LEAD: Min-Max Optimization from a Physical Perspective

Reyhane Askari Hemmat [*1 2] **Amartya Mitra** [*3] **Guillaume Lajoie** [2 4 5] **Ioannis Mitliagkas** [1 2 5]

Abstract

Adversarial formulations have rekindled interest in two-player min-max games. A central obstacle in the optimization of such games is the rotational dynamics that hinder their convergence. In this paper, we show that game optimization shares dynamic properties with particle systems subject to multiple forces, and one can leverage tools from physics to improve optimization dynamics. Inspired by the physical framework, we propose LEAD, an optimizer for min-max games. Next, using Lyapunov stability theory from dynamical systems as well as spectral analysis, we study LEAD's convergence properties in continuous and discrete time settings for a class of quadratic min-max games to demonstrate linear convergence to the Nash equilibrium. Finally, we empirically evaluate our method on synthetic setups and CIFAR-10 image generation to demonstrate improvements in GAN training.

1. Introduction

Much of the advances in traditional machine learning can be attributed to the success of gradient-based methods. Modern machine learning systems such as GANs (Goodfellow et al., 2014), multi-task learning, and multi-agent settings (Sener & Koltun, 2018) in reinforcement learning (Bu et al., 2008) require joint optimization of two or more objectives which can often be formulated as games. In these *game* settings, best practices and methods developed for single-objective optimization are observed to perform noticeably poorly (Mescheder et al., 2017; Balduzzi et al., 2018; Gidel et al., 2019). Specifically, they exhibit rotational dynamics in parameter space about the *Nash Equilibria* (Mescheder et al., 2017), slowing down convergence. Recent work in game optimization (Wang et al., 2019; Mazumdar et al., 2019; Mescheder et al., 2017; Balduzzi et al., 2018; Abernethy et al., 2019; Loizou et al., 2020) demonstrates that introducing additional second-order terms in the optimization algorithm helps to suppress these rotations, thereby improving convergence.

Taking inspiration from recent work in single-objective optimization that re-derives existing accelerated methods from a variational perspective (Wibisono et al., 2016; Wilson et al., 2016), in this work, we adopt a similar approach in the context of games. To do so, we borrow formalism from physics by likening the gradient-based optimization of two-player (zero-sum) games to the dynamics of a system where we introduce relevant forces that helps curb these rotations. We consequently utilize the dynamics of this resultant system to propose our novel second-order optimizer for games, *LEAD*.

Next, using Lyapunov and spectral analysis, we demonstrate linear convergence of our optimizer (LEAD) in both continuous and discrete-time settings for a class of quadratic min-max games. In terms of empirical performance, LEAD achieves an FID of 10.49 on CIFAR-10 image generation, outperforming existing baselines such as BigGAN (Brock et al., 2018), which is approximately 30-times larger than our baseline ResNet architecture.

What distinguishes LEAD from other second-order optimization methods for min-max games such as (Mescheder et al., 2017; Wang et al., 2019; Mazumdar et al., 2019; Schäfer & Anandkumar, 2019) is its computational complexity. All these different methods involve Jacobian (or Jacobian-inverse) vector-product computation commonly implemented using a form of approximation. Thus making a majority of them intractable in real-world large scale problems. On the other hand, LEAD involves computing only *one-block* of the full Jacobian of the gradient vector-field multiplied by a vector. This makes our method significantly cheaper and comparable to several first-order methods, as we show in Section 5. We summarize our contributions as following: 1) In Section 3, we model gradient descent-ascent as a physical system. Armed with the physical model, we introduce counter-rotational forces to curb the existing rotations in the system. Next, we employ the principle of

[*]Equal contribution [1]Department of Computer Science and Operations Research University of Montreal [2]Mila, Quebec AI Institute [3]University of California, Riverside [4]Department of Mathematics and Statistics University of Montreal [5]Canada CIFAR AI chair. Correspondence to: Reyhane Askari Hemmat <reyhane.askari.hemmat@umontreal.ca>.

Workshop on New Frontiers in Learning, Control, and Dynamical Systems at the International Conference on Machine Learning (ICML), Honolulu, Hawaii, USA, 2023.

least action to determine the (continuous-time) dynamics. We then accordingly discretize these resultant dynamics to obtain our optimization scheme, Least Action Dynamics (LEAD). 2) In Section 4, we use Lyapunov stability theory to prove a linear convergence of LEAD in continuous for quadratic min-max games. 3) Finally, in Section 5, we empirically demonstrate that LEAD is computationally efficient. Additionally, we demonstrate that LEAD improves the performance of GANs on different tasks such as 8-Gaussians and CIFAR-10 while comparing the performance of our method against other first and second-order methods.

Note that we also study LEAD in discrete-time using spectral analysis. See Appendix B.

2. Problem Setting

Notation Continuous time scalar variables are in uppercase letters (X), discrete-time scalar variables are in lower case (x) and vectors are in boldface ($\mathbf{A}$). Matrices are in blackboard bold ($\mathbb{M}$) and derivatives w.r.t. time are denoted as an over-dot ($\dot{x}$). Furthermore, off-diag$[\mathbb{M}]_{i,j}$ is equal to $\mathbb{M}_{i,j}$ for $i \neq j$, and equal to 0 for $i = j$ where $i, j = 1, 2, \ldots, n$.

