Dissipative Gradient Descent Ascent Method: A Control Theory Inspired Algorithm for Min-max Optimization

Tianqi Zheng[1], Nicolas Loizou[1], Pengcheng You[2] and Enrique Mallada[1]

Abstract— **Gradient Descent Ascent (GDA) methods for min-max optimization problems typically produce oscillatory behavior that can lead to instability, e.g., in bilinear settings. To address this problem, we introduce a dissipation term into the GDA updates to dampen these oscillations. The proposed Dissipative GDA (DGDA) method can be seen as performing standard GDA on a state-augmented and regularized saddle function that does not strictly introduce additional convexity/concavity. We theoretically show the linear convergence of DGDA in the bilinear and strongly convex-strongly concave settings and assess its performance by comparing DGDA with other methods such as GDA, Extra-Gradient (EG), and Optimistic GDA. Our findings demonstrate that DGDA surpasses these methods, achieving superior convergence rates. We support our claims with two numerical examples that showcase DGDA's effectiveness in solving saddle point problems.**

I. Introduction

In recent years, there has been a significant focus on solving saddle point problems, namely min-max optimization problems [1]–[6]. These problems have garnered considerable attention, particularly in fields such as Generative Adversarial Networks (GANs) [6]–[8], Reinforcement Learning (RL) [9], and Constrained RL (C-RL) [10], [11]. However, a major challenge that persists in these approaches is the instability of the training process. That is, solving the min-max optimization problem via running the standard Gradient Descent Ascent (GDA) algorithm often leads to unstable oscillatory behavior rather than convergence to the optimal solution. This is particularly illustrated in bilinear min-max problems, such as the training Wasserstein GANs [12] or solving C-RL problems in the occupancy measure space [13], for which the standard GDA fails to converge [1]–[3].

In order to understand the instability of the GDA method and further tackle its limitation, we draw inspiration from the control-theoretic notions of dissipativity [14], which enables the design of stabilizing controllers using dynamic (state-augmented) components that seek to dissipate the energy generated by the unstable process. This aligns with recent work that leverages control theory tools in the analysis and design of optimization algorithms [15]–[20]. From a dynamical system point of view, dissipativity theory characterizes the manner in which energy dissipates within the system and drives it towards equilibrium. It provides a direct way to construct a Lyapunov function, which further relates the rate of decrease of this internal energy to the rate of convergence of the algorithm.

We motivate our developments by looking first at a simple scalar bilinear problem wherein, the energy of the system, expressed as the square 2-norm distance to the saddle, is shown to strictly increase on every iteration, leading to oscillations of increasing amplitude. To tackle this unstable oscillating behavior, we propose the Dissipative GDA method, which, as the name suggests, incorporates a simple friction term to GDA updates to dissipate the internal energy and stabilize the system. Our algorithm can be seen as a discrete-time version of [21], which has been applied to solve the C-RL problems [10]. In this work, we build on this literature, making the following contributions:

1. Novel control theory inspired algorithm: We illustrate how to use control theoretic concepts of dissipativity theory to design an algorithm that can stabilize the unstable behavior of GDA. Particularly, we show that by introducing a friction term, the proposed DGDA algorithm dissipates the stored internal energy and converges toward equilibrium.

2. Theoretical analysis with better rates: We establish the linear convergence of the DGDA method for bilinear and strongly convex-strongly concave saddle point problems. In both settings, we show that DGDA method outperforms other state-of-the-art first-order explicit methods, surpassing the standard known linear convergence rate (see Table I and II).

3. Numerical Validation: We corroborate our theoretical results with numerical experiments by evaluating the performance of the DGDA method with GDA, EG, and OGDA methods. When applied to solve bilinear and strongly convex-strongly concave saddle point problems, the DGDA method systematically outperforms other methods in terms of convergence rate.

Outline: The rest of the paper is organized as follows. In Section II, we provide some preliminary definitions and background. In Section III, we leverage tools from dissipativity theory and propose the Dissipative GDA (DGDA) algorithm to tackle the unstable oscillatory behavior of GDA methods. In Section IV, we establish its linear convergence rate for bilinear and strongly convex-strongly concave problems, which outperforms state-of-the-art first-order explicit algorithms, including GDA, EG and OGDA methods. In Section V, we support our claims with two numerical examples. We close the paper with concluding remarks and future research directions in Section VI.

[1] T. Zheng, and E. Mallada are with the Department of Electrical and Computer Engineering, N. Loizou is with the Department of Applied Mathematics and Statistics at Johns Hopkins University, Baltimore, MD 21218, USA {tzheng8,nloizou,mallada}@jhu.edu

[2] P. You is with the Department of Industrial Engineering and Management at Peking University, Beijing, China pcyou@pku.edu.cn

Bil.	Mokhtari, 2020	Azizian, 2020	This Work
EG	$\frac{\kappa^{-1}}{20}$	$\frac{\kappa^{-1}}{64}$	-
OG	$\frac{\kappa^{-1}}{800}$	$\frac{\kappa^{-1}}{128}$	-
DG	-	-	$\frac{\kappa^{-1}}{4}$

TABLE I: Summary of the global convergence results presented in Section IV for EG, GDA, and DGDA methods with bilinear objective functions. If a result shows that the iterates converge as $\mathcal{O}((1-r)^t)$, the quantity r is reported (the larger the better). κ represents the condition number.

S.C	Zhang, 2021	Mokhtari, 2020	Azizian, 2020	This Work
GD	κ^{-2}	-	-	-
EG	-	$\frac{\kappa^{-1}}{4}$	$\frac{\kappa^{-1}}{4}+\epsilon$	-
OG	-	$\frac{\kappa^{-1}}{4}$	$\frac{\kappa^{-1}}{4}+\epsilon$	-
DG	-	-	-	$\kappa^{-1}- \mathcal{O}(\kappa^{-2})$

TABLE II: Summary of the global convergence results presented in Section IV for GDA, EG, OGDA, and DGDA methods with strongly convex-strongly concave and L-Lipschitz objective functions. The table reports the term r of a $(1-r)$ linear rate. The constant $\epsilon > 0$ depends on the problem.

II. Problem Formulation

In this paper, we study the problem of finding saddle points in the min-max optimization problem:

$$\min_{x\in\mathbb{R}^n}\max_{y\in\mathbb{R}^m} f(x,y), \tag{1}$$

where the function $f:\mathbb{R}^n\times\mathbb{R}^m\to\mathbb{R}$ is a convex-concave function. Precisely, $f(\cdot,y)$ is convex for all $y\in\mathbb{R}^m$ and $f(x,\cdot)$ is concave for all $x\in\mathbb{R}^n$. We seek to develop a novel optimization algorithm that converges to some saddle point (x^*,y^*) of Problem 1.

Definition 1 (Saddle Point): A point $(x^*,y^*)\in\mathbb{R}^n\times\mathbb{R}^m$ is a saddle point of convex-concave function (1) if and only if it satisfies $f(x^*,y)\leq f(x^*,y^*)\leq f(x,y^*)$ for all $x\in\mathbb{R}^n, y\in\mathbb{R}^m$.

Throughout this paper, we consider two specific instances of Problem 1 commonly studied in related literature: strongly convex-strongly concave and bilinear functions. Herein, we briefly present some definitions and properties used in our results.

Definition 2 (Strongly Convex): A differentiable function $f:\mathbb{R}^n\to\mathbb{R}$ is said to be μ-strongly convex if $f(w)\geq f(w')+\nabla f(w)^T(w-w')+\frac{\mu}{2}\|w-w'\|^2$.

