

Accelerated Proximal Alternating Gradient-Descent-Ascent for Nonconvex Minimax Machine Learning

Ziyi Chen

Electrical & Computer Engineering
University of Utah
Salt Lake City, US
u1276972@utah.edu

Shaocong Ma

Electrical & Computer Engineering
University of Utah
Salt Lake City, US
s.ma@utah.edu

Yi Zhou

Electrical & Computer Engineering
University of Utah
Salt Lake City, US
yi.zhou@utah.edu

Abstract—Alternating gradient-descent-ascent (AltGDA) is an optimization algorithm that has been widely used for model training in various machine learning applications, which aims to solve a nonconvex minimax optimization problem. However, the existing studies show that it suffers from a high computation complexity in nonconvex minimax optimization. In this paper, we develop a single-loop and fast AltGDA-type algorithm that leverages proximal gradient updates and momentum acceleration to solve regularized nonconvex minimax optimization problems. By leveraging the momentum acceleration technique, we prove that the algorithm converges to a critical point in nonconvex minimax optimization and achieves a computation complexity in the order of $\mathcal{O}(\kappa^{\frac{11}{6}}\epsilon^{-2})$, where ϵ is the desired level of accuracy and κ is the problem’s condition number. Such a computation complexity improves the state-of-the-art complexities of single-loop GDA and AltGDA algorithms (see the summary of comparison in Table I). We demonstrate the effectiveness of our algorithm via an experiment on adversarial deep learning.

Index Terms—Minimax optimization, alternating gradient descent ascent, proximal gradient, momentum, complexity.

I. INTRODUCTION

Minimax optimization is an emerging and important optimization framework that covers a variety of modern machine learning applications. Some popular application examples include generative adversarial networks (GANs) [13], adversarial machine learning [37], game theory [10], reinforcement learning [34], etc. A standard minimax optimization problem is written as follows, where f is a differentiable bivariate function.

$$\min_{x \in \mathbb{R}^m} \max_{y \in \mathcal{Y}} f(x, y).$$

A basic algorithm for solving the above minimax optimization problem is the gradient-descent-ascent (GDA), which simultaneously performs gradient descent update and gradient ascent update on the variables x and y , respectively, i.e., $x_{t+1} = x_t - \eta_x \nabla_1 f(x_t, y_t)$, $y_{t+1} = y_t + \eta_y \nabla_2 f(x_t, y_t)$. Here, ∇_1 and ∇_2 denote the gradient operator with regard to the first and the second variable, respectively. The convergence rate of GDA has been studied under various types of geometries of the minimax problem, including strongly-convex-strongly-concave geometry [9], nonconvex-(strongly)-concave geometry [23]

and Lojasiewicz-type geometry [5]. Recently, by leveraging the popular momentum technique [3], [11], [20], [21], [28], [38] for accelerating gradient-based algorithms, accelerated variants of GDA have been proposed for strongly-convex-strongly-concave [45] and nonconvex-strongly-concave [17] minimax optimization. There are other accelerated GDA-type algorithms that achieve a near-optimal complexity [22], [46], but they involve complex nested-loop structures and require function smoothing with many fine-tuned hyper-parameters, which are not used in practical minimax machine learning.

Another important variant of GDA that has been widely used in practical training of minimax machine learning is the alternating-GDA (AltGDA) algorithm, which updates the two variables x and y alternatively via $x_{t+1} = x_t - \eta_x \nabla_1 f(x_t, y_t)$, $y_{t+1} = y_t + \eta_y \nabla_2 f(x_{t+1}, y_t)$. Compared with the update of GDA, the y -update of AltGDA uses the fresh x_{t+1} instead of the previous x_t , and it is shown to converge faster than the standard GDA algorithm [2], [4], [12], [42]. Despite the popularity of the AltGDA algorithm, it is shown to suffer from a high computation complexity in nonconvex minimax optimization. Therefore, this study aims to improve the complexity of AltGDA by leveraging momentum acceleration techniques. In particular, in the existing literature, the convergence of momentum accelerated AltGDA is only established for convex-concave minimax problems [43] and bilinear minimax problems [12], and it has not been explored in nonconvex minimax optimization that covers modern machine learning applications. Therefore, this study aims to fill in this gap by developing a single-loop proximal-AltGDA algorithm with momentum acceleration for solving a class of regularized nonconvex minimax problems, and analyze its convergence and computation complexity.

A. Our Contribution

We are interested in a class of regularized nonconvex-strongly-concave minimax optimization problems, where the regularizers are convex functions that can be possibly non-smooth. To solve this class of minimax problems, we propose a single-loop proximal-AltGDA with momentum algorithm

(referred to as proximal-AltGDAm). The algorithm takes single-loop updates that consist of a proximal gradient descent update with the heavy-ball momentum, and a proximal gradient ascent update with the Nesterov's momentum. Our algorithm extends the applicability of the conventional momentum acceleration schemes (heavy-ball and Nesterov's momentum) for nonconvex minimization to nonconvex minimax optimization.

We study the convergence property of Proximal-AltGDAm. Specifically, under standard smoothness assumptions on the objective function and by viewing the accelerated alternating GDA updates as inexact accelerated gradient updates, we develop an analysis to show that every limit point of the parameter sequences generated by the algorithm is a critical point of the nonconvex regularized minimax problem. Moreover, to achieve an ϵ -accurate critical point, the overall computation complexity (i.e., number of gradient and proximal mapping evaluations) is of the order $\mathcal{O}(\kappa^{\frac{11}{6}}\epsilon^{-2})$, where $\kappa > 1$ is the condition number of the problem. Thanks to momentum acceleration and a tight analysis in our technical proof, such a computation complexity is lower than that of the existing single-loop GDA-type algorithms. See Table I for a summary of comparison of the computation complexities and Appendix E for their derivation.

Table I
COMPARISON OF COMPUTATION COMPLEXITY (NUMBER OF GRADIENT AND PROXIMAL MAPPINGS EVALUATIONS) OF THE EXISTING SINGLE-LOOP GDA-TYPE ALGORITHMS IN NONCONVEX-STRONGLY-CONCAVE MINIMAX OPTIMIZATION, WHERE $\kappa \geq 1$ IS THE PROBLEM CONDITION NUMBER.

	Alternating update	Momentum acceleration	Computation complexity
(Chen, et.al) [5]	×	×	$\mathcal{O}(\kappa^6\epsilon^{-2})$
(Huang, et.al) [17]	×	✓	$\mathcal{O}(\kappa^3\epsilon^{-2})$
(Lin, et.al) [23]	×	×	$\mathcal{O}(\kappa^2\epsilon^{-2})$
(Xu, et.al) [42]	✓	×	$\mathcal{O}(\kappa^5\epsilon^{-2})$
(Boč and Böhm) [4]	✓	×	$\mathcal{O}(\kappa^2\epsilon^{-2})$
This work	✓	✓	$\mathcal{O}(\kappa^{\frac{11}{6}}\epsilon^{-2})$

B. Other Related Work

GDA-type algorithms: [27] studied a slight variant of GDA by replacing gradients with subgradients for convex-concave non-smooth minimax optimization. [42] studied AltGDA which applies ℓ_2 regularizer to the local objective function of GDA followed by projection onto the constraint sets and obtained its convergence rate under nonconvex-concave and convex-nonconcave geometry. [26] also studied two variants of GDA, Extra-gradient and Optimistic GDA, and obtained linear convergence rate under bilinear geometry and strongly-convex-strongly-concave geometry. [6], [18] studied GDA in continuous time dynamics using differential equations. [1] analyzed a second-order variant of the GDA algorithm.

Many other studies have developed stochastic GDA-type algorithms. [23], [44] analyzed stochastic GDA and stochastic AltGDA respectively. Variance reduction techniques have been applied to stochastic minimax optimization, including SVRG-

based [8], [44], SARAH/SPIDER-based [25], [41], STORM [34] and its gradient free version [16].

GDA-type algorithms with momentum: For strongly-convex-strongly-concave problems, [45] studied accelerated GDA with negative momentum. [22], [39] developed nested-loop AltGDA algorithms with Nesterov's Accelerated Gradient Descent that achieve improved complexities. For convex-concave problems, [7] analyzed stable points of both GDA and optimistic GDA that apply negative momentum for acceleration. Moreover, for nonconvex-(strongly)-concave problems, [40] developed a single-loop GDA with momentum and Hessian preconditioning and studied its convergence to a local minimax point. [17] developed a mirror descent ascent algorithm with momentum which includes GDAm as a special case. [30] studied nested-loop GDA where multiple gradient ascent steps are performed, and they also studied the momentum-accelerated version. Similarly, [16], [34] developed GDA with momentum scheme and STORM variance reduction, and a similar version of this algorithm is extended to minimax optimization on Riemann manifold [15]. [14] developed a single-loop Primal-Dual Stochastic Momentum algorithm with ADAM-type methods.

II. PROBLEM FORMULATION AND PRELIMINARIES

In this section, we introduce the problem formulation and technical assumptions. We consider the following class of regularized minimax optimization problems.

$$\min_{x \in \mathbb{R}^m} \max_{y \in \mathcal{Y}} f(x, y) + g(x) - h(y), \quad (\text{P})$$

where $f : \mathbb{R}^m \times \mathcal{Y} \rightarrow \mathbb{R}$ is differentiable and nonconvex-strongly-concave, g, h are possibly non-smooth convex regularizers, and \mathcal{Y} is a bounded set with diameter R . In particular, define the envelope function $\Phi(x) := \max_{y \in \mathcal{Y}} f(x, y) - h(y)$, then the problem (P) can be rewritten as the minimization problem $\min_{x \in \mathbb{R}^m} \Phi(x) + g(x)$.