Setting In this work, we study the optimization problem of two-player zero-sum games,

$$\min_X \max_Y f(X, Y), \tag{1}$$

where $f : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$, and is assumed to be a convex-concave function which is continuous and twice differentiable w.r.t. $X, Y \in \mathbb{R}$. It is to be noted that though in developing our framework below, X, Y are assumed to be scalars, it is nevertheless found to hold for the more general case of vectorial X and Y, as we demonstrate both analytically (Appendix D) and empirically, our theoretical analysis is found to hold.

3. Optimization Mechanics

In our effort to study min-max optimization from a physical perspective, we note from classical physics the following: under the influence of a net force F, the equation of motion of a physical object of mass m, is determined by Newton's 2nd Law,

$$m\ddot{X} = F, \tag{2}$$

with the object's coordinate expressed as $X_t \equiv X$. According to the *principle of least action*[1] (Landau & Lifshitz, 1960), nature "selects" this particular trajectory over other possibilities, as a quantity called the *action* is extremized along it.

[1]Also referred to as the Principle of Stationary Action.

We start with a simple observation that showcases the connection between optimization algorithms and physics. Polyak's heavy-ball momentum (Polyak, 1964) is often perceived from a physical perspective as a ball moving in a "potential" well (cost function). In fact, it is straightforward to show that Polyak momentum is a discrete counterpart of a continuous-time equation of motion governed by Newton's 2nd Law. For single-objective minimization of an objective function $f(x)$, Polyak momentum follows:

$$x_{k+1} = x_k + \beta(x_k - x_{k-1}) - \eta \nabla_x f(x_k), \tag{3}$$

where η is the learning rate and β is the momentum coefficient. For simplicity, setting β to one, and moving to continuous time, one can rewrite this equation as,

$$\frac{(x_{k+\delta} - x_k) - (x_k - x_{k-\delta})}{\delta^2} = -\frac{\eta}{\delta^2} \nabla_x f(x_k), \tag{4}$$

and in the limit $\delta, \eta \to 0$, Eq.(4) then becomes ($x_k \to X(t) \equiv X$),

$$m\ddot{X} = -\nabla_X f(X). \tag{5}$$

This is equivalent to Newton's 2nd Law of motion (Eq.(2)) of a particle of mass $m = \delta^2/\eta$, and identifying $F = -\nabla_X f(X)$ (i.e. $f(X)$ acting as a *potential* function (Landau & Lifshitz, 1960)). Thus, Polyak's heavy-ball method Eq.(3) can be interpreted as an object (ball) of mass m rolling down under a potential $f(X)$ to reach the minimum while accelerating.

Armed with this observation, we perform an extension of 5 to our min-max setup,

$$\begin{aligned} m\ddot{X} &= -\nabla_X f(X, Y), \\ m\ddot{Y} &= \nabla_Y f(X, Y), \end{aligned} \tag{6}$$

which represents the dynamics of an object moving under a *curl force* (Berry & Shukla, 2016): $\boldsymbol{F}_{\text{curl}} = (-\nabla_X f, \nabla_Y f)$ in the 2-dimensional $X - Y$ plane. Furthermore, it is to be noted that discretization of Eq.(6) corresponds to Gradient Descent-Ascent (GDA) with momentum 1. Authors in (Gidel et al., 2019) found that this optimizer is divergent in the prototypical min-max objective, $f(X, Y) = XY$, thus indicating the need for further improvement.

To this end, we note that the failure modes of the optimizer obtained from the discretization of Eq.(6), can be attributed to: *(a)* an outward rotatory motion by our particle of mass m, accompanied by *(b)* an increase in its velocity over time. Following these observations, we aim to introduce suitable *counter-rotational* and *dissipative* forces to our system above, in order to tackle *(a)* and *(b)* in an attempt to achieve converging dynamics.

Specifically, as an initial consideration, we choose to add to our system, two ubiquitous forces:

- magnetic force,

$$\boldsymbol{F}_{\text{mag}} = \Big(-q\nabla_{XY} f\, \dot{Y}, q\nabla_{XY} f\, \dot{X} \Big) \tag{7}$$

known to produce rotational motion (in charged particles), to counteract the rotations introduced by $\boldsymbol{F}_{\text{curl}}$. Here, q is the charge imparted to our particle.

- friction,

$$\boldsymbol{F}_{\text{fric}} = \big(\mu\dot{X}, \mu\dot{Y}\big) \tag{8}$$

to prevent the increase in velocity of our particle (μ: coefficient of friction).

Assimilating all the above forces $\boldsymbol{F}_{\text{curl}}$, $\boldsymbol{F}_{\text{mag}}$ and $\boldsymbol{F}_{\text{fric}}$, the equations of motion (EOMs) of our crafted system then becomes,

$$\begin{aligned} m\ddot{X} &= \boldsymbol{F}_{\text{curl}} + \boldsymbol{F}_{\text{mag}} + \boldsymbol{F}_{\text{fric}}, \\ m\ddot{Y} &= \boldsymbol{F}_{\text{curl}} + \boldsymbol{F}_{\text{mag}} + \boldsymbol{F}_{\text{fric}}. \end{aligned} \tag{9}$$

Or equivalently,

$$\begin{aligned} m\ddot{X} &= -\mu\dot{X} - \nabla_X f - q\nabla_{XY} f\dot{Y}, \\ m\ddot{Y} &= -\mu\dot{Y} + \nabla_Y f + q\nabla_{XY} f\dot{X}. \end{aligned} \tag{10}$$

Without loss of generality, from hereon we set the mass of our object to be unity. In the rest of this work, we study the above EOMs in continuous and discrete time for min-max games.

