f(\mathbf{x}, \mathbf{y}) \;\middle|\; (\mathbf{x}, \mathbf{y}) \in Z\right\},$$
$$\mathcal{S}_y(Z) \triangleq \operatorname{span}\left\{\mathbf{y} - \mathbf{y}^0,\ \mathbf{P}_y^{-1}\nabla_y f(\mathbf{x}, \mathbf{y}) \;\middle|\; (\mathbf{x}, \mathbf{y}) \in Z\right\}.$$

Definition 1. An algorithm $\mathscr{M}$ is called a *distributed gradient-span algorithm* for problem class $\mathscr{P}_{\text{SP}}$ if, when applied to any instance $P \in \mathscr{P}_{\text{SP}}$, its trajectories satisfy the following conditions:

1. For all rounds $t \in \{0, \dots, T-1\}$ and local steps $l \in \{0, \dots, \tau_x^t - 1\}$, there exist $\beta \geq 0$, $\mathbf{u}_x \in \mathbf{x}^0 + \mathcal{S}_x(Z_x^{t,l})$, and $\mathbf{u}_y \in \mathbf{y}^0 + \mathcal{S}_y(Z_y^{t-1})$ such that the points in the x-trajectory satisfy

$$\mathbf{x}^{t,l} = \operatorname*{arg\,min}_{\mathbf{x} \in \operatorname{dom}\psi_x} \left[\beta\psi_x(\mathbf{x}) + \frac{1}{2}\|\mathbf{x} - \mathbf{u}_x\|_x^2\right] \quad \text{and} \quad \hat{\mathbf{y}}^{t,l} = \operatorname*{arg\,min}_{\mathbf{y} \in \operatorname{dom}\psi_y} \|\mathbf{y} - \mathbf{u}_y\|_y;$$

   and symmetrically, for all local steps $l \in \{0, \dots, \tau_y^t - 1\}$, there exist $\beta \geq 0$, $\mathbf{u}_y \in \mathbf{y}^0 + \mathcal{S}_y(Z_y^{t,l})$, and $\mathbf{u}_x \in \mathbf{x}^0 + \mathcal{S}_x(Z_x^{t-1})$ such that the points in the y-trajectory satisfy

$$\mathbf{y}^{t,l} = \operatorname*{arg\,min}_{\mathbf{y} \in \operatorname{dom}\psi_y} \left[\beta\psi_y(\mathbf{y}) + \frac{1}{2}\|\mathbf{y} - \mathbf{u}_y\|_y^2\right] \quad \text{and} \quad \hat{\mathbf{x}}^{t,l} = \operatorname*{arg\,min}_{\mathbf{x} \in \operatorname{dom}\psi_x} \|\mathbf{x} - \mathbf{u}_x\|_x.$$

2. After each round $t \in \{0, \dots, T-1\}$, $\mathscr{M}$ returns a solution $\bar{\mathbf{z}}^{t+1} = (\bar{\mathbf{x}}^{t+1}, \bar{\mathbf{y}}^{t+1})$ such that

$$\bar{\mathbf{x}}^{t+1} \in \operatorname{dom}\psi_x \cap \left(\mathbf{x}^0 + \mathcal{S}_x(Z_x^t)\right) \quad \text{and} \quad \bar{\mathbf{y}}^{t+1} \in \operatorname{dom}\psi_y \cap \left(\mathbf{y}^0 + \mathcal{S}_y(Z_y^t)\right).$$

When $\psi_x = \psi_y = 0$, Definition 1 reduces to the standard condition that all points lie in the span of the evaluated partial gradients (see Section B).

We remark that the span condition imposed by Definition 1 is not absolutely necessary for defining distributed algorithms or proving lower bounds. This might be avoided by a more sophisticated reasoning of information-based complexity and resisting oracles. However, we adopt the gradient-span framework in this paper since it holds naturally for the majority of practical distributed algorithms.

Communication and oracle costs. For a given instance $P \in \mathscr{P}_{\text{SP}}$, we define the *communication cost* required by a distributed gradient-span algorithm $\mathscr{M}$ on P, denoted by $T_P^{\mathscr{M}}$, as the smallest integer $k \in \{1, \cdots, T\}$ such that the generated solution $\bar{\mathbf{z}}^k$ satisfies the target accuracy. Similarly, the total number of oracle queries with respect to x and y for instance P up to this point is given by the sizes of the accumulated histories, denoted respectively by $N_{x,P}^{\mathscr{M}} = \left|Z_x^{T_P^{\mathscr{M}}-1}\right|$ and $N_{y,P}^{\mathscr{M}} = \left|Z_y^{T_P^{\mathscr{M}}-1}\right|$.

Let c_x and c_y denote the respective computational costs of evaluating a single partial gradient $\nabla_x f$ and $\nabla_y f$. The *(worst-case) communication cost* and *oracle cost* of algorithm $\mathscr{M}$ over the entire problem class $\mathscr{P}_{\text{SP}}$ are defined by taking the supremum over all instances:

$$T_{\mathscr{P}_{\text{SP}}}^{\mathscr{M}} = \sup_{P \in \mathscr{P}_{\text{SP}}} T_P^{\mathscr{M}} \quad \text{and} \quad N_{\mathscr{P}_{\text{SP}}}^{\mathscr{M}} = \sup_{P \in \mathscr{P}_{\text{SP}}} \left(c_x N_{x,P}^{\mathscr{M}} + c_y N_{y,P}^{\mathscr{M}}\right).$$

Throughout the distributed environment considered in this paper, *we treat the communication cost as the primary performance metric*, as network communication typically forms the main bottleneck. The oracle cost serves as the secondary metric to measure the local computational effort.[1]

2.3 Existing algorithms from literature

In this section, we review existing algorithms for solving SPs and analyze their communication and oracle costs within the distributed gradient-span algorithm framework. To keep the presentation concise, we summarize the methods and their limitations below, and defer their detailed algorithmic formulations, trajectories, and complexity propositions to Section A.

Extragradient (EG). The classic EG method [Nemirovski, 2004, Juditsky et al., 2011] naturally fits our framework. Its distributed execution requires two communication rounds per iteration to evaluate coupled partial gradients at both the current and extrapolated points. It provides a robust and natural baseline for communication and oracle costs.

Decoupled GDA (DGDA). The DGDA method [Zindari et al., 2025] attempts to reduce communication overhead by freezing the remote variable and taking multiple local gradient steps. While it achieves a fast logarithmic communication cost, its applications are highly restrictive: it only converges for weakly coupled strongly convex-strongly concave instances. For general problem class $\mathscr{P}_{\text{SP}}$, the delayed remote variables cause the local updates to drift, leading the method to diverge.

Catalyst acceleration. Using a Catalyst wrapper around EG (Cat-EG) [Lin et al., 2020, Yang et al., 2020, Lan and Li, 2026] accelerates the algorithm's dependence on the diagonal conditioning. However, this comes with **five significant caveats**: (i) it requires a complicated, multi-loop communication protocol and careful parameter tuning; (ii) it is highly sensitive to the inexactness of the diameter estimates $\hat{D}_x$ and $\hat{D}_y$; (iii) it introduces multiplicative logarithmic factors in the complexity; (iv) under certain conditioning, its theoretical complexity can be strictly worse than the unaccelerated EG baseline; and (v) it does not support extensions to multi-agent scenarios (cf. Section 5).

Four-loop method. The Cat-Cat-DAGDA method [Wang and Li, 2020] applies double Catalyst wrappers around a decoupled accelerated GDA to further accelerate the cross-coupling term. Despite this theoretical improvement, it shares all five caveats of Cat-EG, introduces even more complicated nested loops into the communication protocol, and adds further logarithmic factors. Consequently, it serves primarily as a theoretical benchmark rather than a practical method in our setting.

Other distributed stochastic gradient methods. Some recent papers [Zhang et al., 2024, Yoon et al., 2025, Yoon and Loizou, 2025] consider distributed SPs with stochastic gradient oracles. They propose different communication-efficient approaches; however, when applied to standard deterministic oracles considered in this paper, these methods fail to outperform EG.

Consequently, as summarized in Table 1, the classic EG method remains a formidable baseline for $\mathscr{P}_{\text{SP}}$, and *improving its communication and oracle complexity remains a significant challenge*.

3 Lower complexity bounds for SPs

In this section, we present lower complexity bounds for all algorithms in the distributed gradient-span family (cf. Section 2.2). We restrict our attention to this algorithm family because it is a key framework that captures a wide range of practical algorithms, as illustrated in Section 2.3.

In the standard non-distributed setting with full-gradient oracles, the lower complexity bounds for SPs have been well-studied in the literature [Nemirovsky, 1992, Zhang et al., 2022]. These bounds are typically expressed in terms of the number of full-gradient oracle queries and does not directly translate to the distributed setting. To this end, we carefully construct the worst-case instances and analyze the trajectories of the algorithms to establish the desired lower bounds.

Theorem 1. *Let the dimensions of $\mathcal{E}_x$ and $\mathcal{E}_y$ be sufficiently large such that $n_x \geq \frac{L_{xy}D_xD_y}{2\epsilon} + \sqrt{\frac{3L_xD_x^2}{16\epsilon}} + 1$ and $n_y \geq \frac{L_{xy}D_xD_y}{2\epsilon} + \sqrt{\frac{3L_yD_y^2}{16\epsilon}} + 1$. For any distributed gradient-span algorithm $\mathscr{M}$*

[1] Our work is complementary to other communication-saving methods, such as message quantization [Nguyen et al., 2025].

for $\mathscr{P}_{\text{SP}}$, we have:

$$T^{\mathscr{M}}_{\mathscr{P}_{\text{SP}}} \geq \frac{L_{xy} D_x D_y}{2\epsilon} - 2,$$

$$N^{\mathscr{M}}_{\mathscr{P}_{\text{SP}}} \geq \frac{c_x + c_y}{12} \frac{L_{xy} D_x D_y}{\epsilon} + \frac{c_x}{3} \sqrt{\frac{3 L_x D_x^2}{64\epsilon}} + \frac{c_y}{3} \sqrt{\frac{3 L_y D_y^2}{64\epsilon}} - \frac{2c_x + 2c_y}{3}.$$

Main proof idea. We now provide a simplified intuition for our lower bounds by focusing on standard Euclidean geometries with $n_x = n_y = n$, though our formal analysis is more general.

To establish the communication lower bound, we construct a difficult bilinear instance $F_{xy}(\mathbf{x}, \mathbf{y}) = \langle \mathbf{A}\mathbf{x} - \mathbf{b}, \mathbf{y} \rangle$. The upper bidiagonal matrix $\mathbf{A}$ and vector $\mathbf{b}$ are constructed as follows:

$$\mathbf{A} \propto \begin{pmatrix} 1 & -1 & 0 & \dots & 0 \\ 0 & 1 & -1 & \ddots & \vdots \\ 0 & 0 & 1 & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & -1 \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{b} \propto (1, 1, \dots, 1)^\top, \quad \text{so that} \quad \mathbf{A}^\top \mathbf{b} \propto (1, 0, \dots, 0)^\top.$$

For this instance, the Agents $\mathbf{x}$ and $\mathbf{y}$ have to communicate back and forth to sequentially pass the new non-zero coordinates between each other, which leads to the communication lower bound. For the oracle lower bound, we additionally leverage quadratic instances for $\mathbf{x}$ and $\mathbf{y}$ respectively, and take the maximum cost over these three instances. We defer the detailed proofs to Section B. □

Remark 1. Theorem 1 presents a communication lower bound depending only on the cross-coupled conditioning. In contrast, none of the existing algorithms reviewed in Section 2.3 match this lower bound. Specifically, the communication cost of the EG baseline is suboptimal due to its dependence on the diagonal conditioning. While advanced frameworks like Cat-Cat-DAGDA successfully isolate the communication cost from the diagonal conditioning, they suffer from highly complicated nested-loop designs and introduce poly-logarithmic overheads. This gap from the theoretical limit highlights the insufficiency of existing approaches and motivates our subsequent algorithmic developments.

4 Decoupled method for SPs

When designing a communication-efficient method, the primary challenge is enabling distributed agents to compute local solutions independently despite the presence of cross-coupled functions. To address this, we propose a clean algorithmic template (or communication protocol) *that reduces an SP into a sequence of coordinate-wise computational tasks*. We highlight the key results and insights below, deferring the detailed derivation to Section C.

Assembled norm. Given parameters $\alpha_x, \alpha_y > 0$ (to be specified later), we equip the joint space $\mathcal{E} = \mathcal{E}_x \times \mathcal{E}_y$ with the assembled norm:

$$\|\mathbf{z}\|_{\mathcal{E}} = \langle \mathbf{P}\mathbf{z}, \mathbf{z} \rangle^{\frac{1}{2}} = \sqrt{\alpha_x \|\mathbf{x}\|_x^2 + \alpha_y \|\mathbf{y}\|_y^2} \quad \text{for all } \mathbf{z} \in \mathcal{E}, \tag{2}$$

which corresponds to the block diagonal linear operator $\mathbf{P} = \alpha_x \mathbf{P}_x \oplus \alpha_y \mathbf{P}_y$.

Template DM-SP. Algorithm 1 outlines the Decoupled Method for Saddle Problems (DM-SP), which adapts the abstract framework of the Reduced-Operator Method [Nesterov, 2023] for distributed environments. The algorithm maintains a sequence of anchor points $\mathbf{v}^t$ and proceeds iteratively.

First, the agents decouple the joint problem by fixing the remote variable at the current anchor $\mathbf{v}^t$. This allows Agent x and Agent y to independently and concurrently solve their respective regularized local subproblems up to target accuracies δ_x^{t+1} and δ_y^{t+1} (Lines 4 and 5). Specifically, Agent x aims to approximately compute $\arg\min_{\mathbf{x} \in \operatorname{dom} \psi_x} [f(\mathbf{x}, \mathbf{v}_y^t) + \frac{\alpha_x \lambda_{t+1}}{2} \|\mathbf{x} - \mathbf{v}_x^t\|_x^2 + \psi_x(\mathbf{x})]$ by finding a point $\mathbf{x}^{t+1}$ whose regularized subgradient norm satisfies the exact mathematical bound specified in Line 4. Agent y symmetrically performs an approximate minimization for its corresponding objective $-f(\mathbf{v}_x^t, \mathbf{y}) + \frac{\alpha_y \lambda_{t+1}}{2} \|\mathbf{y} - \mathbf{v}_y^t\|_y^2 + \psi_y(\mathbf{y})$.

Following this local computation phase, the agents perform exactly two communication rounds to complete the iteration. In the first round (Line 6), the agents exchange their locally computed approximate solutions to assemble the joint intermediate point $\mathbf{z}^{t+1} = (\mathbf{x}^{t+1}, \mathbf{y}^{t+1})$. In the second round (Line 7), they use this assembled point to evaluate their local partial gradients, which they then exchange to form the full joint operator $V_\psi(\mathbf{z}^{t+1})$.

Finally, using this assembled operator, the agents compute a closed-form step size a_{t+1}, update the running ergodic average $\bar{\mathbf{z}}^{t+1}$, and perform an extragradient-like step to generate the next anchor $\mathbf{v}^{t+1}$ (Lines 8 and 9). By structuring the method this way, DM-SP cleanly reduces the coupled SP into isolated coordinate-wise tasks with minimal communication overhead.

Algorithm 1 DM-SP$(f, (\psi_x, \psi_y), \mathbf{z}^0, (\lambda_t)_{t\geq 1}, (\alpha_x, \alpha_y))$

1: $\mathbf{v}^0 = (\mathbf{v}_x^0, \mathbf{v}_y^0) = \mathbf{z}^0$.
2: **for** $t = 0, 1, \ldots, T-1$ **do**
3: Let $\delta_x^{t+1} = \frac{\alpha_x \lambda_{t+1}}{2}$ and $\delta_y^{t+1} = \frac{\alpha_y \lambda_{t+1}}{2}$.
4: Agent x finds $\mathbf{x}^{t+1}$ and $\psi_x'(\mathbf{x}^{t+1}) \in \partial\psi_x(\mathbf{x}^{t+1})$ such that

$$\|\nabla_x f(\mathbf{x}^{t+1}, \mathbf{v}_y^t) + \alpha_x \lambda_{t+1} \mathbf{P}_x(\mathbf{x}^{t+1} - \mathbf{v}_x^t) + \psi_x'(\mathbf{x}^{t+1})\|_{x^*} \leq \delta_x^{t+1} \|\mathbf{x}^{t+1} - \mathbf{v}_x^t\|_x.$$

5: Agent y finds $\mathbf{y}^{t+1}$ and $\psi_y'(\mathbf{y}^{t+1}) \in \partial\psi_y(\mathbf{y}^{t+1})$ such that

$$\|-\nabla_y f(\mathbf{v}_x^t, \mathbf{y}^{t+1}) + \alpha_y \lambda_{t+1} \mathbf{P}_y(\mathbf{y}^{t+1} - \mathbf{v}_y^t) + \psi_y'(\mathbf{y}^{t+1})\|_{y^*} \leq \delta_y^{t+1} \|\mathbf{y}^{t+1} - \mathbf{v}_y^t\|_y.$$

6: Exchange $\mathbf{x}^{t+1}$ and $\mathbf{y}^{t+1}$ to assemble $\mathbf{z}^{t+1} = (\mathbf{x}^{t+1}, \mathbf{y}^{t+1})$.
7: Calculate corresponding coordinates of $V_\psi(\mathbf{z}^{t+1})$, then exchange to assemble:

$$V_\psi(\mathbf{z}^{t+1}) = \big(\nabla_x f(\mathbf{z}^{t+1}) + \psi_x'(\mathbf{x}^{t+1}), -\nabla_y f(\mathbf{z}^{t+1}) + \psi_y'(\mathbf{y}^{t+1})\big).$$

8: Let $a_{t+1} = \frac{2\langle V_\psi(\mathbf{z}^{t+1}), \mathbf{v}^t - \mathbf{z}^{t+1}\rangle}{\|V_\psi(\mathbf{z}^{t+1})\|_{\mathcal{E}^*}^2}$ and generate solution $\bar{\mathbf{z}}^{t+1} = \left(\sum_{i=1}^{t+1} a_i\right)^{-1} \sum_{i=1}^{t+1} a_i \mathbf{z}^i$.
9: $\mathbf{v}^{t+1} = \arg\min_{\mathbf{v}\in Q} \left[a_{t+1}\langle V_\psi(\mathbf{z}^{t+1}), \mathbf{v}\rangle + \frac{1}{2}\|\mathbf{v} - \mathbf{v}^t\|_{\mathcal{E}}^2\right]$.
10: **end for**

We refer to Algorithm 1 as a template method because we have not yet specified the implementations for the local computations in Lines 4 and 5. Provided that the trajectories of these local computations satisfy the span conditions outlined in Definition 1, the template DM-SP procedure formally qualifies as a distributed gradient-span algorithm.

Theorem 2. *Consider the* DM-SP *template applied to $\mathscr{P}_{\text{SP}}$, assuming its local trajectories satisfy Definition 1. With the parameter choices of $\alpha_x = \frac{L_{xy} D_y}{D_x}$, $\alpha_y = \frac{L_{xy} D_x}{D_y}$, and $\lambda_t \equiv \lambda = 2$, we have:*

$$T_{\mathscr{P}_{\text{SP}}}^{\text{DM-SP}} \leq 2 + 4\frac{L_{xy} D_x D_y}{\epsilon}.$$

Remark 2 (Communication optimality). Theorem 2 shows that the communication cost of DM-SP depends only on the cross-coupled conditioning $L_{xy} D_x D_y$, independent of the diagonal conditioning. As established in Section 3, this $\mathcal{O}\big(\frac{L_{xy} D_x D_y}{\epsilon}\big)$ communication cost matches the theoretical lower bound within the family of distributed gradient-span algorithms. To our knowledge, DM-SP is the first communication-optimal algorithm for distributed SPs.