Notice that if $\mu=0$, then we recover the definition of convexity for a continuously differentiable function and $f(w)$ is μ-strongly concave if $-f(w)$ is μ-strongly convex. Another important property commonly used in the convergence analysis of optimization algorithms is the Lipschitz-ness of the gradient $\nabla f(w)$.

Definition 3 (L-Lipschitz): A function $F:\mathbb{R}^n\to\mathbb{R}^m$ is L-Lipschitz if $\forall w,w'\in\mathbb{R}^n$, we have $\|F(w)-F(w')\|\leq L\|w-w'\|$.

Combining the above two properties leads to the first important class of problem that has been extensively studied [1]–[3], [22].

Assumption 1: (Strongly strongly convex-concave functions with L-Lipschitz Gradient) The function $f:\mathbb{R}^n\times\mathbb{R}^m\to\mathbb{R}$ is continuously differentiable, μ strongly convex in x, and μ strongly concave in y. Further, the gradient vector $(\nabla_x f(x,y);-\nabla_y f(x,y))$ is L-Lipschitz.

It is also crucial to consider situations where the objective function is bilinear. Such bilinear min-max problems often appear when solving constrained reinforcement learning problems [10], [23], and training of WGANs [12].

Assumption 2 (Bilinear function): The function $f:\mathbb{R}^n\times\mathbb{R}^m\to\mathbb{R}$ is a bilinear function if it can be written in the form $f(x,y)=x^TAy$. For simplicity, we further assume that the matrix $A\in\mathbb{R}^{m\times n}$ is full rank, with $m\leq n$.

As seen in Table I and II as well as in Section IV, the linear convergence rates of existing algorithms are frequently characterized by the *condition number* κ. Specifically, when the objective function is bilinear, the condition number is defined as $\kappa:=\sigma^2_{\max}(A)/\sigma^2_{\min}(A)$, where $\sigma_{\max}(M)$ and $\sigma_{\min}(M)$ denote the largest singular value and smallest singular of a matrix M respectively. When the objective function is strongly convex-strongly concave with the L-Lipschitz gradient, the condition number of the problem is defined as $\kappa:=L/\mu$.

III. Dissipative Gradient Descent Ascent Algorithm

This section introduces the proposed first-order method for solving the min-max optimization problem 1. The algorithm can be seen as a discretization of the algorithm proposed by [21], wherein a regularization framework was introduced for continuous saddle flow dynamics that guarantees asymptotic convergence to a saddle point under mild assumptions. However, the continuous-time analysis presented in [21] does not generally extend to discrete time. In this paper, we show the linear convergence of the discrete-time version of this algorithm.

A. Algorithm Design

Our results build on gaining an intuitive understanding of the problems that one encounters when applying the vanilla GDA method to solve saddle point problems (1):

Gradient Descent Ascent (GDA)

$$\begin{aligned} x_{k+1} &= x_k-\eta\nabla_x f(x_k,y_k),\\ y_{k+1} &= y_k+\eta\nabla_y f(x_k,y_k). \end{aligned}$$

When (1) is strongly convex-strongly concave, and has L-Lipschitz gradients, the GDA method provides linear convergence, with step size $\eta=\mu/L^2$ and a know rate estimate of $1-1/\kappa^2$ [24]. However, when the problem is bilinear, the standard GDA method fails to converge. This is illustrated in the top plot in Figure 3.

Our proposed algorithm draws inspiration from dissipative theory in control by introducing two dynamic feedback controllers (friction) to dissipate the energy stored and amplified by the GDA algorithm. This is implemented in the form of high pass filters of the form

$$\begin{aligned}\zeta_{k+1} &= \zeta_k - \rho(\zeta_k - v_k)\\ w_k &= \rho(v_k - \zeta_k),\end{aligned}$$

with transfer function $\hat{w}(z) = \frac{z-1}{z-(1-\rho)}\hat{v}(z)$, that is interconnected in negative feedback to attenuate dampen the oscillations of both x_k and y_k. This modification leads to the proposed algorithm, effectively dampening the oscillations in our illustrative example in Figure1, and is formally introduced next.

Dissipative gradient descent ascent (DGDA):

$$\begin{bmatrix} x_{k+1}\\ \hat{x}_{k+1}\\ y_{k+1}\\ \hat{y}_{k+1}\end{bmatrix} = \begin{bmatrix} x_k - \eta\nabla_x f(x_k, y_k) - \rho(x_k - \hat{x}_k)\\ \hat{x}_k - \rho(\hat{x}_k - x_k)\\ y_k + \eta\nabla_y f(x_k, y_k) - \rho(y_k - \hat{y}_k)\\ \hat{y}_k - \rho(\hat{y}_k - y_k)\end{bmatrix} \tag{2}$$

Particularly, for f as in (1), in (2) we introduce two new sets of variables $\hat{x} \in \mathbb{R}^n$ and $\hat{y} \in \mathbb{R}^m$ and a damping parameter $\rho > 0$. One important observation is that, due to the high-pass filter structure of the feedback, once the system reaches equilibrium, i.e., $x_{k+1} = x_k$, $y_{k+1} = y_k$, $\hat{x}_{k+1} = \hat{x}_k$, $\hat{y}_{k+1} = \hat{y}_k$, one necessarily has $\hat{x}_k = x_k$ and $\hat{y}_k = y_k$, which ensures that the fixed point is necessarily a critical point of the saddle function.

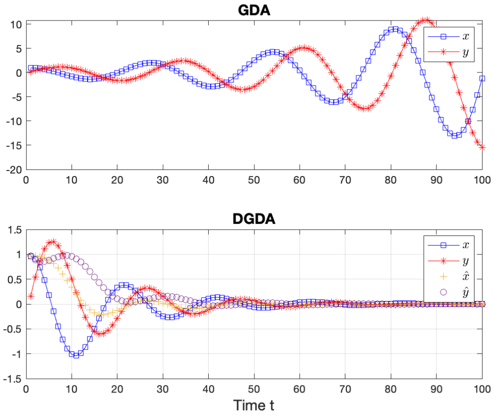

Fig. 1: Trajectories of states for GDA and DGDA for the simple bilinear objective function $f(x, y) := xy$.

In the remaining part of this section, we provide several key properties of the proposed DGDA update and discuss its differences with existing related algorithms. Then, in the next section, we formally prove that the proposed DGDA algorithm provides a linear convergence guarantee for both bilinear and strongly convex-strongly concave functions.

B. Key Properties and Related Algorithms

The first important observation is that the above DGDA update could be considered as applying a vanilla GDA update to the following regularized surrogate for $f(x, y)$:

$$f(x, y, \hat{x}, \hat{y}) := f(x, y) + \frac{\rho}{2}\|x - \hat{x}\|^2 - \frac{\rho}{2}\|y - \hat{y}\|^2. \tag{3}$$

We note that this is different from the *Proximal Point Method* [22], [25] or introducing a L_2 *regularization* [26], [27].

Differences with L_2 regularization. A commonly used method to ensure convergence is to introduce a L_2 regularization term in x and y [26], [27]:

$$\min_{x\in\mathbb{R}^n}\max_{y\in\mathbb{R}^m} f(x, y) + \frac{\rho}{2}\|x\|^2 - \frac{\rho}{2}\|y\|^2.$$

Although the augmented objective function becomes strongly convex-strongly concave and the vanilla GDA updates will converge, this regularization changes the saddle points. While our algorithm also introduces two regularizing terms, the following Lemma verifies the fixed positions of saddle points between $f(x, y)$ and $f(x, y, \hat{x}, \hat{y})$ with virtual variables aligned with original variables.

