Escaping Saddle Point Efficiently in Minimax and Bilevel Optimizations

Wenhan Xian[1] , **Feihu Huang**[2] , **Heng Huang**[1]
[1]University of Maryland, College Park
[2]University of Pittsburgh
wxian1@umd.edu, huangfeihu2018@gmail.com, heng@umd.edu

Abstract

Hierarchical optimization is attracting significant attentions as it can be applied to a broad range of machine learning tasks. Recently, many algorithms are proposed to improve the theoretical results of minimax and bilevel optimizations. Among these works, a core issue that has not been well studies is to escape saddle point and find local minimum. In this paper, thus, we investigate the methods to achieve second-order optimality for nonconvex minimax and bilevel optimization. Specifically, we propose a new algorithm named PRGDA without the computation of second order derivative of the primal function. In nonconvex-strongly-concave minimax optimization, we prove that our algorithm can find a second-order stationary point with the gradient complexity that matches state-of-the-art result to find first-order stationary point. To our best knowledge, PRGDA is the first stochastic algorithm that is guaranteed to obtain the second-order stationary point for nonconvex minimax problems. In nonconvex-strongly-convex bilevel optimization, our method also achieves better gradient complexity to find local minimum. Finally, we conduct two numerical experiments to validate the performance of our new method.

1 Introduction

Hierarchical optimization (including minimax and bilevel optimization) is a popular and important optimization framework which is applied to a wide range of machine learning problems, such as Generative Adversarial Net [Goodfellow *et al.*, 2014], adversarial training [Madry *et al.*, 2018], multi-agent reinforcement learning [Wai *et al.*, 2018], meta-learning [Franceschi *et al.*, 2018; Bertinetto *et al.*, 2018] and hyperparameter optimization [Shaban *et al.*, 2019; Feurer and Hutter, 2019]. In this paper, we study the following stochastic hierarchical optimization problem

$$\min_{x\in\mathbb{R}^{d_1}} \Phi(x) := f(x, y^*(x)) = \mathbb{E}_{\xi\in\mathcal{D}}[F(x, y^*(x); \xi)] \quad (1)$$
$$\text{s.t. } y^*(x) = \arg\min_{y\in\mathbb{R}^{d_2}} g(x, y) = \mathbb{E}_{\zeta\in\mathcal{D}'}[G(x, y; \zeta)],$$

where the upper-level function $f(x, y^*(x))$ is smooth and possibly nonconvex, and the lower-level function $g(x, y)$ is smooth and strongly-convex w.r.t. variable y so that $y^*(x)$ and $\Phi(x)$ can be well defined. ξ and ζ are samples drawn from data distribution $\mathcal{D}$ and $\mathcal{D}'$. Stochastic problem is a general form that covers a couple of optimization tasks, including online optimization and finite-sum optimization. When $g(x, y) = -f(x, y)$, the above bilevel optimization problem is reduced to a standard minimax optimization which can be rewritten as Eq. (2)

$$\min_{x\in\mathbb{R}^{d_1}} \max_{y\in\mathcal{Y}} f(x, y) = \mathbb{E}_{\xi\in\mathcal{D}}[F(x, y; \xi)] \quad (2)$$

where $\mathcal{Y}$ is a convex domain (not required to be compact). The loss function $f(x, y)$ is smooth, nonconvex w.r.t. x and strongly-concave w.r.t. y.

1.1 Minimax Optimzation

Recently, there are plenty of works studying minimax optimization problem in a variety of research fields in machine learning. Many deterministic and stochastic algorithms with asymptotic or non-asymptotic convergence analysis have been developed, such as Gradient Descent Ascent (GDA) [Du and Hu, 2019; Nemirovski, 2004] and Stochastic Gradient Descent Ascent (SGDA) [Lin *et al.*, 2020b]. Some algorithms adopt a single loop structure [Heusel *et al.*, 2017; Lin *et al.*, 2020b; Xu *et al.*, 2023] while the others use a nested loop to update y more frequently so that they can obtain a better estimation of the maximum $y^*(x)$ [Jin *et al.*, 2020; Nouiehed *et al.*, 2019].

Besides, some algorithms have been proposed to improve the theoretical results of minimax optimization, such as SREDA [Luo *et al.*, 2020] and Acc-MDA [Huang *et al.*, 2022] which take advantage of variance reduction to accelerate the convergence rate and reduce the gradient complexity. Moreover, on deterministic setting some recently proposed algorithms [Lin *et al.*, 2020a] have already matched the optimal lower bound [Zhang *et al.*, 2021].

However, most of these works only consider the criterion of finding first-order stationary point. In nonconvex setting, convergence to first-order stationary point is not always satisfactory because a first-order stationary point could be a local minimum, saddle point or even local maximum. Therefore, second-order stationary point that reaches local minimum becomes a popular and important issue in nonconvex optimiza-

Name	Reference	Stochastic	Local Minimum	Pure First-Order
SGDA	[Lin *et al.*, 2020b]	$\surd$	$\times$	$\surd$
Cubic-GDA	[Chen *et al.*, 2021]	$\times$	$\surd$	$\times$
MCN	[Luo and Chen, 2022]	$\times$	$\surd$	$\times$
Perturbed GDmax	[Huang *et al.*, 2025]	$\times$	$\surd$	$\surd$
PRGDA	(ours)	$\surd$	$\surd$	$\surd$

Table 1: Comparison of properties between related algorithms for minimax optimization.

tion. Since finding global minimum in nonconvex optimization is usually an NP-hard problem [Hillar and Lim, 2013], in some situations we attempt to find a local minimum instead. Moreover, in some machine learning tasks such as tensor decomposition [Ge *et al.*, 2015], matrix sensing [Bhojanapalli *et al.*, 2016; Park *et al.*, 2017], and matrix completion [Ge *et al.*, 2016], finding local minimum is equivalent to finding global minimum, which makes second-order stationary point more crucial.

Therefore, we are motivated to study the second-order optimality for minimax optimization which captures the local minimum of $\Phi(x)$. We will discuss the relation between the second-order stationary point of $\Phi(x)$ and the local equilibrium of $f(x, y)$ in Section 3.2. In Section 3.1 we can see that under certain conditions the primal function $\Phi(x)$ is twice differentiable. An $O(\epsilon, \epsilon_H)$ second-order stationary point satisfies $\|\nabla\Phi(x)\| \leq O(\epsilon)$ and $\lambda_{min}(\nabla^2\Phi(x)) \geq -\epsilon_H$ where $\lambda_{min}(\cdot)$ means the smallest eigenvalue.

Although several recent works have been proposed to study the second-order stationary point for nonconvex-strongly-concave minimax optimization based on cubic-regularized gradient descent ascent [Chen *et al.*, 2021; Luo and Chen, 2022] or perturbed gradient [Huang *et al.*, 2025], they are only adaptive to deterministic gradient oracle and finite-sum problem. The study of the second-order stationary point for stochastic nonconvex minimax problem where the full gradient is not available is still limited. A comparison between related minimax algorithms is demonstrated in Table 1.

Thus, to fill in this gap, we propose a new algorithm named Perturbed Recursive Gradient Descent Ascent (PRGDA) to search second-order stationary point for stochastic nonconvex problem (2). To our best knowledge, PRGDA is the first algorithm that is guaranteed to obtain second-order stationary point for stochastic nonconvex minimax optimization problems. Furthermore, our method is a pure first-order algorithm that only requires the computation of gradient oracle, which makes our method more efficient to implement. We also provide the analysis to show that the gradient complexity of our algorithm is $\tilde{O}(\kappa^3\epsilon^{-3})$ to achieve $O(\epsilon, \sqrt{\rho_\Phi\epsilon})$ second-order stationary point where κ and ρ_Φ are defined in Section 3.1, which matches the best result of finding first-order stationary point for the same stochastic nonconvex minimax problem.