Remark 3 (Robustness to inexact distance estimates). Let us consider a practical scenario where the algorithm may not have the precise values of D_x and D_y in advance, but it has access to upper estimates $\hat{D}_x \geq D_x$ and $\hat{D}_y \geq D_y$. Let $\boxed{\theta \triangleq \frac{D_x \hat{D}_y}{\hat{D}_x D_y} + \frac{D_y \hat{D}_x}{\hat{D}_y D_x}}$, which quantifies the disproportionality between the true distance parameters and their estimates. Note that $\theta \geq 2$ with equality if and only if the estimates are proportional, i.e., $\hat{D}_x / D_x = \hat{D}_y / D_y$. Now, consider the DM-SP template applied to $\mathscr{P}_{\text{SP}}$. With the parameter choices of $\alpha_x = \frac{L_{xy} \hat{D}_y}{\hat{D}_x}$, $\alpha_y = \frac{L_{xy} \hat{D}_x}{\hat{D}_y}$, and $\lambda_t \equiv \lambda = 2$, we have:

$$T_{\mathscr{P}_{\text{SP}}}^{\text{DM-SP}} \leq 2 + 2\theta\frac{L_{xy} D_x D_y}{\epsilon}.$$

In particular, θ provides a *scale-invariant* robustness compared to existing accelerated frameworks. As shown in Table 1, the communication complexities of Cat-EG and Cat-Cat-DAGDA scale directly with the product of the estimates, $\hat{D}_x\hat{D}_y$. Consequently, if both agents conservatively overestimate their domain sizes by a uniform factor $c \gg 1$ (i.e., $\hat{D}_x = cD_x$ and $\hat{D}_y = cD_y$), the communication cost of Catalyst-based methods inflates by a massive factor of c^2. For DM-SP, however, this uniform overestimation perfectly cancels out, yielding $\theta = 2$.

Concrete implementation. Instantiating the template with the local solvers in Section D (Eq. (9)) gives the concrete DM-SP implementation. The bound below includes all local partial-gradient queries, with ARM [Lan et al., 2023] accelerating the decoupled convex subproblems without additional communication.

Theorem 3. *Consider the* DM-SP *algorithm equipped with the local solvers in Eq. (9), applied to* $\mathscr{P}_{\text{SP}}$. *With the same choices of* α_x, α_y, *and* $(\lambda_t)_{t\geq 0}$ *as in Theorem 2, we have:*

$$N^{\text{DM-SP}}_{\mathscr{P}_{\text{SP}}} = (c_x + c_y)\frac{2L_{xy}D_xD_y}{\epsilon} + 102\Big(\frac{L_{xy}D_xD_y}{\epsilon}\Big)^{\frac{1}{2}}\Big(c_x\Big(\frac{L_xD_x^2}{\epsilon}\Big)^{\frac{1}{2}} + c_y\Big(\frac{L_yD_y^2}{\epsilon}\Big)^{\frac{1}{2}}\Big).$$

We compare the total oracle cost with the EG baseline (Section 2.3).

Remark 4. Let us compare the oracle cost of DM-SP against the EG baseline, which requires

$$N^{\text{EG}}_{\mathscr{P}^{\circ}_{\text{SP}}} = (c_x + c_y)\cdot\Big(\frac{L_{xy}D_xD_y}{\epsilon} + \frac{L_xD_x^2}{\epsilon} + \frac{L_yD_y^2}{\epsilon}\Big)$$

oracle costs (Proposition 5). We conclude that the computational cost of DM-SP is consistently bounded by that of EG up to absolute constants. Furthermore, it yields a substantial improvement when

$$L_xD_x^2 + L_yD_y^2 \gg L_{xy}D_xD_y + \sqrt{L_{xy}D_xD_y}\cdot\Big(\frac{c_x}{c_x+c_y}\sqrt{L_xD_x^2} + \frac{c_y}{c_x+c_y}\sqrt{L_yD_y^2}\Big).$$

For instance, assuming uniform oracle costs ($c_x = c_y$), this improvement occurs when the diagonal conditioning dominates the cross-coupled conditioning, i.e., $L_xD_x^2 + L_yD_y^2 \gg L_{xy}D_xD_y$. To our knowledge, DM-SP is the first method to consistently improve upon the EG oracle cost for $\mathscr{P}_{\text{SP}}$. We note, however, that a gap remains between this achieved oracle cost and the theoretical oracle lower bound established in Theorem 1, which is a known open question even for non-distributed SPs.

Novelty. Our DM-SP is built upon a novel multi-stage reduction. We first leverage Reduced-Operator Method to reduce the problem to a Monteiro-Svaiter Subproblem (MSS). Then and crucially, we show that when this subproblem is weakly coupled, it can be solved by a Fully Decoupled Solver in one communication round. Consequently, the problem is further reduced to coordinate-wise Minimization of Residual Norms (MRNs). Finally, by exploiting the strong maximal monotonicity (cf. (Â2)), the agents can apply existing accelerated methods to solve the MRNs to desired accuracy.

5 Variational inequality problems with distributed oracles

Motivation. Thus far, we have studied SPs, which naturally model two-player zero-sum games. To capture more complex multiagent interactions (such as equilibrium computation in multiplayer general-sum games, network routing, and multiagent reinforcement learning), we extend our algorithmic framework to the broader class of monotone Variational Inequality Problems (VIPs) with separable composite terms and distributed oracles. Due to lack of space, we briefly outline the problem class and our results here, while defer the detailed presentation to Section E.

Problem class. We consider a distributed multiagent setting with a star communication network over a product space $\mathcal{E} = \mathcal{E}_1 \times \cdots \times \mathcal{E}_K$, where a joint decision variable is partitioned among K autonomous agents as $\mathbf{z} = (\mathbf{z}_1, \ldots, \mathbf{z}_K)$. Let us consider the problem class $\mathscr{P}_{\text{VIP}}$ as follows:

- **Operators and local components:** The problem is governed by a joint monotone operator $V(\mathbf{z}) = (V_1(\mathbf{z}), \cdots, V_K(\mathbf{z}))$ and a separable local composite function $\psi(\mathbf{z}) = \sum_{i=1}^K \psi_i(\mathbf{z}_i)$ defined on $Q = \operatorname{dom}\psi_1 \times \cdots \times \operatorname{dom}\psi_K$. We assume the solution set is bounded by local

distance parameters $D_i > 0$ for each agent. Furthermore, the operator satisfies block-wise Lipschitz continuity: for any fixed $\mathbf{z}_{-j}$, the mapping $V_i(\mathbf{z}_j; \mathbf{z}_{-j})$ is L_{ij}-Lipschitz continuous with respect to $\mathbf{z}_j$.

- **Distributed oracles:** Each Agent $i \in [K]$ controls its local variable $\mathbf{z}_i \in \operatorname{dom} \psi_i$, has access to its private function ψ_i and a partial oracle $\mathscr{O}_i(\mathbf{z}) = V_i(\mathbf{z})$.
- **Accuracy measure:** The goal is to find an ϵ-approximate solution $\bar{\mathbf{z}} \in Q$ such that the restricted gap $\Delta(\bar{\mathbf{z}}) \triangleq \sup_{\mathbf{z} \in \mathcal{B} \cap Q} [\langle V(\mathbf{z}), \bar{\mathbf{z}} - \mathbf{z} \rangle + \psi(\bar{\mathbf{z}}) - \psi(\mathbf{z})]$ satisfies $\Delta(\bar{\mathbf{z}}) \leq \epsilon$, where the bounded domain $\mathcal{B}$ is defined by the initial point $\mathbf{z}^0$ and distance parameters D_i, $i \in [K]$.

Conditionings. Let $\bar{L}_{ij} \triangleq \max\{L_{ij}, L_{ji}\}$. Similar to SPs, we hereby define the *cross-coupled conditioning*, denoted by $\sum_{i \in [K]} A_i$ with $A_i \triangleq D_i \sum_{j \neq i} \bar{L}_{ij} D_j$, that quantifies the cross-dependencies between the agents over the network. The *diagonal conditioning*, denoted by $\sum_{i \in [K]} B_i$ with $B_i \triangleq L_{ii} D_i^2$, measures the self-dependency within a single agent's domain.

New state-of-the-art communication cost. In the multiagent setting, Extragradient (EG) remains the state-of-the-art method, requiring $T^{\text{EG}}_{\mathscr{P}_{\text{VIP}}} = \mathcal{O}\big(\sum_{i \in [K]} \frac{A_i + B_i}{\epsilon}\big)$ communication cost, which depends on both the cross-coupled and diagonal conditionings. By extending our decoupled template to distributed VIPs, we propose DM-VIP, and establish a much better communication cost.

Theorem 4. *Consider the* DM-VIP *algorithm for problem class* $\mathscr{P}_{\text{VIP}}$. *We have*

$$T^{\text{DM-VIP}}_{\mathscr{P}_{\text{VIP}}} = \mathcal{O}\Big(\sum_{i \in [K]} A_i / \epsilon\Big).$$

Remark 5. The communication cost in Theorem 4 completely drops the dependence on the diagonal conditioning. Consequently, we have substantially improved the state-of-the-art EG cost when the diagonal conditioning dominates, i.e., when $\sum_i B_i \gg \sum_i A_i$.

6 Conclusion and limitations

This paper studies communication and oracle costs in distributed SPs and VIPs. For the class of SPs, we settle the communication complexity in the distributed setup within gradient-span framework, and consistently improve the long-standing oracle cost of EG method. For the class of distributed VIPs, we improve the state-of-the-art communication cost. The following directions are not addressed in this paper and are left for future work: (a) closing the gap of oracle costs; (b) showing lower bound for non-zero-sum games; and (c) showing information-theoretic lower bounds for randomized methods.

Acknowledgments and Disclosure of Funding

The authors thank Ali Zindari, Ehsan Goharshady, and Krishnendu Chatterjee for their helpful discussions and suggestions on this paper. RL acknowledges the support of ERC CoG 863818 (ForM-SMArt) and Austrian Science Fund (FWF) 10.55776/COE12. The authors used Gemini 3.1 Pro for writing assistance and formatting checks; the lower bound proof, the derivation of the Decoupled Method, and the convergence analysis are the authors' own work.

References

Aharon Ben-Tal and Arkadi Nemirovski. Robust optimization–methodology and applications. *Mathematical programming*, 92(3):453–480, 2002. 1

Aleksandr Beznosikov, Valentin Samokhin, and Alexander Gasnikov. Distributed saddle point problems: lower bounds, near-optimal and robust algorithms. *Optimization Methods and Software*, pages 1–18, 2025. 1

Radu Ioan Boţ and Enis Chenchene. Extra-gradient method with flexible anchoring: Strong convergence and fast residual decay. *arXiv preprint arXiv:2410.14369*, 2024. E.4.2

Vincent Conitzer and Tuomas Sandholm. Communication complexity as a lower bound for learning in games. In *Proceedings of the twenty-first international conference on Machine learning*, page 24, 2004. 1

Yuyang Deng and Mehrdad Mahdavi. Local stochastic gradient descent ascent: Convergence analysis and communication efficiency. In *International Conference on Artificial Intelligence and Statistics*, pages 1387–1395. PMLR, 2021. 1

Nikita Doikov. Lecture notes on continuous optimization: Algorithms and complexity. Cornell ORIE 6365, 2026. URL https://doikov.com/Doikov_Optimization_2026.pdf. B.1

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 1

Sergiu Hart and Yishay Mansour. How long to equilibrium? the communication complexity of uncoupled equilibrium procedures. *Games and Economic Behavior*, 69(1):107–126, 2010. 1

Junling Hu, Michael P Wellman, et al. Multiagent reinforcement learning: theoretical framework and an algorithm. In *ICML*, volume 98, pages 242–250, 1998. 1

Anatoli Juditsky, Arkadi Nemirovski, et al. First order methods for nonsmooth convex large-scale optimization, ii: utilizing problems structure. *Optimization for Machine Learning*, 30(9):149–183, 2011. 2.3, 5, E, 20

Guanghui Lan and Yan Li. A novel catalyst scheme for stochastic minimax optimization. *Mathematical Programming*, pages 1–49, 2026. 2.3, 6

Guanghui Lan, Yuyuan Ouyang, and Zhe Zhang. Optimal and parameter-free gradient minimization methods for convex and nonconvex optimization. *arXiv preprint arXiv:2310.12139*, 2023. 4, C.4, C.4, 18, C.4

Tianyi Lin, Chi Jin, and Michael I Jordan. Near-optimal algorithms for minimax optimization. In *Conference on learning theory*, pages 2738–2779. PMLR, 2020. 2.3

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017. 1

Renato DC Monteiro and Benar Fux Svaiter. An accelerated hybrid proximal extragradient method for convex optimization and its implications to second-order methods. *SIAM Journal on Optimization*, 23(2):1092–1125, 2013. C.2

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. 1, 2.3, E

Arkadi S Nemirovsky. Information-based complexity of linear operator equations. *Journal of Complexity*, 8(2): 153–175, 1992. 3

Yurii Nesterov. High-order reduced-gradient methods for composite variational inequalities. *arXiv preprint arXiv:2311.15154*, 2023. 4, C.2

Yurii E. Nesterov. *Introductory Lectures on Convex Optimization - A Basic Course*, volume 87 of *Applied Optimization*. Springer, 2004. ISBN 978-1-4613-4691-3. doi: 10.1007/978-1-4419-8853-9. URL https://doi.org/10.1007/978-1-4419-8853-9. 2.2, B.1

Anh Duc Nguyen, Ilia Markov, Zhengqing Wu, Ali Ramezani-Kebrya, Kimon Antonakopoulos, Dan Alistarh, and Volkan Cevher. Layer-wise quantization for Quantized Optimistic Dual Averaging. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 46026–46072. PMLR, 2025. URL https://proceedings.mlr.press/v267/nguyen25d.html. 1

Noam Nisan and Ilya Segal. The communication requirements of efficient allocations and supporting prices. *Journal of Economic Theory*, 129(1):192–224, 2006. 1

J Ben Rosen. Existence and uniqueness of equilibrium points for concave n-person games. *Econometrica: Journal of the Econometric Society*, pages 520–534, 1965. 1

Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1):83–112, 2017. 1

Paul Tseng. On linear convergence of iterative methods for the variational inequality problem. *Journal of Computational and Applied Mathematics*, 60(1-2):237–252, 1995. 1

John von Neumann and Oskar Morgenstern. *Theory of games and economic behavior*. Princeton university press, 1947. 1

Yuanhao Wang and Jian Li. Improved algorithms for convex-concave minimax optimization. *Advances in Neural Information Processing Systems*, 33:4800–4810, 2020. 2.3, A, 7

Junchi Yang, Siqi Zhang, Negar Kiyavash, and Niao He. A catalyst framework for minimax optimization. *Advances in Neural Information Processing Systems*, 33:5667–5678, 2020. 2.3, 6

TaeHo Yoon and Nicolas Loizou. Pearl-prox: Proximal algorithm for resolving player drift in multiplayer federated learning. In *OPT 2025: Optimization for Machine Learning*, 2025. 1, 2.3

TaeHo Yoon, Sayantan Choudhury, and Nicolas Loizou. Multiplayer federated learning: Reaching equilibrium with less communication. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=9JX8XrTVEz. 1, 2.3

Junyu Zhang, Mingyi Hong, and Shuzhong Zhang. On lower iteration complexity bounds for the convex concave saddle point problems. *Mathematical Programming*, 194(1):901–935, 2022. 3

Siqi Zhang, Sayantan Choudhury, Sebastian U Stich, and Nicolas Loizou. Communication-efficient gradient descent-ascent methods for distributed variational inequalities: Unified analysis and local updates. In *The Twelfth International Conference on Learning Representations*, 2024. 1, 2.3

Ali Zindari, Parham Yazdkhasti, Anton Rodomanov, Tatjana Chavdarova, and Sebastian U Stich. Decoupled SGDA for games with intermittent strategy communication. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=ZYkFTSEZ6k. 1, 2.3, A

A Detailed review of existing algorithms

In this section, we provide the detailed formulations, distributed trajectories, and formal complexity results for the algorithms summarized in Section 2.3.

We recall that $\mathscr{P}_{\text{SP}}^{\circ}$ denotes the subclass of $\mathscr{P}_{\text{SP}}$ where the local components ψ_x and ψ_y are zero. In this setting, the problem reduces to finding a saddle point of a smooth convex-concave function f. We note that many algorithms discussed below do not handle general composite functions and only deal with problem instances from this non-composite subclass.

As in Remark 3, let us consider a practical scenario where the algorithms may not have the precise values of D_x and D_y in advance, but they have access to upper estimates $\hat{D}_x \geq D_x$ and $\hat{D}_y \geq D_y$. Let

$$\boxed{\theta \triangleq \frac{D_x \hat{D}_y}{\hat{D}_x D_y} + \frac{D_y \hat{D}_x}{\hat{D}_y D_x},}$$

which quantifies the disproportionality between the true distance parameters and their estimates.

Extragradient (EG). Suppose we apply EG to a non-composite problem instance $P = (f, 0, 0, \mathbf{z}^0) \in \mathscr{P}_{\text{SP}}^{\circ}$. The method iteratively maintains and updates two sequences of variables $\mathbf{v}^k, \mathbf{z}^k \in \mathcal{E}_x \times \mathcal{E}_y$. Initializing with $\mathbf{v}^0 = \mathbf{z}^0$, the updates for iteration $k = 0, 1, \ldots, K-1$ are given by:

$$\mathbf{z}^{k+1} = \mathbf{v}^k - \eta^k \mathbf{P}^{-1} V^f(\mathbf{v}^k),$$
$$\mathbf{v}^{k+1} = \mathbf{v}^k - \eta^k \mathbf{P}^{-1} V^f(\mathbf{z}^{k+1}),$$

where $\eta^k > 0$ is the step size at iteration k.

Although the EG method is conventionally formulated as a K-iteration loop, its distributed execution requires $2K$ communication rounds. In each iteration, the agents must synchronize twice to evaluate the coupled partial gradients at $\mathbf{v}^k$ and at $\mathbf{z}^{k+1}$. Thus, the EG method corresponds to a distributed gradient-span algorithm with $T = 2K$ rounds and local step lengths $\tau_x^t = \tau_y^t = 1$, where the trajectories are given by:

$$\hat{Z}_x^{2k} = \hat{Z}_y^{2k} = \{\mathbf{v}^k\} \quad \text{and} \quad \hat{Z}_x^{2k+1} = \hat{Z}_y^{2k+1} = \{\mathbf{z}^{k+1}\},$$

for $k = 0, \ldots, K-1$.

To align with Definition 1, the method generates a solution $\bar{\mathbf{z}}^{t+1}$ after each round $t \in \{0, \ldots, T-1\}$. Because each iteration requires two communication rounds, the algorithm effectively updates its output only every two rounds. Specifically, after completing iteration k (which corresponds to round $t = 2k+1$), it outputs the ergodic average $\bar{\mathbf{z}}^{2k+2} = \frac{1}{\sum_{i=0}^{k} \eta^i} \left(\sum_{i=0}^{k} \eta^i \mathbf{z}^{i+1}\right)$. During the intermediate rounds (at $t = 2k$), it simply retains the previous solution by setting $\bar{\mathbf{z}}^{2k+1} = \bar{\mathbf{z}}^{2k}$ (where $\bar{\mathbf{z}}^0 = \mathbf{z}^0$). Since $\psi_x = \psi_y = 0$, all minimizations in Definition 1 return $\mathbf{u}_x$ or $\mathbf{u}_y$. Since the trajectories and the generated solutions are formed by linear combinations of the evaluated gradients, they satisfy the span conditions in Definition 1.