Lemma 1 (Saddle Point Invariance): [21, Lemma 6] For problem 1, a point (x^*, y^*) is a saddle point of $f(x, y)$ if and only if $(x^*, y^*, \hat{x}^*, \hat{y}^*)$ is a saddle point of $f(x, y, \hat{x}, \hat{y})$, with $\hat{x}^* = x^*$ and $\hat{y}^* = y^*$.

More interestingly, the regularization terms, $\frac{\rho}{2}\|x - \hat{x}\|^2$ and $\frac{\rho}{2}\|y - \hat{y}\|^2$, do not introduce extra strong convexity-stong concavity to the original problem. Precisely, the augmented problem $f(x, y, \hat{x}, \hat{y})$ is neither strongly convex on $(x, \hat{x})$ nor strongly concave on $(y, \hat{y})$. Indeed, on the hyperplane of $x = \hat{x}$ and $y = \hat{y}$, the augmented problem recovers the original problem $f(x, y, \hat{x}, \hat{y}) = f(x, y)$. Additionally, the introduced regularization terms of the DGDA method are separable and local, thus preserving the distributed structure that original systems may have. Consequently, it can be seamlessly integrated into a fully distributed approach.

Differences with Proximal Point Method. The Proximal Point Method for saddle point problems [22] shares a similar structure with DGDA algorithm. In the proximal method, the next iterates (x_{k+1}, y_{k+1}) is the unique solution to the saddle point problem

$$\begin{aligned}&(x_{k+1}, y_{k+1}) = \text{prox}_\eta(x_k, y_k)\\ &:= \arg\min_{x\in\mathbb{R}^n}\max_{y\in\mathbb{R}^m} f(x, y) + \frac{\eta}{2}\|x - x_k\|^2 - \frac{\eta}{2}\|y - y_k\|^2\end{aligned} \tag{4}$$

Using the optimality conditions of (4), the update of the Proximal Point method can be written as:

$$\begin{aligned}x_{k+1} &= x_k - \eta\nabla_x f(x_{k+1}, y_{k+1}),\\ y_{k+1} &= y_k + \eta\nabla_y f(x_{k+1}, y_{k+1}).\end{aligned}$$

As such, the Proximal Point method is an implicit method. Although Implicit methods are known to be more stable and to benefit from better convergence properties [6], [25], implementing the above updates requires computing the operators $(I + \eta\nabla_x f)^{-1}$ and $(I + \eta\nabla_y f)^{-1}$, and therefore may be computationally intractable. In contrast, the DGDA algorithm is an explicit algorithm, which applies a vanilla GDA update to the augmented objective function (3).

Thus, as shown in Table I and II, and in the next section, we choose to compare the convergence of DGDA only with other explicit algorithms for saddle point problems, such as the GDA, Extra-Gradient (EG), and Optimistic GDA (OGDA) methods which have comparable per-iteration computational requirements; although the DGDA algorithm has twice as many state variables, it only requires a single gradient computation per update. Moreover, there is no need for retaining and reemploying the extrapolated gradient, which also sets it apart from the OGDA method.

Smoothed GDA We finalize this section comparing DGDA with recent efforts leveraging *Moreau-Yosida* smoothing techniques to solve nonconvex-concave [28]–[30], nonconvex-nonconcave [31] min-max optimization problems. Unfortunately, the setting where such algorithm has been studied is different from the one considered in this paper, which difficult the comparison with the present work. A thorough comparison with DGDA is subject of current research.

IV. Convergence Analysis

In this section, we provide a theoretical analysis of the proposed algorithm. For the purpose of our analysis, we consider a quadratic Lyapunov function to track the energy dissipation of the DGDA updates

$$V_k := \|x_k - x^*\|^2 + \|y_k - y^*\|^2 + \|\hat{x}_k - \hat{x}^*\|^2 + \|\hat{y}_k - \hat{y}^*\|^2,$$

which denotes the square 2-norm distance to the saddle point at the k-th iteration. The goal is, therefore, to find some $0to \le \alpha < 1$ such that:

$$V_{k+1} \le \alpha V_k$$

where α denotes the linear convergence rate.

A. Convergence Analysis for Bilinear Functions

When applied to bilinear min-max optimization problem $f(x, y) = x^T Ay$, the DGDA update (2) is equivalent to a linear dynamical system. Specifically, denote $z = [x, y]^T, \hat{z} = [\hat{x}, \hat{y}]^T$ yields:

$$\begin{bmatrix} z_{k+1} - z^* \\ \hat{z}_{k+1} - \hat{z}^* \end{bmatrix} = \begin{bmatrix} (1-\rho)I - \eta M & \rho I \\ \rho I & (1-\rho)I \end{bmatrix} \begin{bmatrix} z_k - z^* \\ \hat{z}_k - \hat{z}^* \end{bmatrix}, \quad (5)$$

where

$$M = \begin{bmatrix} \mathbf{0} & A \\ -A^T & \mathbf{0} \end{bmatrix}.$$

As a result, the linear convergence of DGDA, as well as its convergence rate can be derived from the analysis of the spectrum of the associated matrix that defines the DGDA update in (5). This yields the following theorem.

Theorem 2: (Linear convergence of DGDA, bilinear case) Let Assumption 2 hold. Then the updates 2 of DGDA with $0 < \eta \le \frac{2\rho}{\sigma_{\max}(A)}$ and $\rho > 0$ provide linearly converging iterates. That is, there is a constant $\beta > 0$, independent of V_0 such that

$$V_k \le \mathcal{O}\left(\left(1-2\rho+2\rho^2+(1-\rho)\sqrt{4\rho^2-\eta^2\sigma^2_{\min}(A)}\right)^k\right) V_0,$$

Particularly, setting $\rho = 1/2$ and $\eta = 1/\sigma_{\max}(A)$ we have

$$V_k \le \mathcal{O}\left(\left(1 - \tfrac{1}{4\kappa}\right)^k\right) V_0.$$

Proof: The proof can be found in the appendix B

The first important observation is that linear convergence requires $\rho > 0$. This is not surprising since GDA could be interpreted as DGDA method when $\rho = 0$. More importantly, by choosing the optimal step size $\rho = 1/2, \eta = 1/\sigma_{\max}(A)$, DGDA method achieves a better linear convergence rate than the EG and OGDA methods (see Table I). Specifically, the above Theorem provides a linear convergence rate estimate of $\mathcal{O}\left(1 - \frac{1}{4\kappa}\right)$.

B. Strongly Convex Stronly Concave

We now consider the case of strongly convex-strongly concave min-max problems. Let $F(z_k) := (\nabla_x f(x_k, y_k), -\nabla_y f(x_k, y_k))$. The the DGDA updates can be written as follows:

$$\begin{bmatrix} z_{k+1} \\ \hat{z}_{k+1} \end{bmatrix} = \begin{bmatrix} z_k - \eta F(z_k) - \rho(z_k - \hat{z}_k) \\ \hat{z}_k - \rho(\hat{z}_k - z_k) \end{bmatrix}$$

Because of the existence of the nonlinear term $F(z_k)$, we cannot analyze the spectrum as in the previous bilinear case. This is indeed a common challenge in analyzing most optimization algorithms beyond a neighborhood of the fixed point. We circumvent this problem by leveraging recent results on the analysis of variational mappings as $F(\cdot)$ via integral quadratic constraint [17]–[19]. More details can be found in the Appendix where we prove the following theorem.

Theorem 3: (Linear convergence of DGDA, strongly convex-strongly concave case) Let Assumption 1 hold. Then the updates (2) with $\rho = 1/2$ and $\eta = 1/(L + \mu)$ of the Dissipative GDA algorihtm provide linearly converging iterates:

$$V_k \le \left(1 - \kappa^{-1} + \mathcal{O}(\kappa^{-2})\right)^k V_0 \quad (6)$$

Proof: The proof can be found in Appendix C.