1.2 Bilevel Optimization

Recently, many bilevel algorithms are proposed, such as deterministic algorithms AID-BiO and ITD-BiO [Ji *et al.*, 2021], and stochastic algorithms BSA [Ghadimi and Wang, 2018] and StocBiO [Ji *et al.*, 2021]. These methods are proposed to improve the convergence analysis of bilevel optimization since most earlier works [Domke, 2012; Pedregosa, 2016] only provide the asymptotic convergence analysis.

StocBiO algorithm [Ji *et al.*, 2021] is a recent work to solve stochastic nonconvex-strongly-convex bilevel optimization via AID. In this paper, we also study the convergence of our method under this condition where $\Phi(x)$ is stochastic and probably nonconvex. According to previous studies of bilevel optimization, when $f(x, y)$ and $g(x, y)$ are differentiable and $g(x, y)$ is strongly-convex with respect to y, $\Phi(x)$ is also differentiable and automatically $\|\nabla\Phi(x)\| \leq \epsilon$ is a criterion of first-order stationary point. Notice that in [Ji *et al.*, 2021] $\|\nabla\Phi(x)\|^2 \leq \epsilon$ is used as the criterion. In this paper, we will uniformly adopt $\|\nabla\Phi(x)\| \leq \epsilon$ as the convergence criterion. More recently, many stochastic algorithms with variance reduction are proposed, such as RSVRB [Guo *et al.*, 2021], SUSTAIN [Khanduri *et al.*, 2021], MRBO and VRBO [Yang *et al.*, 2021]. The gradient complexity of bilevel optimization is enhanced to $O(\epsilon^{-3})$, which is the best theoretical result as far as we know. StocBiO with iNEON [Huang *et al.*, 2025] is a recent work that combines StocBiO with pure first-order method inexact negative curvature originated from noise (iNEON) to escape saddle point and find second-order stationary point for nonconvex-strongly-convex bilevel optimization.

Although these works are proposed to improve the performance of algorithms for bilevel optimization, the complexity of current methods that achieve second-order stationary point are still high. Actually, the complexity of StocBiO with iNEON is even higher than the standard StocBiO algorithm in order to find a local minimum with high probability. Thus, to fill these gap, we are motivated to propose an accelerated algorithm with variance reduction that requires lower complexity to find second-order stationary point for stochastic nonconvex-strongly-convex bilevel optimization.

The comparison between our method and related works to find $O(\epsilon)$ first-order stationary point or $O(\epsilon, \sqrt{\rho_\Phi\epsilon})$ second-order stationary point is shown in Table 2, where $Gc(f, \epsilon)$ and $Gc(g, \epsilon)$ are the numbers of gradient evaluations of function $f(x, y)$ and $g(x, y)$ respectively. The last column represents whether the algorithm can escape saddle point and find local minimum. Notation $\tilde{O}(\cdot)$ hides the logarithm term. StocBiO with iNEON and our PRGDA algorithm involve a logarithm term in the complexity because they converge to second-order stationary point with high probability.

From Table 2 we can see our PRGDA algorithm improves the gradient complexity $Gc(f, \epsilon)$ and $Gc(g, \epsilon)$ of StocBiO with iNEON algorithm significantly and matches state-of-the-art complexity $O(\epsilon^{-3})$, which is one of the most impor-

Name	Reference	$Gc(f,\epsilon)$	$Gc(g,\epsilon)$	Local Minimum
StocBiO	[Ji *et al.*, 2021]	$O(\kappa^5\epsilon^{-4})$	$O(\kappa^9\epsilon^{-4})$	$\times$
SUSTAIN	[Khanduri *et al.*, 2021]	$O(p(\kappa)\epsilon^{-3})$	$O(p(\kappa)\epsilon^{-3})$	$\times$
MRBO/VRBO	[Yang *et al.*, 2021]	$O(p(\kappa)\epsilon^{-3})$	$O(p(\kappa)\epsilon^{-3})$	$\times$
StocBiO + iNEON	[Huang *et al.*, 2025]	$\tilde{O}(\kappa^5\epsilon^{-4})$	$\tilde{O}(\kappa^{10}\epsilon^{-4})$	$\surd$
PRGDA	(ours)	$\tilde{O}(\kappa^3\epsilon^{-3})$	$\tilde{O}(\kappa^7\epsilon^{-3})$	$\surd$

Table 2: Comparison between related bilevel algorithms. We use $p(\kappa)$ for some algorithms that do not provide the explicit dependence on κ.

tant contribution of this paper.

1.3 Contributions

We summarize our main contributions as follows:

- We propose a new PRGDA algorithm which is the first algorithm to reach second-order stationary point for stochastic nonconvex minimax optimization problem. Our method is pure first-order and does not require any calculation of second-order derivatives. Our method does not involve nested loops either, which makes it more efficient to implement.
- We prove that the gradient complexity of our algorithm is $\tilde{O}(\kappa^3\epsilon^{-3})$ to achieve $O(\epsilon, \sqrt{\epsilon})$ second-order stationary point in stochastic nonconvex minimax optimization, which matches the best result of finding first-order stationary point in the same problem.
- PRGDA can also be applied to nonconvex bilevel optimization and we can prove that the gradient complexity is $Gc(f,\epsilon) = \tilde{O}(\kappa^3\epsilon^{-3})$ and $Gc(g,\epsilon) = \tilde{O}(\kappa^7\epsilon^{-3})$ to find $O(\epsilon, \sqrt{\epsilon})$ second-order stationary point in stochastic nonconvex bilevel optimization, which outperforms the previous best theoretical results and matches state-of-the-art to find first-order stationary point.

2 Related Work

2.1 Stochastic Minimax Optimization

Many algorithms are proposed to solve stochastic nonconvex-strongly-concave minimax problem, including intuitive methods SGDmax [Jin *et al.*, 2020] and Stochastic Gradient Descent Ascent (SGDA) [Lin *et al.*, 2020b]. Recently, some methods integrate variance reduction with minimax problem to accelerate the convergence, such as Stochastic Recursive gradiEnt Descent Ascent (SREDA) [Luo *et al.*, 2020], Hybrid Variance-Reduced SGD [Tran Dinh *et al.*, 2020] and Acc-MDA [Huang *et al.*, 2022]. There are also some works that study the weakly-convex concave minimax optimization such as [Rafique *et al.*, 2022] and [Yan *et al.*, 2020]. More related to this work, Cubic-Regularized Gradient Descent Ascent (Cubic-GDA) [Chen *et al.*, 2021] and Minimax Cubic Newton (MCN) [Luo and Chen, 2022] are two recent algorithms that can reach the second-order stationary point in nonconvex-strongly-concave minimax optimization.

2.2 Perturbed Gradient Descent

Perturbed Gradient Descent (PGD) [Jin *et al.*, 2017] was proposed to find second-order stationary point for nonconvex optimization which introduces a perturbation under specific condition. It is a deterministic gradient based algorithm and only involves first-order oracle. Perturbed Gradient Descent algorithm consists of two phases, a descent phase and an escaping phase. In the descent phase, the algorithm runs gradient descent to make the function value decrease until the magnitude of the gradient is smaller than a certain threshold. In the escaping phase, it first introduces a perturbation drawn from a uniform distribution on the ball $B_0(r)$ with center $\mathbf{0}$ and radius r. After certain iterations of gradient descent, if the function value is reduced by a significant threshold then it indicates that the algorithm escapes a saddle point and it will do the descent phase again. Otherwise, it can be proven that the point where the last descent terminates is second-order stationary with high probability. To extend PGD to the stochastic setting and incorporate it with variance reduction, SSRGD [Li, 2019] was proposed to reach second-order stationary point with stochastic first-order oracle (SFO) of $O(\epsilon^{-3.5})$. After that Pullback algorithm [Chen *et al.*, 2022] was proposed to improve the complexity to $O(\epsilon^{-3})$.