The complexity of the EG method provides a natural baseline. Translating the classic results into our distributed complexity measures yields the following upper bounds.

Proposition 5 (Juditsky et al. 2011, Eq. (6.21))**.** *Consider the* EG *method applied to* $\mathscr{P}_{\text{SP}}^{\circ}$*. With the parameter choices of* $\alpha_x = \frac{L_x \hat{D}_x + L_{xy} \hat{D}_y}{\hat{D}_x}$, $\alpha_y = \frac{L_y \hat{D}_y + L_{xy} \hat{D}_x}{\hat{D}_y}$, *and* $\eta^k \equiv 1$*, we have*

$$T_{\mathscr{P}_{\text{SP}}^{\circ}}^{\text{EG}} = \theta \cdot \frac{L_{xy} D_x D_y}{\epsilon} + \frac{L_x D_x^2}{\epsilon} + \frac{L_y D_y^2}{\epsilon},$$
$$N_{\mathscr{P}_{\text{SP}}^{\circ}}^{\text{EG}} = (c_x + c_y) \cdot \left(\theta \cdot \frac{L_{xy} D_x D_y}{\epsilon} + \frac{L_x D_x^2}{\epsilon} + \frac{L_y D_y^2}{\epsilon}\right).$$

Decoupled GDA. Let us apply DGDA to an instance $P \in \mathscr{P}_{\text{SP}}^{\circ}$ with a fixed local trajectory length τ. The algorithm maintains local iterates $\mathbf{x}^{t,l}$ and $\mathbf{y}^{t,l}$ for round $t = 0, \ldots, T-1$ and local step

$l = 0, \dots, \tau$. At the beginning of round t, the agents synchronize by exchanging their latest local iterates. Specifically, Agent x receives $\hat{\mathbf{y}}^t$ and Agent y receives $\hat{\mathbf{x}}^t$, defined as:

$$\hat{\mathbf{x}}^t \triangleq \begin{cases} \mathbf{x}^0 & \text{if } t = 0 \\ \mathbf{x}^{t-1,\tau} & \text{if } t > 0 \end{cases} \quad \text{and} \quad \hat{\mathbf{y}}^t \triangleq \begin{cases} \mathbf{y}^0 & \text{if } t = 0 \\ \mathbf{y}^{t-1,\tau} & \text{if } t > 0 \end{cases}.$$

The agents then initialize their local variables for the current round as $\mathbf{x}^{t,0} = \hat{\mathbf{x}}^t$ and $\mathbf{y}^{t,0} = \hat{\mathbf{y}}^t$. With the remote variables firmly fixed, the agents execute τ local gradient steps. For $l = 0, \dots, \tau - 1$, the local updates are given by:

$$\mathbf{x}^{t,l+1} = \mathbf{x}^{t,l} - \eta_x \mathbf{P}_x^{-1} \nabla_x f(\mathbf{x}^{t,l}, \hat{\mathbf{y}}^t),$$
$$\mathbf{y}^{t,l+1} = \mathbf{y}^{t,l} + \eta_y \mathbf{P}_y^{-1} \nabla_y f(\hat{\mathbf{x}}^t, \mathbf{y}^{t,l}),$$

where $\eta_x, \eta_y > 0$ are the local step sizes.

The DGDA algorithm yields the following trajectories:

$$\hat{Z}_x^t = \left\{(\mathbf{x}^{t,l}, \hat{\mathbf{y}}^t)\right\}_{l=0}^{\tau-1} \quad \text{and} \quad \hat{Z}_y^t = \left\{(\hat{\mathbf{x}}^t, \mathbf{y}^{t,l})\right\}_{l=0}^{\tau-1}.$$

After each round $t \in \{0, \dots, T-1\}$, the method returns the updated local iterates as the current solution $\bar{\mathbf{z}}^{t+1} = (\mathbf{x}^{t,\tau}, \mathbf{y}^{t,\tau})$. Since $\psi_x = \psi_y = 0$, all minimizations in Definition 1 return $\mathbf{u}_x$ or $\mathbf{u}_y$. Because the trajectories and the generated solutions are formed entirely by linear combinations of the evaluated gradients, DGDA is indeed a distributed gradient-span algorithm by Definition 1.

While DGDA lies perfectly in our framework, its theoretical guarantees are highly restrictive. The algorithm is only proven to converge for restricted strongly convex-strongly concave problem instances where the cross-coupling between the variables is sufficiently weak [Zindari et al., 2025]. In this narrowly defined regime, DGDA achieves a logarithmic communication complexity of $\mathcal{O}\big(\log \frac{1}{\epsilon}\big)$, which is a clear improvement over the EG baseline. However, for general problem instances in $\mathscr{P}_{\text{SP}}^{\circ}$ with stronger coupling, the delayed remote variables cause the local updates to drift, ultimately leading the method to diverge. Consequently, DGDA fails to provide any meaningful complexity guarantee for the general problem class $\mathscr{P}_{\text{SP}}^{\circ}$ under consideration.

Catalyst acceleration. Catalyst methods first add small regularization terms to the objective: $f(\mathbf{x}, \mathbf{y}) + \frac{\epsilon}{4\hat{D}_x^2}\|\mathbf{x} - \mathbf{x}^0\|_x^2 - \frac{\epsilon}{4\hat{D}_y^2}\|\mathbf{y} - \mathbf{y}^0\|_y^2$, reducing the problem to a strongly convex-strongly concave one. Then, Catalyst introduces an outer loop, indexed by $k = 0, 1, \dots, K-1$, designed to balance the conditioning between the two variables. For instance, when the conditioning of x is worse (i.e. $L_x \hat{D}_x^2 \geq L_y \hat{D}_y^2$), the method carefully maintains an extrapolation sequence $(\tilde{\mathbf{x}}^k)_{k=0}^{K-1}$ and, in each outer iteration, adds a proximal term $\frac{\lambda_x}{2}\|\mathbf{x} - \tilde{\mathbf{x}}^k\|_x^2$ to the objective. Conversely, if the conditioning of y is worse, the outer loop would instead maintain an extrapolation sequence for y and add a corresponding regularization term for y. An inner base algorithm (EG in this case) is then deployed to solve this regularized subproblem to a specified accuracy.

To simplify the notation, let $L_{\max} = \max\{L_x, L_y, L_{xy}\}$.

Proposition 6 (Yang et al. [2020], Lan and Li [2026]). *Consider the Catalyst framework equipped with the* EG *method as the inner solver, denoted by Cat-EG, applied to $\mathscr{P}_{\text{SP}}^{\circ}$. Then, we have:*

$$T_{\mathscr{P}_{\text{SP}}^{\circ}}^{\text{Cat-EG}} = \mathcal{O}\left(\left(\frac{L_{\max}\hat{D}_x\hat{D}_y}{\epsilon} + \sqrt{\frac{L_x\hat{D}_x^2}{\epsilon}} + \sqrt{\frac{L_y\hat{D}_y^2}{\epsilon}}\right)\log^2\left(\frac{1}{\epsilon}\right)\right),$$
$$N_{\mathscr{P}_{\text{SP}}^{\circ}}^{\text{Cat-EG}} = (c_x + c_y)\cdot\mathcal{O}\left(\left(\frac{L_{\max}\hat{D}_x\hat{D}_y}{\epsilon} + \sqrt{\frac{L_x\hat{D}_x^2}{\epsilon}} + \sqrt{\frac{L_y\hat{D}_y^2}{\epsilon}}\right)\log^2\left(\frac{1}{\epsilon}\right)\right).$$

Now, let us explain the caveats we mentioned in Section 2.3 regarding the Cat-EG method in more detail.

To explain the second caveat of Cat-EG, its sensitivity to inexact diameter estimates, we compare it with the EG baseline and our proposed method. Because Catalyst uses $\hat{D}_x$ and $\hat{D}_y$ to set the initial regularization, its complexity scales directly with these estimates rather than the true distances D_x and D_y. Specifically, the diagonal terms in EG or DM-SP depend strictly on the true distances,

whereas in Cat-EG they scale with $\hat{D}_x$ and $\hat{D}_y$. Similarly, for the cross-coupled term, EG or DM-SP depends on the true distances multiplied by the proportionality ratio θ. If the estimates are loose but proportional (e.g., $\hat{D}_x = cD_x$ and $\hat{D}_y = cD_y$), θ remains 2, leaving the complexity unaffected by the overestimation factor c. In contrast, the Cat-EG coupled term scales with $\hat{D}_x\hat{D}_y$, meaning any overestimation of $\hat{D}_x \gg D_x$ or $\hat{D}_y \gg D_y$ directly inflates the bound. Consequently, EG and our proposed DM-SP method are much more robust to inexact distance estimates, provided the estimates are roughly proportional.

Furthermore, we note that the fourth caveat of Cat-EG (performing worse than unaccelerated EG) can be easily verified. For instance, consider a problem instance where $L_x = 10^6$, $L_{xy} = 1$, $L_y = 10^{-6}$, $D_x = 10^{-3}$, and $D_y = 10^3$. Under this conditioning, the theoretical upper bound of Cat-EG significantly exceeds that of standard EG.

Four-loop method. The Cat-Cat-DAGDA method [Wang and Li, 2020] first adds small $\mathcal{O}(\epsilon)$ regularizations to both $\mathbf{x}$ and $\mathbf{y}$ to ensure the objective is strongly convex-strongly concave. It then executes four nested loops, which justifies our naming convention: (i) The first loop is a Catalyst outer loop adding a proximal regularization term to the primal variable; (ii) The second loop is another Catalyst outer loop adding a proximal regularization term to the dual variable; (iii) The third loop manages communication by exchanging and freezing the remote variables, identical to DGDA; and (iv) The fourth loop performs local computations, but unlike the standard gradient steps in DGDA, it employs an accelerated gradient method (hence DAGDA) to solve the inner subproblems.

While the fourth loop of the method is a local computation loop, the first three loops all require communication rounds. Let us state the complexity results of this method without going into the tedious details of the algorithm.

Proposition 7 (Wang and Li [2020])**.** *Consider the Cat-Cat-DAGDA method applied to $\mathscr{P}_{\mathrm{SP}}^{\circ}$. Then, we have:*

$$T_{\mathscr{P}_{\mathrm{SP}}^{\circ}}^{\text{Cat-Cat-DAGDA}} = \mathcal{O}\left(\frac{L_{xy}\hat{D}_x\hat{D}_y}{\epsilon}\log^3\left(\frac{1}{\epsilon}\right)\right),$$

$$N_{\mathscr{P}_{\mathrm{SP}}^{\circ}}^{\text{Cat-Cat-DAGDA}} = (c_x + c_y)\cdot\mathcal{O}\left(\left(\frac{\sqrt{L_{\max}L_{xy}}\hat{D}_x\hat{D}_y}{\epsilon} + \sqrt{\frac{L_x\hat{D}_x^2}{\epsilon}} + \sqrt{\frac{L_y\hat{D}_y^2}{\epsilon}}\right)\log^4\left(\frac{1}{\epsilon}\right)\right).$$

To derive the communication bound in Proposition 7, the initial regularization strengths are of order $\epsilon/\hat{D}_x^2$ and $\epsilon/\hat{D}_y^2$. For sufficiently small ϵ, the two Catalyst loops require respectively $\mathcal{O}(\sqrt{L_{xy}\hat{D}_x^2/\epsilon}\,\log(1/\epsilon))$ and $\mathcal{O}(\sqrt{L_{xy}\hat{D}_y^2/\epsilon}\,\log(1/\epsilon))$ iterations, while the best-response loop requires $\mathcal{O}(\log(1/\epsilon))$ communication rounds per call. Multiplying these three counts gives

$$\begin{aligned} T_{\mathscr{P}_{\mathrm{SP}}^{\circ}}^{\text{Cat-Cat-DAGDA}} &= \mathcal{O}\left(\sqrt{\frac{L_{xy}\hat{D}_x^2}{\epsilon}}\sqrt{\frac{L_{xy}\hat{D}_y^2}{\epsilon}}\log^3\left(\frac{1}{\epsilon}\right)\right) \\ &= \mathcal{O}\left(\frac{L_{xy}\hat{D}_x\hat{D}_y}{\epsilon}\log^3\left(\frac{1}{\epsilon}\right)\right). \end{aligned}$$

The innermost accelerated gradient loop uses only local oracle queries and contributes no additional communication rounds.

B Lower complexity bounds

In this section, we provide detailed proofs for the lower complexity bounds of distributed gradient-span algorithms applied to $\mathscr{P}_{\mathrm{SP}}$. We focus on a subclass of unconstrained SPs with $\psi_x = \psi_y = 0$, denoted $\mathscr{P}_{\mathrm{SP}}^{\circ}$. We assume the initial points are $\mathbf{x}^0 = \mathbf{0}$ and $\mathbf{y}^0 = \mathbf{0}$. Lower bounds established for this subclass hold for the general class $\mathscr{P}_{\mathrm{SP}}$.

On this subclass, we have $\operatorname{dom}\psi_x = \mathcal{E}_x$ and $\operatorname{dom}\psi_y = \mathcal{E}_y$, so all minimizations in Definition 1 return $\mathbf{u}_x$ or $\mathbf{u}_y$. Hence, for all histories $Z_x^{t,l}$ of Agent x and $Z_y^{t,l}$ of Agent y, we have

$$\begin{aligned} \mathcal{S}_x(Z_x^{t,l}) &= \mathbf{P}_x^{-1}\operatorname{span}\{\nabla_x f(\mathbf{z}) \mid \mathbf{z} \in Z_x^{t,l}\}, \\ \mathcal{S}_y(Z_y^{t,l}) &= \mathbf{P}_y^{-1}\operatorname{span}\{\nabla_y f(\mathbf{z}) \mid \mathbf{z} \in Z_y^{t,l}\}. \end{aligned} \tag{3}$$

Consequently, on $\mathscr{P}_{\text{SP}}^{\circ}$, every query point, remote variable, and generated solution lies in the initial point plus the span of the corresponding evaluated partial gradients. This is the form of Definition 1 used in the lower bound proofs below.

Geometry. Let us consider the spaces $\mathcal{E}_x = \mathbb{R}^{n_x}$ and $\mathcal{E}_y = \mathbb{R}^{n_y}$ equipped with the standard Euclidean inner product $\langle \cdot, \cdot \rangle$ and the corresponding Euclidean norm $\|\cdot\|$. The cases with general preconditioned norms $\|\cdot\|_x$ and $\|\cdot\|_y$ can be proven similarly.

Function subfamilies. To establish the overall lower bound, we decompose the general function family into three distinct subfamilies:

Definition 2. Let $\mathscr{F}_x = \mathscr{F}(L_x, 0, 0, D_x, 0)$ denote the set of functions

$$F_x(\mathbf{x}, \mathbf{y}) = \frac{1}{2}\|\mathbf{A}_x\mathbf{x} - \mathbf{b}_x\|^2,$$

where the matrix $\mathbf{A}_x \in \mathbb{R}^{n_x \times n_x}$ satisfies $\|\mathbf{A}_x\|^2 \leq L_x$, and the vector $\mathbf{b}_x \in \mathbb{R}^{n_x}$ is such that the linear system $\mathbf{A}_x\mathbf{x} = \mathbf{b}_x$ has a solution $\mathbf{x}^* \in \mathcal{E}_x$ satisfying $\|\mathbf{x}^*\| \leq D_x$.

Definition 3. Let $\mathscr{F}_y = \mathscr{F}_y(0, 0, L_y, 0, D_y)$ denote the set of functions

$$F_y(\mathbf{x}, \mathbf{y}) = -\frac{1}{2}\|\mathbf{A}_y\mathbf{y} - \mathbf{b}_y\|^2,$$

where the matrix $\mathbf{A}_y \in \mathbb{R}^{n_y \times n_y}$ satisfies $\|\mathbf{A}_y\|^2 \leq L_y$, and the vector $\mathbf{b}_y \in \mathbb{R}^{n_y}$ is such that the linear system $\mathbf{A}_y\mathbf{y} = \mathbf{b}_y$ has a solution $\mathbf{y}^* \in \mathcal{E}_y$ satisfying $\|\mathbf{y}^*\| \leq D_y$.

Definition 4. Let $\mathscr{F}_{xy} = \mathscr{F}_{xy}(0, L_{xy}, 0, D_x, D_y)$ denote the set of functions

$$F_{xy}(\mathbf{x}, \mathbf{y}) = \langle \mathbf{A}_{xy}\mathbf{x} - \mathbf{b}_{xy}, \mathbf{y} \rangle,$$

where the matrix $\mathbf{A}_{xy} \in \mathbb{R}^{n_y \times n_x}$ satisfies $\|\mathbf{A}_{xy}\| \leq L_{xy}$, and the vector $\mathbf{b}_{xy} \in \mathbb{R}^{n_y}$ is such that the linear system $\mathbf{A}_{xy}\mathbf{x} = \mathbf{b}_{xy}$ has a solution $\mathbf{x}^* \in \mathcal{E}_x$ satisfying $\|\mathbf{x}^*\| \leq D_x$.

Distributed oracles. We consider the pairs of distributed oracles $(\mathscr{O}_x, \mathscr{O}_y)$ that return the partial gradients of the functions in these subfamilies:

- For $F_x \in \mathscr{F}_x$: $\mathscr{O}_x(\mathbf{x}, \mathbf{y}) = \nabla_x F_x(\mathbf{x}, \mathbf{y}) = \mathbf{A}_x^\top(\mathbf{A}_x\mathbf{x} - \mathbf{b}_x)$ and $\mathscr{O}_y(\mathbf{x}, \mathbf{y}) = -\nabla_y F_x(\mathbf{x}, \mathbf{y}) = \mathbf{0}$.
- For $F_y \in \mathscr{F}_y$: $\mathscr{O}_x(\mathbf{x}, \mathbf{y}) = \nabla_x F_y(\mathbf{x}, \mathbf{y}) = \mathbf{0}$ and $\mathscr{O}_y(\mathbf{x}, \mathbf{y}) = -\nabla_y F_y(\mathbf{x}, \mathbf{y}) = \mathbf{A}_y^\top(\mathbf{A}_y\mathbf{y} - \mathbf{b}_y)$.
- For $F_{xy} \in \mathscr{F}_{xy}$: $\mathscr{O}_x(\mathbf{x}, \mathbf{y}) = \nabla_x F_{xy}(\mathbf{x}, \mathbf{y}) = \mathbf{A}_{xy}^\top\mathbf{y}$ and $\mathscr{O}_y(\mathbf{x}, \mathbf{y}) = -\nabla_y F_{xy}(\mathbf{x}, \mathbf{y}) = \mathbf{b}_{xy} - \mathbf{A}_{xy}\mathbf{x}$.

Accuracy measures. For any approximate solution $\bar{\mathbf{z}} = (\bar{\mathbf{x}}, \bar{\mathbf{y}}) \in \mathcal{E}_x \times \mathcal{E}_y$, the restricted primal-dual gaps over the bounded sets evaluate to:

- For $F_x \in \mathscr{F}_x$: $\Delta_{F_x}(\bar{\mathbf{z}}) = F_x(\bar{\mathbf{x}}, \bar{\mathbf{y}}) - \min_{\mathbf{x} \in \mathcal{B}_x} F_x(\mathbf{x}, \bar{\mathbf{y}}) = \frac{1}{2}\|\mathbf{A}_x\bar{\mathbf{x}} - \mathbf{b}_x\|^2$.
- For $F_y \in \mathscr{F}_y$: $\Delta_{F_y}(\bar{\mathbf{z}}) = \max_{\mathbf{y} \in \mathcal{B}_y} F_y(\bar{\mathbf{x}}, \mathbf{y}) - F_y(\bar{\mathbf{x}}, \bar{\mathbf{y}}) = \frac{1}{2}\|\mathbf{A}_y\bar{\mathbf{y}} - \mathbf{b}_y\|^2$.
- For $F_{xy} \in \mathscr{F}_{xy}$: $\Delta_{F_{xy}}(\bar{\mathbf{z}}) = \max_{\mathbf{y} \in \mathcal{B}_y} \langle \mathbf{A}_{xy}\bar{\mathbf{x}} - \mathbf{b}_{xy}, \mathbf{y} \rangle = D_y\|\mathbf{A}_{xy}\bar{\mathbf{x}} - \mathbf{b}_{xy}\|$.