Similarly, as in the bilinear case, we remark on the importance of the dissipation component. When $\rho = 0$ in (2), a similar analysis as in the proof of the theorem recovers the lower bound of the convergence rate of GDA $(1 - \kappa^{-2})$ as shown in [20, 3.1]. Thus, our DGDA method provides a better convergence rate estimate than GDA, since clearly $\kappa \in [1, \infty)$, and therefore $\kappa^{-2} \le \kappa^{-1}$.

It is important to point out, however, that while the rate obtain in Theorem 3 is clearly better than those of the EG and OGDA methods for large condition numbers κ (see Table II), the theorem fails to quantify the comparative performance of DGDA for small values of κ. The following corollary shows that indeed, the rate of DGDA is provably better for all $\kappa \ge 2$.

Corollary 4 (SCSC, comparison with known rates): Let Assumption 1 hold, and suppose that $L \ge 2m$, i.e., $\kappa \ge 2$. Then, the linear convergence rate estimate of DGDA (6) is smaller (better) than that of of EG and OGDA, i.e., $1 - \kappa^{-1}/4$ (Theorem 6&7 [3] and Theorem 4&7 [2]).

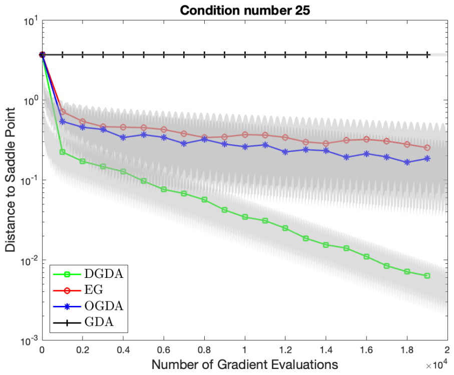

Fig. 2: Convergence of GDA, EG, OGDA, and DGDA in terms of the number of gradient evaluations for the bilinear problem in V-A. GDA diverges and the error is not shown. All other three algorithms converge linearly, where the DGDA method provides the best performance.

V. Numerical Experiments

In this section, we compare the performance of the proposed Dissipative gradient descent (DGDA) method with the Extra-gradient (EG), Gradient descent ascent (GDA), and Optimistic gradient descent ascent (OGDA) methods.

A. Bilinear problem

We first consider the following bilinear min-max optimization problem:

$$\min_{x\in\mathbb{R}^n}\max_{y\in\mathbb{R}^m} x^T Ay$$

where $A \in \mathbb{R}^{m\times n}$ is full-rank. The simulation results are illustrated in Figure 2 and Figure 3. In this experiment, we set the dimension of the problem to $m = n = 10$ and the iterates are initialized at x_0, y_0, which are randomly drawn from the uniform distribution on the open interval $(0, 1)$.

We plot the errors (distance to saddle points) of DGDA, EG, and OGDA versus the number of gradient evaluations for this problem in Figure 2. The solid line and grey-shaded error bars represent the average trajectories and standard deviations of 20 trials, where in each trial the randomly generated matrix A has a fixed condition number, i.e., $\kappa = \sigma^2_{\max}(A)/\sigma^2_{\min}(A) = 25$. The key motivation is that all three algorithms' convergence rates critically depend on κ^{-1}, and by fixing the condition number, we provide an explicit comparison of their convergence speed.

We pick the step size for different methods according to theoretical findings. That is, we select $\rho = 1/2$ and $\eta = 1/\sigma_{\max}(A)$ for DGDA (Theorem 2), $\eta = 1/4L = 1/4\sigma_{\max}(A)$ for EG and OGDA (Theorem 6&7 [3] and Theorem 4&7 [2]). In Figure 2, we do not show the error of GDA since it diverges for this bilinear saddle point problem. All other three algorithms converge linearly, with the DGDA method providing the best performance.

Finally, to provide a qualitative demonstration of how DGDA fares with other existing algorithms, we further plot the sample trajectories of GDA, EG, OGDA, and EGDA on a simple 2D bilinear min-max problem, with $m = n = 1$. In Figure 3, we observe that while GDA diverges, the trajectories of all other three algorithms converge linearly to the saddle point $(x^*, y^*) = (0, 0)$. Interestingly, our proposed algorithm (DGDA) despite taking larger steps, exhibits faster linear convergence.

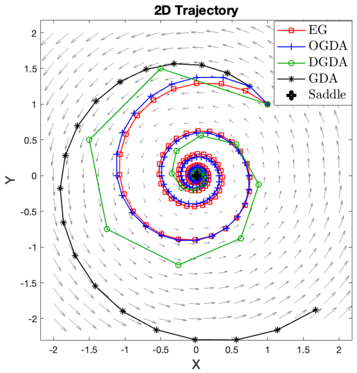

Fig. 3: Trajectories of GDA, EG, OGDA, and DGDA for a 2d bilinear problem. GDA diverges and all other three algorithms converge linearly, where the DGDA method provides the best performance.

B. Strongly convex-strongly concave problem

In the second numerical example, we focus on a strongly convex-strongly strongly-concave quadratic problem of the following form:

$$\min_{x\in\mathbb{R}^n}\max_{y\in\mathbb{R}^m} \frac{1}{2}x^T Ax - \frac{1}{2}y^T By + x^T Cy, \tag{7}$$

where the matrices satisfy $\mu_A I \preceq A \preceq L_A I$, $\mu_B I \preceq B \preceq L_B I$, $\mu_c^2 I \preceq C^T C \preceq L_c^2 I$. As a result, the problem (7) satisfy Assumption 1. In this experiment, we set the dimension of the problem to $n = 50, m = 10$, and the iterates are initialized at x_0, y_0, which are randomly drawn from the uniform distribution on the open interval $(0, 1)$. We plot the errors (distance to saddle points) of GDA, DGDA, EG, and OGDA versus the number of gradient evaluations for this problem in Figure 4. Again, the solid line and grey-shaded error bars represent the average trajectories and standard deviations of 20 trials, where in each trial the randomly generated matrix

$$\begin{bmatrix} A & C \\ -C^T & B \end{bmatrix}$$

is chosen such that the condition number of (7) remains constant, i.e., $\kappa = L/\mu = 31$. Similarly as in the bilinear problem in Section V-A, we pick the step size for the DGDA method according to our theoretical finding in Theorem 3. The step size of the GDA method is selected as $\eta = \mu/L^2$ (Theorem 5 [32]). The step sizes for EG and OGDA methods are selected as $\eta = 1/4L$ (Theorem 6&7 [3] and Theorem 4&7 [2]). According to the plots, all algorithms converge linearly, and the DGDA method has the best performance.

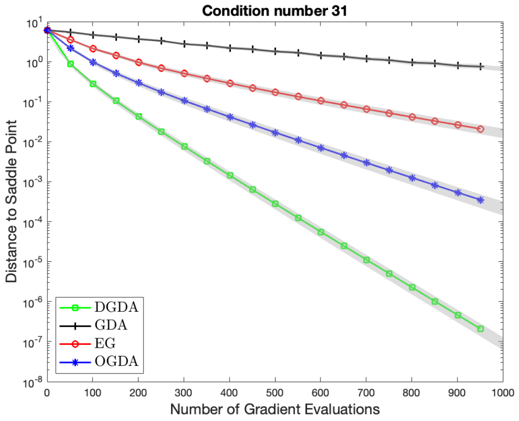

Fig. 4: Convergence of GDA, EG, OGDA, and DGDA in terms of the number of gradient evaluations for the strongly convex-strongly concave problem in 7. All algorithms converge linearly, and the DGDA method has the best performance.