2.3 Cubic-GDA and Minimax Cubic Newton

Cubic-Regularized Gradient Descent Ascent (Cubic-GDA) [Chen *et al.*, 2021] and Minimax Cubic Newton (MCN) [Luo and Chen, 2022] are two recent algorithms that can reach the second-order stationary point in nonconvex-strongly-concave minimax optimization. Both of these two algorithms are inspired by cubic regularization and designed for deterministic problem. Cubic regularization was first proposed in [Nesterov and Polyak, 2006] which is a standard method that converges to second-order stationary point in conventional nonconvex optimization. However, Cubic-GDA and MCN are both designed for deterministic problem and neither of them works for the stochastic minimax problem (2) considered in this paper. Therefore, we are motivated to propose an algorithm that is suitable for the stochastic problem. Besides, Cubic-GDA and MCN involves the calculation of second-order oracle or Hessian vector product while our method only requires the first-order information, which indicates that our method is more efficient to implement because the computation cost of Hessian matrix could be high.

2.4 StocBiO with iNEON

In [Huang *et al.*, 2025], algorithms for both minimax and bilevel optimization are proposed to find second-order stationary point. However, for minimax optimization only the deterministic problem is studied. For bilevel optimization, the stochastic problem is considered and the StocBiO with

Algorithm 1 Perturbed Recursive Gradient Descent Ascent

Input: initial value x_0, y_0
Parameter: stepsize η and η_H, perturbation radius r, escaping phase threshold t_{thres}, average movement $\bar{D}$, tolerance ϵ, maximum iteration T.

Set $escape = false$, $s = 0$, $esc = 0$.
for $t = 0, 1, \ldots, T-1$ **do**
  Minimax: Update y_{t+1}, v_t, u_t from Algorithm 2.
  Bilevel: Update y_{t+1}, v_t, u_t from Algorithm 3.
  if $escape = false$ **then**
    if $\|v_t\| \geq \epsilon$ **then**
      Update $x_{t+1} = x_t - (\eta/\|v_t\|)v_t$.
    else
      Let $m_s = t$, $s = s + 1$.
      Set $escape = true$, $esc = 0$.
      Draw perturbation $\xi \sim B_0(r)$.
      update $x_{t+1} = x_t + \xi$.
    end if
  else
    Compute $D = \sum_{j=m_s+1}^{t} \eta_H^2 \|v_j\|^2$.
    if $D > (t - m_s)\bar{D}$ **then**
      Set η_t s.t. $\sum_{j=m_s+1}^{t} \eta_j^2 \|v_j\|^2 = (t - m_s)\bar{D}$.
      Update $x_{t+1} = x_t - \eta_t v_t$. Set $escape = false$.
    else
      Set $\eta_t = \eta_H$.
      Update $x_{t+1} = x_t - \eta_t v_t$, $esc = esc + 1$.
      Return x_{m_s} **if** $esc = t_{thres}$.
    end if
  end if
end for

Output: x_{m_s}

iNEON algorithm is proposed. The algorithm is inspired by NEON [Xu *et al.*, 2018; Allen-Zhu and Li, 2018], which is a method to find local minimum merely based on first-order oracles. Inexact NEON is a variant of NEON since the exact gradient in bilevel optimization is unavailable. However, it requires an extra nested loop to solve a subproblem that extracts a negative curvature descent direction. Besides, the gradient complexity of StocBiO with iNEON is also higher than the vanilla StocBiO. Therefore, we are motivated to propose a more efficient bilevel optimization algorithm that converges to second-order stationary point.

3 Preliminary

3.1 Notations and Assumptions

In this section we will present the notations used in this paper and introduce some basic assumptions to further illustrate the problem setting. In this paper we assume that upper-level function $f(x, y)$ is twice differentiable. Lower-level $g(x, y)$ is three times differentiable (only required in bilevel optimization). The partial derivative is denoted by ∇_x and ∇_y, *e.g.*, $\nabla f(x, y) = [\nabla_x f(x, y), \nabla_y f(x, y)]$. Similarly, ∇_x^2 and ∇_y^2 represent the Hessian. ∇_{xy}^2 and ∇_{yx}^2 represent the Jacobian. We use $\|\cdot\|_2$ and $\|\cdot\|_F$ to denote the spectral norm and Frobenius norm of matrix respectively. Notation $\tilde{O}(\cdot)$ means the complexity after hiding logarithm terms. First, we assume that lower-level function $g(x, y)$ is strongly-convex with respect to y so that $y^*(x)$ and $\Phi(x)$ can be well defined.

Assumption 1. *The lower-level function $g(x, y)$ is μ-strongly-convex with respect to y, i.e., there exists a constant μ such that*

$$g(x, y) + \langle \nabla_y g(x, y), y' - y \rangle + \frac{\mu}{2}\|y' - y\|^2 \leq g(x, y') \quad (3)$$

for any x, y and y'.

Notice that in minimax optimization $g(x, y)$ is the same as $-f(x, y)$ so we merge these two cases into one statement. With Assumption 1, objective function $\Phi(x)$ is also differentiable and the gradient is formulated as follows [Ji *et al.*, 2021].

$$\nabla\Phi(x) = \nabla_x f(x, y^*(x)) - \nabla_{xy}^2 g(x, y^*(x)) \cdot [\nabla_y^2 g(x, y^*(x))]^{-1} \nabla_y f(x, y^*(x)) \quad (4)$$

We can see the Hessian of g is automatically involved in the gradient of Φ. **Notice** that in this paper first-order method means only using the first-order information of Φ. In minimax optimization, since we always have $\nabla_y f(x, y^*(x)) = 0$, the expression of $\nabla\Phi(x)$ is simplified by

$$\nabla\Phi(x) = \nabla_x f(x, y^*(x)) \quad (5)$$

Next, we introduce the following assumptions about Lipschitz continuity of first and second order derivatives. These assumptions are commonly used in the convergence analysis of minimax and bilevel optimization [Luo *et al.*, 2020; Luo and Chen, 2022; Ji *et al.*, 2021; Huang *et al.*, 2025].

Assumption 2. *The gradients of component functions $F(x, y; \xi)$ and $G(x, y; \zeta)$ are L-Lipschitz continuous, i.e., there exists a constant L such that*

$$\|\nabla F(z; \xi) - \nabla F(z'; \xi)\| \leq L\|z - z'\|,$$
$$\|\nabla G(z; \zeta) - \nabla G(z'; \zeta)\| \leq L\|z - z'\| \quad (6)$$

for any $z = (x, y)$ and $z' = (x', y')$.

Assumption 3. *The second order derivatives $\nabla_x^2 f(x, y)$, $\nabla_{xy}^2 f(x, y)$, $\nabla_y^2 f(x, y)$, $\nabla_{xy}^2 g(x, y)$ and $\nabla_y^2 g(x, y)$ are ρ-Lipschitz continuous.*

The condition number κ of the hierarchical optimization problem is defined by $\kappa = L/\mu$. According to previous works, in minimax optimization under Assumptions 1, 2 and 3, $\Phi(x)$ is twice differentiable. $y^*(x)$ is κ-Lipschitz continuous, $\nabla\Phi(x)$ is L_Φ-Lipschitz continuous and $\nabla^2\Phi(x)$ is ρ_Φ-Lipschitz continuous.

According to [Ghadimi and Wang, 2018; Ji *et al.*, 2021], we know that in bilevel optimization function $y^*(x)$ is also κ-Lipschitz continuous, but we need an additional Assumptions 4 to guarantee $\Phi(x)$ has L_Φ-Lipschitz gradient.

Assumption 4. *The upper-level function $f(x, y)$ is M-Lipschitz continuous, i.e., there exists a constant M such that*

$$\|f(z) - f(z')\| \leq M\|z - z'\| \quad (7)$$

for any $z = (x, y)$ and $z' = (x', y')$.

Since in this paper we study the convergence to second-order stationary point, we also need the following Assumption 5 which is also assumed in [Huang *et al.*, 2025] that makes function $\Phi(x)$ twice differentiable and have ρ_Φ-Lipschitz Hessian. We should notice that Assumption 4 and 5 are **only** used for bilevel optimization.