Throughout the paper, we adopt the following assumptions on the problem (P). These are standard assumptions that have been considered in the existing literature [5], [23].

Assumption 1. *The minimax problem (P) satisfies:*

- 1) *Function $f(\cdot, \cdot)$ is L -smooth and function $f(x, \cdot)$ is μ -strongly concave for all x ;*
- 2) *Function $(\Phi + g)(x)$ is bounded below and has compact sub-level sets;*
- 3) *Function g, h are proper and convex.*

In particular, item 3 of the above assumption allows the regularizers g, h to be any convex function that can be possibly non-smooth. Next, consider the mapping $y^*(x) = \arg \max_{y \in \mathcal{Y}} f(x, y) - h(y)$, which is uniquely defined due to the strong concavity of $f(x, \cdot)$. The following proposition is proved in [5], [23] that characterizes the Lipschitz continuity of the mapping $y^*(x)$ and the smoothness of the function $\Phi(x)$. Throughout the paper, we denote $\kappa = L/\mu > 1$ as the condition number, and denote $\nabla_1 f(x, y), \nabla_2 f(x, y)$ as the gradients with respect to the first and the second input variables, respectively.

Proposition 1 ([5], [23]). *Let Assumption 1 hold. Then, the mapping $y^*(x)$ is κ -Lipschitz continuous and the function $\Phi(x)$ is $L(1 + \kappa)$ -smooth with $\nabla\Phi(x) = \nabla_1 f(x, y^*(x))$.*

Lastly, recall that the minimax problem (P) is equivalent to the minimization problem $\min_{x \in \mathbb{R}^m} \Phi(x) + g(x)$. Therefore, the optimization goal of the minimax problem (P) is to find a critical point x^* of the nonconvex function $\Phi(x) + g(x)$ that satisfies the optimality condition $\mathbf{0} \in \partial(\Phi + g)(x^*)$, where ∂ denotes the subdifferential operator.

III. PROXIMAL ALTERNATING GDA WITH MOMENTUM

In this section, we propose a single-loop proximal alternating-GDA with momentum (proximal-AltGDAm) algorithm to solve the regularized minimax problem (P).

We first recall the update rules of the basic proximal alternating-GDA (proximal-AltGDA) algorithm [4] for solving the problem (P). Specifically, proximal-AltGDA alternates between the following two proximal-gradient updates (a.k.a. forward-backward splitting updates [24]).

(Proximal-AltGDA):

$$\begin{cases} x_{t+1} = \text{prox}_{\eta_x g}(x_t - \eta_x \nabla_1 f(x_t, y_t)), \\ y_{t+1} = \text{prox}_{\eta_y h}(y_t + \eta_y \nabla_2 f(x_{t+1}, y_t)). \end{cases}$$

To elaborate, the first update is a proximal gradient descent update that aims to minimize the nonconvex function $f(x, y_t) + g(x)$ from the current point x_t , and the second update is a proximal gradient ascent update that aims to maximize the strongly-concave function $f(x_{t+1}, y) - h(y)$ from the current point y_t . More specifically, the two proximal gradient mappings are formally defined as

$$\begin{aligned} \text{prox}_{\eta_x g}(x_t - \eta_x \nabla_1 f(x_t, y_t)) \\ := \underset{u \in \mathbb{R}^m}{\text{argmin}} \left\{ g(u) + \frac{1}{2\eta_x} \|u - x_t + \eta_x \nabla_1 f(x_t, y_t)\|^2 \right\}, \\ \text{prox}_{\eta_y h}(y_t + \eta_y \nabla_2 f(x_{t+1}, y_t)) \\ := \underset{v \in \mathcal{Y}}{\text{argmin}} \left\{ h(v) + \frac{1}{2\eta_y} \|v - y_t - \eta_y \nabla_2 f(x_{t+1}, y_t)\|^2 \right\}. \end{aligned}$$

Compared with the standard (proximal) GDA algorithm [5], [23], the proximal ascent step of proximal-AltGDA evaluates the gradient at the freshest point x_{t+1} instead of x_t . Such an alternative update is widely used in practice and usually leads to better convergence properties [2], [4], [12], [42].

Next, we introduce momentum schemes to accelerate the convergence of proximal-AltGDA. As the two proximal gradient steps of proximal-AltGDA are used to solve two different types of optimization problems, namely, the nonconvex problem $f(x, y_t) + g(x)$ and the strongly-concave problem $f(x_{t+1}, y) - h(y)$, we consider applying different momentum schemes to accelerate these proximal gradient updates. Specifically, the proximal gradient descent step minimizes a composite nonconvex function, and we apply the heavy-ball momentum scheme [33] that was originally designed for accelerating nonconvex optimization. Therefore, we propose the following proximal gradient descent with heavy-ball momentum update for minimizing the nonconvex part of the problem (P).

(Heavy-ball momentum):

$$\begin{cases} \tilde{x}_t = x_t + \beta(x_t - x_{t-1}), \\ x_{t+1} = \text{prox}_{\eta_x g}(\tilde{x}_t - \eta_x \nabla_1 f(x_t, y_t)). \end{cases}$$

To explain, the first step is an extrapolation step that applies the momentum term $\beta(x_t - x_{t-1})$ (with momentum coefficient $\beta > 0$), and the second proximal gradient step updates the extrapolation point \tilde{x}_t using the original gradient $\nabla_1 f(x_t, y_t)$. In conventional gradient-based optimization, gradient descent with such a heavy-ball momentum is guaranteed to find a critical point of smooth nonconvex functions [31], [32].

On the other hand, as the proximal gradient ascent step of proximal-AltGDA maximizes a composite strongly-concave function, we are motivated to apply the popular Nesterov's momentum scheme [29], which was originally designed for accelerating strongly-concave (convex) optimization. Specifically, we propose the following proximal gradient ascent with Nesterov's momentum update for maximizing the strongly-concave part of the problem (P).

(Nesterov's momentum):

$$\begin{cases} \tilde{y}_t = y_t + \gamma(y_t - y_{t-1}), \\ y_{t+1} = \text{prox}_{\eta_y h}(\tilde{y}_t + \eta_y \nabla_2 f(x_{t+1}, \tilde{y}_t)). \end{cases} \quad (1)$$

To elaborate, the first step is a regular extrapolation step that involves momentum, which is the same as the first step of the heavy-ball scheme. The only difference from the heavy-ball scheme is that the starting point of the second proximal gradient step changes from y_t to its extrapolated point \tilde{y}_t .

We refer to the above algorithm design as **proximal-AltGDA with momentum (proximal-AltGDAm)**, and the algorithm updates are formally presented in Algorithm 1. It can be seen that proximal-AltGDAm is a simple algorithm that has a single loop structure, and adopts alternating updates with momentum acceleration. More importantly, it involves only standard hyperparameters such as the learning rates and momentum parameters and therefore is easy to implement in practice. Such a practical algorithm is much simpler than the other accelerated GDA-type algorithms that involve complex nested-loop structure and require fine-tuned function smoothing [22], [46].

Algorithm 1 Proximal Alternating GDA with Momentum (proximal-AltGDAm)

Input: Initialization $x_{-1} = x_0, y_{-1} = y_0$, learning rates η_x, η_y , momentum parameters β, γ .

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

$$\begin{aligned} \tilde{x}_t &= x_t + \beta(x_t - x_{t-1}), \\ x_{t+1} &= \text{prox}_{\eta_x g}(\tilde{x}_t - \eta_x \nabla_1 f(x_t, y_t)), \\ \tilde{y}_t &= y_t + \gamma(y_t - y_{t-1}), \\ y_{t+1} &= \text{prox}_{\eta_y h}(\tilde{y}_t + \eta_y \nabla_2 f(x_{t+1}, \tilde{y}_t)). \end{aligned}$$

end

Output: x_T, y_T .

IV. CONVERGENCE AND COMPUTATION COMPLEXITY OF PROXIMAL-ALTGDAm

Although the proposed proximal-AltGDAm algorithm applies the popular heavy-ball momentum and Nesterov's momentum to the GDA updates in a straightforward way, its convergence analysis cannot directly follow from the existing studies of momentum accelerated gradient descent algorithms. To explain more specifically, notice that in the x -proximal gradient update of Algorithm 1, it involves the partial gradient $\nabla_1 f(x_t, y_t)$, which corresponds to the gradient of the time-varying function $f(\cdot, y_t)$ (since y_t changes over time t). Similarly, the y -proximal gradient update involves the gradient of another time-varying function $f(x_{t+1}, \cdot)$. Therefore, both momentum accelerated proximal gradient updates are actually applied to time-varying functions due to the nature of GDA updates. In this sense, the existing analysis of momentum accelerated gradient descent algorithms cannot be directly applied to analyze this algorithm.

To analyze the convergence of Algorithm 1, we first study the x -proximal gradient update with heavy-ball momentum and obtain the following characterization of per-iteration progress on the objective function value.