VI. Conclusion and Future Work

In this work, we present the Dissipative GDA (DGDA) algorithm, a novel method for solving min-max optimization problems. Drawing inspiration from dissipativity theory and control theory, we address the challenge of diverging oscillations in bilinear min-max optimization problems when using the Gradient Descent Ascent (GDA) method. Particularly, we introduce a friction term into the GDA updates aiming to dissipate the internal energy and drive the system towards equilibrium. By incorporating a state-augmented regularization, our proposed DGDA method can be seen as performing standard GDA on an extended saddle function without introducing additional convexity. We further establish the superiority of the convergence rate of the proposed DGDA method when compared with other established methods including GDA, Extra-Gradient (EG), and Optimistic GDA. The analysis is further supported by two numerical examples, demonstrating its effectiveness in solving saddle point problems. Our future work includes studying the DGDA method in a stochastic setting and its application in solving Constrained Reinforcement learning problems in the policy space.

References

[1] P. Tseng, "On linear convergence of iterative methods for the variational inequality problem," *Journal of Computational and Applied Mathematics*, vol. 60, no. 1-2, pp. 237–252, 1995.

[2] A. Mokhtari, A. Ozdaglar, and S. Pattathil, "A unified analysis of extra-gradient and optimistic gradient methods for saddle point problems: Proximal point approach," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 1497–1507.

[3] W. Azizian, I. Mitliagkas, S. Lacoste-Julien, and G. Gidel, "A tight and unified analysis of gradient-based methods for a whole spectrum of differentiable games," in *International conference on artificial intelligence and statistics*. PMLR, 2020, pp. 2863–2873.

[4] T. Liang and J. Stokes, "Interaction matters: A note on non-asymptotic local convergence of generative adversarial networks," in *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 2019, pp. 907–915.

[5] A. Beznosikov, E. Gorbunov, H. Berard, and N. Loizou, "Stochastic gradient descent-ascent: Unified theory and new efficient methods," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2023, pp. 172–235.

[6] G. Gidel, H. Berard, G. Vignoud, P. Vincent, and S. Lacoste-Julien, "A variational inequality perspective on generative adversarial networks," *arXiv preprint arXiv:1802.10551*, 2018.

[7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[8] C. Daskalakis, A. Ilyas, V. Syrgkanis, and H. Zeng, "Training gans with optimism," *arXiv preprint arXiv:1711.00141*, 2017.

[9] D. Pfau and O. Vinyals, "Connecting generative adversarial networks and actor-critic methods," *arXiv preprint arXiv:1610.01945*, 2016.

[10] T. Zheng, P. You, and E. Mallada, "Constrained reinforcement learning via dissipative saddle flow dynamics," in *2022 56th Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2022, pp. 1362–1366.

[11] D. Ding, C.-Y. Wei, K. Zhang, and A. Ribeiro, "Last-iterate convergent policy gradient primal-dual methods for constrained mdps," *arXiv preprint arXiv:2306.11700*, 2023.

[12] J. Adler and S. Lunz, "Banach wasserstein gan," *Advances in neural information processing systems*, vol. 31, 2018.

[13] E. Altman, *Constrained Markov decision processes: stochastic modeling*. Routledge, 1999.

[14] S. Sastry, *Nonlinear systems: analysis, stability, and control*. Springer Science & Business Media, 2013, vol. 10.

[15] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*. SIAM, 1994.

[16] Z. E. Nelson and E. Mallada, "An integral quadratic constraint framework for real-time steady-state optimization of linear time-invariant systems," in *2018 annual American control conference (ACC)*. IEEE, 2018, pp. 597–603.

[17] B. Hu and L. Lessard, "Dissipativity theory for nesterov's accelerated method," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1549–1557.

[18] M. Fazlyab, A. Ribeiro, M. Morari, and V. M. Preciado, "Analysis of optimization algorithms via integral quadratic constraints: Nonstrongly convex problems," *SIAM Journal on Optimization*, vol. 28, no. 3, pp. 2654–2689, 2018.

[19] L. Lessard, B. Recht, and A. Packard, "Analysis and design of optimization algorithms via integral quadratic constraints," *SIAM Journal on Optimization*, vol. 26, no. 1, pp. 57–95, 2016.

[20] G. Zhang, X. Bao, L. Lessard, and R. Grosse, "A unified analysis of first-order methods for smooth games via integral quadratic constraints," *The Journal of Machine Learning Research*, vol. 22, no. 1, pp. 4648–4686, 2021.

[21] P. You and E. Mallada, "Saddle flow dynamics: Observable certificates and separable regularization," in *2021 American Control Conference (ACC)*. IEEE, 2021, pp. 4817–4823.

[22] R. T. Rockafellar, "Monotone operators and the proximal point algorithm," *SIAM journal on control and optimization*, vol. 14, no. 5, pp. 877–898, 1976.

[23] M. Wang, "Randomized linear programming solves the markov decision problem in nearly linear (sometimes sublinear) time," *Mathematics of Operations Research*, vol. 45, no. 2, pp. 517–546, 2020.

[24] B. Grimmer, H. Lu, P. Worah, and V. Mirrokni, "The landscape of the proximal point method for nonconvex–nonconcave minimax optimization," *Mathematical Programming*, vol. 201, no. 1-2, pp. 373–407, 2023.

[25] N. Parikh, S. Boyd *et al.*, "Proximal algorithms," *Foundations and trends® in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[26] A. Krogh and J. Hertz, "A simple weight decay can improve generalization," *Advances in neural information processing systems*, vol. 4, 1991.

[27] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2009, vol. 2.

[28] J. Zhang, P. Xiao, R. Sun, and Z. Luo, "A single-loop smoothed gradient descent-ascent algorithm for nonconvex-concave min-max problems," *Advances in neural information processing systems*, vol. 33, pp. 7377–7389, 2020.

[29] 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," *Mathematical Programming*, pp. 1–72, 2023.

[30] J. Yang, A. Orvieto, A. Lucchi, and N. He, "Faster single-loop algorithms for minimax optimization without strong concavity," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2022, pp. 5485–5517.

[31] T. Zheng, L. Zhu, A. M.-C. So, J. Blanchet, and J. Li, "Universal gradient descent ascent method for nonconvex-nonconcave minimax optimization," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[32] A. Beznosikov, B. Polyak, E. Gorbunov, D. Kovalev, and A. Gasnikov, "Smooth monotone stochastic variational inequalities and saddle point problems: A survey," *European Mathematical Society Magazine*, no. 127, pp. 15–28, 2023.

[33] G. M. Korpelevich, "The extragradient method for finding saddle points and other problems," *Matecon*, vol. 12, pp. 747–756, 1976.

[34] R. T. Rockafellar, "Augmented lagrangians and applications of the proximal point algorithm in convex programming," *Mathematics of operations research*, vol. 1, no. 2, pp. 97–116, 1976.

[35] J. Zhang and Z.-Q. Luo, "A proximal alternating direction method of multiplier for linearly constrained nonconvex minimization," *SIAM Journal on Optimization*, vol. 30, no. 3, pp. 2272–2302, 2020.

[36] G. Zhang and Y. Yu, "Convergence of gradient methods on bilinear zero-sum games," in *International Conference on Learning Representations*, 2019.

[37] J. P. Hespanha, *Linear systems theory*. Princeton university press, 2018.

Appendix

A. First Order Algorithms for Saddle Point Problems

In this section, we introduce several popular first-order methods for solving the min-max problem 1 in the machine learning community. Precisely, we focus on Gradient Descent-Ascent (GDA), Extra-gradient (EG), Optimistic Gradient Descent-Ascent (OGDA), Proximal Point and Smoothed-GDA methods.