Assumption 5. *The third order derivatives $\nabla^3_{xyx}g$, $\nabla^3_{yxy}g$ and $\nabla^3_y g$ are ν-Lipschitz continuous.*

3.2 Relations Between Local Nash Equilibrium

In nonconvex minimax optimization, local Nash equilibrium is an important concept about the convergence criterion, which is defined as follows in [Jin *et al.*, 2020].

Definition 1. *(Local Nash equilibrium) A point (x^*, y^*) is a local Nash equilibrium of Problem (2) if $f(x^*, y) \leq f(x^*, y^*) \leq f(x, y^*)$ for any x, y that satisfy $\|x - x^*\| \leq \delta$ and $\|y - y^*\| \leq \delta$.*

However, as it is indicated in [Jin *et al.*, 2020], a local Nash equilibrium may not exist in sequential games such as GAN. Hence a necessary condition of local Nash equilibrium is provided.

Definition 2. *(Local minimax point) A point (x^*, y^*) is a local minimax point of Problem (2) if y^* is a local maximum of function $f(x^*, \cdot)$ and there exists a constant $\delta_0 > 0$ such that x^* is a local minimum of function $g_\delta(x) = \max_{\|y - y^*\| \leq \delta} f(x, y)$ for any $0 < \delta \leq \delta_0$.*

Next, we will prove that a saddle point of the primal function $\Phi(x)$ is not a local minimax point, which indicates the importance to find second-order stationary point of $\Phi(x)$ in minimax optimization.

Suppose x^* is a saddle point of $\Phi(x)$ and y^* is the maximum of $f(x^*, \cdot)$. By the definition of saddle point, we know for $\forall \delta > 0$, there exists $\|x - x^*\| < \delta$ and $\Phi(x) < \Phi(x^*)$. Hence for $\forall \delta_0 > 0$ and $\forall \delta > 0$, there exists x' such that $\|x - x^*\| < \min\{\delta_0/\kappa, \delta\}$ and $\Phi(x') < \Phi(x^*)$. Let $y' = \max f(x', \cdot)$. According to Proposition 1, we have $g_{\delta_0}(x') = f(x', y') = \Phi(x') < \Phi(x^*) = g_{\delta_0}(x^*)$, which means x^* is not a local minimum of function $g_{\delta_0}(x)$. Therefore, a saddle point of the primal function $\Phi(x)$ will never be a local minimax point of Problem (2).

Proposition 1. *(Lemma 4.3 in [Lin* et al.*, 2020b]) Suppose function f satisfies Assumption 2 and Assumption 1. Then function $y^*(x)$ is κ-Lipschitz continuous, i.e.,*

$$\|y^*(x_1) - y^*(x_2)\| \leq \kappa \|x_1 - x_2\|$$

for $\forall x_1, x_2 \in \mathbb{R}^{d_1}$. Function $\Phi(x)$ is differentiable with gradient $\nabla\Phi(x) = \nabla_x f(x, y^*(x))$ and the gradient is L_Φ-Lipschitz continuous where $L_\Phi = L + \kappa L$.

4 Algorithm for Minimax Optimization

In this section, we will propose our PRGDA algorithm for the special case of minimax optimization. The description of our PRGDA algorithm is demonstrated in Algorithm 1. Similar to SREDA, the initial value y_0 is also yield by PiSARAH algorithm to make it close to $y^*(x_0)$, which is a conventional strongly-convex optimization subproblem. In our convergence analysis this step costs the gradient complexity of $\tilde{O}(\kappa^2\epsilon^{-2})$. We use v_t and u_t to represent the gradient estimator of $\nabla_x f(x_t, y_t)$ and $\nabla_y f(x_t, y_t)$ respectively. In each iteration, y_{t+1}, v_t and u_t are computed by an inner loop updater with K iterations, which is shown in Algorithm 2.

Algorithm 2 Updater of Inner Loop (Minimax)

Input: status x_t, x_{t-1}, y_t, v_{t-1}, u_{t-1} and t
Parameter: stepsize λ, inner loop size K, batchsize S_1 and S_2, period q.

Set $x_{t,-1} = x_{t-1}$, $x_{t,k} = x_t$ when $k \geq 0$, $y_{t,-1} = y_{t,0} = y_t$.
if $mod(t, q) = 0$ **then**
  Draw S_1 samples $\{\xi_1, \ldots, \xi_{S_1}\}$
  $v_{t,-1} = \frac{1}{S_1}\sum_{i=1}^{S_1} \nabla_x F(x_t, y_t; \xi_i)$,
  $u_{t,-1} = \frac{1}{S_1}\sum_{i=1}^{S_1} \nabla_y F(x_t, y_t; \xi_i)$.
else
  $v_{t,-1} = v_{t-1}$, $u_{t,-1} = u_{t-1}$.
end if
for $k = 0$ **to** $K - 1$ **do**
  Draw S_2 samples $\{\xi_1, \ldots, \xi_{S_2}\}$
  $v_{t,k} = v_{t,k-1} + \frac{1}{S_2}\sum_{i=1}^{S_2}(\nabla_x F(x_{t,k}, y_{t,k}; \xi_i) - \nabla_x F(x_{t,k-1}, y_{t,k-1}; \xi_i))$
  $u_{t,k} = u_{t,k-1} + \frac{1}{S_2}\sum_{i=1}^{S_2}(\nabla_y F(x_{t,k}, y_{t,k}; \xi_i) - \nabla_y F(x_{t,k-1}, y_{t,k-1}; \xi_i))$
  $y_{t,k+1} = \prod_{\mathcal{Y}}(y_{t,k} + \lambda u_{t,k})$.
end for
Select $s_t = \arg\min_k \|\tilde{\mathcal{G}}_\lambda(y_{t,k})\|$.
Let $y_{t+1} = y_{t,s_t}$, $v_t = v_{t,s_t}$, $u_t = u_{t,s_t}$.

Output: y_{t+1}, v_t, u_t.

$$\tilde{\mathcal{G}}_\lambda(y_{t,k}) = \frac{y_{t,k} - \Pi_{\mathcal{Y}}(y_{t,k} + \lambda u_{t,k})}{\lambda} \tag{8}$$

In Algorithm 2, we use the SPIDER gradient estimator to update $y_{t,k}$, $v_{t,k}$ and $u_{t,k}$. S_1 is the large batchsize that is loaded every q iterations of t. S_2 is the small batchsize. λ is the stepsize to update variable y. The output of the inner loop updater depends on the minimum value of the norm of $\tilde{\mathcal{G}}_\lambda(y_{t,k})$ and its corresponding index (defined in Eq. (8)). We will show that gradient estimator v_t satisfies $\|v_t - \nabla\Phi(x_t)\| \leq O(\epsilon)$ based on this inner loop updater.

Inspired by perturbed gradient descent, our PRGDA is also composed of a descent phase and an escaping phase. In the descent phase our PRGDA algorithm follows the iterative update rule of SPIDER that $x_{t+1} = x_t - (\eta/\|v_t\|)v_t$ until the norm of v_t satisfies $\|v_t\| \leq O(\epsilon)$. After the descent phase is terminated, we use m_s to denote the current counter t and uniformly draw a perturbation ξ from ball $B_0(r)$ where parameter r is the perturbation radius. We add the perturbation to the current status x_t and start the escaping phase. In the escaping phase, parameter t_{thres} is maximum number of iterations of the phase and $\bar{D}$ is the average moving distance which is used to determine if the escaping phase should be stopped. The stepsize of x in this phase is denoted by η_H which is typically larger than η in the descent phase. We use D to denote the accumulated squared moving distance. If the averaged

squared moving distance is larger than $\bar{D}$ then we pull it back (line 17 in Algorithm 1) and break the escaping phase. In this case we consider x_{m_s} as a saddle point and continue to run next descent phase. Otherwise, if the escaping phase is not broken after t_{thres} iterations, we claim that x_{m_s} is a second-order stationary point with high probability. This is because when $\lambda_{min}(\nabla^2\Phi(x_{m_s})) < -\epsilon_H$, the stuck region $\mathcal{S}$ defined by the area $\{\xi \in B_0(r)|$ the sequence started from $x_{m_s+1} = x_{m_s} + \xi$ does not break the escaping phase$\}$ has a small volume. Informally, the stuck region $\mathcal{S}$ must be contained in a "narrow band" or "thin disk" in a higher dimensional sphere. Since the perturbation ξ is uniformly drawn from ball $B_0(r)$, the probability that ξ belongs to the stuck region is low.