3.1. Discretization

The continuous-time EOMs (10) can be discretized in an implicit-explicit way, to yield,

$$\begin{aligned} x_{k+1} &= x_k + \beta(x_k - x_{k-1}) - \eta\nabla_x f\,(x_k, y_k) \\ &\quad - \alpha\nabla_{xy} f\,(x_k, y_k)\,(y_k - y_{k-1}), \\ y_{k+1} &= y_k + \beta(y_k - y_{k-1}) + \eta\nabla_y f\,(x_k, y_k) \\ &\quad + \alpha\nabla_{yx} f\,(x_k, y_k)\,(x_k - x_{k-1}), \end{aligned} \tag{11}$$

We name this algorithm *Least Action Dynamics (LEAD)* as it corresponds to the trajectory of a charged particle under a curl, magnetic and frictional force, as governed by the principle of least action.

4. Convergence Analysis

We study the behavior of LEAD on the quadratic min-max game,

$$f\,(\mathbf{X}, \mathbf{Y}) = \frac{h}{2}||\mathbf{X}||^2 - \frac{h}{2}||\mathbf{Y}||^2 + \mathbf{X}^T\mathbb{A}\mathbf{Y}, \tag{12}$$

where $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^n$, $\mathbb{A} \in \mathbb{R}^n \times \mathbb{R}^n$ is a (constant) coupling matrix and h is a scalar constant. Let us further define the *vector field* $\mathbf{v}$ of the above game, f, as,

$$\boldsymbol{v} = \begin{bmatrix} \nabla_{\mathbf{X}} f\,(\mathbf{X}, \mathbf{Y}) \\ -\nabla_{\mathbf{Y}} f\,(\mathbf{X}, \mathbf{Y}) \end{bmatrix} = \begin{bmatrix} h\boldsymbol{X} + \mathbb{A}\mathbf{Y} \\ h\boldsymbol{Y} - \mathbb{A}^\top\mathbf{X} \end{bmatrix}. \tag{13}$$

.

For this quadratic min-max game, Eq.(10) generalizes to,

$$\begin{aligned} \ddot{\boldsymbol{X}} &= -\mu\dot{\boldsymbol{X}} - (h + \mathbb{A})\boldsymbol{Y} - q\mathbb{A}\dot{\boldsymbol{Y}} \\ \ddot{\boldsymbol{Y}} &= -\mu\dot{\boldsymbol{Y}} - (h - \mathbb{A}^T)\boldsymbol{X} + q\mathbb{A}^T\dot{\boldsymbol{X}}, \end{aligned} \tag{14}$$

Theorem 4.1. *For the dynamics of Eq.(14),*

$$\begin{aligned} \mathcal{E}_t &= \frac{1}{2}\Big(\dot{\boldsymbol{X}} + \mu\boldsymbol{X} + \mu\mathbb{A}\boldsymbol{Y}\Big)^T\Big(\dot{\boldsymbol{X}} + \mu\boldsymbol{X} + \mu\mathbb{A}\boldsymbol{Y}\Big) \\ &\quad + \frac{1}{2}\Big(\dot{\boldsymbol{Y}} + \mu\boldsymbol{Y} - \mu\mathbb{A}^T\boldsymbol{X}\Big)^T\Big(\dot{\boldsymbol{X}} + \mu\boldsymbol{Y} - \mu\mathbb{A}^T\boldsymbol{X}\Big) \\ &\quad + \frac{1}{2}\Big(\dot{\boldsymbol{X}}^T\dot{\boldsymbol{X}} + \dot{\boldsymbol{Y}}^T\dot{\boldsymbol{Y}}\Big) + \boldsymbol{X}^T(h + \mathbb{A}\mathbb{A}^T)\boldsymbol{X} + \boldsymbol{y}^T(h + \mathbb{A}^T\mathbb{A})\boldsymbol{Y} \end{aligned} \tag{15}$$

is a Lyapunov function of the system. Furthermore, setting $q = (2/\mu) + \mu$, we find $\dot{\mathcal{E}}_t \leq -\rho\mathcal{E}_t$ for

$$\rho \leq \min\left\{\frac{\mu}{1+\mu},\ \frac{2\mu(\sigma_{\min}^2 + h)}{(1 + \sigma_{\min}^2 + 2h)\,(\mu^2 + \mu) + 2\sigma_{\min}^2}\right\}$$

with $\sigma_{\min}$ being the smallest singular value of $\mathbb{A}$. This consequently ensures linear convergence of the dynamics of Eq. (14),

$$\boxed{||\boldsymbol{X}||^2 + ||\boldsymbol{Y}||^2 \leq \frac{\mathcal{E}_0}{h + \sigma_{min}^2}\exp\left(-\rho t\right)}. \tag{16}$$

5. Experiments

We evaluate LEAD (implemented on top of Adam) on complex, deep architectures. We adapt the ResNet architecture in SN-GAN (Miyato et al., 2018). We compare with several existing results on the task of image generation on CIFAR-10 using ResNets. See Table 1 for a full comparison. Note that, Style-GAN based models (Sauer et al., 2022; Kang et al., 2021; Lee et al., 2021) or BigGAN based models (Brock et al., 2018; Lorraine & Duvenaud, 2022) use architectures that are ¿ 30X larger than the architecture that we have chosen to test our method on. Our method obtains a competitive FID of 10.49.