Problem subclasses. These constructions yield three distinct problem subclasses:

- $\mathscr{P}_x = \big(\mathscr{F}_x, \mathscr{O}_x, \mathscr{O}_y, \epsilon\big)$,
- $\mathscr{P}_y = \big(\mathscr{F}_y, \mathscr{O}_x, \mathscr{O}_y, \epsilon\big)$, and
- $\mathscr{P}_{xy} = \big(\mathscr{F}_{xy}, \mathscr{O}_x, \mathscr{O}_y, \epsilon\big)$.

The worst-case complexity for general SPs is bounded below by the maximum complexity among these three subclasses.

Structure of this section. We present our lower bound proofs by establishing a connection from distributed SPs to convex minimization. In Section B.1, we present a properly rescaled variant of Nesterov's construction for unconstrained convex optimization. We analyze the three problem subclasses separately in Sections B.2.1 and B.2.2, mapping the lower bound for each subclass to this core convex minimization problem. Finally, we combine the results to establish the overall lower bound for distributed SPs in Section B.3.

B.1 The "worst function in the world" (with proper rescaling)

While the original formulation (Theorem 2.1.7 by Nesterov [2004]) establishes a lower bound with respect to an arbitrary initial distance $\|\mathbf{v}^0 - \mathbf{v}^*\|$, this is insufficient for the lower bound proof in this paper. We are working on a lower bound analysis with a more specific problem subclass, where the algorithms may be designed with prior knowledge of a bound D on the distance to the optimum. To this end, we use a refined variant of Nesterov's construction [Nesterov, 2004, Section 2.1.2], also presented in Doikov [2026, Section 3.4.1], such that the distance to the optimum is bounded by D.

For any given matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ and vector $\mathbf{b} \in \mathbb{R}^m$, we define the k-th Krylov subspace as follows:

$$\mathcal{H}^k(\mathbf{A}, \mathbf{b}) \triangleq \operatorname{span}\left\{\mathbf{A}^\top \mathbf{b}, (\mathbf{A}^\top \mathbf{A})\mathbf{A}^\top \mathbf{b}, \ldots, (\mathbf{A}^\top \mathbf{A})^{k-1}\mathbf{A}^\top \mathbf{b}\right\}.$$

Proposition 8. *Let $L > 0$, $D > 0$, and integer $0 \le k \le \frac{\min\{m,n\}-1}{2}$. Then, there exist a matrix $\mathbf{A} = \mathbf{A}(L,k) \in \mathbb{R}^{m\times n}$ with $\|\mathbf{A}\| \le L$ and a vector $\mathbf{b} = \mathbf{b}(L, D, k) \in \mathbb{R}^m$, such that*

$$\min_{\mathbf{v}\in\mathcal{H}^k(\mathbf{A},\mathbf{b})} \frac{1}{2}\|\mathbf{A}\mathbf{v} - \mathbf{b}\|^2 \ge \frac{3L^2D^2}{64(k+1)^2},$$

and the linear system $\mathbf{A}\mathbf{v} = \mathbf{b}$ has a solution $\mathbf{v}^* \in \mathbb{R}^n$ satisfying $\|\mathbf{v}^*\| \le D$.

Proof. Let $p = 2k + 1$. By the condition $k \le \frac{\min\{m,n\}-1}{2}$, we have $p \le \min\{m, n\}$. Let $\mathbf{M}_p \in \mathbb{R}^{p\times p}$ be the symmetric tridiagonal matrix defined as:

$$\mathbf{M}_p = \begin{bmatrix} 1 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \cdots & 0 \\ 0 & -1 & 2 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & -1 \\ 0 & 0 & \cdots & -1 & 2 \end{bmatrix}.$$

Let $\mathbf{B}_p \in \mathbb{R}^{p\times p}$ be the upper bidiagonal matrix such that $\mathbf{B}_p^\top \mathbf{B}_p = \mathbf{M}_p$, defined as:

$$\mathbf{B}_p = \begin{bmatrix} 1 & -1 & 0 & \cdots & 0 \\ 0 & 1 & -1 & \ddots & \vdots \\ 0 & 0 & 1 & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & -1 \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix}.$$

We define the matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ as the block matrix:

$$\mathbf{A} = \frac{L}{2}\begin{bmatrix} \mathbf{B}_p & \mathbf{0}_{p\times(n-p)} \\ \mathbf{0}_{(m-p)\times p} & \mathbf{0}_{(m-p)\times(n-p)} \end{bmatrix}.$$

The matrix $\mathbf{A}^\top \mathbf{A} \in \mathbb{R}^{n\times n}$ is given by the block-diagonal matrix:

$$\mathbf{A}^\top \mathbf{A} = \frac{L^2}{4}\begin{bmatrix} \mathbf{M}_p & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

Since the absolute values of the entries in each row of $\mathbf{M}_p$ sum to at most 4, the Gershgorin circle theorem gives $\lambda_{\max}(\mathbf{M}_p) \le 4$. Hence, the spectral norm of $\mathbf{A}$ is bounded as $\|\mathbf{A}\| = \sqrt{\lambda_{\max}(\mathbf{A}^\top \mathbf{A})} = \frac{L}{2}\sqrt{\lambda_{\max}(\mathbf{M}_p)} \le L$.

Let $\gamma = D\sqrt{\frac{6}{p(p+1)(2p+1)}}$ and $\mathbf{1}_p = (1, \ldots, 1)^\top \in \mathbb{R}^p$. We define $\mathbf{b} \in \mathbb{R}^m$ as the block vector:

$$\mathbf{b} = \gamma \frac{L}{2} \begin{bmatrix} \mathbf{1}_p \\ \mathbf{0}_{m-p} \end{bmatrix}.$$

Let $\mathbf{v}^* \in \mathbb{R}^n$ have the coordinates $v_i^* = \gamma(p+1-i)$ for $1 \le i \le p$, and 0 otherwise. Since $v_i^* - v_{i+1}^* = \gamma$ for $1 \le i \le p-1$ and $v_p^* = \gamma$, we have $\mathbf{A}\mathbf{v}^* = \mathbf{b}$. Its squared norm evaluates to:

$$\|\mathbf{v}^*\|^2 = \gamma^2 \sum_{j=1}^{p} j^2 = \gamma^2 \frac{p(p+1)(2p+1)}{6} = D^2.$$

The coordinates of $\mathbf{B}_p^\top \mathbf{1}_p$ are 1 for $i = 1$ and $1 - 1 = 0$ for $2 \le i \le p$. Thus, the initial vector of the Krylov subspace evaluates to $\mathbf{A}^\top \mathbf{b} = \gamma \frac{L^2}{4} \mathbf{e}_1^{(n)}$. Since $\mathbf{M}_p$ is tridiagonal, each multiplication by $\mathbf{A}^\top \mathbf{A}$ expands the non-zero support by at most one standard basis vector. Thus,

$$\mathcal{H}^k(\mathbf{A}, \mathbf{b}) \subseteq \operatorname{span}\{\mathbf{e}_1^{(n)}, \mathbf{e}_2^{(n)}, \ldots, \mathbf{e}_k^{(n)}\}.$$

Let $\mathbf{v} \in \mathcal{H}^k(\mathbf{A}, \mathbf{b})$. Then, $v_i = 0$ for $i > k$. Since $\mathbf{B}_p$ is upper bidiagonal, the i-th coordinate of $\mathbf{A}\mathbf{v}$ is zero for $k < i \le p$, while the i-th coordinate of $\mathbf{b}$ is $\gamma \frac{L}{2}$. Hence,

$$\begin{aligned} \frac{1}{2}\|\mathbf{A}\mathbf{v} - \mathbf{b}\|^2 &\ge \frac{1}{2}(p-k)\gamma^2 \frac{L^2}{4} = \frac{3(k+1)L^2D^2}{4(2k+1)(2k+2)(4k+3)} \\ &= \frac{3L^2D^2}{8(8k^2+10k+3)} \ge \frac{3L^2D^2}{64(k+1)^2}, \end{aligned}$$

where the last step uses $8k^2 + 10k + 3 \le 8(k+1)^2$. □

B.2 Three subclasses

B.2.1 Quadratic subclasses $\mathscr{P}_x$ and $\mathscr{P}_y$

In this section, we apply Proposition 8 to establish lower bounds for the quadratic subclasses $\mathscr{P}_x$ and $\mathscr{P}_y$.

Notice that the gradient span sequences are naturally confined to the standard Krylov subspaces. Specifically, for $\mathbf{A}_x \in \mathbb{R}^{n_x \times n_x}$ and $\mathbf{b}_x \in \mathbb{R}^{n_x}$, the subspace satisfies the algebraic progression:

$$(\mathbf{A}_x^\top \mathbf{A}_x)\mathcal{H}^m(\mathbf{A}_x, \mathbf{b}_x) \subseteq \mathcal{H}^{m+1}(\mathbf{A}_x, \mathbf{b}_x) \quad \text{and} \quad \mathbf{A}_x^\top \mathbf{b}_x \in \mathcal{H}^{m+1}(\mathbf{A}_x, \mathbf{b}_x).$$

Proposition 9. *Let $\mathscr{M}$ be a distributed gradient-span algorithm operating on an instance $F_x \in \mathscr{P}_x$ of the form $F_x(\mathbf{x}) = \frac{1}{2}\|\mathbf{A}_x\mathbf{x} - \mathbf{b}_x\|^2$. Let $\bar{\mathbf{x}}$ be the approximate solution generated by Agent x after evaluating N_x partial gradients. Then, $\bar{\mathbf{x}} \in \mathcal{H}^{N_x}(\mathbf{A}_x, \mathbf{b}_x)$.*

Proof. For $F_x \in \mathscr{P}_x$, we have $\psi_x = \psi_y = 0$. Let $\mathbf{z}_1, \ldots, \mathbf{z}_{N_x}$ be the sequence of query points evaluated by Agent x, where $\mathbf{z}_i = (\mathbf{x}_i, \mathbf{y}_i)$. Since $\mathbf{x}^0 = \mathbf{0}$, Definition 1 and (3) require that each query point $\mathbf{x}_{m+1}$ and the generated solution $\bar{\mathbf{x}}$ reside in the span of historical gradients. Let $S_m \triangleq \operatorname{span}\{\nabla_x F_x(\mathbf{z}_i) \mid 1 \le i \le m\} \subseteq \mathcal{E}_x$. We have $\mathbf{x}_{m+1} \in S_m$ and $\bar{\mathbf{x}} \in S_{N_x}$.

We show $S_m \subseteq \mathcal{H}^m(\mathbf{A}_x, \mathbf{b}_x)$ by induction. The base case $m = 0$ holds since $S_0 = \{\mathbf{0}\} \subseteq \mathcal{H}^0(\mathbf{A}_x, \mathbf{b}_x)$.

Assume $S_m \subseteq \mathcal{H}^m(\mathbf{A}_x, \mathbf{b}_x)$ for some $m \ge 0$. The gradient at $\mathbf{x}_{m+1}$ evaluates to $\nabla_x F_x(\mathbf{x}_{m+1}) = \mathbf{A}_x^\top \mathbf{A}_x \mathbf{x}_{m+1} - \mathbf{A}_x^\top \mathbf{b}_x$. Since $\mathbf{x}_{m+1} \in S_m \subseteq \mathcal{H}^m(\mathbf{A}_x, \mathbf{b}_x)$, applying the algebraic progression properties yields:

$$\nabla_x F_x(\mathbf{x}_{m+1}) \in (\mathbf{A}_x^\top \mathbf{A}_x)\mathcal{H}^m(\mathbf{A}_x, \mathbf{b}_x) - \mathbf{A}_x^\top \mathbf{b}_x \subseteq \mathcal{H}^{m+1}(\mathbf{A}_x, \mathbf{b}_x).$$

Thus, $S_{m+1} = S_m + \operatorname{span}\{\nabla_x F_x(\mathbf{x}_{m+1})\} \subseteq \mathcal{H}^{m+1}(\mathbf{A}_x, \mathbf{b}_x)$.

By induction, $S_{N_x} \subseteq \mathcal{H}^{N_x}(\mathbf{A}_x, \mathbf{b}_x)$, which implies $\bar{\mathbf{x}} \in \mathcal{H}^{N_x}(\mathbf{A}_x, \mathbf{b}_x)$. □

Analogously, for any instance $F_y \in \mathscr{P}_y$ evaluated by Agent y, of the form $F_y(\mathbf{x}, \mathbf{y}) = -\frac{1}{2}\|\mathbf{A}_y\mathbf{y} - \mathbf{b}_y\|^2$, the approximate solution generated after N_y queries satisfies $\bar{\mathbf{y}} \in \mathcal{H}^{N_y}(\mathbf{A}_y, \mathbf{b}_y)$.

Theorem 10. *Let $\mathscr{M}$ be a distributed gradient-span algorithm for problem class $\mathscr{P}_{\text{SP}}$, where $n_x \geq 2\sqrt{\frac{3L_xD_x^2}{64\epsilon}} + 1$ and $n_y \geq 2\sqrt{\frac{3L_yD_y^2}{64\epsilon}} + 1$. Then, we have*

$$N^{\mathscr{M}}_{\mathscr{P}_x} \geq c_x\left(\sqrt{\frac{3L_xD_x^2}{64\epsilon}} - 1\right),$$

and analogously,

$$N^{\mathscr{M}}_{\mathscr{P}_y} \geq c_y\left(\sqrt{\frac{3L_yD_y^2}{64\epsilon}} - 1\right).$$

Proof. We prove the bound for $\mathscr{P}_x$. Let $s = \sqrt{\frac{3L_xD_x^2}{64\epsilon}}$. If $s \leq 1$, the bound holds trivially. Otherwise, let $K = \lceil s \rceil - 2 \geq 0$, so that $K + 1 < s$. Since $n_x \geq 2s + 1 > 2K + 1$, we apply Proposition 8 to obtain $\mathbf{A}_K = \mathbf{A}(\sqrt{L_x}, K)$ and $\mathbf{b}_K = \mathbf{b}(\sqrt{L_x}, D_x, K)$ constructed in $\mathbb{R}^{n_x \times n_x}$ and $\mathbb{R}^{n_x}$.

We set $\mathbf{A}_x = \mathbf{A}_K$, $\mathbf{b}_x = \mathbf{b}_K$, and define $F_K(\mathbf{x}, \mathbf{y}) = \frac{1}{2}\|\mathbf{A}_x\mathbf{x} - \mathbf{b}_x\|^2$. By Proposition 8, $\|\mathbf{A}_x\|^2 \leq L_x$, and the linear system $\mathbf{A}_x\mathbf{x} = \mathbf{b}_x$ has a solution $\mathbf{x}^*$ satisfying $\|\mathbf{x}^*\| \leq D_x$. Hence, $F_K \in \mathscr{P}_x$.

When $\mathscr{M}$ is applied to P_K, suppose it generates an ϵ-saddle point $\bar{\mathbf{z}}$ utilizing N_x queries to $\mathscr{O}_x$. Suppose that $N_x \leq K$. Then, Proposition 9 implies $\bar{\mathbf{x}} \in \mathcal{H}^{N_x}(\mathbf{A}_x, \mathbf{b}_x) \subseteq \mathcal{H}^{K}(\mathbf{A}_x, \mathbf{b}_x)$. Bounding the restricted duality gap via Proposition 8 yields:

$$\epsilon \geq \Delta_{F_K}(\bar{\mathbf{z}}) = \frac{1}{2}\|\mathbf{A}_x\bar{\mathbf{x}} - \mathbf{b}_x\|^2 \geq \min_{\mathbf{v} \in \mathcal{H}^K(\mathbf{A}_x, \mathbf{b}_x)} \frac{1}{2}\|\mathbf{A}_x\mathbf{v} - \mathbf{b}_x\|^2 \geq \frac{3L_xD_x^2}{64(K+1)^2} = \frac{s^2\epsilon}{(K+1)^2} > \epsilon,$$

which is a contradiction. Hence, $N_x \geq K + 1 = \lceil s \rceil - 1 \geq s - 1$. Multiplying by c_x yields $N^{\mathscr{M}}_{\mathscr{P}_x} \geq c_x\big(\sqrt{\frac{3L_xD_x^2}{64\epsilon}} - 1\big)$.

The proof for $\mathscr{P}_y$ is symmetric, setting $\mathbf{A}_y = \mathbf{A}(\sqrt{L_y}, K)$ and $\mathbf{b}_y = \mathbf{b}(\sqrt{L_y}, D_y, K)$ for $K = \lceil\sqrt{\frac{3L_yD_y^2}{64\epsilon}}\rceil - 2$. □

B.2.2 Bilinear subclass $\mathscr{P}_{xy}$

In this section, we establish the lower bounds for the bilinear subclass $\mathscr{P}_{xy}$.

Consider any function from $\mathscr{P}_{xy}$ of the form $F(\mathbf{x}, \mathbf{y}) = \langle \mathbf{A}\mathbf{x} - \mathbf{b}, \mathbf{y}\rangle$, where $\mathbf{A} \in \mathbb{R}^{n_y \times n_x}$ and $\mathbf{b} \in \mathbb{R}^{n_y}$. The partial gradients are $\nabla_x F(\mathbf{x}, \mathbf{y}) = \mathbf{A}^\top\mathbf{y}$ and $\nabla_y F(\mathbf{x}, \mathbf{y}) = \mathbf{A}\mathbf{x} - \mathbf{b}$. For any $\bar{\mathbf{z}} = (\bar{\mathbf{x}}, \bar{\mathbf{y}}) \in \mathcal{E}_x \times \mathcal{E}_y$, the restricted primal-dual gap evaluates to the residual norm:

$$\Delta_F(\bar{\mathbf{z}}) = \max_{\mathbf{y} \in \mathcal{B}_y} \langle \mathbf{A}\bar{\mathbf{x}} - \mathbf{b}, \mathbf{y}\rangle = D_y\|\mathbf{A}\bar{\mathbf{x}} - \mathbf{b}\|.$$

We define the coupled Krylov subspaces in $\mathcal{E}_y$ and $\mathcal{E}_x$ as:

$$\mathcal{H}_y^k(\mathbf{A}, \mathbf{b}) \triangleq \operatorname{span}\big\{\mathbf{b}, (\mathbf{A}\mathbf{A}^\top)\mathbf{b}, \ldots, (\mathbf{A}\mathbf{A}^\top)^{k-1}\mathbf{b}\big\},$$

$$\mathcal{H}_x^k(\mathbf{A}, \mathbf{b}) \triangleq \operatorname{span}\big\{\mathbf{A}^\top\mathbf{b}, (\mathbf{A}^\top\mathbf{A})\mathbf{A}^\top\mathbf{b}, \ldots, (\mathbf{A}^\top\mathbf{A})^{k-1}\mathbf{A}^\top\mathbf{b}\big\} \equiv \mathcal{H}^k(\mathbf{A}, \mathbf{b}).$$

By definition, these subspaces satisfy the alternating properties:

$$\mathbf{A}\mathcal{H}_x^m(\mathbf{A}, \mathbf{b}) + \operatorname{span}\{\mathbf{b}\} = \mathcal{H}_y^{m+1}(\mathbf{A}, \mathbf{b}) \quad \text{and} \quad \mathbf{A}^\top\mathcal{H}_y^m(\mathbf{A}, \mathbf{b}) = \mathcal{H}_x^m(\mathbf{A}, \mathbf{b}) \subseteq \mathcal{H}_x^{m+1}(\mathbf{A}, \mathbf{b}).$$

Proposition 11. *Let $\mathscr{M}$ be a distributed gradient-span algorithm operating on $F(\mathbf{x}, \mathbf{y}) = \langle \mathbf{A}\mathbf{x} - \mathbf{b}, \mathbf{y}\rangle$. Let $\bar{\mathbf{x}}$ be the approximate solution generated after T communication rounds. Then, $\bar{\mathbf{x}} \in \mathcal{H}_x^{\lceil (T-1)/2 \rceil}(\mathbf{A}, \mathbf{b})$.*

Proof. Let $S_x^t \triangleq \operatorname{span}\{\nabla_x F(\mathbf{z}) \mid \mathbf{z} \in Z_x^t\} \subseteq \mathcal{E}_x$ and $S_y^t \triangleq \operatorname{span}\{-\nabla_y F(\mathbf{z}) \mid \mathbf{z} \in Z_y^t\} \subseteq \mathcal{E}_y$. We show by induction that $S_x^t \subseteq \mathcal{H}_x^{\lceil t/2 \rceil}(\mathbf{A}, \mathbf{b})$ and $S_y^t \subseteq \mathcal{H}_y^{\lfloor t/2 \rfloor + 1}(\mathbf{A}, \mathbf{b})$.