Proposition 2. *Let Assumption 1 hold. Then, the parameter sequences $\{x_t, y_t\}_t$ generated by proximal-AltGDAm satisfy, for all $t = 0, 1, 2, \dots$,*

$$\begin{aligned} & \Phi(x_{t+1}) + g(x_{t+1}) \\ & \leq \Phi(x_t) + g(x_t) - \left(\frac{1-\beta}{2\eta_x} - 2L\kappa^{\frac{11}{6}} \right) \|x_{t+1} - x_t\|^2 \\ & \quad + \frac{\beta}{2\eta_x} \|x_t - x_{t-1}\|^2 + \frac{L}{4\kappa^{\frac{11}{6}}} \|y^*(x_t) - y_t\|^2. \end{aligned} \quad (2)$$

The above proposition tracks the per-iteration optimization progress made by the x -proximal gradient update with heavy-ball momentum. To elaborate, the increment terms $\|x_{t+1} - x_t\|^2, \|x_t - x_{t-1}\|^2$ are induced by the heavy-ball momentum scheme. Moreover, since the x -update uses the partial gradient $\nabla_1 f(x_t, y_t)$ to approximate the exact gradient $\nabla \Phi(x_t) = \nabla_1 f(x_t, y^*(x_t))$, it naturally induces a tracking error term $\|y_t - y^*(x_t)\|^2$ that tracks the optimization gap of the y -update. Hence, we need to further bound this tracking error term by analyzing the y -proximal gradient update with Nesterov's momentum, which we explore next.

As explained earlier, the momentum accelerated y -updates in proximal-AltGDAm are applied to a time-varying strongly-concave function $f(x_{t+1}, \cdot)$. Hence, the tracking error term $\|y_t - y^*(x_t)\|^2$ cannot be directly bounded using the standard convergence result of Nesterov's accelerated proximal gradient algorithm [28]. Instead, we can view these y -updates as inexact accelerated proximal gradient updates. To elaborate, note that in the t -th iteration, the y -proximal gradient update is applied to the function $f(x_{t+1}, \cdot)$. Then, we can rewrite the y -updates in all the previous iterations $k = 0, 1, \dots, t-1$ as follows.

$$\begin{aligned} y_{k+1} &= \text{prox}_{\eta_y h}(\tilde{y}_k + \eta_y \nabla_2 f(x_{t+1}, \tilde{y}_k) \\ &\quad + \eta_y \underbrace{[\nabla_2 f(x_{k+1}, \tilde{y}_k) - \nabla_2 f(x_{t+1}, \tilde{y}_k)]}_{\mathbf{e}_{k+1}}). \end{aligned} \quad (3)$$

That is, we can view all the previous y -updates as applied to the fixed function $f(x_{t+1}, \cdot)$ with an inexactness \mathbf{e}_{k+1} involved in the computation of the partial gradient $\nabla_2 f(x_{t+1}, \tilde{y}_k)$. In summary, the y -updates of proximal-AltGDAm can be understood as inexact accelerated gradient updates applied to the function $f(x_{t+1}, \cdot)$ at time t . In particular, under the smoothness condition in Assumption 1, the inexactness is bounded as $\|\mathbf{e}_{k+1}\| \leq L\|x_{k+1} - x_{t+1}\|$. Consequently, we can leverage the existing convergence result of inexact accelerated gradient algorithm [36] to bound the optimality gap $\|y_t - y^*(x_t)\|^2$ as follows.

Proposition 3. *Let Assumption 1 hold. Choose learning rate $\eta_y = \frac{1}{L}$ and momentum parameter $\gamma = \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$. Then, the parameter sequences $\{x_t, y_t\}$ generated by proximal-AltGDAm satisfy, for all $t = 0, 1, 2, \dots$*

$$\begin{aligned} \|y_{t+1} - y^*(x_{t+1})\|^2 &\leq \frac{2R\kappa}{\sqrt{L}} (1 - \kappa^{-0.5})^{t+1} \\ &\quad + \frac{6\kappa^2}{\sqrt{L}} \sum_{j=1}^t (1 - \kappa^{-0.5})^{t+1-j/2} \|x_{j+1} - x_j\|. \end{aligned} \quad (4)$$

Intuitively, in the above bound, the first term on the right hand side corresponds to the normal accelerated convergence rate, and the other term is induced by the inexactness \mathbf{e}_k . As both terms are scaled by the accelerated convergence factor $(1 - \kappa^{-0.5})$, we expect that the above bound converges fast and further facilitates the convergence of eq. (2). Next, substituting eq. (4) into eq. (2) and telescoping, we obtain the following asymptotic stability properties of proximal-AltGDAm.

Corollary 1. *Under the conditions of Proposition 3 and stepsize $\eta_x \leq 1/(16L\kappa^{\frac{11}{6}})$, the sequences $\{x_t, y_t\}_t$ generated by proximal-AltGDAm satisfy*

$$\|x_{t+1} - x_t\|, \|y_t - y^*(x_t)\|, \|y_{t+1} - y_t\| \xrightarrow{t} 0.$$

Remark 1. In [5], the proximal-GDA algorithm (without alternating update and momentum) uses a small learning rate $\eta_x \leq \mathcal{O}(L^{-2}\kappa^{-3})$ to establish convergence. As a comparison, proximal-AltGDAm allows to choose a much larger learning rate $\eta_x \leq \mathcal{O}(L^{-1}\kappa^{-\frac{11}{6}})$ to guarantee a faster convergence (proved later), thanks to the momentum acceleration schemes.

Therefore, if we can show that $\{x_t\}_t$ converges to a desired critical point x^* , then the above stability properties of proximal-AltGDAm implies that $\{y_t\}_t$ converges to the corresponding best response point $y^*(x^*)$.

To further characterize the global convergence property of proximal-AltGDAm, we define the following proximal gradient mapping associated with the objective function $\Phi(x) + g(x)$.

$$G(x) := \frac{1}{\eta_x} \left(x - \text{prox}_{\eta_x g}(x - \eta_x \nabla \Phi(x)) \right). \quad (5)$$

The proximal gradient mapping is a standard metric for evaluating the optimality of nonconvex composite optimization problems [28]. It can be shown that x is a critical point of $(\Phi + g)(x)$ if and only if $G(x) = 0$, and it reduces to the normal gradient when there is no regularizer g . Hence, our

convergence criterion is to find a near-critical point x that satisfies $\|G(x)\| \leq \epsilon$ for some pre-determined accuracy $\epsilon > 0$.

Next, by leveraging Proposition 2 and Proposition 3, we obtain the following global convergence result of proximal-AltGDA and characterize its computational complexity (number of gradient and proximal mapping evaluations).

Theorem 1 (Global convergence). *Under the conditions of Proposition 3 and stepsize $\eta_x \leq 1/(16L\kappa^{11/6})$, the sequence $\{x_t\}_t$ generated by proximal-AltGDA is bounded and has a compact set of limit points. Also, every such limit point is a critical point of $(\Phi + g)(x)$. Moreover, the total number of iterations T required to achieve $\min_{0 \leq t \leq T} \|G(x_t)\| \leq \epsilon$ is $T = \mathcal{O}(\kappa^{11/6}\epsilon^{-2})$, which is also the order of the required computational complexity.*

To elaborate, the first statement of Theorem 1 shows that the sequence generated by proximal-AltGDA converges to critical points of the minimax problem. The second statement proves that the computation complexity of proximal-AltGDA for finding a near-critical point is of the order $\mathcal{O}(\kappa^{11/6}\epsilon^{-2})$, which strictly improves the complexity $\mathcal{O}(\kappa^2\epsilon^{-2})$ of both GDA [23] and proximal-AltGDA [4] in nonconvex-strongly concave minimax optimization. We note that the improvement is substantial when the problem condition number $\kappa = L/\mu$ is large, while the dependence on ϵ^{-2} is generally unimprovable in nonconvex optimization. In addition, thanks to the momentum acceleration schemes, our choice of learning rate $\eta_x = \mathcal{O}(L^{-1}\kappa^{-11/6})$ is more flexible than that $\eta_x = \mathcal{O}(L^{-1}\kappa^{-2})$ adopted by these GDA-type algorithms. These improvements are not only attributed to momentum acceleration, but also to the elaborate selection of the coefficients in the proof that aims to minimize the dependence of the complexity on κ .

V. EXPERIMENTS

In this section, we compare proximal-AltGDA with the standard proximal-GDA/AltGDA algorithm [5] and the single-loop accelerated AltGDA algorithm (APDA) [47]. All these algorithms have a single-loop structure.

We consider solving the following regularized Wasserstein robustness model (WRM) [37] using the MNIST dataset [19].

$$\min_{\theta} \max_{\{\xi_i\}_{i=1}^n} \frac{1}{n} \sum_{i=1}^n \left[\ell(h_{\theta}(\xi_i), y_i) - \lambda \|\xi_i - x_i\|^2 \right] - \lambda_1 \sum_{i=1}^n \|\xi_i\|_1 + \frac{\lambda_2}{2} \|\theta\|^2, \quad (6)$$

where $n = 60k$ is the number of training samples, θ is the model parameter, (x_i, y_i) corresponds to the i -th data sample and label, respectively, and ξ_i is the adversarial sample corresponding to x_i . We choose the cross-entropy loss function ℓ . We add an ℓ_1 regularization to impose sparsity on the adversarial examples, and add an ℓ_2 regularization to prevent divergence of the model parameters.

We set $\lambda = 1.0$ that suffices to make the maximization part be strongly-concave, and set $\lambda_1 = \lambda_2 = 10^{-4}$. We use a convolution network that consists of two convolution blocks

followed by two fully connected layers. Specifically, each convolution block contains a convolution layer, a max-pooling layer with stride step 2, and a ReLU activation layer. The convolution layers in the two blocks have 1, 10 input channels and 10, 20 output channels, respectively, and both of them have kernel size 5, stride step 1 and no padding. The two fully connected layers have input dimensions 320, 50 and output dimensions 50, 10, respectively.