Gradient descent ascent (GDA):

$$\begin{aligned} x_{k+1} &= x_k - \eta \nabla_x f(x_k, y_k), \\ y_{k+1} &= y_k + \eta \nabla_y f(x_k, y_k). \end{aligned}$$

When the problem is strongly convex-strongly concave and L-Lipschitz, the GDA method provides linear convergence, with step size $\eta = \mu/L^2$ and a know rate estimate of $1 - 1/\kappa^2$ [20], [24]. However, when the problem is bilinear, the standard GDA method fails to converge. Therefore, variants of the gradient method such as Extra-Gradient and Optimistic Gradient Descent-Ascent methods have attracted much attention in recent literature because of their superior empirical performance in solving min-max optimization problems such as training GANs and solving C-RL problems.

Extra-gradient (EG):

$$\begin{aligned} x_{k+1/2} &= x_k - \eta \nabla_x f(x_k, y_k), \\ y_{k+1/2} &= y_k + \eta \nabla_y f(x_k, y_k), \\ x_{k+1} &= x_k - \eta \nabla_x f(x_{k+1/2}, y_{k+1/2}), \\ y_{k+1} &= y_k + \eta \nabla_y f(x_{k+1/2}, y_{k+1/2}). \end{aligned}$$

Extra-gradient is a classical method introduced in [33], where its linear rate of convergence for bilinear functions and smooth strongly convex-strongly concave functions have been established in many recent works (see Table I and II). The Extra-gradient method first computes an extrapolated point $(x_{k+1/2}, y_{k+1/2})$ by performing a GDA update. Then the gradients evaluated at the extrapolated point are used to compute the new iterates (x_{k+1}, y_{k+1}).

The linear convergence rate of EG for strongly convex-strongly concave is established, with a standard known rate of $1 - 1/4\kappa$; see e.g. [2], [3]. One issue with the Extra-gradient method is that, as the name suggests, each update requires evaluation of extra gradients at the extrapolated point $(x_{k+1/2}, y_{k+1/2})$, which doubles the computational complexity of EG method compared to vanilla GDA method.

Optimistic gradient descent ascent (OGDA):

$$\begin{aligned} x_{k+1} &= x_k - 2\eta \nabla_x f(x_k, y_k) + \eta \nabla_x f(x_{k-1}, y_{k-1}), \\ y_{k+1} &= y_k + 2\eta \nabla_y f(x_k, y_k) - \eta \nabla_y f(x_{k-1}, y_{k-1}). \end{aligned}$$

The Optimistic gradient descent ascent (OGDA) method adds a "negative-momentum" term to each of the updates, which differentiates the OGDA method from the vanilla GDA method. Meanwhile, the OGDA method stores and re-uses the extrapolated gradient for the extrapolation, which only requires a single gradient computation per update.

The convergence properties of OGDA were also recently investigated in (refer to Table I and II), demonstrating linear convergence rates with smooth and bilinear functions, as well as strongly convex-strongly concave functions.

Proximal Point (PP): The proximal point method for convex minimization has been extensively studied [25], [34] and extended to solve saddle point problems in [22]. Define the iterates $\{x_{k+1}, y_{k+1}\}$ as the unique solution to the saddle point problem

$$\begin{aligned} &(x_{k+1}, y_{k+1}) = \text{prox}_\eta(x_k, y_k) \\ &:= \arg \min_{x \in \mathbb{R}^n} \max_{y \in \mathbb{R}^m} f(x, y) + \frac{\eta}{2}\|x - x_k\|^2 - \frac{\eta}{2}\|y - y_k\|^2 \end{aligned}$$

Using the optimality conditions of (4), the update of the Proximal Point method can be written as:

$$\begin{aligned} x_{k+1} &= x_k - \eta \nabla_x f(x_{k+1}, y_{k+1}), \\ y_{k+1} &= y_k + \eta \nabla_y f(x_{k+1}, y_{k+1}). \end{aligned}$$

This expression shows that in contrast to explicit methods such as GDA, EG, and OGDA methods, the Proximal Point method is an implicit method. Although Implicit methods are known to be more stable and to benefit from better convergence properties [6], [25], implementing the above updates requires computing the operators $(I + \eta \nabla_x f)^{-1}$ and $(I + \eta \nabla_y f)^{-1}$, and therefore may be computationally intractable. Notably, in [2], the authors show the EG and OGDA methods can be interpreted as an approximation of the PP method, and therefore exhibits similar convergence performance.

Smoothed-GDA method:

$$\begin{aligned} x_{k+1} &= x_k - c\nabla_x K(x_k, z_k; y_k), \\ y_{k+1} &= y_k + \alpha \nabla_K(x_{k+1}, z_k; y_k) \\ z_{k+1} &= z_k + \beta(x_{k+1} - z_k) \end{aligned}$$

where $K(x, z; y) = f(x, y) + \frac{p}{2}\|x - z\|^2$.

The Smoothed-GDA was independently introduced by Jiawei et al. in [35] and later [28]. It was originally motivated by ADMM to solve the linearly constrained nonconvex differentiable minimization problem [35], where they introduce an extra quadratic proximal term for the equality constraints and an extra sequence $\{z_k\}$. They claim this smoothing or exponential averaging scheme is necessary for the convergence of the proximal ADMM when the objective is nonconvex. Later on, this scheme is further extended to solve the nonconvex-concave min-max optimization problem [28].

B. Proof of Theorem 2

We consider, for ease of presentation, the case when $A \in \mathbb{R}^{m\times m}$ is a square matrix. The extension for non-square matrices is straightforward and has been covered in the literature [36, Appendix G]. Applying the updates 2 to $f(x,y) = x^T Ay$ and denoting $z = [x,y]^T, \hat{z} = [\hat{x},\hat{y}]^T$ yields:

$$\begin{aligned}\begin{bmatrix} z_{k+1}-z^* \\ \hat{z}_{k+1}-\hat{z}^*\end{bmatrix} &= \begin{bmatrix} z_k - \eta M z_k - \rho(z_k-\hat{z}_k) \\ \hat{z}_k - \rho(\hat{z}_k - z_k)\end{bmatrix} \\ &= \begin{bmatrix}(1-\rho)I - \eta M & \rho I \\ \rho I & (1-\rho)I\end{bmatrix}\begin{bmatrix} z_k - z^* \\ \hat{z}_k - \hat{z}^*\end{bmatrix},\end{aligned} \tag{8}$$

where

$$M = \begin{bmatrix}\mathbf{0} & A \\ -A^T & \mathbf{0}\end{bmatrix}$$

According to [3, Lemma 7]

$$\mathrm{Sp}(M) = \{\pm i\sigma | \sigma^2 \in \mathrm{Sp}(AA^T)\}.$$

We will use $\sigma_{\max}$ and $\sigma_{\min}$ to denote the largest singular value and smallest singular of matrix A, respectively. And according to Assumption 2, we have $\sigma_{\min} > 0$. Since M is a normal matrix and diagonalizable, we can compute the eigenvalues of the linear system (8) using the following similarity transformation

$$\begin{aligned}&\begin{bmatrix}(1-\rho)I - \eta M & \rho I \\ \rho I & (1-\rho)I\end{bmatrix} = \\ &\begin{bmatrix}U^{-1} & \mathbf{0} \\ \mathbf{0} & U^{-1}\end{bmatrix}\begin{bmatrix}(1-\rho)I - \eta\mathbf{\Lambda} & \rho I \\ \rho I & (1-\rho)I\end{bmatrix}\begin{bmatrix}U & \mathbf{0} \\ \mathbf{0} & U\end{bmatrix},\end{aligned} \tag{9}$$

where $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1,...,\lambda_{2m})$, $\lambda_{2j-1} = +i\sigma_j$, $\lambda_{2j} = -i\sigma_j$, with $\pm i\sigma_j \in \mathrm{Sp}(M), j = \{1,...,m\}$. In order to show linear convergence, we want to show that $\max_{j\in[m]} |\mu_j|^2 < 1$, where μ_j are the eigenvalues of the above matrix (9) (i.e., the spectral radius of matrix (9)). As straight forward computation leads to

$$\begin{aligned}\mu_j =& \frac{1}{2}(2-2\rho-\eta\lambda_j \pm \sqrt{\eta^2\lambda_j^2 + 4\rho^2}) \\ =& 1-\rho \pm i(\frac{1}{2}\eta\sigma_j) \pm \frac{1}{2}\sqrt{4\rho^2 - \eta^2\sigma_j^2}\end{aligned}$$