The base case $t = -1$ holds since $S_x^{-1} = \{\mathbf{0}\} \subseteq \mathcal{H}_x^0(\mathbf{A}, \mathbf{b})$ and $S_y^{-1} = \{\mathbf{0}\} \subseteq \mathcal{H}_y^0(\mathbf{A}, \mathbf{b})$.

Assume the claim holds for round $t-1$. In round t, Agent x queries points using remote variables $\hat{\mathbf{y}} \in S_y^{t-1}$. Applying the alternating properties, the evaluated gradient satisfies:

$$\nabla_x F(\mathbf{x}, \hat{\mathbf{y}}) = \mathbf{A}^\top \hat{\mathbf{y}} \in \mathbf{A}^\top \mathcal{H}_y^{\lfloor (t-1)/2 \rfloor + 1}(\mathbf{A}, \mathbf{b}) = \mathcal{H}_x^{\lfloor (t-1)/2 \rfloor + 1}(\mathbf{A}, \mathbf{b}) = \mathcal{H}_x^{\lceil t/2 \rceil}(\mathbf{A}, \mathbf{b}).$$

Thus, $S_x^t \subseteq \mathcal{H}_x^{\lceil t/2 \rceil}(\mathbf{A}, \mathbf{b})$. Similarly, Agent y queries points using remote variables $\hat{\mathbf{x}} \in S_x^{t-1}$. The evaluated gradient satisfies:

$$\begin{aligned} -\nabla_y F(\hat{\mathbf{x}}, \mathbf{y}) &= \mathbf{b} - \mathbf{A}\hat{\mathbf{x}} \in \operatorname{span}\{\mathbf{b}\} + \mathbf{A}\mathcal{H}_x^{\lceil (t-1)/2 \rceil}(\mathbf{A}, \mathbf{b}) \\ &= \mathcal{H}_y^{\lceil (t-1)/2 \rceil + 1}(\mathbf{A}, \mathbf{b}) = \mathcal{H}_y^{\lfloor t/2 \rfloor + 1}(\mathbf{A}, \mathbf{b}). \end{aligned}$$

Thus, $S_y^t \subseteq \mathcal{H}_y^{\lfloor t/2 \rfloor + 1}(\mathbf{A}, \mathbf{b})$. By induction, the claim holds for all t. The generated approximate solution satisfies $\bar{\mathbf{x}} \in S_x^{T-1} \subseteq \mathcal{H}_x^{\lceil (T-1)/2 \rceil}(\mathbf{A}, \mathbf{b})$. □

Theorem 12. *Let $\mathscr{M}$ be a distributed gradient-span algorithm for problem class $\mathscr{P}_{xy}$, where $\min\{n_x, n_y\} \geq \frac{L_{xy} D_x D_y}{2\epsilon} + 1$. Then, we have*

$$T_{\mathscr{P}_{xy}}^{\mathscr{M}} \geq \frac{L_{xy} D_x D_y}{2\epsilon} - 2.$$

Proof. Let $s = \frac{L_{xy} D_x D_y}{4\epsilon}$. If $s \leq 1$, the bound holds trivially. Otherwise, let $K = \lceil s \rceil - 2 \geq 0$, so that $K + 1 < s$. Since $n_x, n_y \geq 2s + 1 > 2K + 1$, we apply Proposition 8 to obtain $\mathbf{A}_K = \mathbf{A}(L_{xy}, K) \in \mathbb{R}^{n_y \times n_x}$ and $\mathbf{b}_K = \mathbf{b}(L_{xy}, D_x, K) \in \mathbb{R}^{n_y}$.

We set $\mathbf{A}_{xy} = \mathbf{A}_K$ and $\mathbf{b}_{xy} = \mathbf{b}_K$, and define $F_K(\mathbf{x}, \mathbf{y}) = \langle \mathbf{A}_{xy}\mathbf{x} - \mathbf{b}_{xy}, \mathbf{y} \rangle$. By Proposition 8, $\|\mathbf{A}_{xy}\| = \|\mathbf{A}_K\| \leq L_{xy}$. The linear system $\mathbf{A}_{xy}\mathbf{x} = \mathbf{b}_{xy}$ has a solution $\mathbf{x}^*$ satisfying $\|\mathbf{x}^*\| \leq D_x$. Thus, $F_K \in \mathscr{P}_{xy}$.

When $\mathscr{M}$ is applied to P_K, suppose it generates an ϵ-saddle point $(\bar{\mathbf{x}}, \bar{\mathbf{y}})$ after T communication rounds, and let $k = \lceil (T-1)/2 \rceil$. Suppose that $k \leq K$. Then, Proposition 11 implies $\bar{\mathbf{x}} \in \mathcal{H}_x^k(\mathbf{A}_{xy}, \mathbf{b}_{xy}) \subseteq \mathcal{H}^K(\mathbf{A}_K, \mathbf{b}_K)$. The restricted duality gap on P_K evaluates to:

$$\Delta_{F_K}(\bar{\mathbf{x}}, \bar{\mathbf{y}}) = D_y \|\mathbf{A}_K \bar{\mathbf{x}} - \mathbf{b}_K\|.$$

Applying Proposition 8 yields:

$$\begin{aligned} \epsilon \geq \Delta_{F_K}(\bar{\mathbf{x}}, \bar{\mathbf{y}}) &\geq D_y \min_{\mathbf{v} \in \mathcal{H}^K(\mathbf{A}_K, \mathbf{b}_K)} \|\mathbf{A}_K \mathbf{v} - \mathbf{b}_K\| \\ &\geq D_y \sqrt{\frac{3 L_{xy}^2 D_x^2}{32(K+1)^2}} \geq \frac{L_{xy} D_x D_y}{4(K+1)} = \frac{s\epsilon}{K+1} > \epsilon, \end{aligned}$$

where the fourth inequality uses $\frac{3}{32} \geq \frac{1}{16}$. This is a contradiction. Hence, $k \geq K+1 = \lceil s \rceil - 1 \geq s-1$. Because $k = \lceil (T-1)/2 \rceil \leq \frac{T}{2}$, we have $T \geq 2k \geq 2s - 2 = \frac{L_{xy} D_x D_y}{2\epsilon} - 2$. □

The oracle lower bounds can be proved using identical subspace confinement arguments. Rather than repeating the proof, we provide the main proposition and theorem here.

Proposition 13. *Let $\mathscr{M}$ be a distributed gradient-span algorithm operating on $F(\mathbf{x}, \mathbf{y}) = \langle \mathbf{A}\mathbf{x} - \mathbf{b}, \mathbf{y} \rangle$. Let $(\bar{\mathbf{x}}, \bar{\mathbf{y}})$ be the solution generated after N_x and N_y queries to $\mathscr{O}_x$ and $\mathscr{O}_y$, respectively. Then, $\bar{\mathbf{x}} \in \mathcal{H}_x^k(\mathbf{A}, \mathbf{b})$, where $k = \min(N_x, N_y)$.*

Theorem 14. *Let $\mathscr{M}$ be a distributed gradient-span algorithm for problem class $\mathscr{P}_{xy}$, where $\min\{n_x, n_y\} \geq \frac{L_{xy} D_x D_y}{2\epsilon} + 1$. Then, we have*

$$N_{\mathscr{P}_{xy}}^{\mathscr{M}} \geq (c_x + c_y)\left(\frac{L_{xy} D_x D_y}{4\epsilon} - 1\right).$$

B.3 Proof of Theorem 1

Finally, we derive the lower bound for $\mathscr{P}_{\text{SP}}$ by assembling the lower bounds obtained from the three subclasses.

Proof of Theorem 1. The general class of convex-concave saddle point problems $\mathscr{P}_{\text{SP}}$ contains the three unregularized subclasses constructed in the previous sections: the $\mathbf{x}$-quadratic subclass $\mathscr{P}_x$, the $\mathbf{y}$-quadratic subclass $\mathscr{P}_y$, and the bilinear subclass $\mathscr{P}_{xy}$. The worst-case complexity for an algorithm operating over the entire class $\mathscr{P}_{\text{SP}}$ is bounded from below by the maximum of the complexities required for these individual subclasses.

By Theorem 12, the communication complexity over the class is bounded by the communication complexity of the bilinear subclass:

$$T^{\mathscr{M}}_{\mathscr{P}_{\text{SP}}} \geq T^{\mathscr{M}}_{\mathscr{P}_{xy}} \geq \frac{L_{xy} D_x D_y}{2\epsilon} - 2.$$

For the computational complexity, we combine the independent lower bounds established in Theorem 10 and Theorem 14. The total computational complexity is bounded by the maximum of the three individual requirements:

$$N^{\mathscr{M}}_{\mathscr{P}_{\text{SP}}} \geq \max\left\{ N^{\mathscr{M}}_{\mathscr{P}_{xy}}, N^{\mathscr{M}}_{\mathscr{P}_x}, N^{\mathscr{M}}_{\mathscr{P}_y} \right\}.$$

Using the algebraic property $\max\{a,b,c\} \geq \frac{1}{3}(a+b+c)$, we obtain the lower bound:

$$\begin{aligned}
N^{\mathscr{M}}_{\mathscr{P}_{\text{SP}}} &\geq \frac{1}{3}\left[N^{\mathscr{M}}_{\mathscr{P}_{xy}} + N^{\mathscr{M}}_{\mathscr{P}_x} + N^{\mathscr{M}}_{\mathscr{P}_y} \right] \\
&\geq \frac{1}{3}\left[(c_x + c_y)\left(\frac{L_{xy} D_x D_y}{4\epsilon} - 1 \right) + c_x \left(\sqrt{\frac{3L_x D_x^2}{64\epsilon}} - 1 \right) + c_y \left(\sqrt{\frac{3L_y D_y^2}{64\epsilon}} - 1 \right) \right] \\
&= \frac{c_x + c_y}{12} \frac{L_{xy} D_x D_y}{\epsilon} + \frac{c_x}{3}\sqrt{\frac{3L_x D_x^2}{64\epsilon}} + \frac{c_y}{3}\sqrt{\frac{3L_y D_y^2}{64\epsilon}} - \frac{2c_x + 2c_y}{3}.
\end{aligned}$$

This establishes the stated lower bounds for the general problem class and concludes the proof. □

C Multi-stage reduction for distributed SPs

We first introduce the general notion of composite variational inequality problem (VIP) in Section C.1, which is the backbone of our problems. Then, in Sections C.2 to C.4, we introduce the subsequent technical components of our algorithm. Unless otherwise specified, in this section, we work with (general) finite-dimensional real vector space $\mathcal{E}$ equipped with the norm $\|\mathbf{z}\|_{\mathcal{E}} = \langle \mathbf{P}\mathbf{z}, \mathbf{z} \rangle^{\frac{1}{2}}$, where $\mathbf{P}\colon \mathcal{E} \to \mathcal{E}^*$ is a self-adjoint positive definite linear operator.

C.1 Preliminary: Variational inequality problems

For any operator $V(\cdot)\colon \operatorname{dom}\psi \to \mathcal{E}^*$ and any function $\psi(\cdot)\colon \mathcal{E} \to \mathbb{R} \cup \{+\infty\}$, we say that $\mathbf{z}^* \in \mathcal{E}$ is a *solution* of the VIP of (V, ψ) if

$$\langle V(\mathbf{z}^*), \mathbf{z} - \mathbf{z}^* \rangle + \psi(\mathbf{z}) \geq \psi(\mathbf{z}^*), \text{ for all } \mathbf{z} \in \operatorname{dom}\psi. \tag{4}$$

Assumption for VIPs. Let us introduce the following assumption:

(A1') The function ψ is a (simple) proper closed convex function. The operator V is continuous and monotone over $\operatorname{dom}\psi$: that is, $\langle V(\mathbf{z}') - V(\mathbf{z}), \mathbf{z}' - \mathbf{z} \rangle \geq 0$, for all $\mathbf{z}', \mathbf{z} \in \operatorname{dom}\psi$.

The solution of VIP defined in Eq. (4) is often called a "strong solution". There is another notion of "weak solution", but under (A1'), these two notions coincide. Meanwhile, under (A1'), a point $\mathbf{z}^* \in \mathcal{E}$ is the solution of (4) if and only if $\mathbf{0} \in V(\mathbf{z}^*) + \partial\psi(\mathbf{z}^*)$.

Indeed, for the saddle problem given by (f, ψ_x, ψ_y), let us consider the VIP given by (V^f, ψ_z), where

$$V^f(\mathbf{z}) \equiv (\nabla_x f(\mathbf{z}), -\nabla_y f(\mathbf{z})) \text{ and } \psi_z(\mathbf{z}) \equiv \psi_x(\mathbf{x}) + \psi_y(\mathbf{y}), \text{ for all } \mathbf{z} \in Q. \tag{5}$$

Moreover, for the saddle problem under (A1), the above VIP satisfies (A1'), and therefore, the solution of the VIP coincides with the saddle point.

C.2 Reduced-operator method for VIPs

Now, we introduce the Reduced-Operator Method (ROM) recently proposed in [Nesterov, 2023]. In particular, we apply this general framework in a special way so as to reduce the VIP to a sequence of Monteiro-Svaiter Subproblems (MSSs) [Monteiro and Svaiter, 2013]. The MSS asks to find a point for the regularized function such that the subgradient norm at this point is small compared to the distance from the initial point.

Monteiro-Svaiter Subproblem. Given an operator $V\colon \operatorname{dom}\psi \to \mathcal{E}^*$, a function $\psi\colon \mathcal{E} \to \mathbb{R}\cup\{+\infty\}$, a reference point $\mathbf{v}\in\operatorname{dom}\psi$, and a real number $\lambda>0$, we say $(\mathbf{z}^+, \psi'(\mathbf{z}^+))$ is a *solution* of the MSS if $\mathbf{z}^+\in\operatorname{dom}\psi$, $\psi'(\mathbf{z}^+)\in\partial\psi(\mathbf{z}^+)$, and

$$\|V(\mathbf{z}^+)+\psi'(\mathbf{z}^+)+\lambda\mathbf{P}(\mathbf{z}^+-\mathbf{v})\|_{\mathcal{E}^*}\le\lambda\|\mathbf{z}^+-\mathbf{v}\|_{\mathcal{E}}. \tag{6}$$

We will discuss how to solve the MSSs later in Section C.3. But for now, let us assume there exists a solver

$$\mathcal{M}^{\mathrm{MS}}(V,\psi,v,\lambda)$$

for the MSSs, which takes an MSS given by (V,ψ,v,λ) and returns a solution of it. Built upon such a solver $\mathcal{M}^{\mathrm{MS}}$, we now introduce ROM in Algorithm 2. At each iteration t: the solver $\mathcal{M}^{\mathrm{MS}}$ returns a solution $(\mathbf{z}^{t+1},\psi'(\mathbf{z}^{t+1}))$ for the MSS built at reference point $\mathbf{v}^t$; this solution is used as a midpoint to compute subgradient $V_\psi(\mathbf{z}^{t+1})$; then the 'extragradient-like' step is taken with the stepsize a_{t+1}.

Algorithm 2 $\mathrm{ROM}_{\|\cdot\|_{\mathcal{E}}}(V,\psi,\mathbf{z}^0,(\lambda_t)_{t\ge1}\mid\mathcal{M}^{\mathrm{MS}})$

Require: A solver $\mathcal{M}^{\mathrm{MS}}$ for the MSSs.

$\mathbf{v}^0=\mathbf{z}^0$.
for $t=0,1,\cdots$ **do**
  $(\mathbf{z}^{t+1},\psi'(\mathbf{z}^{t+1}))=\mathcal{M}^{\mathrm{MS}}(V,\psi,\mathbf{v}^t,\lambda_{t+1})$.
  $V_\psi(\mathbf{z}^{t+1})=V(\mathbf{z}^{t+1})+\psi'(\mathbf{z}^{t+1})$.
  $a_{t+1}=\frac{2\langle V_\psi(\mathbf{z}^{t+1}),\mathbf{v}^t-\mathbf{z}^{t+1}\rangle}{\|V_\psi(\mathbf{z}^{t+1})\|^2_{\mathcal{E}^*}}$.
  $\mathbf{v}^{t+1}=\arg\min_{\mathbf{v}\in\operatorname{dom}\psi}\left[a_{t+1}\langle V_\psi(\mathbf{z}^{t+1}),\mathbf{v}\rangle+\frac12\|\mathbf{v}-\mathbf{v}^t\|^2_{\mathcal{E}}\right]$.
end for

Next, let us show the convergence of ROM.

Lemma 15. ROM *(Algorithm 2) ensures for all* $\mathbf{v}\in\operatorname{dom}\psi$ *and for all* $T\ge1$,

$$\sum_{t=0}^{T-1}a_{t+1}\langle V_\psi(\mathbf{z}^{t+1}),\mathbf{z}^{t+1}-\mathbf{v}\rangle\le\frac12\|\mathbf{v}^0-\mathbf{v}\|^2_{\mathcal{E}}-\frac12\|\mathbf{v}^T-\mathbf{v}\|^2_{\mathcal{E}}.$$

Moreover, we have $a_{t+1}\ge\frac{1}{\lambda_{t+1}}$, *for all* $t\ge0$.