5 Algorithm for Bilevel Optimization

In this section we propose our PRGDA algorithm to solve the more general bilevel optimization. Actually, we only need to switch the inner loop updater in Algorithm 2 to the bilevel mode, which is demonstrated in Algorithm 3. Similar to the case of minimax optimization, here we also need a initialization algorithm to initialize y_0 with the cost of $Gc(g,\epsilon) = \tilde{O}(\kappa^6\epsilon^{-2})$ in the convergence analysis. Next we will elaborate the inner loop updater for bilevel optimization. We also use the update rule of SPIDER to compute $v_{t,k}^{(1)}$, $v_{t,k}^{(2)}$ and $u_{t,k}$, which represent the estimator of $\nabla_x f(x,y)$, $\nabla_y f(x,y)$ and $\nabla_y g(x,y)$ respectively. We should notice that the large and small batchsize of computing $u_{t,k}$ are different from that of $v_{t,k}^{(1)}$ or $v_{t,k}^{(2)}$. After the inner loop to compute y_{t+1}, we calculate the Jacobian J_t with a batch of size S_5. Then we compute v_t, the estimator of $\nabla\Phi(x)$ via AID. Here we follow the method used in StocBiO, which is

$$v_t = v_t^{(1)} - \alpha J_t \sum_{q=-1}^{Q-1} \prod_{j=Q-q}^{Q} (I - \alpha\nabla_y^2 G(x_t, y_{t+1}; \mathcal{B}_j)) v_t^{(2)} \tag{9}$$

where $\mathcal{B}_j$ is the set of samples to calculate the stochastic estimator of Hessian $\nabla_y^2 g(x_t, y_{t+1})$.

6 Convergence Analysis

In this section we will illustrate the convergence analysis of our algorithm. First, we need to assume that $\Phi(x)$ is lower bounded by Φ^*. Then we will present the main theorems of our PRGDA algorithm. In this paper, we set $\epsilon_H = \sqrt{\rho_\Phi \epsilon}$ as the tolerance of the second-order stationary point.

6.1 Main Theorem for Minimax Optimization

Theorem 1. *Under Assumption 1, 2 and 3, we set stepsize $\eta = \tilde{O}(\frac{\epsilon}{\kappa L})$, $\eta_H = \tilde{O}(\frac{1}{\kappa L})$ and $\lambda = O(\frac{1}{L})$, batchsize $S_1 = \tilde{O}(\kappa^2\epsilon^{-2})$ and $S_2 = \tilde{O}(\kappa\epsilon^{-1})$, period $q = O(\epsilon^{-1})$, inner loop $K = O(\kappa)$, perturbation radius $r = \min\{\tilde{O}(\sqrt{\frac{\epsilon}{\kappa^3\rho}}), \tilde{O}(\frac{\epsilon}{\kappa L})\}$, threshold $t_{thres} = \tilde{O}(\frac{L}{\sqrt{\kappa\rho\epsilon}})$ and average movement $\bar{D} = \tilde{O}(\frac{\epsilon^2}{\kappa^2 L^2})$. Then our PRGDA algorithm requires $\tilde{O}(\kappa^3\epsilon^{-3})$ SFO to achieve $O(\epsilon, \sqrt{\rho_\Phi\epsilon})$ second-order stationary point with high probability.*

Algorithm 3 Updater of Inner Loop (Bilevel)

Input: status x_t, x_{t-1}, y_t, $v_{t-1}^{(1)}$, $v_{t-1}^{(2)}$, u_{t-1} and t
Parameter: stepsize λ and α, inner loop size K and Q, batchsize B, S_1, S_2, S_3, S_4 and S_5, period q.

Set $x_{t,-1} = x_{t-1}$, $x_{t,k} = x_t$ when $k \geq 0$, $y_{t,-1} = y_{t,0} = y_t$.
if $mod(t,q) = 0$ **then**
  Draw samples with size S_1 and S_3.
  $v_{t,-1}^{(1)} = \frac{1}{S_1}\sum_{i=1}^{S_1} \nabla_x F(x_t, y_t; \xi_i)$,
  $v_{t,-1}^{(2)} = \frac{1}{S_1}\sum_{i=1}^{S_1} \nabla_y F(x_t, y_t; \xi_i)$,
  $u_{t,-1} = \frac{1}{S_3}\sum_{i=1}^{S_3} \nabla_y G(x_t, y_t; \zeta_i)$.
else
  $v_{t,-1}^{(1)} = v_{t-1}^{(1)}$, $v_{t,-1}^{(2)} = v_{t-1}^{(2)}$, $u_{t,-1} = u_{t-1}$.
end if
for $k = 0$ **to** $K-1$ **do**
  Draw samples with size S_2 and S_4.
  $v_{t,k}^{(1)} = v_{t,k-1}^{(1)} + \frac{1}{S_2}\sum_{i=1}^{S_2}(\nabla_x F(x_{t,k}, y_{t,k}; \xi_i) - \nabla_x F(x_{t,k-1}, y_{t,k-1}; \xi_i))$
  $v_{t,k}^{(2)} = v_{t,k-1}^{(2)} + \frac{1}{S_2}\sum_{i=1}^{S_2}(\nabla_y F(x_{t,k}, y_{t,k}; \xi_i) - \nabla_y F(x_{t,k-1}, y_{t,k-1}; \xi_i))$
  $u_{t,k} = u_{t,k-1} + \frac{1}{S_4}\sum_{i=1}^{S_4}(\nabla_y G(x_{t,k}, y_{t,k}; \zeta_i) - \nabla_y G(x_{t,k-1}, y_{t,k-1}; \zeta_i))$
  $y_{t,k+1} = y_{t,k} - \lambda u_{t,k}$.
end for
Select $s_t = \arg\min_k \|\tilde{\mathcal{G}}_\lambda(y_{t,k})\|$. Let $y_{t+1} = y_{t,s_t}$, $v_t^{(1)} = v_{t,s_t}^{(1)}$, $v_t^{(2)} = v_{t,s_t}^{(2)}$, $u_t = u_{t,s_t}$.
Compute Jacobian $J_t = \frac{1}{S_5}\sum_{i=1}^{S_5} \nabla_{xy}^2 G(x_t, y_{t+1}; \zeta_i)$.
Compute v_t via AID in Eq. (9).

Output: y_{t+1}, v_t, u_t.

6.2 Main Theorem for Bilevel Optimization

Theorem 2. *Under Assumption 1, 2, 3, 4 and 5, we set stepsize $\eta = \tilde{O}(\frac{\epsilon}{\kappa^3 L})$, $\eta_H = \tilde{O}(\frac{1}{\kappa^3 L})$, $\lambda = O(\frac{1}{L})$ and $\alpha = O(\frac{1}{L})$, batchsize $S_1 = \tilde{O}(\kappa^2\epsilon^{-2})$, $S_2 = \tilde{O}(\kappa^{-1}\epsilon^{-1})$, $S_3 = \tilde{O}(\kappa^6\epsilon^{-2})$, $S_4 = \tilde{O}(\kappa^3\epsilon^{-1})$, $S_5 = \tilde{O}(\kappa^2\epsilon^{-2})$ and $B = \tilde{O}(\kappa^2\epsilon^{-2})$, period $q = O(\kappa^2\epsilon^{-1})$, inner loop $K = O(\kappa)$ and $Q = \tilde{O}(\kappa)$, perturbation radius $r = \min\{\tilde{O}(\sqrt{\frac{\epsilon}{\rho_\Phi}}), \tilde{O}(\frac{\epsilon}{\kappa^3 L})\}$, threshold $t_{thres} = \tilde{O}(\frac{\kappa^3 L}{\sqrt{\rho_\Phi\epsilon}})$ and average movement $\bar{D} = \tilde{O}(\frac{\epsilon^2}{\kappa^6 L^2})$. Then our PRGDA algorithm requires complexity of $Gc(f,\epsilon) = \tilde{O}(\kappa^3\epsilon^{-3})$, $Gc(g,\epsilon) = \tilde{O}(\kappa^7\epsilon^{-3})$, $JV(g,\epsilon) = \tilde{O}(\kappa^5\epsilon^{-4})$ and $HV(g,\epsilon) = \tilde{O}(\kappa^6\epsilon^{-4})$ to achieve $O(\epsilon, \sqrt{\rho_\Phi\epsilon})$ second-order stationary point with high probability.*