Since, for complex number c, $|c|^2 = c\bar{c}$, the magnitude of eigenvalues $|\mu_j|^2$ are given by,

$$\begin{aligned}|\mu_j|^2 =& \left(1-\rho+i\frac{1}{2}\eta\sigma_j \pm \frac{1}{2}\sqrt{4\rho^2-\eta^2\sigma_j^2}\right)\times \\ & \left(1-\rho-i\frac{1}{2}\eta\sigma_j \pm \frac{1}{2}\overline{\sqrt{4\rho^2-\eta^2\sigma_j^2}}\right) \\ =& (1-\rho)^2 + \frac{1}{4}\eta^2\sigma_j^2 + \frac{1}{4}|4\rho^2 - \eta^2\sigma_j^2 \pm (1-\rho)\Re(\sqrt{4\rho^2-\eta^2\sigma_j^2}) \\ & \pm \frac{i}{4}\eta\sigma_j\overline{\sqrt{4\rho^2-\eta^2\sigma_j^2}} \mp \frac{i}{4}\eta\sigma_j\sqrt{4\rho^2-\eta^2\sigma_j^2} \\ =& \begin{cases} 1-2\rho+2\rho^2 \pm (1-\rho)\sqrt{4\rho^2-\eta^2\sigma_j^2}, & \text{if } 4\rho^2-\eta^2\sigma_j^2 \geq 0 \\ 1-2\rho+\frac{1}{2}\eta^2\sigma_j^2 \pm \frac{1}{2}\eta\sigma_j\sqrt{\eta^2\sigma_j^2-4\rho^2}, & \text{if } \eta^2\sigma_j^2-4\rho^2 \geq 0\end{cases}\end{aligned}$$

Suppose that for all $j \in [m]$, we choose $0 < \eta \leq \frac{2\rho}{\sigma_{\max}} \leq \frac{2\rho}{\sigma_j}$ and $\rho > 0$, which implies $4\rho^2 - \eta^2\sigma_j^2 \geq 0$. From (B), $\forall j \in [m]$ we have,

$$\begin{aligned}|\mu_j|^2 &= 1-2\rho+2\rho^2 \pm (1-\rho)\sqrt{4\rho^2-\eta^2\sigma_j^2} \\ &\leq 1-2\rho+2\rho^2 + (1-\rho)\sqrt{4\rho^2-\eta^2\sigma_j^2} \\ &\leq 1-2\rho+2\rho^2 + (1-\rho)\sqrt{4\rho^2-\eta^2\sigma_{\min}^2} \\ &< 1-2\rho+2\rho^2 + (1-\rho)\sqrt{4\rho^2} \\ &= 1\ .\end{aligned}$$

According to classical linear system theory, e.g. [37, Theorem 8.3], the above spectral radius analysis of the linear system (8) results in the following linear convergence rate estimate:

$$V_k \leq \mathcal{O}\left(\left(1-2\rho+2\rho^2+(1-\rho)\sqrt{4\rho^2-\eta^2\sigma_{\min}^2}\right)^k\right)V_0,$$

where $V_k := \|x_k - x^*\|^2 + \|y_k - y^*\|^2 + \|\hat{x}_k - \hat{x}^*\|^2 + \|\hat{y}_k - \hat{y}^*\|^2$.

Furthermore, we want to select the optimal step size ρ, η. The immediate step is to substitute the optimal $\eta = \frac{2\rho}{\sigma_{\max}}$, which yields the following inequality:

$$|\mu_j|^2 \leq 1-2\rho+2\rho^2+(1-\rho)\sqrt{4\rho^2 - \frac{4\rho^2}{\sigma_{\max}^2}\sigma_j^2}\ ,\forall j \in [m].$$

The spectral radius is therefore given by choosing $\sigma_j = \sigma_{\min}$ above, i.e.,

$$\begin{aligned}\max_{j\in[m]}|\mu_j|^2 &= 2\rho^2 - 2\rho + 1 + (1-\rho)2\rho\sqrt{1-\frac{\sigma_{\min}^2}{\sigma_{\max}^2}} \\ &= \rho^2\left(2-2\sqrt{1-\frac{\sigma_{\min}^2}{\sigma_{\max}^2}}\right) - \rho\left(2-2\sqrt{1-\frac{\sigma_{\min}^2}{\sigma_{\max}^2}}\right)+1 \\ &\leq 1 - \frac{1}{2}(1-\sqrt{1-\frac{\sigma_{\min}^2}{\sigma_{\max}^2}}) \\ &= \frac{1}{2}+\frac{1}{2}\sqrt{1-\frac{\sigma_{\min}^2}{\sigma_{\max}^2}}\ ,\forall j \in [m]\end{aligned}$$

where the last inequality comes from selecting optimal $\rho = \frac{1}{2}$ of a quadratic polynomial of ρ. Using the fact that $\sqrt{1-x} \leq 1 - x/2$, we have

$$\max_{j\in[m]} |\mu_j|^2 \leq 1 - \frac{1}{4}\frac{\sigma_{\min}^2}{\sigma_{\max}^2}$$

Again, this results in the following linear convergence rate estimate:

$$V_k \leq \mathcal{O}\left(\left(1 - \tfrac{1}{4\kappa}\right)^k\right) V_0.$$

Remark 1: We could also choose $\eta = \frac{2\rho}{\sigma_{\min}}$ such that $\eta^2\sigma_j^2 - 4\rho^2 \geq 0$. And we could construct a similar linear convergence rate by repeating the above process. However, in practice, we found that the step sizes $\eta = \frac{2\rho}{\sigma_{\max}}, \rho = 1/2$ always perform better in numerical experiments. Therefore, we choose this pair of step sizes by default.

Remark 2: Since GDA method could be interpreted as a special case of DGDA method when selecting $\rho = 0$, the above step proves that when $\eta > 0$, the GDA method diverges for a bilinear objective function. Specifically, when $\rho = 0, \eta > 0$, we have $\eta^2\sigma_j^2 - 4\rho^2 > 0$ and

$$|\mu_j|^2 = 1 \pm \frac{1}{2}\eta^2\sigma_j^2,$$

C. Proof of Theorem 3

The proof relies on the application of dissipativity theory to construct Lyapunov functions and establish linear convergence. For more detailed information, refer to [17].