Proof. By the optimality of $\mathbf{v}^{t+1}$, we have for all $\mathbf{v}\in\operatorname{dom}\psi$,

$$a_{t+1}\langle V_\psi(\mathbf{z}^{t+1}),\mathbf{v}-\mathbf{v}^{t+1}\rangle+\frac12\|\mathbf{v}^t-\mathbf{v}\|^2_{\mathcal{E}}\ge\frac12\|\mathbf{v}^{t+1}-\mathbf{v}\|^2_{\mathcal{E}}+\frac12\|\mathbf{v}^t-\mathbf{v}^{t+1}\|^2_{\mathcal{E}},$$

and therefore,

$$\begin{aligned}
&a_{t+1}\langle V_\psi(\mathbf{z}^{t+1}),\mathbf{v}-\mathbf{z}^{t+1}\rangle+\frac12\|\mathbf{v}^t-\mathbf{v}\|^2_{\mathcal{E}}\\
&\ge a_{t+1}\langle V_\psi(\mathbf{z}^{t+1}),\mathbf{v}^{t+1}-\mathbf{z}^{t+1}\rangle+\frac12\|\mathbf{v}^{t+1}-\mathbf{v}\|^2_{\mathcal{E}}+\frac12\|\mathbf{v}^t-\mathbf{v}^{t+1}\|^2_{\mathcal{E}}\\
&=a_{t+1}\langle V_\psi(\mathbf{z}^{t+1}),\mathbf{v}^t-\mathbf{z}^{t+1}\rangle+\frac12\|\mathbf{v}^{t+1}-\mathbf{v}\|^2_{\mathcal{E}}\\
&\quad+a_{t+1}\langle V_\psi(\mathbf{z}^{t+1}),\mathbf{v}^{t+1}-\mathbf{v}^t\rangle+\frac12\|\mathbf{v}^t-\mathbf{v}^{t+1}\|^2_{\mathcal{E}}\\
&\ge a_{t+1}\langle V_\psi(\mathbf{z}^{t+1}),\mathbf{v}^t-\mathbf{z}^{t+1}\rangle+\frac12\|\mathbf{v}^{t+1}-\mathbf{v}\|^2_{\mathcal{E}}-\frac{a^2_{t+1}}{2}\|V_\psi(\mathbf{z}^{t+1})\|^2_{\mathcal{E}^*}\\
&\ge\frac12\|\mathbf{v}^{t+1}-\mathbf{v}\|^2_{\mathcal{E}},
\end{aligned}$$

where the last inequality follows from the assignment of a_{t+1} in Line 5 of Algorithm 2. Then, the desired bound follows from summing the above inequality over t from 0 to $T-1$.

Next, we show the lower bound for a_t. For all $t \geq 1$, we have

$$\begin{aligned}
&\langle V_\psi(\mathbf{z}^t), \mathbf{v}^{t-1} - \mathbf{z}^t \rangle - \frac{1}{2\lambda_t} \|V_\psi(\mathbf{z}^t)\|_{\mathcal{E}^*}^2 \\
\equiv\ &\frac{\lambda_t}{2} \|\mathbf{z}^t - \mathbf{v}^{t-1}\|_{\mathcal{E}}^2 - \frac{1}{2\lambda_t} \|V_\psi(\mathbf{z}^t) + \lambda_t \mathbf{P}(\mathbf{z}^t - \mathbf{v}^{t-1})\|_{\mathcal{E}^*}^2 \\
\geq\ &0,
\end{aligned}$$

where the last inequality follows from Eq. (6). Therefore, we have

$$a_t = \frac{2\langle V_\psi(\mathbf{z}^t), \mathbf{v}^t - \mathbf{z}^t \rangle}{\|V_\psi(\mathbf{z}^t)\|_{\mathcal{E}^*}^2} \geq \frac{1}{\lambda_t}.$$

□

C.3 Fully decoupled solver for MSSs with weak couplings

We now address the MSSs introduced in Section C.2. Specifically, we focus on the subproblems arising from applying ROM to the saddle problem of (f, ψ_x, ψ_y). Therefore, we consider the MSS given by

$$(V^f, \psi_z, \mathbf{v}, \lambda), \tag{7}$$

where V^f and ψ_z are defined in Eq. (5), the reference point $\mathbf{v} = (\mathbf{v}_x, \mathbf{v}_y) \in \operatorname{dom} \psi_x \times \operatorname{dom} \psi_y$, and the assembled norm $\|\cdot\|_{\mathcal{E}}$ is associated with parameters α_x and α_y.

We say that an MSS has a *weak coupling* if $\lambda \geq \boxed{2\bar{L}_{\mathsf{c}} \triangleq \frac{2L_{xy}}{\sqrt{\alpha_x \alpha_y}}}$. In this section, we will introduce a Fully Decoupled Solver (FDS), which reduces the weakly-coupled MSSs to coordinate-wise Minimization of Residual Norms (MRNs).

Let us first define the problem of MRN. The MRN asks to find an approximate solution $\mathbf{w}^+$ of the VIP of (V_w, ψ_w) with the accuracy specified as follows:

Minimization of residual norm. Given an operator $V_w \colon \operatorname{dom} \psi_w \to \mathcal{E}_w^*$, a function $\psi_w \colon \mathcal{E}_w \to \mathbb{R} \cup \{+\infty\}$, a reference point $\mathbf{v}_w \in \operatorname{dom} \psi_w$, and an accuracy $\delta > 0$, we say $(\mathbf{w}^+, \psi_w'(\mathbf{w}^+))$ minimizes the residual norm to δ-*relative distance accuracy*, if $\mathbf{w}^+ \in \operatorname{dom} \psi_w$, $\psi_w'(\mathbf{w}^+) \in \partial\psi_w(\mathbf{w}^+)$, and

$$\|V_w(\mathbf{w}^+) + \psi_w'(\mathbf{w}^+)\|_{w^*} \leq \delta \|\mathbf{w}^+ - \mathbf{v}_w\|_w.$$

Let us, again, defer the discussion of solving MRNs to Section C.4. But for now, let us assume there exist solvers

$$\mathcal{M}_x^{\text{MRN}}(V_x, \hat{\psi}_x, \mathbf{v}_x, \delta_x) \quad \text{and} \quad \mathcal{M}_y^{\text{MRN}}(V_y, \hat{\psi}_y, \mathbf{v}_y, \delta_y)$$

for the coordinate-wise MRNs in spaces $\mathcal{E}_x$ and $\mathcal{E}_y$. In particular, these solvers take a coordinate-wise MRN problem and returns a point and a subgradient satisfying the desired accuracy.

Crucially, a key result in our proof is that, for an MSS with a weak coupling, we apply Algorithm 3—Fully Decoupled Solver (FDS)—that solves for each decision variable separately. We show in Lemma 16 that FDS returns the correct solution *in one round*.

Lemma 16. *Consider the saddle problem given by (f, ψ_x, ψ_y) which satisfies (A3). For $\lambda \geq 2\bar{L}_{\mathsf{c}}$, FDS (Algorithm 3) returns a solution of the MSS introduced in Eq. (7).*

The correctness of the FDS for SPs can be implied as a direct consequence of the correctness of FDS for the (more general) VIPs proven in Section E. We will get to this result later in Lemma 22.

C.4 Minimization of residual norms

We arrive at the last building component, the Minimization of Residual Norms (MRNs).

Algorithm 3 $\mathrm{FDS}_{\|\cdot\|_\mathcal{E}}((\nabla_x f, -\nabla_y f), (\psi_x, \psi_y), \mathbf{v}, \lambda \mid (\mathcal{M}_x^{\mathrm{MRN}}, \mathcal{M}_y^{\mathrm{MRN}}))$

Require: Solvers $\mathcal{M}_x^{\mathrm{MRN}}$ and $\mathcal{M}_y^{\mathrm{MRN}}$ for the coordinate-wise MRNs.

1: $\hat{\psi}_x = \psi_x + \frac{\alpha_x \lambda}{2}\|\cdot - \mathbf{v}_x\|_x^2$ and $\hat{\psi}_y = \psi_y + \frac{\alpha_y \lambda}{2}\|\cdot - \mathbf{v}_y\|_y^2$.

2: Agent x and Agent y respectively compute

$$(\mathbf{x}^+, \hat{\psi}_x'(\mathbf{x}^+)) = \mathcal{M}_x^{\mathrm{MRN}}\big(\nabla_x f(\cdot, \mathbf{v}_y), \hat{\psi}_x, \mathbf{v}_x, \delta_x\big) \text{ and}$$
$$(\mathbf{y}^+, \hat{\psi}_y'(\mathbf{y}^+)) = \mathcal{M}_y^{\mathrm{MRN}}\big(-\nabla_y f(\mathbf{v}_x, \cdot), \hat{\psi}_y, \mathbf{v}_y, \delta_y\big),$$

where $\delta_x = \frac{\alpha_x \lambda}{2}$ and $\delta_y = \frac{\alpha_y \lambda}{2}$.

3: $\psi_x'(\mathbf{x}^+) = \hat{\psi}_x'(\mathbf{x}^+) - \alpha_x \lambda \mathbf{P}_x(\mathbf{x}^+ - \mathbf{v}_x)$ and $\psi_y'(\mathbf{y}^+) = \hat{\psi}_y'(\mathbf{y}^+) - \alpha_y \lambda \mathbf{P}_y(\mathbf{y}^+ - \mathbf{v}_y)$.

4: **return** $(\mathbf{z}^+, \psi'(\mathbf{z}^+))$, where $\mathbf{z}^+ = (\mathbf{x}^+, \mathbf{y}^+)$ and $\psi'(\mathbf{z}^+) = (\psi_x'(\mathbf{x}^+), \psi_y'(\mathbf{y}^+))$.

Assumptions for MRNs. Let us introduce the following assumptions:

(Â1) The function ψ_w is a (simple) proper closed convex function. The operator V_w is monotone over $\operatorname{dom}\psi_w$.

(Â2) The set-valued operator $V_w + \partial\psi_w$ is μ-strongly maximally monotone over $\operatorname{dom}\psi_w$. That is,

$$\big\langle V_w(\mathbf{w}') + \psi_w'(\mathbf{w}') - V_w(\mathbf{w}) - \psi_w'(\mathbf{w}), \mathbf{w}' - \mathbf{w}\big\rangle \geq \mu\|\mathbf{w}' - \mathbf{w}\|_w^2,$$

for all $\mathbf{w}', \mathbf{w} \in \operatorname{dom}\psi_w$, $\psi_w'(\mathbf{w}') \in \partial\psi_w(\mathbf{w}')$, and $\psi_w'(\mathbf{w}) \in \partial\psi_w(\mathbf{w})$,

(Â3) The operator $V_w(\mathbf{w})$ is L-Lipschitz continuous over $\mathbf{w} \in \operatorname{dom}\psi_w$.

(Â4) The operator $V_w = \nabla f_w$, where f_w is a continuously differentiable function defined on an open set containing $\operatorname{dom}\psi_w$.

The theoretical guarantee provided in the literature is usually based on the distance-to-solution accuracy (cf. Definition 5). We show in Lemma 17 that, under strong maximal monotonicity, the relative distance accuracy required in this paper can be implied from distance-to-solution accuracy.

Definition 5 (Distance-to-solution accuracy)**.** We say that $(\mathbf{w}^+, \psi_w'(\mathbf{w}^+))$ satisfies *ξ-distance-to-solution accuracy* if $\mathbf{w}^+ \in \operatorname{dom}\psi_w$, $\psi_w'(\mathbf{w}^+) \in \partial\psi_w(\mathbf{w}^+)$, and $\|V_w(\mathbf{w}^+) + \psi_w'(\mathbf{w}^+)\|_{w^*} \leq \xi\|\mathbf{v}_w - \tilde{\mathbf{w}}\|_w$ for some $\tilde{\mathbf{w}}$ in the solution set of the VIP of (V_w, ψ_w).

Lemma 17. *Consider the MRN problem given by $(V_w, \psi_w, \mathbf{v}_w, \delta)$ which satisfies (Â2). Let $\xi \leq \frac{\mu\delta}{\mu+\delta}$. If $(\mathbf{w}^+, \psi_w'(\mathbf{w}^+))$ satisfies ξ-distance-to-solution accuracy, then $(\mathbf{w}^+, \psi_w'(\mathbf{w}^+))$ is a solution of the MRN problem.*

Proof. In view of the triangle inequality and the μ-strong maximal monotonicity of $V_w + \psi_w$, we have

$$\begin{aligned}\|\mathbf{v}_w - \tilde{\mathbf{w}}\|_w &\leq \|\mathbf{w}^+ - \mathbf{v}_w\|_w + \|\mathbf{w}^+ - \tilde{\mathbf{w}}\|_w \\ &\leq \|\mathbf{w}^+ - \mathbf{v}_w\|_w + \frac{1}{\mu}\|V_w(\mathbf{w}^+) + \psi_w'(\mathbf{w}^+)\|_{w^*} \\ &\leq \|\mathbf{w}^+ - \mathbf{v}_w\|_w + \frac{\xi}{\mu}\|\mathbf{v}_w - \tilde{\mathbf{w}}\|_w.\end{aligned}$$

Then, we have

$$\|\mathbf{v}_w - \tilde{\mathbf{w}}\|_w \leq \frac{\mu}{\mu - \xi}\|\mathbf{w}^+ - \mathbf{v}_w\|_w.$$

Therefore, we have

$$\|V_w(\mathbf{w}^+) + \psi_w'(\mathbf{w}^+)\|_{w^*} \leq \xi\|\mathbf{v}_w - \tilde{\mathbf{w}}\|_w \leq \frac{\mu\xi}{\mu - \xi}\|\mathbf{w}^+ - \mathbf{v}_w\|_w \leq \delta,$$

where the last inequality follows from the assignment $\xi \leq \frac{\mu\delta}{\mu+\delta}$. □

We will leverage efficient existing solvers for the MRN problems. In particular, we are to deal with the specific MRNs given in Line 2 in Algorithm 3, where the corresponding coordinate-wise operators are gradients of smooth convex functions. Therefore, we can leverage the existing accelerated gradient methods from the literature.

Let us apply, for instance, the Accumulative Regularization Method (ARM) from Lan et al. [2023], whose detailed pseudocode is presented in Algorithm 4 for self-consistency. Let us denote this algorithm as

$$\text{ARM}\ (\nabla f_w, \psi_w, \mathbf{v}_w, \xi \mid L), \tag{8}$$

which takes an MRN instance of interest, has knowledge of the parameter L in (Â3), and returns a solution that satisfies ξ-distance-to-solution accuracy.

Algorithm 4 ARM$\big(\nabla f_w, \psi_w, \mathbf{v}_w, \xi \mid L\big)$

Set $\tau = 2 + \max\big\{0, \big\lceil \log_4\big(\frac{3L}{2\xi}\big)\big\rceil\big\}$ and $\sigma^{(0)} = 0$.
Set $\sigma^{(k)} = 4^{k-3}\frac{2\xi}{3}$ and $N_k = \left\lceil 16\sqrt{\frac{L}{\sigma^{(k)}}}\right\rceil$ for $k = 1, \dots, \tau$.
Initialize $\bar{\mathbf{w}}^{(0)} = \mathbf{w}^{(0)} = \mathbf{v}_w$.
for $k = 1, \dots, \tau$ **do**
  $\gamma^{(k)} = 1 - \frac{\sigma^{(k-1)}}{\sigma^{(k)}}$
  $\bar{\mathbf{w}}^{(k)} = (1-\gamma^{(k)})\bar{\mathbf{w}}^{(k-1)} + \gamma^{(k)}\mathbf{w}^{(k-1)}$
  % Begin Inner Subroutine: Nesterov's Accelerated Gradient Method
  Initialize $\mathbf{x}_k^{(0)} = \mathbf{w}^{(k-1)}$, $\mathbf{y}_k^{(0)} = \mathbf{w}^{(k-1)}$, $t_0 = 1$, and $L_k = L + \sigma^{(k)}$.
  for $i = 0, \dots, N_k - 1$ **do**
    $\nabla f_w^{(k)}(\mathbf{y}_k^{(i)}) = \nabla f_w(\mathbf{y}_k^{(i)}) + \sigma^{(k)}(\mathbf{y}_k^{(i)} - \bar{\mathbf{w}}^{(k)})$
    $\mathbf{x}_k^{(i+1)} = \arg\min_{\mathbf{w}\in w}\big\{\langle \nabla f_w^{(k)}(\mathbf{y}_k^{(i)}), \mathbf{w}\rangle + \frac{L_k}{2}\|\mathbf{w} - \mathbf{y}_k^{(i)}\|_w^2 + \psi_w(\mathbf{w})\big\}$
    $t_{i+1} = \frac{1+\sqrt{1+4t_i^2}}{2}$
    $\mathbf{y}_k^{(i+1)} = \mathbf{x}_k^{(i+1)} + \frac{t_i - 1}{t_{i+1}}(\mathbf{x}_k^{(i+1)} - \mathbf{x}_k^{(i)})$
  end for
  % End Inner Subroutine
  $\mathbf{w}^{(k)} = \mathbf{x}_k^{(N_k)}$
  $\psi_w'(\mathbf{w}^{(k)}) = -\nabla f_w^{(k)}(\mathbf{y}_k^{(N_k-1)}) - L_k(\mathbf{x}_k^{(N_k)} - \mathbf{y}_k^{(N_k-1)})$
end for
return $(\mathbf{w}^{(\tau)}, \psi_w'(\mathbf{w}^{(\tau)}))$

Now, we state the gradient query complexity with distance-to-solution accuracy. The original result of Lan et al. [2023] is given in projected gradient norm, which can be converted to the subgradient norm considered in this paper.

Lemma 18 (Lan et al. 2023)**.** *Assume (Â1), (Â3), (Â4), and that the solution set of the VIP of (V_w, ψ_w) is non-empty. Let*

$$(\mathbf{w}^+, \psi_w'(\mathbf{w}^+)) = \text{ARM}\ (\nabla f_w, \psi_w, \mathbf{v}_w, \xi \mid L).$$

Then, ARM *takes no more than $34\cdot\sqrt{\frac{3L}{2\xi}}$ queries to $\nabla f_w(\cdot)$ and ensures that $(\mathbf{w}^+, \psi_w'(\mathbf{w}^+))$ satisfies ξ-distance-to-solution accuracy.*

Proof. It was originally shown in [Lan et al., 2023, Theorem 3.1] that ARM takes no more than $34\sqrt{\frac{3L}{2\xi}}$ gradient queries and obtains $\mathbf{w}' \in \operatorname{dom}\psi_w$, such that there exists $\tilde{\mathbf{w}} \in \operatorname{dom}\psi_w$ with $\nabla f_w(\tilde{\mathbf{w}}) \in -\partial\psi_w(\tilde{\mathbf{w}})$, and

$$\|2L(\mathbf{w}^+ - \mathbf{w}')\|_w \le \frac{2}{3}\xi\|\mathbf{v}_w - \tilde{\mathbf{w}}\|_w,$$

where

$$\mathbf{w}^+ = \operatorname*{arg\,min}_{\mathbf{w}\in\operatorname{dom}\psi_w}\big[\langle\nabla f_w(\mathbf{w}'), \mathbf{w}\rangle + \psi_w(\mathbf{w}) + L\|\mathbf{w} - \mathbf{w}'\|_w^2\big].$$

By the optimality of $\mathbf{w}^+$, there exists $\psi_w'(\mathbf{w}^+) \in \partial\psi_w(\mathbf{w}^+)$ such that

$$\nabla f_w(\mathbf{w}') + \psi_w'(\mathbf{w}^+) + 2L\mathbf{P}(\mathbf{w}^+ - \mathbf{w}') = \mathbf{0}.$$

Then, we have

$$\begin{aligned}
&\|\nabla f_w(\mathbf{w}^+) + \psi'_w(\mathbf{w}^+)\|_{w^*} \\
&\le \|\nabla f_w(\mathbf{w}') + \psi'_w(\mathbf{w}^+)\|_{w^*} + \|\nabla f_w(\mathbf{w}^+) - \nabla f_w(\mathbf{w}')\|_{w^*} \\
&= \|2L\mathbf{P}(\mathbf{w}^+ - \mathbf{w}')\|_{w^*} + \|\nabla f_w(\mathbf{w}^+) - \nabla f_w(\mathbf{w}')\|_{w^*} \\
&\le 3L\|\mathbf{w}^+ - \mathbf{w}'\|_w \\
&\le \xi\|\mathbf{v}_w - \tilde{\mathbf{w}}\|_w.
\end{aligned}$$

□

D Decoupled method for saddle problems: Concrete implementation

We are now back to considering the SPs in Eq. (1). Let us combine the technical components in Section C and present the final, implementable algorithm.