6. Related Work

Game Optimization: With increasing interest in games, significant effort is being spent in understanding common issues affecting optimization in this domain. These issues range from convergence to non-Nash equilibrium points, to exhibiting rotational dynamics around the equilibrium which hampers convergence. Authors in (Mescheder et al., 2017) discuss how the eigenvalues of the Jacobian govern the local convergence properties of GANs. They argue that the presence of eigenvalues with zero real-part and

Table 1: Performance of several methods on CIFAR-10 image generation task. The FID and IS is reported over 50k samples unless mentioned otherwise.

DCGAN	FID (↓)	IS (↑)
Adam (Radford et al., 2015)	24.38 ± 0.13	6.58
LEAD-Adam	19.27 ± 0.10	7.58± 0.11
CGD-WGAN (Schäfer & Anandkumar, 2019)	21.3	7.2
OMD (Daskalakis et al., 2018)	29.6 ± 0.19	5.74 ± 0.1
ResNet		
SNGAN	12.10 ± 0.31	8.58 ± 0.03
LEAD-Adam (ours)	**10.49** ± 0.11	**8.82** ± 0.05
ExtraAdam (Gidel et al., 2018)	16.78 ± 0.21	8.47 ± 0.1
LA-GAN (Chavdarova et al., 2020)	12.67 ± 0.57	8.55 ± 0.04
ODE-GAN (Qin et al., 2020)	11.85 ± 0.21	8.61 ± 0.06
Evaluated with 5k samples		
SN-GAN (DCGAN) (Miyato et al., 2018)	29.3	7.42 ± 0.08
SN-GAN (ResNet) (Miyato et al., 2018)	21.7 ± 0.21	8.22 ± 0.05

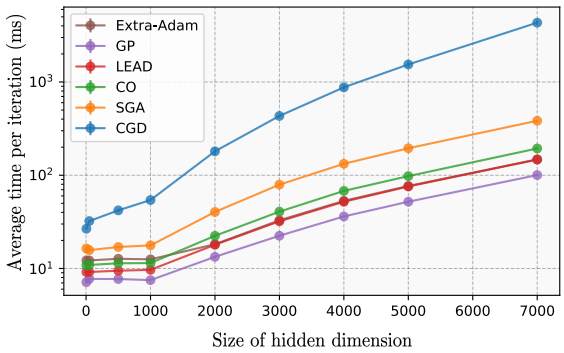

Figure 1: Average computational cost per iteration of several well-known methods for (non-saturating) GAN optimization. The numbers are reported on the 8-Gaussians generation task and averaged over 1000 iterations. Note that the y-axis is log-scale. We compare Competitive Gradient Descent (CGD) (38) (using official CGD optimizer code), Symplectic Gradient Adjustment (SGA) (4), Consensus Optimization (CO) (28), Extra-gradient with Adam (Extra-Adam) (13), WGAN with Gradient Penalty (WGAN GP) (16).

large imaginary part results in oscillatory behavior. To mitigate this issue, they propose Consensus Optimization (CO). Along similar lines, (Balduzzi et al., 2018; Gemp & Mahadevan, 2018; Letcher et al., 2019; Loizou et al., 2020) use the *Hamiltonian* of the gradient vector-field, to improve the convergence in games through disentangling the convergent parts of the dynamics from the rotations. Another line of attack taken in (Schäfer & Anandkumar, 2019) is to use second-order information as a regularizer of the dynamics and motivate the use of Competitive Gradient Descent (CGD). In (Wang et al., 2019), Follow the Ridge (FtR) is proposed. They motivate the use of a second order term for one of the players (follower) as to avoid the rotational dynamics in a sequential formulation of the zero-sum game. See appendix K for full discussion on the comparison of LEAD versus other second-order methods.

Another approach taken by (Gidel et al., 2019), demonstrate how applying negative momentum over GDA can improve convergence in min-max games, while also proving a linear rate of convergence in the case of bilinear games. More recently, (Zhang & Wang, 2021) have shown the suboptimality of negative momentum in specific settings. Furthermore, in (Lorraine & Duvenaud, 2022) authors carry-out an extensive study on the effect of momentum in games and specifically show that complex momentum is optimal in many games ranging from adversarial to non-adversarial settings.

Single-objective Optimization and Dynamical Systems: The authors of (Su et al., 2014) started a new trend in single-objective optimization by studying the continuous-time dynamics of Nesterov's accelerated method (Nesterov, 2013). Their analysis allowed for a better understanding of the much-celebrated Nesterov's method. In a similar spirit, (Wibisono et al., 2016; Wilson et al., 2016) study continuous-time accelerated methods within a Lagrangian framework, while analyzing their stability using Lyapunov analysis. These work show that a family of discrete-time methods can be derived from their corresponding continuous-time formalism using various discretization schemes. Additionally, several recent work (Muehlebach & Jordan, 2019; Bailey & Piliouras, 2019; Maddison et al., 2018; Ryu et al., 2019) cast game optimization algorithms as dynamical systems so to leverage its rich theory, to study the stability and convergence of various continuous-time methods. (Nagarajan & Kolter, 2017) also analyzes the local stability of GANs as an approximated continuous dynamical system.