We implement all three algorithms using full gradients with the whole training set of 60k images. We choose the same learning rates $\eta_x = \eta_y = 10^{-3}$ for all algorithms. For proximal-AltGDA, we choose momentum $\beta = 0.25$ and $\gamma = 0.75$. For APDA, we choose the fine-tuned $\eta = 2\eta_x$. As the function $\Phi(x)$ cannot be exactly evaluated, we run 100 steps of stochastic gradient ascent updates with learning rate 0.1 to maximize $f(x_t, y) - h(y)$ and obtain an approximated $y^*(x_t)$, which is used to estimate $\Phi(x) + g(x)$.

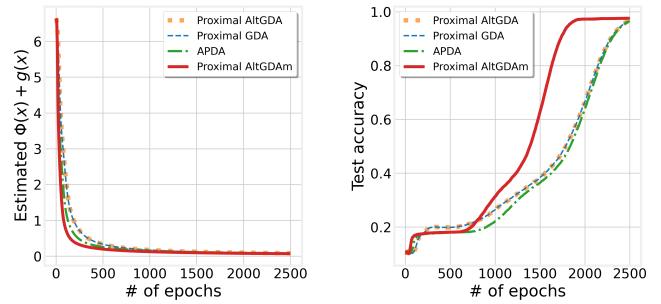


Figure 1. Left: comparison of $\Phi(x) + g(x)$ of all four algorithms. Right: comparison of the corresponding accuracy on the test dataset.

Figure 1 (Left) compares the estimated objective function value achieved by all the three algorithms. It can be seen that proximal-AltGDAm achieves the fastest convergence among these algorithms and is significantly faster than the proximal-GDA and the proximal-AltGDA. Figure 1 (Right) further demonstrates the advantage of proximal-AltGDAm in the test accuracy. It can be seen that the robust model trained by proximal-AltGDAm achieves a much higher test accuracy.

VI. CONCLUSION

We develop a single-loop and fast AltGDA algorithm that leverages proximal gradient updates and momentum acceleration to solve general regularized nonconvex-strongly-concave minimax optimization problems. By viewing the GDA updates of the algorithm as inexact accelerated gradient updates, we prove that the algorithm converges to a ϵ -critical point with a computational complexity $\mathcal{O}(\kappa^{11/6}\epsilon^{-2})$, which substantially improves the state-of-the-art result. In the future work, it is interesting to develop a stochastic variant of this algorithm to further improve the sample complexity.

ACKNOWLEDGMENT

The work of Ziyi Chen and Yi Zhou was supported in part by U.S. National Science Foundation under the Grants CCF-2106216 and DMS-2134223.

REFERENCES

[1] L. Adolphs, H. Daneshmand, A. Lucchi, and T. Hofmann. Local saddle point optimization: A curvature exploitation approach. In *Proc. International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 486–495, 2019.

[2] J. P. Bailey, G. Gidel, and G. Pilouras. Finite regret and cycles with fixed step-size via alternating gradient descent-ascent. In *Proc. Conference on Learning Theory (COLT)*, pages 391–407, 2020.

[3] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, Mar. 2009.

[4] R. I. Bot and A. Böhm. Alternating proximal-gradient steps for (stochastic) nonconvex-concave minimax problems. *ArXiv:2007.13605*, 2020.

[5] Z. Chen, Y. Zhou, T. Xu, and Y. Liang. Proximal gradient descent-ascent: Variable convergence under kl geometry. In *Proc. International Conference on Learning Representations (ICLR)*, 2021.

[6] A. Cherukuri, B. Gharesifard, and J. Cortes. Saddle-point dynamics: conditions for asymptotic stability of saddle points. *SIAM Journal on Control and Optimization*, 55(1):486–511, 2017.

[7] C. Daskalakis and I. Panageas. The limit points of (optimistic) gradient descent in min-max optimization. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pages 9236–9246, 2018.

[8] S. S. Du and W. Hu. Linear convergence of the primal-dual gradient method for convex-concave saddle point problems without strong convexity. In *Proc. International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 196–205, 2019.

[9] A. Fallah, A. Ozdaglar, and S. Pattathil. An optimal multistage stochastic gradient method for minimax problems. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 3573–3579. IEEE, 2020.

[10] M. A. M. Ferreira, M. Andrade, M. C. P. Matos, J. A. Filipe, and M. P. Coelho. Minimax theorem and nash equilibrium. *International Journal of Latest Trends in Finance & Economic Sciences*, 2012.

[11] S. Ghadimi and G. Lan. Accelerated gradient methods for nonconvex nonlinear and stochastic programming. *Mathematical Programming*, 156(1-2):59–99, Mar. 2016.

[12] G. Gidel, R. A. Hemmat, M. Pezeshki, R. Le Priol, G. Huang, S. Lacoste-Julien, and I. Mitliagkas. Negative momentum for improved game dynamics. In *Proc. International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1802–1811, 2019.

[13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pages 2672–2680, 2014.

[14] Z. Guo, Y. Xu, W. Yin, R. Jin, and T. Yang. On stochastic moving-average estimators for non-convex optimization. *ArXiv:2104.14840*, 2021.

[15] F. Huang, S. Gao, and H. Huang. Gradient descent ascent for min-max problems on riemannian manifolds. *ArXiv:2010.06097*, 2020.

[16] F. Huang, S. Gao, J. Pei, and H. Huang. Accelerated zeroth-order and first-order momentum methods from mini to minimax optimization. *ArXiv:2008.08170*, 2020.

[17] F. Huang, X. Wu, and H. Huang. Efficient mirror descent ascent methods for nonsmooth minimax problems. In *Proc. International Conference on Neural Information Processing Systems (Neurips)*, volume 34, 2021.

[18] C. Jin, P. Netrapalli, and M. I. Jordan. What is local optimality in nonconvex-nonconcave minimax optimization? In *Proc. International Conference on Machine Learning (ICML)*, pages 4880–4889, 2020.

[19] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.

[20] H. Li and Z. Lin. Accelerated proximal gradient methods for nonconvex programming. In *Proc. International Conference on Neural Information Processing Systems (Neurips)*, pages 379–387, 2015.

[21] Q. Li, Y. Zhou, Y. Liang, and P. K. Varshney. Convergence analysis of proximal gradient with momentum for nonconvex optimization. In *Proc. International Conference on Machine Learning (ICML)*, volume 70, pages 2111–2119, Aug 2017.

[22] T. Lin, C. Jin, and M. I. Jordan. Near-optimal algorithms for minimax optimization. In *Proc. Annual Conference on Learning Theory (COLT)*, pages 2738–2779, 2020.

[23] T. Lin, C. Jin, and M. I. Jordan. On gradient descent ascent for nonconvex-concave minimax problems. In *Proc. International Conference on Machine Learning (ICML)*, pages 6083–6093, 2020.

[24] P.-L. Lions and B. Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Numerical Analysis*, 16(6):964–979, 1979.

[25] L. Luo, H. Ye, Z. Huang, and T. Zhang. Stochastic recursive gradient descent ascent for stochastic nonconvex-strongly-concave minimax problems. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020.

[26] A. Mokhtari, A. Ozdaglar, and S. Pattathil. A unified analysis of extragradient and optimistic gradient methods for saddle point problems: Proximal point approach. In *Proc. International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1497–1507, 2020.

[27] A. Nedić and A. Ozdaglar. Subgradient methods for saddle-point problems. *Journal of optimization theory and applications*, 142(1):205–228, 2009.

[28] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Springer, 2014.

[29] Y. E. Nesterov. A method for solving the convex programming problem with convergence rate $O(1/k^2)$. In *Dokl. akad. nauk SSSR*, volume 269, pages 543–547, 1983.

[30] M. Nouiehed, M. Sanjabi, T. Huang, J. D. Lee, and M. Razaviyayn. Solving a class of non-convex min-max games using iterative first order methods. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pages 14934–14942, 2019.

[31] P. Ochs. Local convergence of the heavy-ball method and iPiano for non-convex optimization. *Journal of Optimization Theory and Applications*, 177(1):153–180, Apr 2018.

[32] P. Ochs, Y. Chen, T. Brox, and T. Pock. iPiano: Inertial proximal algorithm for nonconvex optimization. *SIAM Journal on Imaging Sciences*, 7(2):1388–1419, 2014.

[33] B. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

[34] S. Qiu, Z. Yang, X. Wei, J. Ye, and Z. Wang. Single-timescale stochastic nonconvex-concave optimization for smooth nonlinear td learning. *ArXiv:2008.10103*, 2020.

[35] R. T. Rockafellar. *Convex analysis*. Number 28. Princeton university press, 1970.

[36] M. Schmidt, N. Le Roux, and F. Bach. Convergence rates of inexact proximal-gradient methods for convex optimization. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2011.

[37] A. Sinha, H. Namkoong, and J. C. Duchi. Certifying some distributional robustness with principled adversarial training. In *Proc. International Conference on Learning Representations (ICLR)*, 2018.

[38] P. Tseng. Approximation accuracy, gradient methods, and error bound for structured convex optimization. *Mathematical Programming*, 125(2):263–295, Oct 2010.

[39] Y. Wang and J. Li. Improved algorithms for convex-concave minimax optimization. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[40] Y. Wang, G. Zhang, and J. Ba. On solving minimax optimization locally: A follow-the-ridge approach. In *Proc. International Conference on Learning Representations (ICLR)*, 2019.

[41] T. Xu, Z. Wang, Y. Liang, and H. V. Poor. Enhanced first and zeroth order variance reduced algorithms for min-max optimization. *ArXiv:2006.09361*, 2020.

[42] Z. Xu, H. Zhang, Y. Xu, and G. Lan. A unified single-loop alternating gradient projection algorithm for nonconvex-concave and convex-nonconcave minimax problems. *ArXiv:2006.02032*, 2020.