Implementation of DM-SP. We use the ARM in Lemma 18 for Minimization of Residual Norms:

$$\begin{aligned}
\mathcal{M}_x^{\text{MRN}}(V_x, \hat{\psi}_x, \mathbf{v}_x, \delta_x) &\triangleq \text{ARM}\big(V_x, \hat{\psi}_x, \mathbf{v}_x, \frac{2\delta_x}{3} \mid L_x\big), \\
\mathcal{M}_y^{\text{MRN}}(V_y, \hat{\psi}_y, \mathbf{v}_y, \delta_y) &\triangleq \text{ARM}\big(V_y, \hat{\psi}_y, \mathbf{v}_y, \frac{2\delta_y}{3} \mid L_y\big).
\end{aligned} \tag{9}$$

Consider the assembled norm $\|\cdot\|_{\mathcal{E}}$ with parameters α_x and α_y. Then, for any Monteiro-Svaiter Subproblem given by $(V^f, \psi_z, \mathbf{v}, \lambda)$, we leverage the solver

$$\text{FDS-ARM}(V^f, \psi_z, \mathbf{v}, \lambda) = \text{FDS}_{\|\cdot\|_{\mathcal{E}}}\big(V^f, \psi_z, \mathbf{v}, \lambda \mid (\mathcal{M}_x^{\text{MRN}}, \mathcal{M}_y^{\text{MRN}})\big)$$

Finally, we obtain the concrete algorithm DM-SP as follows:

$$\boxed{\text{ROM}_{\|\cdot\|_{\mathcal{E}}}\big(V^f, \psi_z, \mathbf{z}^0, (\lambda_t)_{t\ge 1} \mid \text{FDS-ARM}\big).} \tag{10}$$

Combining Lemmas 15 to 18, we obtain the following result on the computational cost.

Let us prove the main convergence lemma for SPs in Lemma 19.

Lemma 19. *Under (A1) to (A3), for* $\lambda_{t+1} \equiv \lambda \ge \frac{2L_{xy}}{\sqrt{\alpha_x \alpha_y}}$, DM-SP *with the implementation in Eq. (10) takes no more than* $2T$ *communication rounds, no more than*

$$T \cdot \big(1 + 34\sqrt{\frac{9L_x}{2\alpha_x\lambda}}\big)$$

queries to $\mathcal{O}_x$*, and no more than*

$$T \cdot \big(1 + 34\sqrt{\frac{9L_y}{2\alpha_y\lambda}}\big)$$

queries to $\mathcal{O}_y$*, and obtains an* ϵ*-saddle point* $\bar{\mathbf{z}}^T$*, where*

$$T = \Big\lceil \frac{\alpha_x\lambda D_x^2 + \alpha_y\lambda D_y^2}{2\epsilon} \Big\rceil.$$

Proof. By (A1), we have

$$\Delta(\bar{\mathbf{z}}^T) \le \Big(\sum_{t=0}^{T-1} a_{t+1}\Big)^{-1} \max_{\mathbf{z}\in\mathcal{B}\cap Q} \Big[\sum_{t=0}^{T-1} a_{t+1}\langle V_\psi(\mathbf{z}^{t+1}), \mathbf{z}^{t+1} - \mathbf{z}\rangle\Big].$$

Further, with $\lambda \geq 2\bar{L}_{\mathrm{c}}$, by Lemmas 15 and 16, we have

$$\begin{aligned}\Delta(\bar{\mathbf{z}}^T) &\leq \Big(\sum_{t=0}^{T-1} a_{t+1}\Big)^{-1} \max_{\mathbf{z}\in\mathcal{B}\cap Q} \Big[\sum_{t=0}^{T-1} a_{t+1}\langle V_\psi(\mathbf{z}^{t+1}), \mathbf{z}^{t+1}-\mathbf{z}\rangle\Big] \\ &\leq \Big(\sum_{t=0}^{T-1} a_{t+1}\Big)^{-1} \max_{\mathbf{z}\in\mathcal{B}\cap Q} \Big[\frac{\alpha_x}{2}\|\mathbf{x}^0-\mathbf{x}\|_x^2 + \frac{\alpha_y}{2}\|\mathbf{y}^0-\mathbf{y}\|_y^2\Big] \\ &\leq \Big(\sum_{t=0}^{T-1} \frac{1}{\lambda_{t+1}}\Big)^{-1} \cdot \frac{1}{2}(\alpha_x D_x^2 + \alpha_y D_y^2) \leq \epsilon,\end{aligned}$$

where the last inequality follows from the assignments of $(\lambda_t)_{t\geq 1}$ and T. Therefore, the number of communication rounds is bounded by $2T$.

Now we count the number of gradient queries. By Lemma 17, ARM always returns the solution with the required relative distance accuracy; and in view of Lemma 18, it takes no more than $34\sqrt{\frac{3L_x}{\alpha_x\lambda}}$ gradient queries to $\nabla_x f$ and no more than $34\sqrt{\frac{3L_y}{\alpha_y\lambda}}$ gradient queries to $\nabla_y f$. Therefore, the numbers of gradient queries to $\nabla_x f$ and $\nabla_y f$ are bounded by $T\cdot\big(1+34\sqrt{\frac{3L_x}{\alpha_x\lambda}}\big)$ and $T\cdot\big(1+34\sqrt{\frac{3L_y}{\alpha_y\lambda}}\big)$, respectively. □

The main convergence result for SPs in Theorems 2 and 3 can be directly implied from Lemma 19.

E Monotone composite variational inequality problems

In this section, we study variational inequality problems (VIPs) [Nemirovski, 2004, Juditsky et al., 2011], a generalization of SPs that captures, for instance, multiplayer general-sum games.

E.1 Problem formulation

VIPs (with separable composite terms). Let us consider the VIP in Eq. (4), where $\mathcal{E} = \mathcal{E}_1 \times \cdots \times \mathcal{E}_K$ is the direct product of K finite-dimensional real vector spaces. For all $i \in [K]$: let the mapping $V_i\colon \operatorname{dom}\psi \to \mathcal{E}_i^*$, and let the function $\psi_i\colon \mathcal{E}_i \to \mathbb{R}\cup\{+\infty\}$. We consider the decomposition of $V(\mathbf{z}) = \big(V_1(\mathbf{z}), \cdots, V_K(\mathbf{z})\big)$ and $\psi(\mathbf{z}) = \psi_1(\mathbf{z}_1) + \cdots + \psi_K(\mathbf{z}_K)$, for all $\mathbf{z} = (\mathbf{z}_1, \cdots, \mathbf{z}_K) \in \mathcal{E}$. Moreover, we denote $\operatorname{dom}\psi = \operatorname{dom}\psi_1 \times \cdots \times \operatorname{dom}\psi_K \triangleq Q$.

Assumptions for VIPs. Let us make the following assumptions:

(A2') Let $\mathbf{z}^0 = (\mathbf{z}_1^0, \cdots, \mathbf{z}_K^0) \in Q$ be a given point. There exists $\mathbf{z}^* = (\mathbf{z}_1^*, \cdots, \mathbf{z}_K^*) \in Q$ in the solution set of the VIP of (V, ψ), such that for all $i \in [K]$: $\mathbf{z}_i^* \in \mathcal{B}_i$, where $\mathcal{B}_i \triangleq \{\mathbf{z}_i \in \mathcal{E}_i \mid \|\mathbf{z}_i^0 - \mathbf{z}_i\|_i \leq D_i\}$ and $D_i > 0$ is a given distance.

(A3') The operator $V_i(\mathbf{z}_j; \mathbf{z}_{-j})$ is L_{ij}-Lipschitz continuous in $\mathbf{z}_j \in \operatorname{dom}\psi_j$ for any fixed $\mathbf{z}_{-j} \in \operatorname{dom}\psi_1 \times \cdots \times \operatorname{dom}\psi_{j-1} \times \operatorname{dom}\psi_{j+1} \times \cdots \times \operatorname{dom}\psi_K$. [2]

Let the operator family $\mathscr{V}_{\mathrm{VIP}}$ be comprised of all the operators with initialization points $\big((V_i)_{i\in[K]}, \mathbf{z}^0\big) \in \mathscr{P}_{\mathrm{VIP}}$, such that Assumptions (Â1) to (Â3) are satisfied.

Notations. To simplify the notations, let us denote $\mathbf{z} \triangleq (\mathbf{z}_1, \cdots, \mathbf{z}_K) \in Q$ in the context of VIPs. Let us denote

$$\bar{L}_{ij} \triangleq \max\{L_{ij}, L_{ji}\},\ A_i \triangleq D_i\Big(\sum_{j\in[K]\setminus\{i\}} \bar{L}_{ij} D_j\Big),\ \text{and } B_i \triangleq \bar{L}_{ii} D_i^2,\ \text{for all } i, j \in [K].$$

We refer to $\sum_i A_i$ as the diagonal conditioning, and we say that the diagonal conditioning dominates when $\sum_i A_i \gg \sum_i B_i$.

[2] For all $\mathbf{z} = (\mathbf{z}_1, \cdots, \mathbf{z}_K) \in Q$, we use the following notations for simplicity: $(\mathbf{z}_j; \mathbf{z}_{-j}) \triangleq \mathbf{z}$ and $\mathbf{z}_{-j} \triangleq (\mathbf{z}_1, \cdots, \mathbf{z}_{j-1}, \mathbf{z}_{j+1}, \cdots, \mathbf{z}_K)$.

E.2 Communication and computational costs

Distributed oracles. We consider a distributed setting with K agents very similar to the one in Sections 2.1 and 2.2. For all $i \in [K]$: Agent i controls decision variable $\mathbf{z}_i \in \operatorname{dom} \psi_i$, has direct access to the function ψ_i, and has access to the oracle $\mathscr{O}_i(\mathbf{z}) = V_i(\mathbf{z})$ for $\mathbf{z} \in Q$. We consider per query to $\mathscr{O}_i$ costs $c_i \geq 0$, $i \in [K]$.

Accuracy measure. We consider the following accuracy measure for VIPs:

$$\Delta(\bar{\mathbf{z}}) \triangleq \sup_{\mathbf{z} \in \mathcal{B} \cap Q} \langle V(\mathbf{z}), \bar{\mathbf{z}} - \mathbf{z} \rangle + \psi(\bar{\mathbf{z}}) - \psi(\mathbf{z}), \text{ for all } \mathbf{z} \in Q,$$

where $\mathcal{B} \triangleq \mathcal{B}_1 \times \cdots \times \mathcal{B}_K$. We say that a point $\bar{\mathbf{z}} \in Q$ is an *ϵ-approximate solution* of the VIP if $\Delta(\bar{\mathbf{z}}) \leq \epsilon$. Our goal is to find such an ϵ-approximate solution for any $\epsilon > 0$.

Problem class. We formally define the overall *problem class*, denoted by $\mathscr{P}_{\text{VIP}}\big(\mathcal{V}_{\text{VIP}}, (\mathscr{O}_i)_{i\in[K]}, \epsilon\big)$, or for short $\mathscr{P}_{\text{VIP}}$. A specific problem instance $P \in \mathscr{P}_{\text{VIP}}$ is constructed by drawing an operator instance (with initial point) V from the operator family $\mathscr{F}$, equipping it with the specific distributed oracles $((\mathscr{O}_i)_{i\in[K]})$, and specifying a target accuracy $\epsilon > 0$. Solving the instance P requires an algorithm to output an ϵ-approximate solution of V utilizing the distributed oracles.

Distributed algorithms for VIPs, communication and oracle costs. To provide the formal definitions of distributed algorithms for the problem class $\mathscr{P}_{\text{VIP}}$, we generalize the information-based framework to K agents. Every query point, message, and output is generated as a deterministic mapping of the information available to the agent at that step.

Suppose an algorithm $\mathscr{M}$ proceeds in T rounds. In each round $t \in \{0, \ldots, T-1\}$, each Agent $i \in [K]$ executes multiple local computational steps to generate local query points denoted by

$$\mathbf{z}_i^{t,l} = (\mathbf{z}_{i,1}^{t,l}, \ldots, \mathbf{z}_{i,i}^{t,l}, \ldots, \mathbf{z}_{i,K}^{t,l}) \quad \text{for } l \in \{0, \ldots, \tau_i^t - 1\},$$

where $\mathbf{z}_{i,i}^{t,l}$ is the local variable updated by Agent i, and $\mathbf{z}_{i,j}^{t,l}$ (for $j \neq i$) is the delayed approximation of Agent j's variable utilized by Agent i.

Let $I_i^{t,l}$ denote the accumulated information sequence available to Agent i prior to making its $(l+1)$-th local oracle query in round t. The base case at initialization is $I_i^{0,0} = (\psi_i, \mathbf{z}^0)$. During the local computational steps $l \in \{0, \ldots, \tau_i^t - 1\}$, the information sequence of Agent i updates sequentially by appending the newly acquired oracle response for V_i:

$$I_i^{t,l+1} = \big(I_i^{t,l}, V_i(\mathbf{z}_i^{t,l})\big).$$

After the local steps in round t, the agents exchange messages. Let $M_{j\to i}^t$ denote the message sent from Agent j to Agent i. The information sequence available to Agent i at the beginning of round $t+1$ appends the received messages to its prior local history:

$$I_i^{t+1,0} = \big(I_i^{t,\tau_i^t}, \{M_{j\to i}^t\}_{j\neq i}\big).$$

Definition 6. An algorithm $\mathscr{M}$ is called a *distributed algorithm* for problem class $\mathscr{P}_{\text{VIP}}$ if, when applied to any instance $P \in \mathscr{P}_{\text{VIP}}$, its execution satisfies the following conditions for all $t \in \{0, \ldots, T-1\}$ and agents $i \in [K]$:

1. **Local Computation:** The query points are determined entirely by the locally available information. For all local steps $l \in \{0, \ldots, \tau_i^t - 1\}$, there exists a deterministic mapping function $\mathcal{A}_i^{t,l}$ such that:
$$\mathbf{z}_i^{t,l} = \mathcal{A}_i^{t,l}\big(I_i^{t,l}\big).$$
2. **Communication:** The messages exchanged are produced by deterministic mappings of the sender's local information. For any $j \neq i$, there exists a mapping function $\mathcal{M}_{i\to j}^t$ such that:
$$M_{i\to j}^t = \mathcal{M}_{i\to j}^t\big(I_i^{t,\tau_i^t}\big).$$

3. **Generate Solution:** The candidate output solutions $\bar{\mathbf{z}}^{t+1} = (\bar{\mathbf{z}}_1^{t+1}, \ldots, \bar{\mathbf{z}}_K^{t+1})$ are constructed from the respective agents' updated information sets. There exists a deterministic mapping function $\bar{\mathcal{A}}_i^{t+1}$ such that:
$$\bar{\mathbf{z}}_i^{t+1} = \bar{\mathcal{A}}_i^{t+1}\big(I_i^{t+1,0}\big).$$

For a given instance $P \in \mathscr{P}_{\text{VIP}}$ and a target accuracy $\epsilon > 0$, we define the *communication cost* required by a distributed algorithm $\mathscr{M}$ on P, denoted by $T_P^{\mathscr{M}}$, as the smallest integer $k \in \{1, \ldots, T\}$ such that the generated solution $\bar{\mathbf{z}}^k$ satisfies the target accuracy ϵ.

The total number of local oracle queries evaluated by Agent i for instance P up to this point is given by the cumulative number of local steps taken, denoted by $N_{i,P}^{\mathscr{M}} = \sum_{r=0}^{T_P^{\mathscr{M}}-1} \tau_i^r$.

Let c_i denote the computational cost of evaluating a single partial oracle V_i. The *communication cost* and *oracle cost* of algorithm $\mathscr{M}$ over the entire problem class $\mathscr{P}_{\text{VIP}}$ are defined by taking the supremum over all instances:
$$T_{\mathscr{P}_{\text{VIP}}}^{\mathscr{M}} = \sup_{P \in \mathscr{P}_{\text{VIP}}} T_P^{\mathscr{M}} \quad \text{and} \quad N_{\mathscr{P}_{\text{VIP}}}^{\mathscr{M}} = \sup_{P \in \mathscr{P}_{\text{VIP}}} \Big(\sum_{i \in [K]} c_i N_{i,P}^{\mathscr{M}} \Big).$$

We first state the classic results of the EG method in Proposition 20, which remains the state-of-the-art communication complexity bound.

Proposition 20 (Juditsky et al. 2011, Eq. (6.21))**.** *For any target accuracy $\epsilon > 0$, the communication cost of* EG *is bounded by*
$$\mathcal{O}\big(\sum_{i \in [K]} \frac{A_i + B_i}{\epsilon}\big),$$
and the computational cost of EG *is bounded by*
$$\mathcal{O}\Big(\big(\sum_{i \in [K]} c_i\big)\big(\sum_{i \in [K]} \frac{A_i}{\epsilon}\big) + \big(\sum_{i \in [K]} c_i\big)\big(\sum_{i \in [K]} \frac{B_i}{\epsilon}\big)\Big).$$

E.3 Decoupled method for variational inequality problems

Now, we present our DM-VIP method, which extends the DM-SP into multiplayer general-sum games.

Assembled norm. Given parameters $\alpha_i > 0$ for all $i \in [K]$ (to be specified later), we equip the joint space $\mathcal{E} = \mathcal{E}_1 \times \cdots \times \mathcal{E}_K$ with the assembled norm:
$$\|\mathbf{z}\|_{\mathcal{E}} = \langle \mathbf{P}\mathbf{z}, \mathbf{z} \rangle^{\frac{1}{2}} = \sqrt{\sum_{i=1}^{K} \alpha_i \|\mathbf{z}_i\|_i^2} \quad \text{for all } \mathbf{z} \in \mathcal{E}, \tag{11}$$
which corresponds to the block diagonal linear operator $\mathbf{P} = \alpha_1 \mathbf{P}_1 \oplus \cdots \oplus \alpha_K \mathbf{P}_K$. Accordingly, we equip the dual space $\mathcal{E}^* = \mathcal{E}_1^* \times \cdots \times \mathcal{E}_K^*$ with the corresponding dual norm:
$$\|\mathbf{g}\|_{\mathcal{E}^*} = \langle \mathbf{g}, \mathbf{P}^{-1}\mathbf{g} \rangle^{\frac{1}{2}} = \sqrt{\sum_{i=1}^{K} \alpha_i^{-1} \|\mathbf{g}_i\|_{i^*}^2} \quad \text{for all } \mathbf{g} \in \mathcal{E}^*. \tag{12}$$

Template DM-VIP. To extend our decoupled framework to block composite variational inequality problems (VIPs), we first define the coupled conditioning constant, which characterizes the interaction between the K distinct blocks:
$$\bar{L}_{\mathsf{c}} \triangleq \sqrt{\max_{j \in [K]} \Big[(\alpha_j D_j)^{-1} \sum_{i \in [K] \setminus \{j\}} \frac{\bar{L}_{ij} \big(\sum_{l \in [K] \setminus \{i\}} \bar{L}_{il} D_l \big)}{\alpha_i} \Big]}. \tag{13}$$

Algorithm 5 outlines the Decoupled Method for block composite VIPs (DM-VIP), generalizing the DM-SP procedure. The algorithm maintains a sequence of anchor points $\mathbf{v}^t$ and orchestrates iterative updates among K agents over a distributed network.