7 Experiments

In this section we conduct the matrix sensing [Bhojanapalli *et al.*, 2016; Park *et al.*, 2017] experiment to validate the performance of out PRGDA algorithm for solving both minimax and bilevel problem. As a result of existing study on matrix sensing problem [Ge *et al.*, 2017], there is no spurious local minimum in this circumstance, *i.e.*, every local minimum is a

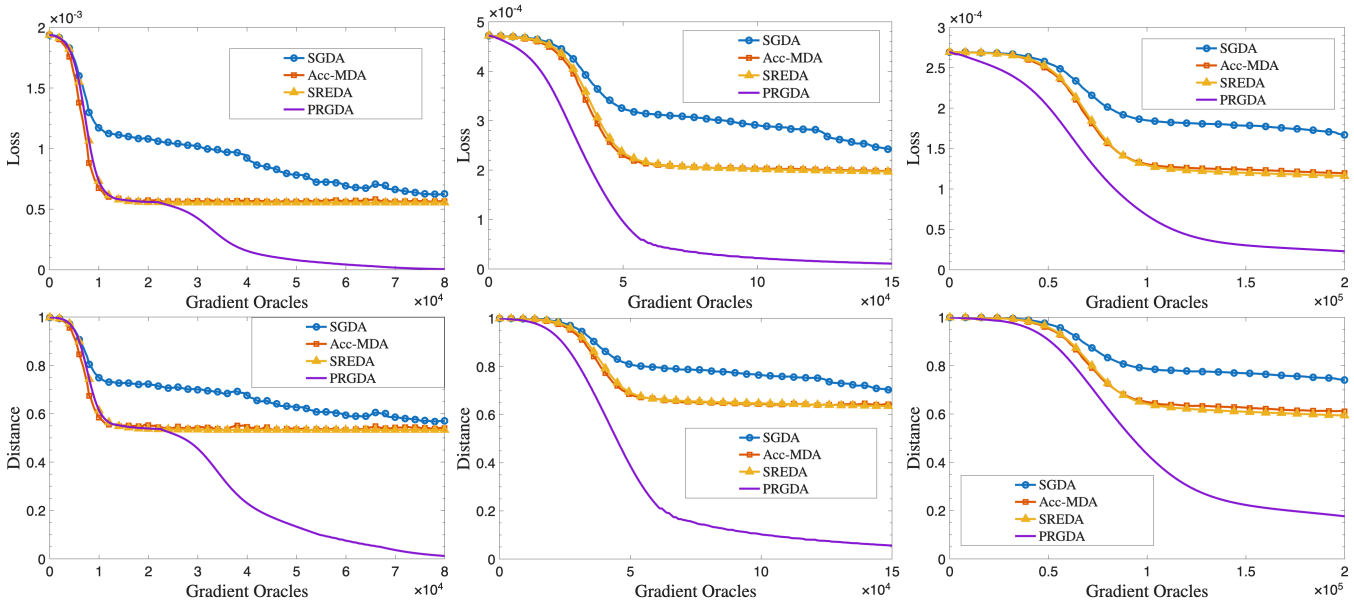

Figure 1: Experimental results of our robust low-rank matrix sensing task. The first three subfigures show the loss function value of $\Phi(U)$ against the number of gradient oracles with $d = 50$, $d = 75$, and $d = 100$ respectively. The last three subfigures show the ratio of distance $\|UU^T - M^*\|_F^2/\|M^*\|_F^2$ against the number of gradient oracles with $d = 50$, $d = 75$, and $d = 100$ respectively.

global minimum. Therefore, the capability of escaping saddle points of our algorithm can be verified by this experiment. We follow the experiment setup of [Chen *et al.*, 2022] to recover a low-rank symmetric matrix $M^* = U^*(U^*)^T$ where $U^* \in \mathbb{R}^{d\times r}$. Suppose we have n sensing matrices $\{A_i\}_{i=1}^n$ with n observations $b_i = \langle A_i, M^*\rangle$. Here the inner product of two matrices is defined by the trace $\langle X, Y\rangle = tr(X^TY)$. Then the optimization problem can be defined by

$$\min_{U\in\mathbb{R}^{d\times r}} \frac{1}{2}\sum_{i=1}^{n} L_i(U),\ L_i(U) = (\langle A_i, UU^T\rangle - b_i)^2 \quad (10)$$

7.1 Robust Optimization

Similar to the problem setting of [Yan *et al.*, 2019], we also introduce another variable y and add a robust term to make the model robust. Therefore, the optimization problem can be formulated by

$$\min_{U\in\mathbb{R}^{d\times r}} \max_{y\in\Delta_n} f(U, y) = \frac{1}{2}\sum_{i=1}^{n} y_i L_i(U) - (y_i - \frac{1}{n})^2 \quad (11)$$

where $\Delta_n = \{y \in \mathbb{R}^n | 0 \le y_i \le 1, \sum_{i=1}^n y_i = 1\}$ is the simplex in $\mathbb{R}^n$ and $L_i(U)$ is defined in Eq. (10). Moreover, it is easy to check there is no spurious local minimum given the strict saddle property in [Ge *et al.*, 2017].

The number of rows of matrix U is set to $d = 50$, $d = 75$ and $d = 100$ respectively and the number of columns is fixed as $r = 3$. The ground truth low-rank matrix M^* is generated by $M^* = U^*(U^*)^T$ where each entry of U^* is drawn from Gaussian distribution $\mathcal{N}(0, 1/d)$ independently. We randomly generate $n = 20d$ samples of sensing matrices $\{A_i\}_{i=1}^n$, $A_i \in \mathbb{R}^{d\times d}$ from standard Gaussian distribution and calculate the corresponding labels $b_i = \langle A_i, M^*\rangle$ hence

Algorithm	$d = 50$	$d = 75$	$d = 100$
SGDA	-0.0819	-0.0434	-0.0330
Acc-MDA	-0.0746	-0.0384	-0.0289
SREDA	-0.0744	-0.0385	-0.0283
PRGDA	**-0.0035**	**-0.0011**	**-0.0011**

Table 3: Smallest eigenvalue of $\nabla^2\Phi(U)$.

there is no noise in the synthetic data. The global minimum of loss function value $\Phi(U)$ should be 0 which can be achieved at point $U = U^*$ and $y = \mathbf{1}/n$.

Following the initialization in [Chen *et al.*, 2022], we randomly generalize a vector u_0 from Gaussian distribution and multiply it by a scalar to satisfy the condition $\|u_0\| \le \lambda_{max}(M^*)$ where we denote $\lambda_{max}(\cdot)$ as the maximum eigenvalue. The initial value is set to $U = [u_0, \mathbf{0}, \mathbf{0}]$. Each optimization algorithm shares the same initialization. Apart from our PRGDA algorithm, we run three baseline algorithms, SGDA, Acc-MDA and SREDA. The code is implemented on matlab. We choose $\eta = 0.001$, $\eta_H = 0.1$, $\lambda = 0.01$, $\bar{D} = r = 0.01$, $t_{thres} = 20$, $K = 5$, $S_2 = 40$ and $q = 25$.

We evaluate the performance of each algorithm by two criteria, loss function value of $\Phi(U)$ and the ratio of distance to the optimum $\|UU^T - M^*\|_F^2/\|M^*\|_F^2$. The experimental results of these two quantities versus the number of gradient oracles are shown in Figure 1.