[43] A. Yadav, S. Shah, Z. Xu, D. Jacobs, and T. Goldstein. Stabilizing adversarial nets with prediction methods. In *Proc. International Conference on Learning Representations (ICLR)*, 2018.

[44] J. Yang, N. Kiyavash, and N. He. Global convergence and variance reduction for a class of nonconvex-nonconcave minimax problems. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[45] G. Zhang and Y. Wang. On the suboptimality of negative momentum for minimax optimization. *ArXiv:2008.07459*, 2020.

[46] S. Zhang, J. Yang, C. Guzmán, N. Kiyavash, and N. He. The complexity of nonconvex-strongly-concave minimax optimization. *ArXiv:2103.15888*, 2021.

[47] Y. Zhu, D. Liu, and Q. Tran-Dinh. A New Primal-Dual Algorithm for a Class of Nonlinear Compositional Convex Optimization Problems. *ArXiv:2006.09263*, June 2020.

APPENDIX A
PROOF OF PROPOSITION 2

Proposition 2. *Let Assumption 1 hold. Then, the parameter sequences $\{x_t, y_t\}_t$ generated by proximal-AltGDAm satisfy, for all $t = 0, 1, 2, \dots$,*

$$\begin{aligned} & \Phi(x_{t+1}) + g(x_{t+1}) \\ & \leq \Phi(x_t) + g(x_t) - \left(\frac{1-\beta}{2\eta_x} - 2L\kappa^{\frac{11}{6}} \right) \|x_{t+1} - x_t\|^2 \\ & \quad + \frac{\beta}{2\eta_x} \|x_t - x_{t-1}\|^2 + \frac{L}{4\kappa^{\frac{11}{6}}} \|y^*(x_t) - y_t\|^2. \end{aligned} \quad (2)$$

Proof. Consider the t -th iteration of PGDAm. As the function Φ is $L(1 + \kappa)$ -smooth (from Proposition 1), we obtain that

$$\Phi(x_{t+1}) \leq \Phi(x_t) + \langle x_{t+1} - x_t, \nabla \Phi(x_t) \rangle + \frac{L(1 + \kappa)}{2} \|x_{t+1} - x_t\|^2. \quad (7)$$

On the other hand, by the definition of the proximal gradient step of x_t , we have that

$$g(x_{t+1}) + \frac{1}{2\eta_x} \|x_{t+1} - \tilde{x}_t + \eta_x \nabla_1 f(x_t, y_t)\|^2 \leq g(x_t) + \frac{1}{2\eta_x} \|x_t - \tilde{x}_t + \eta_x \nabla_1 f(x_t, y_t)\|^2,$$

which further simplifies to

$$\begin{aligned} g(x_{t+1}) & \leq g(x_t) + \frac{1}{2\eta_x} \|x_t - \tilde{x}_t\|^2 + \langle x_t - \tilde{x}_t, \nabla_1 f(x_t, y_t) \rangle \\ & \quad - \frac{1}{2\eta_x} \|x_{t+1} - \tilde{x}_t\|^2 - \langle x_{t+1} - \tilde{x}_t, \nabla_1 f(x_t, y_t) \rangle \\ & \stackrel{(i)}{=} g(x_t) + \frac{\beta^2}{2\eta_x} \|x_t - x_{t-1}\|^2 - \frac{1}{2\eta_x} \|x_{t+1} - x_t - \beta(x_t - x_{t-1})\|^2 \\ & \quad + \langle x_t - x_{t+1}, \nabla_1 f(x_t, y_t) \rangle \\ & = g(x_t) + \frac{\beta^2}{2\eta_x} \|x_t - x_{t-1}\|^2 - \frac{1}{2\eta_x} \|x_{t+1} - x_t\|^2 - \frac{\beta^2}{2\eta_x} \|x_t - x_{t-1}\|^2 \\ & \quad + \frac{\beta}{\eta_x} \langle x_{t+1} - x_t, x_t - x_{t-1} \rangle + \langle x_t - x_{t+1}, \nabla_1 f(x_t, y_t) \rangle \\ & \leq g(x_t) - \frac{1}{2\eta_x} \|x_{t+1} - x_t\|^2 + \frac{\beta}{2\eta_x} \|x_{t+1} - x_t\|^2 + \frac{\beta}{2\eta_x} \|x_t - x_{t-1}\|^2 + \langle x_t - x_{t+1}, \nabla_1 f(x_t, y_t) \rangle, \end{aligned} \quad (8)$$

where (i) uses the fact that $x_t - \tilde{x}_t = \beta(x_{t-1} - x_t)$.

Adding up eq. (7) and eq. (8) yields that

$$\begin{aligned} & \Phi(x_{t+1}) + g(x_{t+1}) \\ & \leq \Phi(x_t) + g(x_t) - \left(\frac{1}{2\eta_x} - \frac{L(1 + \kappa)}{2} \right) \|x_{t+1} - x_t\|^2 + \frac{\beta}{2\eta_x} \|x_{t+1} - x_t\|^2 + \frac{\beta}{2\eta_x} \|x_t - x_{t-1}\|^2 \\ & \quad + \langle x_{t+1} - x_t, \nabla \Phi(x_t) - \nabla_1 f(x_t, y_t) \rangle \\ & \leq \Phi(x_t) + g(x_t) - \left(\frac{1}{2\eta_x} - \frac{L(1 + \kappa)}{2} \right) \|x_{t+1} - x_t\|^2 + \frac{\beta}{2\eta_x} \|x_{t+1} - x_t\|^2 + \frac{\beta}{2\eta_x} \|x_t - x_{t-1}\|^2 \\ & \quad + \|x_{t+1} - x_t\| \|\nabla \Phi(x_t) - \nabla_1 f(x_t, y_t)\| \\ & \leq \Phi(x_t) + g(x_t) - \left(\frac{1}{2\eta_x} - \frac{L(1 + \kappa)}{2} \right) \|x_{t+1} - x_t\|^2 + \frac{\beta}{2\eta_x} \|x_{t+1} - x_t\|^2 + \frac{\beta}{2\eta_x} \|x_t - x_{t-1}\|^2 \\ & \quad + \|x_{t+1} - x_t\| \|\nabla_1 f(x_t, y^*(x_t)) - \nabla_1 f(x_t, y_t)\| \\ & \leq \Phi(x_t) + g(x_t) - \left(\frac{1}{2\eta_x} - L\kappa \right) \|x_{t+1} - x_t\|^2 + \frac{\beta}{2\eta_x} \|x_{t+1} - x_t\|^2 + \frac{\beta}{2\eta_x} \|x_t - x_{t-1}\|^2 \\ & \quad + L \|x_{t+1} - x_t\| \|y^*(x_t) - y_t\| \\ & \stackrel{(i)}{\leq} \Phi(x_t) + g(x_t) - \left(\frac{1-\beta}{2\eta_x} - L\kappa - L\kappa^{\frac{11}{6}} \right) \|x_{t+1} - x_t\|^2 + \frac{\beta}{2\eta_x} \|x_t - x_{t-1}\|^2 + \frac{L}{4\kappa^{\frac{11}{6}}} \|y^*(x_t) - y_t\|^2 \\ & \leq \Phi(x_t) + g(x_t) - \left(\frac{1-\beta}{2\eta_x} - 2L\kappa^{\frac{11}{6}} \right) \|x_{t+1} - x_t\|^2 + \frac{\beta}{2\eta_x} \|x_t - x_{t-1}\|^2 + \frac{L}{4\kappa^{\frac{11}{6}}} \|y^*(x_t) - y_t\|^2. \end{aligned} \quad (9)$$

where (i) uses AM-GM inequality that $ab \leq \frac{Ca^2}{2} + \frac{b^2}{2C}$ for any $a, b, C \geq 0$. This proves eq. (2) \square

APPENDIX B PROOF OF PROPOSITION 3

Proposition 3. *Let Assumption 1 hold. Choose learning rate $\eta_y = \frac{1}{L}$ and momentum parameter $\gamma = \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$. Then, the parameter sequences $\{x_t, y_t\}$ generated by proximal-AltGDAm satisfy, for all $t = 0, 1, 2, \dots$*

$$\begin{aligned} \|y_{t+1} - y^*(x_{t+1})\|^2 &\leq \frac{2R\kappa}{\sqrt{L}}(1 - \kappa^{-0.5})^{t+1} \\ &\quad + \frac{6\kappa^2}{\sqrt{L}} \sum_{j=1}^t (1 - \kappa^{-0.5})^{t+1-j/2} \|x_{j+1} - x_j\|. \end{aligned} \quad (4)$$

Proof. We rewrite the inner accelerated gradient ascent steps in Algorithm 1 as the inexact-proximal gradient method (3). Then, based on Theorem 4 of [36], using $\eta_y = \frac{1}{L}$ and $\gamma = \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$, this method has the following convergence rate.

$$\begin{aligned} &f(x_{t+1}, y_{t+1}) - f(x_{t+1}, y^*(x_{t+1})) \\ &\leq (1 - \kappa^{-0.5})^{t+1} \left(\sqrt{2(f(x_{t+1}, y_{t+1}) - f(x_{t+1}, y^*(x_{t+1})))} + \sqrt{\frac{2}{\mu} \sum_{i=1}^{t+1} \|e_i\| (1 - \kappa^{-0.5})^{-i/2}} \right). \end{aligned} \quad (10)$$

The above convergence rate can be simplified as follows.