At the start of each iteration, the agents decouple the joint problem by fixing their remote variables to the current anchor components $\mathbf{v}_{-i}^t$. This allows each Agent i to independently and concurrently solve its regularized local subproblem. Specifically, each agent invokes an internal solver $\mathcal{M}_i^{\text{MRN}}$ to minimize the local residual norm (MRN) up to a target accuracy $\delta_i^{t+1} = \frac{\alpha_i \lambda_{t+1}}{2}$ (Line 4).

Following this local computation phase, the K agents perform exactly two communication rounds to complete the iteration. In the first round (Line 5), the agents broadcast their locally computed approximate solutions $\mathbf{z}_i^{t+1}$ to assemble the joint intermediate point $\mathbf{z}^{t+1}$. In the second round (Line 6), they use this assembled point to evaluate their local partial operators $V_i(\mathbf{z}^{t+1})$, which they subsequently exchange to form the full joint operator $V_\psi(\mathbf{z}^{t+1})$.

Finally, using this assembled joint operator, the agents compute a closed-form reduced step size a_{t+1}, update the running ergodic average $\bar{\mathbf{z}}^{t+1}$, and perform a joint extragradient-like step to generate the next anchor $\mathbf{v}^{t+1}$ (Lines 7 and 8).

Algorithm 5 DM-VIP$\big(K, (V_i)_{i\in[K]}, (\psi_i)_{i\in[K]}, \mathbf{z}^0, (\lambda_t)_{t\geq 1}, (\alpha_i)_{i\in[K]} \mid (\mathcal{M}_i^{\text{MRN}})_{i\in[K]}\big)$

Require: A local solver $\mathcal{M}_i^{\text{MRN}}$ for the minimization of residual norms.

1: $\mathbf{v}^0 = (\mathbf{v}_1^0, \dots, \mathbf{v}_K^0) = \mathbf{z}^0$.

2: **for** $t = 0, 1, \dots, T-1$ **do**

3: Let $\delta_i^{t+1} = \frac{\alpha_i \lambda_{t+1}}{2}$ for all $i \in [K]$.

4: Concurrently for all $i \in [K]$, Agent i computes

$$\big(\mathbf{z}_i^{t+1}, \psi_i'(\mathbf{z}_i^{t+1})\big) = \mathcal{M}_i^{\text{MRN}}\Big(V_i(\cdot; \mathbf{v}_{-i}^t),\ \psi_i + \frac{\alpha_i \lambda_{t+1}}{2}\|\cdot - \mathbf{v}_i^t\|_i^2,\ \mathbf{v}_i^t,\ \delta_i^{t+1}\Big).$$

5: All agents exchange $\mathbf{z}_i^{t+1}$ to assemble the joint point $\mathbf{z}^{t+1} = (\mathbf{z}_1^{t+1}, \dots, \mathbf{z}_K^{t+1})$.

6: Agents locally compute $V_i(\mathbf{z}^{t+1})$ and exchange to assemble the full operator:

$$V_\psi(\mathbf{z}^{t+1}) = V(\mathbf{z}^{t+1}) + \big(\psi_1'(\mathbf{z}_1^{t+1}), \dots, \psi_K'(\mathbf{z}_K^{t+1})\big).$$

7: Let $a_{t+1} = \frac{2\langle V_\psi(\mathbf{z}^{t+1}), \mathbf{v}^t - \mathbf{z}^{t+1}\rangle}{\|V_\psi(\mathbf{z}^{t+1})\|_{\mathcal{E}^*}^2}$ and generate solution $\bar{\mathbf{z}}^{t+1} = \big(\sum_{i=1}^{t+1} a_i\big)^{-1} \sum_{i=1}^{t+1} a_i \mathbf{z}^i$.

8: $\mathbf{v}^{t+1} = \arg\min_{\mathbf{v}\in Q} \big[a_{t+1}\langle V_\psi(\mathbf{z}^{t+1}), \mathbf{v}\rangle + \frac{1}{2}\|\mathbf{v} - \mathbf{v}^t\|_{\mathcal{E}}^2\big]$.

9: **end for**

Similar to the saddle point setting, we refer to Algorithm 5 as a template method because we have abstracted the exact implementation of the inner solvers. For the theoretical guarantees presented below, we merely assume the existence of an algorithmic oracle $\mathcal{M}_i^{\text{MRN}}(\hat{V}_i, \hat{\psi}_i, \mathbf{v}_i, \delta_i)$ capable of taking an MRN instance in the space $\mathcal{E}_i$ and successfully returning a sufficiently accurate solution. We defer the detailed implementation of these local solvers to Eq. (15) at the end of this section.

Provided that the local updates follow appropriate span conditions, the template DM-VIP algorithm successfully solves the generalized problem with the following optimal communication complexity.

Theorem 21. *Consider the* DM-VIP *template applied to* $\mathscr{P}_{\text{VIP}}$. *With the parameter choices of* $\alpha_i = \frac{\sum_{j\in[K]\setminus\{i\}} \bar{L}_{ij} D_j}{D_i}$ *for all* $i \in [K]$, *and* $\lambda_t \equiv \lambda \geq 2\bar{L}_c$, *we have:*

$$T_{\mathscr{P}_{\text{VIP}}}^{\text{DM-VIP}} \leq 2 + \sum_{\substack{i,j\in[K]\\ i\neq j}} \frac{2\bar{L}_{ij} D_i D_j}{\epsilon}.$$

For the distributed VIP considered in this paper, the classic EG method represents the best known communication cost. Now, let us compare our communication guarantee with that of EG.

Remark 6 (Improved communication). The classic EG method takes

$$\frac{1}{\epsilon}\sum_{i\in[K]} (A_i + B_i)$$

communication rounds (cf. Proposition 20). Our communication cost in Theorem 21 is $\mathcal{O}\big(\sum_{i\in[K]} A_i/\epsilon\big)$, which is consistently no worse compared to that of EG and is substantially faster

when the "diagonal conditioning" dominates—i.e.,

$$\sum_{i\in[K]} B_i \gg \sum_{i\in[K]} A_i.$$

To our knowledge, the rate of DM-VIP has improved the state-of-the-art communication cost for distributed VIPs.

E.4 Detailed proofs

E.4.1 Proof for FDS

Let us provide the detailed pseudocode of FDS for VIPs in Algorithm 6. Then, we prove the correctness of the solution returned by FDS.

Algorithm 6 $\mathrm{FDS}_{\|\cdot\|_\mathcal{E}}\big((V_i)_{i\in[K]}, (\psi_i)_{i\in[K]}, \mathbf{v}, \lambda \mid (\mathcal{M}_i^{\mathrm{MRN}})_{i\in[K]}\big)$

Require: Solver $\mathcal{M}_i^{\mathrm{MRN}}$ for the minimization of residual norms, for all $i \in [K]$.

1: **for** $i \in [K]$ **do**
2: $\quad\delta_i = \frac{\alpha_i\lambda}{2}$.
3: $\quad\hat{\psi}_i = \psi_i + \frac{\alpha_i\lambda}{2}\|\cdot - \mathbf{v}_i\|_i^2$.
4: $\quad(\mathbf{z}_i^+, \psi_i'(\mathbf{z}_i^+)) = \mathcal{M}_i^{\mathrm{MRN}}\big(V_i(\cdot; \mathbf{v}_{-i}), \hat{\psi}_i, \mathbf{v}_i, \delta_i\big)$.
5: $\quad\psi_i'(\mathbf{z}_i^+) = \hat{\psi}_i'(\mathbf{z}_i^+) - \alpha_i\lambda\mathbf{P}_i(\mathbf{z}_i^+ - \mathbf{v}_i)$.
6: **end for**
7: **return** $(\mathbf{z}^+, \psi'(\mathbf{z}^+))$, where $\mathbf{z}^+ = (\mathbf{z}_1^+, \cdots, \mathbf{z}_K^+)$ and $\psi'(\mathbf{z}^+) = (\psi_1'(\mathbf{z}_1^+), \cdots, \psi_K'(\mathbf{z}_K^+))$.

Lemma 22. *Under (A3'), for $\lambda \geq 2\bar{L}_c$,* FDS *(Algorithm 6) returns the correct solution of the MS subproblem given by $(V, \psi, \mathbf{v}, \lambda)$.*

Proof of Lemma 22. For all $i \in [K]$, by (A3') and then by the relative distance accuracy, we have

$$\begin{aligned}
&\|V_i(\mathbf{z}^+) + \psi_i'(\mathbf{z}_i^+) + \alpha_i\lambda\mathbf{P}_i(\mathbf{z}_i^+ - \mathbf{v}_i)\|_{i^*} \\
&\quad\leq \|V_i(\mathbf{z}_i^+; \mathbf{v}_{-i}) + \psi_i'(\mathbf{z}^+) + \alpha_i\lambda\mathbf{P}_i(\mathbf{z}_i^+ - \mathbf{v}_i)\|_{i^*} + \sum_{j\in[K]\setminus\{i\}} L_{ij}\|\mathbf{z}_j^+ - \mathbf{v}_j\|_j \\
&\quad\leq \delta_i\|\mathbf{z}_i^+ - \mathbf{v}_i\|_i + \sum_{j\in[K]\setminus\{i\}} L_{ij}\|\mathbf{z}_j^+ - \mathbf{v}_j\|_j.
\end{aligned} \tag{14}$$

Finally, we assemble the norms:

$$
\begin{aligned}
&\|V(\mathbf{z}^+) + \psi'(\mathbf{z}_i^+) + \lambda \mathbf{P}(\mathbf{z}_i^+ - \mathbf{v}_i)\|_{\mathcal{E}^*}^2 \\
&= \sum_{i\in[K]} \alpha_i^{-1} \|V_i(\mathbf{z}^+) + \psi_i'(\mathbf{z}_i^+) + \alpha_i \lambda \mathbf{P}_i(\mathbf{z}_i^+ - \mathbf{v}_i)\|_{i^*}^2 \\
&\overset{(14)}{\leq} \sum_{i\in[K]} \alpha_i^{-1} \Big(\delta_i \|\mathbf{z}_i^+ - \mathbf{v}_i\|_i + \sum_{j\in[K]\setminus\{i\}} L_{ij} \|\mathbf{z}_j^+ - \mathbf{v}_j\|_j\Big)^2 \\
&\leq \sum_{i\in[K]} 2\alpha_i^{-1} \Big[\delta_i^2 \|\mathbf{z}_i^+ - \mathbf{v}_i\|_i^2 + \Big(\sum_{j\in[K]\setminus\{i\}} L_{ij} \|\mathbf{z}_j^+ - \mathbf{v}_j\|_j\Big)^2\Big] \\
&= \frac{\lambda^2}{2} \sum_{i\in[K]} \Big(\alpha_i \|\mathbf{z}_i^+ - \mathbf{v}_i\|_i^2\Big) + 2 \sum_{i\in[K]} \Big[\alpha_i^{-1} \Big(\sum_{j\in[K]\setminus\{i\}} L_{ij} \|\mathbf{z}_j^+ - \mathbf{v}_j\|_j\Big)^2\Big] \\
&\leq \frac{\lambda^2}{2} \|\mathbf{z}^+ - \mathbf{v}\|_{\mathcal{E}}^2 + 2 \sum_{i\in[K]} \Big[\alpha_i^{-1} \Big(\sum_{l\in[K]\setminus\{i\}} L_{il} D_l\Big) \Big(\sum_{j\in[K]\setminus\{i\}} \frac{L_{ij}}{D_j} \|\mathbf{z}_j^+ - \mathbf{v}_j\|_j^2\Big)\Big] \\
&= \frac{\lambda^2}{2} \|\mathbf{z}^+ - \mathbf{v}\|_{\mathcal{E}}^2 + 2 \sum_{j\in[K]} \Big[\frac{\|\mathbf{z}_j^+ - \mathbf{v}_j\|_j^2}{D_j} \Big(\sum_{i\in[K]\setminus\{j\}} \frac{L_{ij} \big(\sum_{l\in[K]\setminus\{i\}} L_{il} D_l\big)}{\alpha_i}\Big)\Big] \\
&\overset{(13)}{\leq} \frac{\lambda^2}{2} \|\mathbf{z}^+ - \mathbf{v}\|_{\mathcal{E}}^2 + 2\|\mathbf{z}^+ - \mathbf{v}\|_{\mathcal{E}}^2 \bar{L}_{\mathsf{c}}^2 \\
&\leq \frac{\lambda^2}{2} \|\mathbf{z}^+ - \mathbf{v}\|_{\mathcal{E}}^2 + \frac{\lambda^2}{2} \|\mathbf{z}^+ - \mathbf{v}\|_{\mathcal{E}}^2 = \lambda^2 \|\mathbf{z}^+ - \mathbf{v}\|_{\mathcal{E}}^2.
\end{aligned}
$$

□

E.4.2 Proof for MRN

Our algorithm is built upon Lemma 23, the proof of which can be found in [Boţ and Chenchene, 2024, Corollary 2.4].

Lemma 23. *Assume (Â1), (Â3), and that the solution set of the VIP of (V_w, ψ_w) is non-empty. Then, there exists an algorithm, denoted by $(\mathbf{w}^+, \psi_w'(\mathbf{w}^+)) = \mathrm{FEGM}\,(V_w, \psi_w, \mathbf{v}_w, \xi \mid L)$, which takes no more than $C_0 \cdot \frac{L}{\xi}$ operator queries and returns $(\mathbf{w}^+, \psi_w'(\mathbf{w}^+))$ that satisfies ξ-distance-to-solution accuracy, where $C_0 > 0$ is some fixed constant.*

E.4.3 Concrete implementation

We are now back to considering the VIPs. Let us use FEGM in Lemma 23 for the minimization of residual norms:

$$
\mathcal{M}_i^{\mathrm{FEGM}}(\hat{V}_i, \hat{\psi}_i, \mathbf{v}_i, \delta_i) \triangleq \mathrm{FEGM}(\hat{V}_i, \hat{\psi}_i, \mathbf{v}_i, \frac{2\delta_i}{3} \mid L_{ii}).
$$

Then, for any Monteiro-Svaiter Subproblem given by $(V, \psi, \mathbf{v}, \lambda)$, we leverage the solver

$$
\text{FDS-FEGM}(V, \psi, \mathbf{v}, \lambda) = \mathrm{FDS}_{\|\cdot\|_{\mathcal{E}}}\big(V, \psi, \mathbf{v}, \lambda \mid (\mathcal{M}_i^{\mathrm{FEGM}})_{i\in[K]}\big).
$$

Finally, we obtain the concrete algorithm DM-VIP as follows:

$$
\mathrm{ROM}_{\|\cdot\|_{\mathcal{E}}}\Big((V_i)_{i\in[K]}, (\psi_i)_{i\in[K]}, \mathbf{z}^0, (\lambda_t)_{t\geq 1} \mid \text{FDS-FEGM}\Big). \tag{15}
$$

Combining Lemmas 15 to 17 and 23, with the implementation in Eq. (15), we conclude that Lemma 24 holds for the constant C_0 from Lemma 23. We include the complete proof below.

Lemma 24. *Under (A2') and (A3'), for $\lambda_{t+1} \equiv \lambda \geq 2\bar{L}_{\mathsf{c}}$, DM-VIP (Algorithm 5) with the implementation in Eq. (15) takes no more than*

$$
2T
$$

communication rounds and no more than

$$T \cdot \big(1 + C_0 \cdot \frac{3L_{ii}}{\alpha_i \lambda}\big)$$

queries to V_i, for all $i \in [K]$, and obtains an ϵ-approximate solution $\bar{\mathbf{z}}^T$, where

$$T = \Big\lceil \frac{\sum_i \alpha_i \lambda D_i^2}{2\epsilon} \Big\rceil$$

and $C_0 > 0$ is some fixed constant.

Proof of Lemma 24. By (A1'), we have

$$\Delta(\bar{\mathbf{z}}^T) \le \Big(\sum_{t=0}^{T-1} a_{t+1}\Big)^{-1} \max_{\mathbf{z} \in \mathcal{B} \cap Q} \Big[\sum_{t=0}^{T-1} a_{t+1} \langle V_\psi(\mathbf{z}^{t+1}), \mathbf{z}^{t+1} - \mathbf{z}\rangle\Big].$$

Further, with $\lambda \ge 2\bar{L}_{\mathrm{c}}$, by Lemmas 15 and 16, we have

$$\begin{aligned}
\Delta(\bar{\mathbf{z}}^T) &\le \Big(\sum_{t=0}^{T-1} a_{t+1}\Big)^{-1} \max_{\mathbf{z} \in \mathcal{B} \cap Q} \Big[\sum_{t=0}^{T-1} a_{t+1} \langle V_\psi(\mathbf{z}^{t+1}), \mathbf{z}^{t+1} - \mathbf{z}\rangle\Big] \\
&\le \Big(\sum_{t=0}^{T-1} a_{t+1}\Big)^{-1} \Big[\sum_{i \in [K]} \big(\frac{\alpha_i}{2} \max_{\mathbf{z}_i \in \mathcal{B}_i \cap \mathrm{dom}\, \psi_i} \|\mathbf{z}_i^0 - \mathbf{z}_i\|_i^2\big)\Big] \\
&\le \Big(\sum_{t=0}^{T-1} \frac{1}{\lambda_{t+1}}\Big)^{-1} \cdot \frac{1}{2} \sum_{i \in [K]} \alpha_i D_i^2 \le \epsilon,
\end{aligned}$$

where the last inequality follows from the assignments of $(\lambda_t)_{t \ge 1}$ and T. Therefore, the number of communication rounds is bounded by $2T$.

Now we count the number of gradient queries. By Lemma 17, FEGM always returns the solution with the required relative distance accuracy; and in view of Lemma 23, it takes no more than $C_0 \cdot \frac{3L_{ii}}{\alpha_i \lambda}$ gradient queries to V_i, for all $i \in [K]$. Therefore, the numbers of queries to V_i are bounded by $T \cdot \big(1 + C_0 \cdot \frac{3L_{ii}}{\alpha_i \lambda}\big)$, for all $i \in [K]$. □

Remark 7 (Oracle comparison). Under the same choice of parameters as in Theorem 21, the oracle cost of DM-VIP is bounded by

$$\frac{2}{\epsilon}\Big(\sum_{i \in [K]} c_i\Big)\Big(\sum_{i \in [K]} A_i\Big) + \frac{3C_0}{\epsilon}\Big(\sum_{i \in [K]} \frac{B_i c_i}{A_i}\Big)\Big(\sum_{i \in [K]} A_i\Big). \tag{16}$$

Compared to the computational cost of EG, which is given by

$$\frac{1}{\epsilon}\Big(\sum_{i \in [K]} c_i\Big)\Big(\sum_{i \in [K]} A_i\Big) + \frac{1}{\epsilon}\Big(\sum_{i \in [K]} c_i\Big)\Big(\sum_{i \in [K]} B_i\Big),$$

our computational cost in Eq. (16) differs in the second term. Consequently, our DM-VIP may offer an advantage or disadvantage depending on the relative conditioning of A_i, B_i, and c_i for $i \in [K]$.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the claims made, including the contributions made in the paper and important assumptions.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have discussed the limitation and potential future work in the conclusion section of the paper.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The presentation of the results are self-contained and all the assumptions are clearly stated. The proofs are either in the main paper or in the appendix, and they are correct.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [N/A]

   Justification: This paper does not include experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [N/A]

Justification: This paper does not include experiments.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [N/A]

Justification: This paper does not include experiments.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [N/A]

Justification: This paper does not include experiments.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [N/A]

   Justification: This paper does not include experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: This paper is conducted in accordance with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: There is no societal impact of the work performed.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release any data or models.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [N/A]

    Justification: The paper does not use any existing assets.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release any new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not include crowdsourcing experiments or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not include research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are only used for writing, editing, and formatting purposes in this paper, and do not impact the core methodology, scientific rigor, or originality of the research.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.