According to [17], a linear dynamical system of the form:

$$\xi_{k+1} = A\xi_k + Bw_k \tag{10}$$

Here, $\xi \in \mathbb{R}^{n_\xi}$ is the state, $w_k \in \mathbb{R}^{n_w}$ is the input, A is the state transition matrix and B is the input matrix. Suppose that there exist a (Lyapunov) function V, satisfying $V(\xi) \geq 0, \forall \xi \in \mathbb{R}^{n_\xi}$, some $0 \leq \alpha < 1$ and a supply rate function $S(\xi_k, w_k) \leq 0, \forall k$ such that

$$V(\xi_{k+1}) - \alpha^2 V(\xi_k) \leq S(\xi_k, w_k). \tag{11}$$

This dissipation inequality (11) implies that $V(\xi_{k+1}) \leq \alpha^2 V(\xi_k)$, and the state will approach a minimum value ate equilibrium no slower than the linear rate α^2. The flowing theorem states how to construct the dissipation inequality (11) by solving a semidefinite programming problem.

Theorem 5: [17][Theorem 2] Consider the following quadratic supply rate with $X \in \mathbb{R}^{(n_\xi+n_w)\times(n_\xi+n_w)}$ and $X^T = X$

$$S(\xi, w) := \begin{bmatrix}\xi \\ w\end{bmatrix}^T X \begin{bmatrix}\xi \\ w\end{bmatrix}.$$

If there exists matrix $P \in \mathbb{R}^{n_\xi \times n_\xi}$ with $P \succeq 0$ such that

$$\begin{bmatrix} A^TPA - \alpha^2 P & A^TPB \\ B^TPA & B^TPB \end{bmatrix} - X \leq 0, \tag{12}$$

then the dissipation inequality holds for all trajectories of (10) with $V(\xi) = \xi^T P \xi$.

A major benefit of the proposed constructive dissipation approach is that it replaces the trouble some component of a dynamical system (e.g. the gradient term $w = \nabla_\xi f(\xi)$) by a quadratic constraint on its inputs and outputs that is always satisfied, namely the supply rate constraint $S(\xi, w) \leq 0$. This leads to a two-step novel approach to the convergence analysis of optimization algorithms.

1) Choose a proper quadratic supply rate function S such that $S(\xi_k, w_k) \leq 0, \forall k$, that depends on the specific nonlinear term.
2) Solve the Linear Matrix Inequality (12) to obtain a storage function V and finding the linear convergence rate α.

We will apply this methodology to analyze the DGDA update (2). Let $z = [x, y]^T, \hat{z} = [\hat{x}, \hat{y}]^T$ and $F(z_k) = (\nabla_x f(x_k, y_k); -\nabla_y f(x_k, y_k))$, and rewrite (2) as in the form of (10):

$$\begin{bmatrix} z_{k+1} \\ \hat{z}_{k+1}\end{bmatrix} = \begin{bmatrix} z_k - \eta F(z_k) - \rho(z_k - \hat{z}_k) \\ \hat{z}_k - \rho(\hat{z}_k - z_k)\end{bmatrix}$$
$$= \begin{bmatrix} 1-\rho & \rho \\ \rho & 1-\rho \end{bmatrix}\begin{bmatrix} z_k \\ \hat{z}_k \end{bmatrix} + \begin{bmatrix} -\eta \\ 0 \end{bmatrix} w_k$$

where $w_k = F(z_k)$.

According to the previous discussion, the first step would be to choose a proper quadratic supply rate function S such that $S(\xi_k, w_k) \leq 0, \forall k$, where $\xi_k = (z_k; \hat{z}_k)$ that depends on the specific nonlinear term $w_k = F(z_k)$. According to the equations (7) in the work by Hu et al. (2017) [17] and Lemma 6 from the research by Lessard et al. (2016) [19], the following applies to the nonlinear operator $F(z_k)$ that meets the conditions specified in Assumption 1:

$$S(z_k, w_k) = \begin{bmatrix} z_k \\ w_k \end{bmatrix}^T \begin{bmatrix} 2\mu L I & (-\mu + L)I \\ (-\mu + L)I & 2I \end{bmatrix}\begin{bmatrix} z_k \\ w_k \end{bmatrix} \leq 0$$

The conditions in Assumption 1 are also commonly referred to as being L-smooth and m-strongly monotone, as can be found in related literature on variational inequality problems, such as the works by [6], [20]]. Therefore, we could easily extend the above LMI into the following supply rate function for DGDA updates, by augmenting the states $\xi_k = (z_k; \hat{z}_k)$:

$$S(\xi_k, w_k) =$$
$$\begin{bmatrix} z_k \\ \hat{z}_k \\ w_k \end{bmatrix}^T \begin{bmatrix} 2\mu L I & 0 & (-\mu + L)I \\ 0 & 0 & 0 \\ (-\mu + L)I & 0 & 2I \end{bmatrix}\begin{bmatrix} z_k \\ \hat{z}_k \\ w_k \end{bmatrix} \leq 0$$

as a proper quadratic supply rate function $S(\xi_k, w_k) \leq 0$, whenever $w_k = F(z_k)$.

Finally, according to Theorem 5 and the above discussion, proving linear convergence reduces to finding a positive definite a matrix $P \in \mathbf{R}^{2(n+m)\times 2(n+m)}$, $\alpha \in [0, 1)$ such that (12) is satisfied, where the problem parameters are given by

$$A = \begin{bmatrix} 1-\rho & \rho \\ \rho & 1-\rho \end{bmatrix} \otimes I, B = \begin{bmatrix} -\eta \\ 0 \end{bmatrix} \otimes I,$$
$$X = \begin{bmatrix} 2\mu L & 0 & (-\mu + L) \\ 0 & 0 & 0 \\ (-\mu + L) & 0 & 2 \end{bmatrix} \otimes I,$$

where $\otimes$ is the Kronecker product. Due to the Kronecker structure of this problem, this is equivalent to solving an LMI problem of dimension 3 by 3, with design parameters $P = \bar{P} \otimes I$, with $\bar{P} \in \mathbf{R}^{2\times 2}$, $\alpha^2 \in [0, 1)$, ρ and η. Because this Linear Matrix Inequality is simple (3 by 3), it can be solved using analytical methods. This, in turn, results in a feasible solution for the LMI, denoted as follows:

$$\rho = \frac{1}{2}, \qquad \eta = \frac{1}{L+\mu},$$
$$\alpha^2 = \frac{3L^2 + 2L\mu + 3\mu^2 + \sqrt{(L+\mu)^4 + 16L^2\mu^2}}{4(L+\mu)^2},$$
$$P = \begin{bmatrix} (L+\mu)^2 & 0 \\ 0 & (L+\mu)^2 \end{bmatrix} \otimes I.$$

After substituting the definition for condition number $\kappa := L/\mu$, the convergence rate α^2 simplifies to:

$$\alpha^2 = 1 - \kappa^{-1} + \mathcal{O}\big((\frac{\mu}{L})^2\big)$$

D. Proof of Corollary 4

According to Theorem 3, the linear convergence rate estimate of DGDA is

$$\frac{3L^2 + 2L\mu + 3\mu^2 + \sqrt{(L+\mu)^4 + 16L^2\mu^2}}{4(L+\mu)^2}$$
$$= \frac{3\kappa^2 + 2\kappa + 3 + \sqrt{(\kappa+1)^4 + 16\kappa^2}}{4(\kappa+1)^2},$$

where $\kappa = L/\mu$.

According to Theorem 6&7 [3] and Theorem 4&7 [2], the standard known linear convergence rate estimate of EG and OGDA is

$$1 - \frac{\mu}{4L} = 1 - \frac{1}{4\kappa}.$$

By simple algebraic calculation, it can be shown that as a function of κ, when $\kappa \geq 2$, the following polynomial is always nonnegative, i.e.,

$$1 - \frac{1}{4\kappa} - \frac{3\kappa^2 + 2\kappa + 3 + \sqrt{(\kappa+1)^4 + 16\kappa^2}}{4(\kappa+1)^2} \geq 0.$$