$$\begin{aligned} &\frac{\mu}{2} \|y_{t+1} - y^*(x_{t+1})\|^2 \\ &\stackrel{(i)}{\leq} f(x_{t+1}, y_{t+1}) - f(x_{t+1}, y^*(x_{t+1})) \\ &\leq (1 - \kappa^{-0.5})^{t+1} \left(\sqrt{2(f(x_{t+1}, y_{t+1}) - f(x_{t+1}, y^*(x_{t+1})))} + \sqrt{\frac{2}{\mu} \sum_{i=1}^{t+1} \|e_i\| (1 - \kappa^{-0.5})^{-i/2}} \right) \\ &\stackrel{(ii)}{\leq} (1 - \kappa^{-0.5})^{t+1} \sqrt{L \|y_{t+1} - y^*(x_{t+1})\|^2} + \sqrt{\frac{2}{\mu} \sum_{i=1}^{t+1} \|\nabla_2 f(x_i, \tilde{y}_{i-1}) - \nabla_2 f(x_{t+1}, \tilde{y}_{i-1})\| (1 - \kappa^{-0.5})^{t+1-i/2}} \\ &\stackrel{(iii)}{\leq} R\sqrt{L} (1 - \kappa^{-0.5})^{t+1} + \sqrt{\frac{2}{\mu} \sum_{i=1}^{t+1} (1 - \kappa^{-0.5})^{t+1-i/2} \sum_{j=i}^t L \|x_{j+1} - x_j\|} \\ &= R\sqrt{L} (1 - \kappa^{-0.5})^{t+1} + L \sqrt{\frac{2}{\mu} \sum_{j=1}^t \sum_{i=1}^j (1 - \kappa^{-0.5})^{t+1-i/2} \|x_{j+1} - x_j\|} \\ &= R\sqrt{L} (1 - \kappa^{-0.5})^{t+1} + \sqrt{2L\kappa} \sum_{j=1}^t (1 - \kappa^{-0.5})^{t+0.5} \frac{(1 - \kappa^{-0.5})^{-j/2} - 1}{(1 - \kappa^{-0.5})^{-0.5} - 1} \|x_{j+1} - x_j\| \\ &\leq R\sqrt{L} (1 - \kappa^{-0.5})^{t+1} + \sqrt{2L\kappa} \sum_{j=1}^t \frac{(1 - \kappa^{-0.5})^{t+1-j/2}}{1 - (1 - \kappa^{-0.5})^{0.5}} \|x_{j+1} - x_j\| \\ &\stackrel{(iv)}{\leq} R\sqrt{L} (1 - \kappa^{-0.5})^{t+1} + 2\kappa\sqrt{2L} \sum_{j=1}^t (1 - \kappa^{-0.5})^{t+1-j/2} \|x_{j+1} - x_j\|, \end{aligned}$$

where (i) and (ii) use $\nabla_2 f(x_{t+1}, y^*(x_{t+1})) = 0$ and Assumption 1.1 that $f(x, \cdot)$ is L -smooth and μ -strongly concave, (iii) uses the fact that \mathcal{Y} is bounded with diameter R and Assumption 1.1 that f is L -smooth, and (iv) uses $\frac{1}{1 - (1 - \kappa^{-0.5})^{0.5}} = \frac{1 + (1 - \kappa^{-0.5})^{0.5}}{\kappa^{-0.5}} \leq 2\kappa^{0.5}$. Multiplying the above inequality with $2/\mu$ proves Proposition 3. \square

APPENDIX C PROOF OF COROLLARY 1

Corollary 1. *Under the conditions of Proposition 3 and stepsize $\eta_x \leq 1/(16L\kappa^{\frac{11}{6}})$, the sequences $\{x_t, y_t\}_t$ generated by proximal-AltGDAm satisfy*

$$\|x_{t+1} - x_t\|, \|y_t - y^*(x_t)\|, \|y_{t+1} - y_t\| \xrightarrow{t} 0.$$

Proof. Telescoping eq. (4) over $t = 0, 1, \dots, T-1$ yields that

$$\begin{aligned}
& \sum_{t=0}^{T-1} \|y_{t+1} - y^*(x_{t+1})\|^2 \\
& \leq \frac{2R\kappa}{\sqrt{L}} \sum_{t=0}^{T-1} (1 - \kappa^{-0.5})^{t+1} + \frac{6\kappa^2}{\sqrt{L}} \sum_{t=0}^{T-1} \sum_{j=1}^t (1 - \kappa^{-0.5})^{t+1-j/2} \|x_{j+1} - x_j\| \\
& \leq \frac{2R\kappa^{1.5}}{\sqrt{L}} + \frac{6\kappa^2}{\sqrt{L}} \sum_{j=1}^{T-1} \sum_{t=j}^{T-1} (1 - \kappa^{-0.5})^{t+1-j/2} \|x_{j+1} - x_j\| \\
& \leq \frac{2R\kappa^{1.5}}{\sqrt{L}} + \frac{6\kappa^{2.5}}{\sqrt{L}} \sum_{j=1}^{T-1} (1 - \kappa^{-0.5})^{j/2} \|x_{j+1} - x_j\| \\
& \leq \frac{2R\kappa^{1.5}}{\sqrt{L}} + \frac{3\kappa^{2.5}}{\sqrt{L}} \sum_{j=1}^{T-1} \left(\frac{1}{\kappa^{\frac{7}{6}} \sqrt{L}} (1 - \kappa^{-0.5})^j + \kappa^{\frac{7}{6}} \sqrt{L} \|x_{j+1} - x_j\|^2 \right) \\
& \leq \frac{2R\kappa^{1.5}}{\sqrt{L}} + \frac{3\kappa^{\frac{11}{6}}}{L} + 3\kappa^{\frac{11}{3}} \sum_{j=1}^{T-1} \|x_{j+1} - x_j\|^2.
\end{aligned} \tag{11}$$

Then, telescoping eq. (2) over $t = 0, 1, \dots, T-1$ yields that

$$\begin{aligned}
& \Phi(x_T) + g(x_T) - \Phi(x_0) - g(x_0) \\
& \leq -\left(\frac{1-\beta}{2\eta_x} - 2L\kappa^{\frac{11}{6}}\right) \sum_{t=0}^{T-1} \|x_{t+1} - x_t\|^2 + \frac{\beta}{2\eta_x} \sum_{t=0}^{T-1} \|x_t - x_{t-1}\|^2 + \frac{L}{4\kappa^{\frac{11}{6}}} \sum_{t=0}^{T-1} \|y^*(x_t) - y_t\|^2 \\
& \stackrel{(i)}{\leq} -\left(\frac{1-2\beta}{2\eta_x} - 2L\kappa^{\frac{11}{6}}\right) \sum_{t=0}^{T-1} \|x_{t+1} - x_t\|^2 + \frac{L}{4\kappa^{\frac{11}{6}}} \left(R^2 + \frac{2R\kappa^{1.5}}{\sqrt{L}} + \frac{3\kappa^{\frac{11}{6}}}{L} + 3\kappa^{\frac{11}{3}} \sum_{j=1}^{T-1} \|x_{j+1} - x_j\|^2 \right) \\
& \leq -\left(\frac{1-2\beta}{2\eta_x} - 3L\kappa^{\frac{11}{6}}\right) \sum_{t=0}^{T-1} \|x_{t+1} - x_t\|^2 + \frac{LR^2}{4\kappa^{\frac{11}{6}}} + \frac{R\sqrt{L}}{2\kappa^{\frac{1}{3}}} + 1
\end{aligned} \tag{12}$$

where (i) uses $x_{-1} = x_0$, $\|y^*(x_0) - y_0\| \leq R$ and eq. (11). When $\eta_x \leq \frac{1}{16L\kappa^{\frac{11}{6}}}$ and $\beta \leq \frac{1}{4}$, rearranging the above inequality yields that

$$L\kappa^{\frac{11}{6}} \sum_{t=0}^{T-1} \|x_{t+1} - x_t\|^2 \leq \Phi(x_0) + g(x_0) - \inf_{x \in \mathbb{R}^m} (\Phi(x) + g(x)) + \frac{LR^2}{4\kappa^{\frac{11}{6}}} + \frac{R\sqrt{L}}{2\kappa^{\frac{1}{3}}} + 1 < +\infty \tag{13}$$

Letting $T \rightarrow \infty$ in the above inequality yields that $\sum_{t=0}^{\infty} \|x_{t+1} - x_t\|^2 < +\infty$, so $\|x_{t+1} - x_t\| \xrightarrow{t} 0$. Then, letting $T \rightarrow \infty$ in eq. (11) yields that $\sum_{t=0}^{\infty} \|y_{t+1} - y^*(x_{t+1})\|^2 \leq \frac{2R\kappa^{1.5}}{\sqrt{L}} + \frac{3\kappa^{\frac{11}{6}}}{L} + 3\kappa^{\frac{11}{3}} \sum_{j=1}^{\infty} \|x_{j+1} - x_j\|^2 < +\infty$, so $\|y_t - y^*(x_t)\| \xrightarrow{t} 0$.