From the experimental results we can see SGDA, Acc-MDA and SREDA cannot escape saddle points because the loss function value is far away from the global minimum 0, which is equivalent to local minimum in this task because of the strict saddle property. In contrast, we can see our PRGDA algorithm eventually converges to the global optimum U^* and

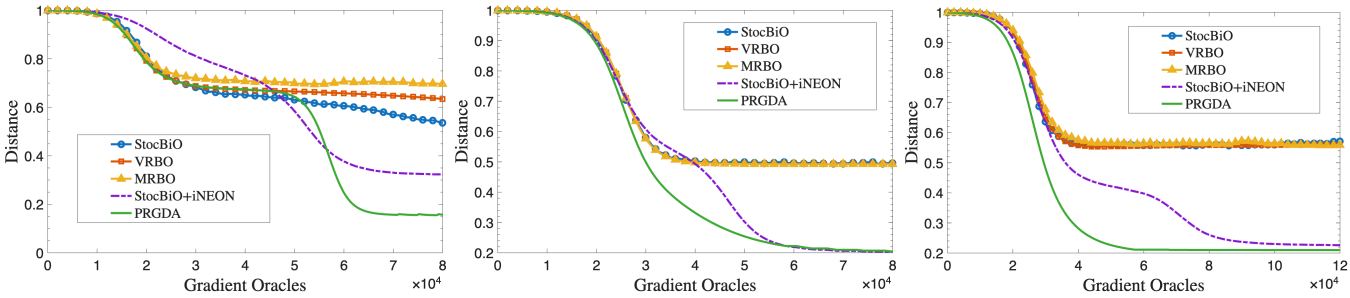

Figure 2: Experimental results of our hyper-representation learning of low-rank matrix sensing task. The ratio of distance $\|UU^T - M^*\|_F^2/\|M^*\|_F^2$ is shown against the number of gradient oracles with $d = 50$, $d = 75$, and $d = 100$ respectively.

achieves the best loss function value that is close to 0, which indicates its ability to escape saddle point. Especially in the case of $d = 50$, we can see clearly that our PRGDA algorithm jumps out of the trap of saddle point. Besides, in our experiment we also list the smallest eigenvalue of the Hessian matrix $\nabla^2\Phi(U)$ for each algorithm after they have converged. The results are shown in Table 3. We can see the value $\lambda_{min}(\nabla^2\Phi(U))$ of our method is the closest to 0 in all cases, which also verifies the performance of our PRGDA algorithm to find second-order stationary point.

7.2 Hyper-Representation Learning

We also conduct a hyper-representation learning experiment a verify the ability of our method to reach second-order stationary point in bilevel optimization. Recently, many methods in meta learning [Finn *et al.*, 2017; Nichol *et al.*, 2018] are designed to learn hyper-representations via two steps and separated dataset. The backbone is trained to extract better feature representations which can be applied to many different tasks. Based on these features a classifier is further learned on specific type of training data, which eventually forms a bilevel problem. In this experiment we also consider the matrix sensing task but conduct it in the hyper-representation learning manner.

The generation of U^*, M^*, A_i and b_i are the same as Section 7.1. We also set $d = 50$, $d = 75$ and $d = 100$. The number of samples is $n = 20d$. We split all samples into two dataset: a train dataset D_1 with 70% data and a validation dataset D_2 with 30% data. We define variable x to be the first $r-1$ columns of U and variable y to be the last column. Then the objective function is formulated by

$$\min_{x\in\mathbb{R}^{d\times(r-1)}} \frac{1}{2|D_1|}\sum_{i\in D_1} L_i(x, y^*(x)),$$
$$\text{where} \quad y^*(x) = \arg\min_{y\in\mathbb{R}^d} \frac{1}{2|D_2|}\sum_{i\in D_2} L_i(x, y) \qquad (12)$$

Here $L_i(\cdot)$ is defined in Eq. (10) since U is the concatenation of x and y.

We follow the initialization in Section 7.1 to set $x = [u_0, \mathbf{0}]$ and $y = \mathbf{0}$. We compare our PRGDA algorithm with four baselines, StocBiO, MRBO, VRBO and StocBiO + iNEON. We choose $\eta = 0.001$, $\eta_H = 0.1$, $\lambda = 0.01$, $\bar{D} = r = 0.01$, $t_{thres} = 20$, $K = 5$, $S_2 = 40$ and $q = 25$. We also use the ration of distance to optimum, *i.e.* $\|UU^T - M^*\|_F^2/\|M^*\|_F^2$ as the metric to evaluate the performance. The experimental results are shown in Figure 2.

From the experimental results we can see our PRGDA algorithm shows the best performance to reach second-order stationary point and approach the expected optimum. MRBO and VRBO do not escape saddle points during the experiment. In the case of $d = 50$, StocBiO performs better than MRBO and VRBO because the randomness of stochastic gradient serves as a kind of perturbation, while in variance-reduced algorithms the gradient estimator is closer to the full gradient. This result indicates the necessity of our method to make variance-reduced bilevel algorithm escape saddle points. StocBiO + iNEON also escapes saddle point probably but its convergence is slower than our method.

8 Conclusion

In this paper, we investigate the methods to achieve second-order optimality for nonconvex minimax and bilevel optimization. We propose a new algorithm PRGDA for stochastic nonconvex hierarchical optimization which is the first algorithm to find second-order stationary point for stochastic nonconvex-strongly-concave minimax optimization. In nonconvex-strongly-convex bilevel optimization, our method also achieves better gradient complexity to find local minimum. We prove that our method obtains the gradient complexity of $\tilde{O}(\epsilon^{-3})$ to achieve $O(\epsilon, \sqrt{\rho_\Phi\epsilon})$ second-order stationary point, which matches the best results of searching first-order stationary point under same conditions. We also conduct two numerical experiments to verify the performance of our algorithm.

Acknowledgements

This work was partially supported by NSF IIS 2347592, 2348169, DBI 2405416, CCF 2348306, CNS 2347617.

References

[Allen-Zhu and Li, 2018] Zeyuan Allen-Zhu and Yuanzhi Li. Neon2: Finding local minima via first-order oracles. *Advances in Neural Information Processing Systems*, 31:3716–3726, 2018.

[Bertinetto *et al.*, 2018] Luca Bertinetto, Joao F Henriques, Philip HS Torr, and Andrea Vedaldi. Meta-learning

with differentiable closed-form solvers. https://doi.org/10.48550/arXiv.1805.08136, 2018.

[Bhojanapalli *et al.*, 2016] Srinadh Bhojanapalli, Behnam Neyshabur, and Nati Srebro. Global optimality of local search for low rank matrix recovery. *Advances in Neural Information Processing Systems*, 29:3873–3881, 2016.

[Chen *et al.*, 2021] Ziyi Chen, Zhengyang Hu, Qunwei Li, Zhe Wang, and Yi Zhou. A cubic regularization approach for finding local minimax points in nonconvex minimax optimization. https://doi.org/10.48550/arXiv.2110.07098, 2021.

[Chen *et al.*, 2022] Zixiang Chen, Dongruo Zhou, and Quanquan Gu. Faster perturbed stochastic gradient methods for finding local minima. *International Conference on Algorithmic Learning Theory*, 167:176–204, 2022.

[Domke, 2012] Justin Domke. Generic methods for optimization-based modeling. *International Conference on Artificial Intelligence and Statistics*, 22:318–326, 2012.

[Du and Hu, 2019] Simon S Du and Wei Hu. Linear convergence of the primal-dual gradient method for convex-concave saddle point problems without strong convexity. *International Conference on Artificial Intelligence and Statistics*, 89:196–205, 2019.

[Feurer and Hutter, 2019] Matthias Feurer and Frank Hutter. *Hyperparameter optimization*. Springer International Publishing, 2019.

[Finn *et al.*, 2017] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *International Conference on Machine Learning*, 70:1126–1135, 2017.

[Franceschi *et al.*, 2018] Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimiliano Pontil. Bilevel programming for hyperparameter optimization and meta-learning. *International Conference on Machine Learning*, 80:1568–1577, 2018.