7. Conclusion

In this paper, we leverage tools from physics to propose a novel second-order optimization scheme LEAD, to address the issue of rotational dynamics in min-max games. By casting min-max game optimization as a physical system, we use the principle of least action to discover an effective optimization algorithm for this setting. Subsequently, with the use of Lyapunov stability theory and spectral analysis, we prove LEAD to be convergent at a linear rate in bilinear min-max games. We supplement our theoretical analysis with experiments on GANs and toy setups, demonstrating improvements over baseline methods. Specifically for GAN training, we observe that our method outperforms other second-order methods, both in terms of sample quality and computational efficiency. Our analysis underlines the advantages of physical approaches in designing novel optimization algorithms for games as well as for traditional optimization tasks. It is important to note in this regard that our crafted physical system is *a* way to model min-max optimization physically. Alternate schemes to perform such modeling can involve other choices of counter-rotational and dissipative forces which can be explored in future work.

References

Abernethy, J., Lai, K. A., and Wibisono, A. Last-iterate convergence rates for min-max optimization. *arXiv preprint arXiv:1906.02027*, 2019.

Azizian, W., Mitliagkas, I., Lacoste-Julien, S., and Gidel, G. A tight and unified analysis of gradient-based methods for a whole spectrum of differentiable games. In *International Conference on Artificial Intelligence and Statistics*, pp. 2863–2873, 2020.

Bailey, J. P. and Piliouras, G. Multi-agent learning in network zero-sum games is a hamiltonian system. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 233–241. International Foundation for Autonomous Agents and Multiagent Systems, 2019.

Balduzzi, D., Racaniere, S., Martens, J., Foerster, J., Tuyls, K., and Graepel, T. The mechanics of n-player differentiable games. In *ICML*, volume 80, pp. 363–372. JMLR. org, 2018.

Berry, M. and Shukla, P. Curl force dynamics: symmetries, chaos and constants of motion. *New Journal of Physics*, 18(6):063018, 2016.

Bertsekas, D. P. *Nonlinear programming*. Athena scientific Belmont, 1999.

Brock, A., Donahue, J., and Simonyan, K. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

Bu, L., Babu, R., De Schutter, B., et al. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(2):156–172, 2008.

Chavdarova, T., Pagliardini, M., Jaggi, M., and Fleuret, F. Taming gans with lookahead. *arXiv preprint arXiv:2006.14567*, 2020.

Daskalakis, C., Ilyas, A., Syrgkanis, V., and Zeng, H. Training gans with optimism. In *International Conference on Learning Representations*, 2018.

Foerster, J., Chen, R. Y., Al-Shedivat, M., Whiteson, S., Abbeel, P., and Mordatch, I. Learning with opponent-learning awareness. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 122–130. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

Gemp, I. and Mahadevan, S. Global convergence to the equilibrium of gans using variational inequalities. *arXiv preprint arXiv:1808.01531*, 2018.

Gidel, G., Berard, H., Vignoud, G., Vincent, P., and Lacoste-Julien, S. A variational inequality perspective on generative adversarial networks. In *International Conference on Learning Representations*, 2018.

Gidel, G., Hemmat, R. A., Pezeshki, M., Le Priol, R., Huang, G., Lacoste-Julien, S., and Mitliagkas, I. Negative momentum for improved game dynamics. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1802–1811, 2019.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pp. 5767–5777, 2017.

Kang, M., Shim, W., Cho, M., and Park, J. Rebooting acgan: Auxiliary classifier gans with stable training. *Advances in Neural Information Processing Systems*, 34:23505–23518, 2021.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Korpelevich, G. The extragradient method for finding saddle points and other problems. *Matecon*, 12:747–756, 1976.

Landau, L. and Lifshitz, E. *Course of theoretical physics. vol. 1: Mechanics*. Oxford, 1960.

Lee, K., Chang, H., Jiang, L., Zhang, H., Tu, Z., and Liu, C. Vitgan: Training gans with vision transformers. *arXiv preprint arXiv:2107.04589*, 2021.

Letcher, A., Balduzzi, D., Racaniere, S., Martens, J., Foerster, J. N., Tuyls, K., and Graepel, T. Differentiable game mechanics. *Journal of Machine Learning Research*, 20 (84):1–40, 2019.

Loizou, N., Berard, H., Jolicoeur-Martineau, A., Vincent, P., Lacoste-Julien, S., and Mitliagkas, I. Stochastic hamiltonian gradient methods for smooth games. *ICML*, 2020.

Lorraine, Jonathan P., D. A. P. V. and Duvenaud, D. Complex momentum for optimization in games. In *International Conference on Artificial Intelligence and Statistics*, pp. 7742–7765. PMLR, 2022.

Maddison, C. J., Paulin, D., Teh, Y. W., O'Donoghue, B., and Doucet, A. Hamiltonian descent methods. *arXiv preprint arXiv:1809.05042*, 2018.