The last term $\|y_{t+1} - y_t\| \xrightarrow{t} 0$ can be proved as follows.

$$\begin{aligned}
\|y_{t+1} - y_t\| & \leq \|y_{t+1} - y^*(x_{t+1})\| + \|y^*(x_t) - y_t\| + \|y^*(x_{t+1}) - y^*(x_t)\| \\
& \stackrel{(i)}{\leq} \|y_{t+1} - y^*(x_{t+1})\| + \|y_t - y^*(x_t)\| + \kappa \|x_{t+1} - x_t\| \xrightarrow{t} 0,
\end{aligned}$$

where (i) uses the fact that y^* is κ -Lipschitz. \square

APPENDIX D PROOF OF THEOREM 1

Theorem 1 (Global convergence). *Under the conditions of Proposition 3 and stepsize $\eta_x \leq 1/(16L\kappa^{\frac{11}{6}})$, the sequence $\{x_t\}_t$ generated by proximal-AltGDAm is bounded and has a compact set of limit points. Also, every such limit point is a critical point of $(\Phi + g)(x)$. Moreover, the total number of iterations T required to achieve $\min_{0 \leq t \leq T} \|G(x_t)\| \leq \epsilon$ is $T = \mathcal{O}(\kappa^{\frac{11}{6}} \epsilon^{-2})$, which is also the order of the required computational complexity.*

Proof. We first prove the existence of the limit points of $\{x_t\}$. Note that in eq. (12), $\frac{1-2\beta}{2\eta_x} - 2L\kappa^{\frac{11}{6}} \geq 0$ since $\eta_x \leq \frac{1}{16L\kappa^{\frac{11}{6}}}$ and $\beta \leq \frac{1}{4}$ as specified in Proposition 3. Hence, for all $T \geq 0$,

$$\Phi(x_T) + g(x_T) \leq \Phi(x_0) + g(x_0) + \frac{LR^2}{4\kappa^{\frac{11}{6}}} + \frac{R\sqrt{L}}{2\kappa^{\frac{1}{3}}} + 1 < +\infty,$$

which implies that $\{\Phi(x_t) + g(x_t)\}_t$ is upper bounded. Hence, based on Assumption 1.2, the sequence $\{x_t\}_t$ is bounded and thus has a compact set of limit points.

Next, we prove that every limit point x of $\{x_t\}_t$ is a critical point of $(\Phi + g)(x)$, i.e., $\mathbf{0} \in \partial(\Phi + g)(x)$. By the optimality condition of the proximal gradient update of x_{t+1} we have

$$\begin{aligned} \mathbf{0} &\in \partial g(x_{t+1}) + \frac{1}{\eta_x} (x_{t+1} - \tilde{x}_t + \eta_x \nabla_1 f(x_t, y_t)) \\ &= \partial g(x_{t+1}) + \frac{1}{\eta_x} (x_{t+1} - x_t - \beta(x_t - x_{t-1}) + \eta_x \nabla_1 f(x_t, y_t)), \end{aligned}$$

which implies that $-\frac{1}{\eta_x} (x_{t+1} - x_t - \beta(x_t - x_{t-1}) + \eta_x \nabla_1 f(x_t, y_t)) \in \partial g(x_{t+1})$ and thus by convexity of g we have

$$g(x) \geq g(x_{t(j)+1}) - \frac{1}{\eta_x} (x_{t+1} - x_t - \beta(x_t - x_{t-1}) + \eta_x \nabla_1 f(x_t, y_t))^{\top} (x - x_{t(j)+1}); \forall x \in \mathbb{R}^m. \quad (14)$$

As $x_{t(j)} \xrightarrow{j} x^*$ and $\|y_{t(j)} - y^*(x_{t(j)})\| \xrightarrow{t} 0$, we have $y_{t(j)} \xrightarrow{t} y^*(x^*)$ due to continuity of $y^*(\cdot)$. Also note that the convex function g is continuous (See Corollary 10.1.1 of [35]). Hence, letting $j \rightarrow \infty$ in eq. (14) yields that

$$g(x) \geq g(x^*) - \nabla_1 f(x^*, y^*(x^*))^{\top} (x - x^*) = g(x^*) - \nabla \Phi(x^*)^{\top} (x - x^*); \forall x \in \mathbb{R}^m, \quad (15)$$

which further implies that $-\nabla \Phi(x^*) \in \partial g(x^*) \Rightarrow \mathbf{0} \in \partial(\Phi + g)(x^*)$. Hence, x^* is a critical point of $(\Phi + g)(x)$.

Finally, we derive the non-asymptotic computational complexity to obtain $\min_{0 \leq t \leq T} \|G(x_t)\| \leq \epsilon$. Note that

$$\begin{aligned} \|G(x_{t+1})\| &= \frac{1}{\eta_x} \|x_{t+1} - \text{prox}_{\eta_x g}(x_{t+1} - \eta_x \nabla \Phi(x_{t+1}))\| \\ &\stackrel{(i)}{\leq} \frac{1}{\eta_x} \|x_{t+1} - \tilde{x}_t + \eta_x [\nabla_1 f(x_t, y_t) - \nabla f_1(x_{t+1}, y^*(x_{t+1}))]\| \\ &\leq \frac{1}{\eta_x} \|x_{t+1} - x_t - \beta(x_t - x_{t-1})\| + L \|x_{t+1} - x_t\| + L \|y^*(x_{t+1}) - y^*(x_t)\| + L \|y^*(x_t) - y_t\| \\ &\stackrel{(ii)}{\leq} \left(\frac{1}{\eta_x} + L + L\kappa \right) \|x_{t+1} - x_t\| + \frac{\beta}{\eta_x} \|x_t - x_{t-1}\| + L \|y^*(x_t) - y_t\|, \end{aligned}$$

where (i) uses $x_{t+1} \in \text{prox}_{\eta_x g}(\tilde{x}_t - \eta_x \nabla_1 f(x_t, y_t))$, $\nabla \Phi(x) = \nabla f_1(x, y^*(x))$ (from Proposition 1) and the non-expansiveness of proximal mapping, (ii) uses the property that y^* is κ -Lipschitz continuous in Proposition 1. Hence,

$$\begin{aligned} &(T-1) \min_{0 \leq t \leq T} \|G(x_t)\|^2 \\ &\leq (T-1) \min_{1 \leq t \leq T-1} \|G(x_{t+1})\|^2 \\ &\leq \sum_{t=1}^{T-1} \|G(x_{t+1})\|^2 \\ &\leq \sum_{t=1}^{T-1} \left[3 \left(\frac{1}{\eta_x} + L + L\kappa \right)^2 \|x_{t+1} - x_t\|^2 + \frac{3\beta^2}{\eta_x^2} \|x_t - x_{t-1}\|^2 + 3L^2 \|y^*(x_t) - y_t\|^2 \right] \\ &\stackrel{(i)}{\leq} 3(18L\kappa^{\frac{11}{6}})^2 \sum_{t=0}^{T-1} \|x_{t+1} - x_t\|^2 + 27L^2\kappa^{\frac{11}{3}} \sum_{t=0}^{T-1} \|x_{t+1} - x_t\|^2 + 3L^2 \sum_{t=0}^{T-1} \|y^*(x_t) - y_t\|^2 \\ &\stackrel{(ii)}{\leq} 999L^2\kappa^{\frac{11}{3}} \sum_{t=0}^{T-1} \|x_{t+1} - x_t\|^2 + 3L^2 \left(\frac{2R\kappa^{1.5}}{\sqrt{L}} + \frac{3\kappa^{\frac{11}{6}}}{L} + 3\kappa^{\frac{11}{3}} \sum_{j=1}^{T-1} \|x_{j+1} - x_j\|^2 \right) + 3L^2 \|y^*(x_0) - y_0\|^2, \\ &\stackrel{(iii)}{\leq} \frac{1008L^2\kappa^{\frac{11}{3}}}{L\kappa^{\frac{11}{6}}} \left(\Phi(x_0) + g(x_0) - \inf_{x \in \mathbb{R}^m} (\Phi(x) + g(x)) + \frac{LR^2}{4\kappa^{\frac{11}{6}}} + \frac{R\sqrt{L}}{2\kappa^{\frac{1}{3}}} + 1 \right) + 6RL^{1.5}\kappa^{1.5} + 9L\kappa^{\frac{11}{6}} + 3L^2R^2 \\ &= \mathcal{O}(\kappa^{\frac{11}{6}}). \end{aligned}$$

where (i) uses $\beta \leq \frac{1}{4}$ and the maximum possible stepsize $\eta_x = \frac{1}{16L\kappa^{\frac{11}{6}}}$, (ii) uses eq. (11), and (iii) uses eq. (13) and the fact that \mathcal{Y} is bounded with diameter R . Based on the above inequality, when the number of iterations $T \geq \mathcal{O}(\kappa^{\frac{11}{6}}\epsilon^{-2})$, $\min_{0 \leq t \leq T} \|G(x_t)\| \leq \sqrt{\mathcal{O}(\kappa^{\frac{11}{6}})/(T-1)} \leq \epsilon$. Since each iteration has $\mathcal{O}(1)$ number of gradient and proximal mapping evaluations, the order of computational complexity is also $\mathcal{O}(\kappa^{\frac{11}{6}}\epsilon^{-2})$. \square

APPENDIX E
DERIVIATION OF COMPUTATIONAL COMPLEXITIES IN TABLE I

In this section, we will derive some computational complexities in Table I that are not directly shown in their corresponding papers. Note that all these GDA-type algorithms in Table I are single-loop. Hence, the computational complexity (the number of gradient evaluations) has the order of $\mathcal{O}(T)$ where T is the number of iterations.

First, the papers in Table I use different convergence measures for computational complexity. Specifically, [5], [17] and our work show computational complexity to achieve $\|G(x)\| \leq \epsilon$ where the proximal gradient mapping G is defined in (5). [4], [23] use the measure $\min_t \text{dist}(\Phi(x_t) + \partial g(x_t), \mathbf{0}) \leq \epsilon$ where $\Phi(x) := \max_{y \in \mathcal{Y}} f(x, y) - h(y)$, ∂g denotes the partial gradient of g and $\text{dist}(A, \mathbf{0})$ denotes the distance between $\mathbf{0}$ and any set A . [42] has no regularizers g, h and uses the convergence measure $\min_t \|\nabla f(x_t, y_t)\| \leq \epsilon$ when $y \in \mathbb{R}^d$ is unconstrained, which does not necessarily yield the desired approximate critical point of Φ .