[Ge *et al.*, 2015] Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points—online stochastic gradient for tensor decomposition. *Conference on Learning Theory*, 40:797–842, 2015.

[Ge *et al.*, 2016] Rong Ge, Jason D Lee, and Tengyu Ma. Matrix completion has no spurious local minimum. *Advances in neural information processing systems*, 29:2973–2981, 2016.

[Ge *et al.*, 2017] Rong Ge, Chi Jin, and Yi Zheng. No spurious local minima in nonconvex low rank problems: A unified geometric analysis. *International Conference on Machine Learning*, 70:1233–1242, 2017.

[Ghadimi and Wang, 2018] Saeed Ghadimi and Mengdi Wang. Approximation methods for bilevel programming. https://doi.org/10.48550/arXiv.1802.02246, 2018.

[Goodfellow *et al.*, 2014] 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*, 27:2672–2680, 2014.

[Guo *et al.*, 2021] Zhishuai Guo, Quanqi Hu, Lijun Zhang, and Tianbao Yang. Randomized stochastic variance-reduced methods for multi-task stochastic bilevel optimization. https://doi.org/10.48550/arXiv.2105.02266, 2021.

[Heusel *et al.*, 2017] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in Neural Information Processing Systems*, 30:6626–6637, 2017.

[Hillar and Lim, 2013] Christopher J Hillar and Lek-Heng Lim. Most tensor problems are np-hard. *Journal of the ACM*, 60(6):1–39, 2013.

[Huang *et al.*, 2022] Feihu Huang, Shangqian Gao, Jian Pei, and Heng Huang. Accelerated zeroth-order and first-order momentum methods from mini to minimax optimization. *Journal of Machine Learning Research*, 23(36):1–70, 2022.

[Huang *et al.*, 2025] Minhui Huang, Xuxing Chen, Kaiyi Ji, Shiqian Ma, and Lifeng Lai. Efficiently escaping saddle points in bilevel optimization. *Journal of Machine Learning Research*, 26(1):1–61, 2025.

[Ji *et al.*, 2021] Kaiyi Ji, Junjie Yang, and Yingbin Liang. Bilevel optimization: Convergence analysis and enhanced design. *International Conference on Machine Learning*, 139:4882–4892, 2021.

[Jin *et al.*, 2017] Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M Kakade, and Michael I Jordan. How to escape saddle points efficiently. *International Conference on Machine Learning*, 70:1724–1732, 2017.

[Jin *et al.*, 2020] Chi Jin, Praneeth Netrapalli, and Michael I Jordan. What is local optimality in nonconvex-nonconcave minimax optimization? *International Conference on Machine Learning*, 119:4880–4889, 2020.

[Khanduri *et al.*, 2021] Prashant Khanduri, Siliang Zeng, Mingyi Hong, Hoi-To Wai, Zhaoran Wang, and Zhuoran Yang. A near-optimal algorithm for stochastic bilevel optimization via double-momentum. *Advances in Neural Information Processing Systems*, 34:30271–30283, 2021.

[Li, 2019] Zhize Li. Ssrgd: Simple stochastic recursive gradient descent for escaping saddle points. *Advances in Neural Information Processing Systems*, 32:1523–1533, 2019.

[Lin *et al.*, 2020a] Tianyi Lin, Chi Jin, and Michael I Jordan. Near-optimal algorithms for minimax optimization. *Conference on Learning Theory*, 125:2738–2779, 2020.

[Lin *et al.*, 2020b] Tianyi Lin, Chi Jin, and Michael I Jordan. On gradient descent ascent for nonconvex-concave minimax problems. *International Conference on Machine Learning*, 119:6083–6093, 2020.

[Luo and Chen, 2022] Luo Luo and Cheng Chen. Finding second-order stationary point for nonconvex-strongly-concave minimax problem. *Advances in Neural Information Processing Systems*, 35:36667–36679, 2022.

[Luo *et al.*, 2020] Luo Luo, Haishan Ye, and Tong Zhang. Stochastic recursive gradient descent ascent for stochastic nonconvex-strongly-concave minimax problems. *Advances in Neural Information Processing Systems*, 33:20566–20577, 2020.

[Madry *et al.*, 2018] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. https://doi.org/10.48550/arXiv.1706.06083, 2018.

[Nemirovski, 2004] 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.

[Nesterov and Polyak, 2006] Yurii Nesterov and Boris T Polyak. Cubic regularization of newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

[Nichol *et al.*, 2018] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. https://doi.org/10.48550/arXiv.1803.02999, 2018.

[Nouiehed *et al.*, 2019] Maher Nouiehed, Maziar Sanjabi, Tianjian Huang, and Jason D. Lee. Solving a class of non-convex min-max games using iterative first order methods. *Advances in Neural Information Processing Systems*, 32:14934–14942, 2019.

[Park *et al.*, 2017] Dohyung Park, Anastasios Kyrillidis, Constantine Carmanis, and Sujay Sanghavi. Non-square matrix sensing without spurious local minima via the burer-monteiro approach. *International Conference on Artificial Intelligence and Statistics*, 54:65–74, 2017.

[Pedregosa, 2016] Fabian Pedregosa. Hyperparameter optimization with approximate gradient. *International Conference on Machine Learning*, 48:737–746, 2016.

[Rafique *et al.*, 2022] Hassan Rafique, Mingrui Liu, Qihang Lin, and Tianbao Yang. Weakly-convex–concave min–max optimization: provable algorithms and applications in machine learning. *Optimization Methods and Software*, 37(3):1087–1121, 2022.

[Shaban *et al.*, 2019] Amirreza Shaban, Ching-An Cheng, Nathan Hatch, and Byron Boots. Truncated back-propagation for bilevel optimization. *International Conference on Artificial Intelligence and Statistics*, 89:1723–1732, 2019.

[Tran Dinh *et al.*, 2020] Quoc Tran Dinh, Deyi Liu, and Lam Nguyen. Hybrid variance-reduced sgd algorithms for minimax problems with nonconvex-linear function. *Advances in Neural Information Processing Systems*, 33:11096–11107, 2020.

[Wai *et al.*, 2018] Hoi-To Wai, Zhuoran Yang, Zhaoran Wang, and Mingyi Hong. Multi-agent reinforcement learning via double averaging primal-dual optimization. *Advances in Neural Information Processing Systems*, 31:9649–9660, 2018.

[Xu *et al.*, 2018] Yi Xu, Rong Jin, and Tianbao Yang. First-order stochastic algorithms for escaping from saddle points in almost linear time. *Advances in Neural Information Processing Systems*, 31:5530–5540, 2018.

[Xu *et al.*, 2023] Zi Xu, Huiling Zhang, Yang Xu, and Guanghui Lan. A unified single-loop alternating gradient projection algorithm for nonconvex–concave and convex–nonconcave minimax problems. *Mathematical Programming*, 201(1):635–706, 2023.

[Yan *et al.*, 2019] Yan Yan, Yi Xu, Qihang Lin, Lijun Zhang, and Tianbao Yang. Stochastic primal-dual algorithms with faster convergence than $O(1/\sqrt{T})$ for problems without bilinear structure. https://doi.org/10.48550/arXiv.1904.10112, 2019.

[Yan *et al.*, 2020] Yan Yan, Yi Xu, Qihang Lin, Wei Liu, and Tianbao Yang. Optimal epoch stochastic gradient descent ascent methods for min-max optimization. *Advances in Neural Information Processing Systems*, 33:5789–5800, 2020.

[Yang *et al.*, 2021] Junjie Yang, Kaiyi Ji, and Yingbin Liang. Provably faster algorithms for bilevel optimization. *Advances in Neural Information Processing Systems*, 34:13670–13682, 2021.

[Zhang *et al.*, 2021] Siqi Zhang, Junchi Yang, Cristóbal Guzmán, Negar Kiyavash, and Niao He. The complexity of nonconvex-strongly-concave minimax optimization. *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 161:482–492, 2021.