Mazumdar, E. V., Jordan, M. I., and Sastry, S. S. On finding local nash equilibria (and only local nash equilibria) in zero-sum games. *arXiv preprint arXiv:1901.00838*, 2019.

Mertikopoulos, P., Lecouat, B., Zenati, H., Foo, C.-S., Chandrasekhar, V., and Piliouras, G. Optimistic mirror descent in saddle-point problems: Going the extra (gradient) mile. *arXiv preprint arXiv:1807.02629*, 2018.

Mescheder, L., Nowozin, S., and Geiger, A. The numerics of gans. In *Advances in Neural Information Processing Systems*, pp. 1825–1835, 2017.

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

Muehlebach, M. and Jordan, M. A dynamical systems perspective on nesterov acceleration. In *International Conference on Machine Learning*, pp. 4656–4662, 2019.

Nagarajan, V. and Kolter, J. Z. Gradient descent gan optimization is locally stable. In *Advances in neural information processing systems*, pp. 5585–5595, 2017.

Nesterov, Y. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

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

Qin, C., Wu, Y., Springenberg, J. T., Brock, A., Donahue, J., Lillicrap, T. P., and Kohli, P. Training generative adversarial networks by solving ordinary differential equations. *arXiv preprint arXiv:2010.15040*, 2020.

Radford, A., Metz, L., and Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Ryu, E. K., Yuan, K., and Yin, W. Ode analysis of stochastic gradient methods with optimism and anchoring for minimax problems and gans. *arXiv preprint arXiv:1905.10899*, 2019.

Sauer, A., Schwarz, K., and Geiger, A. Stylegan-xl: Scaling stylegan to large diverse datasets. In *ACM SIGGRAPH 2022 Conference Proceedings*, pp. 1–10, 2022.

Schäfer, F. and Anandkumar, A. Competitive gradient descent. In *Advances in Neural Information Processing Systems*, pp. 7623–7633, 2019.

Sener, O. and Koltun, V. Multi-task learning as multi-objective optimization. In *Advances in Neural Information Processing Systems*, pp. 527–538, 2018.

Su, W., Boyd, S., and Candes, E. A differential equation for modeling nesterov's accelerated gradient method: Theory and insights. In *Advances in Neural Information Processing Systems*, pp. 2510–2518, 2014.

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

Wibisono, A., Wilson, A. C., and Jordan, M. I. A variational perspective on accelerated methods in optimization. *Proceedings of the National Academy of Sciences*, 113(47): E7351–E7358, 2016.

Wilson, A. C., Recht, B., and Jordan, M. I. A lyapunov analysis of momentum methods in optimization. *arXiv preprint arXiv:1611.02635*, 2016.

Zhang, G. and Wang, Y. On the suboptimality of negative momentum for minimax optimization. In *International Conference on Artificial Intelligence and Statistics*, pp. 2098–2106. PMLR, 2021.

Zhang, G., Wang, Y., Lessard, L., and Grosse, R. B. Near-optimal local convergence of alternating gradient descent-ascent for minimax optimization. In *International Conference on Artificial Intelligence and Statistics*, pp. 7659–7679. PMLR, 2022.

A. Appendix

B. Discrete-Time Analysis

In this section, we next analyze the convergence behavior of LEAD, Eq.(11) in the case of the quadratic min-max game of Eq.(12), using spectral analysis,

$$\begin{aligned}\boldsymbol{x}_{k+1} &= \boldsymbol{x}_k + \beta\Delta\boldsymbol{x}_k - \eta h\boldsymbol{x}_k - \eta\mathbb{A}\boldsymbol{y}_k - \alpha\mathbb{A}\Delta\boldsymbol{y}_k \\ \boldsymbol{y}_{k+1} &= \boldsymbol{y}_k + \beta\Delta\boldsymbol{y}_k - \eta h\boldsymbol{y}_k + \eta\mathbb{A}^T\boldsymbol{x}_k + \alpha\mathbb{A}^T\Delta\boldsymbol{x}_k,\end{aligned} \tag{17}$$

where $\Delta\boldsymbol{x}_k = \boldsymbol{x}_k - \boldsymbol{x}_{k-1}$.

For brevity, consider the joint parameters $\boldsymbol{\omega}_t := (\boldsymbol{x}_t, \boldsymbol{y}_t)$. We start by studying the update operator of simultaneous gradient descent-ascent,

$$F_\eta(\boldsymbol{\omega}_t) = \boldsymbol{\omega}_t - \eta\boldsymbol{v}(\boldsymbol{\omega}_{t-1}).$$

where, the vector-field is given by Eq. 13. Thus, the fixed point $\boldsymbol{\omega}^*$ of $F_\eta(\boldsymbol{\omega}_t)$ satisfies $F_\eta(\boldsymbol{\omega}^*) = \boldsymbol{\omega}^*$.