A. Derivation of complexity in [5]

In [5], Proposition 2 states that the Lyapunov function $H(z_t) := \Phi(x_t) + g(x_t) + (1 - \frac{1}{4\kappa^2})\|y_t - y^*(x_t)\|^2$ where $z_t := (x_t, y_t)$ are generated by GDA decreases at the following rate.

$$H(z_{t+1}) \leq H(z_t) - 2\|x_{t+1} - x_t\|^2 - \frac{1}{4\kappa^2}(\|y_{t+1} - y^*(x_{t+1})\|^2 + \|y_t - y^*(x_t)\|^2). \quad (16)$$

Note that for GDA, the gradient mapping (5) has the following norm bound

$$\begin{aligned} \|G(x_{t+1})\| &= \frac{1}{\eta_x} \|x_{t+1} - \text{prox}_{\eta_x g}(x_{t+1} - \eta_x \nabla \Phi(x_{t+1}))\| \\ &\stackrel{(i)}{\leq} \frac{1}{\eta_x} \|x_{t+1} - x_t + \eta_x [\nabla_1 f(x_t, y_t) - \nabla f_1(x_{t+1}, y^*(x_{t+1}))]\| \\ &\leq \left(\frac{1}{\eta_x} + L\right) \|x_{t+1} - x_t\| + L\|y^*(x_{t+1}) - y^*(x_t)\| + L\|y^*(x_t) - y_t\| \\ &\stackrel{(ii)}{\leq} \left(\frac{1}{\eta_x} + L + L\kappa\right) \|x_{t+1} - x_t\| + L\|y^*(x_t) - y_t\| \end{aligned}$$

where (i) uses the GDA update rule $x_{t+1} \in \text{prox}_{\eta_x g}(x_t - \eta_x \nabla_1 f(x_t, y_t))$, the expression $\nabla \Phi(x) = \nabla f_1(x, y^*(x))$ in Proposition 1 and the non-expansiveness of proximal mapping, (ii) uses the property that y^* is κ -Lipschitz continuous Proposition 1. Hence, we obtain the following convergence rate.

$$\begin{aligned} \min_{0 \leq t \leq T} \|G(x_t)\|^2 &\leq \frac{1}{T} \sum_{t=0}^{T-1} \|G(x_{t+1})\|^2 \\ &\leq \frac{1}{T} \sum_{t=0}^{T-1} \left[2\left(\frac{1}{\eta_x} + L + L\kappa\right)^2 \|x_{t+1} - x_t\|^2 + 2L^2\|y^*(x_t) - y_t\|^2 \right] \\ &\stackrel{(i)}{\leq} \frac{1}{T} \sum_{t=0}^{T-1} \left[\mathcal{O}(\kappa^6) \|x_{t+1} - x_t\|^2 + 2L^2\|y^*(x_t) - y_t\|^2 \right] \\ &\leq \frac{\mathcal{O}(\kappa^6)}{T} \sum_{t=0}^{T-1} \left(2\|x_{t+1} - x_t\|^2 + \frac{1}{4\kappa^2} \|y_{t+1} - y^*(x_{t+1})\|^2 \right) \\ &\stackrel{(ii)}{\leq} \frac{\mathcal{O}(\kappa^6)}{T} \sum_{t=0}^{T-1} (H(z_t) - H(z_{t+1})) \\ &\leq \frac{\mathcal{O}(\kappa^6)}{T} (H(z_0) - H(z_T)) \end{aligned} \quad (17)$$

where (i) uses the maximum possible stepsize $\eta_x = \frac{1}{\kappa^3(L+3)^2} = \mathcal{O}(\kappa^{-3})$ in [5]. Therefore, To let $\min_{0 \leq t \leq T} \|G(x_t)\| \leq \epsilon$, the computational complexity has the order $T = \mathcal{O}(\kappa^6\epsilon^{-2})$.

B. Derivation of complexity in [17]

The mirror descent ascent algorithm (Algorithm 1) in [17] updates the variables x and y simultaneously using proximal mirror descent and momentum accelerated mirror ascent steps respectively. Specifically, using the Bregman functions $\psi_t(x) := \frac{1}{2}\|x\|^2$ and $\phi_t(y) := \frac{1}{2}\|y\|^2$ which are both $\rho = 1$ -strongly convex, this algorithm becomes proximal GDA with momentum on y variable.

Substituting $\rho = 1$ into Theorem 2 that provides convergence rate under deterministic minimax optimization (i.e., there are no stochastic samples in the objective function), we obtain the following hyperparameter choices $\eta = \mathcal{O}(1)$, $L = L_f(1 + \kappa) = \mathcal{O}(L_f \kappa)$ ¹, $\lambda = \mathcal{O}(L_f^{-1})$, $\gamma = \mathcal{O}[\min(L^{-1}, \frac{\mu/L_f}{\kappa^2}, \frac{\mu/L_f}{L_f^2})] = \mathcal{O}[\min(L_f^{-1}\kappa^{-1}, \kappa^{-3}, L_f^{-2}\kappa^{-1})] = \mathcal{O}(\kappa^{-3})$ (Without loss of generality, we assume $\mu \leq 1$ which implies that $\kappa = L_f/\mu \geq L_f$). Then the convergence rate (25) becomes

$$\begin{aligned} \min_{1 \leq t \leq T} \|G(x_t)\| &\leq \frac{1}{T} \sum_{t=1}^T \|\mathcal{G}_t\| \leq \mathcal{O}\left(\frac{\sqrt{\tilde{F}(x_1) - F^*} + \Delta_1}{\sqrt{T\gamma\rho}}\right) \\ &\stackrel{(i)}{=} \mathcal{O}\left(\frac{\sqrt{\Phi(x_1) + g(x_1) - \inf_x (\Phi(x) + g(x))} + \|y_1 - y^*(x_1)\|}{\sqrt{T\kappa^{-3}}}\right) \end{aligned} \quad (18)$$

where (i) uses the notations in [17] that $\mathcal{G}_t = G(x_t)$, $\tilde{F}(x) = \Phi(x) + g(x)$, $\Delta_1 = \|y_1 - y^*(x_1)\|$ and the above hyperparameter choices. Hence, to achieve $\min_{1 \leq t \leq T} \|\mathcal{G}_t\| \leq \epsilon$, the required computation complexity is $T \geq \mathcal{O}(\kappa^3\epsilon^{-2}(\Phi(x_1) + g(x_1) - \inf_x (\Phi(x) + g(x)) + \|y_1 - y^*(x_1)\|^2))$. In Table I, we only keep the dependence of $T(\epsilon)$ on $\epsilon \approx 0$ and $\kappa \gg 1$, which yields $\mathcal{O}(\kappa^3\epsilon^{-2})$.

C. Derivation of complexity in [42]

[42] aims to solve the following minimax optimization

$$\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} f(x, y).$$

where \mathcal{X} and \mathcal{Y} are nonempty closed convex sets and \mathcal{Y} is also compact. The following AltGDA algorithm with projection mappings $\mathcal{P}_{\mathcal{X}}$ and $\mathcal{P}_{\mathcal{Y}}$ is analyzed for nonconvex-strongly concave geometry where f is L -smooth² and $f(\cdot, y)$ is μ -strongly concave for any $y \in \mathcal{Y}$.

$$\begin{cases} x_{k+1} = \mathcal{P}_{\mathcal{X}}(x_k - \eta^{-1}\nabla_x f(x_k, y_k)) \\ y_{k+1} = \mathcal{P}_{\mathcal{Y}}(y_k + \rho\nabla_y f(x_{k+1}, y_k)) \end{cases} \quad (19)$$

Using the largest possible stepsizes $\eta^{-1} = \mathcal{O}(L^{-1}\kappa^{-3})$, $\rho = \mathcal{O}(\mu L^{-2}) = \mathcal{O}(L^{-1}\kappa^{-1})$ that satisfies eq. (3.18), the following key variables in Theorem 3.1 can be computed as follows.

$$\begin{aligned} d_1 &= \mathcal{O}(L^{-1}\kappa^{-5}) \\ F_1 - \bar{F} &= f(x_1, y_1) - \min_{x \in \mathcal{X}, y \in \mathcal{Y}} f(x, y) + \mathcal{O}(L\kappa^3\sigma_y^2) \end{aligned}$$

where σ_y is the diameter of the compact set \mathcal{Y} . Hence the number of iterations (also the order of the computation complexity) required to achieve $\|\nabla_x f(x_k, y_k)\| \leq \epsilon$, $\frac{1}{\rho}\|y_k - \mathcal{P}_{\mathcal{Y}}(y_k + \rho\nabla_y f(x_k, y_k))\| \leq \epsilon$ (note that this does not necessarily yields approximate critical point of $\Phi(x) := \max_{y \in \mathcal{Y}} f(x, y)$) is

$$T(\epsilon) = \mathcal{O}\left(\frac{F_1 - \bar{F}}{d_1\epsilon^2}\right) = \mathcal{O}\left(\epsilon^{-2}L\kappa^5\left(f_1 - \min_{x \in \mathcal{X}, y \in \mathcal{Y}} f(x, y) + 32L\kappa^3\sigma_y^2\right)\right).$$

In Table I, we only keep the dependence of $T(\epsilon)$ on $\epsilon \approx 0$ and $\kappa \gg 1$, which yields $\mathcal{O}(\kappa^5\epsilon^{-2})$.

¹ L_f in [17] has the same meaning as our L , the Lipschitz parameter of ∇f .

²In Assumption 2.1 of [42], let all the Lipschitz smooth parameters $L_x = L_y = L_{12} = L_{21} := L$ for simplicity.