Furthermore, at $\boldsymbol{\omega}^*$, we have,

$$\nabla F_\eta(\boldsymbol{\omega}^*) = \mathbb{I}_n - \eta\nabla\boldsymbol{v}(\boldsymbol{\omega}^*), \tag{18}$$

with $\mathbb{I}_n$ being the $n \times n$ identity matrix. Consequently the spectrum of $\nabla F_\eta(\boldsymbol{\omega}^*)$ in the quadratic game considered, is,

$$\mathrm{Sp}(\nabla F_\eta(\boldsymbol{\omega}^*)) = \{1 - \eta h - \eta\lambda \mid \lambda \in \mathrm{Sp}(\text{off-diag}[\nabla\boldsymbol{v}(\boldsymbol{\omega}^*)])\}, \tag{19}$$

The next proposition outlines the condition under which the fixed point operator is guaranteed to converge around the fixed point.

Proposition 1 (Prop. 4.4.1 (Bertsekas, 1999)). For the spectral radius,

$$\rho_{\max} := \rho\{\nabla F_\eta(\boldsymbol{\omega}^*)\} < 1 \tag{20}$$

and for some $\boldsymbol{\omega}_0$ in a neighborhood of $\boldsymbol{\omega}^*$, the update operator F, ensures linear convergence to $\boldsymbol{\omega}^*$ at a rate,

$$\Delta_{t+1} \leq \mathcal{O}(\rho + \epsilon)\Delta_t \;\forall\, \epsilon > 0,$$

where $\Delta_{t+1} := ||\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}^*||_2^2 + ||\boldsymbol{\omega}_t - \boldsymbol{\omega}^*||_2^2$.

Next, we proceed to define the update operator of Eq.(11) as $F_{\text{LEAD}}\left(\boldsymbol{\omega}_t, \boldsymbol{\omega}_{t-1}\right) = \left(\boldsymbol{\omega}_{t+1}, \boldsymbol{\omega}_t\right)$. For the quadratic min-max game of Eq.(12), the Jacobian of F_{LEAD} takes the form,

$$\nabla F_{\text{LEAD}} = \begin{bmatrix} \mathbb{I}_{2n} + \beta\mathbb{I}_{2n} - (\eta + \alpha)\,\nabla\boldsymbol{v} & -\beta\mathbb{I}_{2n} + \alpha\nabla\boldsymbol{v} \\ \mathbb{I}_{2n} & 0 \end{bmatrix}. \tag{21}$$

In the next Theorem B.1, we find the set of eigenvalues corresponding to the update operator ∇F_{LEAD} which are then used in Theorem B.2, where we show for a selected values of η and α, LEAD attains a linear rate.

Theorem B.1. *The eigenvalues of* $\nabla F_{LEAD}(\boldsymbol{\omega}^*)$ *are,*

$$\mu_\pm = \frac{1 - (\eta + \alpha)\lambda + \beta - \eta h \pm \sqrt{\Delta}}{2} \tag{22}$$

where,

$$\Delta = \left(1 - (\eta + \alpha)\,\lambda + \beta - \eta h\right)^2 - 4\,(\beta - \alpha\lambda)$$

and $\lambda \in$ *Sp*(*off-diag*$[\nabla\boldsymbol{v}(\boldsymbol{\omega}^*)]$).

Furthermore, for $h, \eta, |\alpha|, |\beta| << 1$*, we have,*

$$\mu_+ \approx 1 - \eta h + \frac{(\eta+\alpha)^2 \lambda^2 + \eta^2 h^2 + \beta^2 - 2\eta h \beta}{4} + \lambda \left(\frac{\eta + \alpha}{2} (\eta h - \beta) - \eta \right) \tag{23}$$

and

$$\mu_- \approx \beta - \frac{(\eta+\alpha)^2 \lambda^2 + \eta^2 h^2 + \beta^2 - 2\eta h \beta}{4} + \lambda \left(\frac{\eta + \alpha}{2} (\beta - \eta h) - \alpha \right) \tag{24}$$

See proof in Appendix E.

Theorem B.1 states that the LEAD operator has two eigenvalues μ_+ and μ_- for each $\lambda \in \text{Sp}\left(\text{off-diag}[\nabla \boldsymbol{v}(\boldsymbol{\omega}^*)]\right)$. Specifically, μ_+ can be viewed as a shift of the eigenvalues of GDA in Eq.(19), while additionally being the leading eigenvalue for small values of $h, \eta, |\alpha|$ and $|\beta|$. (See Fig. 2 for a schematic description) Also, for small values of α, μ_+ is the limiting eigenvalue while $\mu_- \approx 0$.

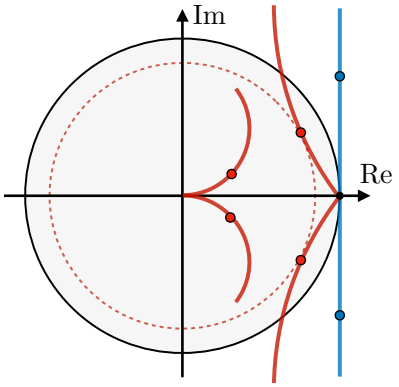

Figure 2: Diagram depicts positioning of the eigenvalues of GDA in blue (Eq. 18) and those of LEAD (Equations (23) and (24) with $\beta = h = 0$) in red. Eigenvalues inside the black unit circle imply convergence such that the closer to the origin, the faster the convergence rate (Prop. 1). Every point on solid blue and red lines corresponds to a specific choice of learning rate. No choice of learning rate results in convergence for gradient ascent descent method as the blue line is tangent to the unit circle. At the same time, for a fixed value of α, LEAD shifts the eigenvalues (μ_+) into the unit circle which leads to a convergence rate proportional to the radius of the red dashed circle. Note that LEAD also introduces an extra set of eigenvalues (μ_-) which are close to zero and do not affect convergence .

In the following Proposition, we next show that locally, a choice of positive α decreases the spectral radius of $\nabla F_\eta\left(\boldsymbol{\omega}^*\right)$ defined as,

$$\rho := \max\{|\mu_+|^2, |\mu_-|^2\} \; \forall \, \lambda.$$

Proposition 2. For any $\lambda \in \text{Sp}(\text{off-diag}[\nabla \boldsymbol{v}(\omega^*)])$,

$$\nabla_\alpha \rho\left(\lambda\right)\big|_{\alpha=0} < 0 \Leftrightarrow \eta \in \left(0, \frac{2}{\text{Im}(\lambda_{\max})}\right), \tag{25}$$

where $\text{Im}(\lambda_{\max})$ is the imaginary component of the largest eigenvalue $\lambda_{\max}$.

See proof in Appendix F.

Having established that a small positive value of α improves the rate of convergence, in the next theorem, we prove that for a specific choice of positive α and η in the quadratic game Eq.(12), a linear rate of convergence to its Nash equilibrium is attained.

Theorem B.2. *Setting $\eta = \alpha = \frac{1}{2(\sigma_{max}(\mathbb{A})+h)}$, then we have $\forall\ \epsilon > 0$,*

$$\Delta_{t+1} \in \mathcal{O}\left(\left(1 - \frac{\sigma_{min}^2}{4(\sigma_{max}+h)^2} - \frac{(1+\beta/2)h}{\sigma_{max}+h} + \frac{3h^2}{8(\sigma_{max}+h)^2}\right)^t \Delta_0\right) \tag{26}$$

where $\sigma_{max}(\sigma_{min})$ is the largest (smallest) eigen value of $\mathbb{A}$ and

$$\Delta_{t+1} := ||\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}^*||_2^2 + ||\boldsymbol{\omega}_t - \boldsymbol{\omega}^*||_2^2.$$

Theorem B.2 ensures a linear convergence of LEAD in the quadratic min-max game. (Proof in Appendix G).

C. Comparison of Convergence Rate for Quadratic Min-Max Game

In this section, we perform a Big-O comparison of convergence rates of LEAD (Eq. 53), with several other existing methods. Below in Table C we summarize the convergence rates for the quadratic min-max game of Eq. 12. For each method that converges at the rate of $\mathcal{O}((1-r)^t)$, we report the quantity r (larger r corresponds to faster convergence). We observe that for the quadratic min-max game, given the analysis in (Azizian et al., 2020), for $h < \sigma_{\max}(\mathbb{A})$ and $\beta > \frac{3h^2}{8(\sigma_{max}+h)}$, $r_{\text{LEAD}} \gtrsim r_{\text{EG}}$ and $r_{\text{LEAD}} \gtrsim r_{\text{OG}}$. Furthermore, for the bilinear case, where $h = 0$, LEAD has a faster convergence rate than EG and OG.

Method	r
Alternating-GDA	$h/2L$
Extra-Gradient (EG)	$\frac{1}{4}(h/L + \sigma_{\min}^2(\mathbb{A})/16L^2)$
Optimistic Gradient(OG)	$\frac{1}{4}(h/L + \sigma_{\min}^2(\mathbb{A})/32L^2)$
Consensus Optimization (CO)	$h^2/2L_H^2 + \sigma_{\min}^2(\mathbb{A})/2L_H^2$
LEAD (Th. B.2)	$(1+\beta/2)h/(\sigma_{max}+h) + \sigma_{min}^2/4(\sigma_{max}+h)^2 - 3h^2/8(\sigma_{max}+h)^2$

Table 2: Big-O comparison of convergence rates of LEAD against EG (Korpelevich, 1976), OG (Mertikopoulos et al., 2018) and CO (Mescheder et al., 2017) for the **quadratic min-max game** of Eq. 12. We report the EG, OG and CO rates from the tight analysis in (Azizian et al., 2020) and Alt-GDA from (Zhang et al., 2022). For each method that converges at the rate of $\mathcal{O}((1-r)^t)$, we report the quantity r (larger r corresponds to faster convergence). Note that $L := \sqrt{2}\max\{h, \sigma_{\max}(\mathbb{A})\}$, is the Lipschitz constant of the vector field and and L_H^2 is the Lipschitz-smoothness of $\frac{1}{2}\|v\|^2$.

D. Continuous-time Convergence Analysis: Quadratic Min-Max Game

Proof. For the class of quadratic min-max games,

$$f(\boldsymbol{X}, \boldsymbol{Y}) = \frac{h}{2}|\boldsymbol{X}|^2 - \frac{h}{2}|\boldsymbol{Y}|^2 + \boldsymbol{X}^T\mathbb{A}\boldsymbol{Y} \tag{27}$$

where $\boldsymbol{X} \equiv \left(X^1, \cdots, X^n\right), \boldsymbol{Y} \equiv \left(Y^1, \cdots, Y^n\right) \in \mathbb{R}^n$ and $\mathbb{A}_{n\times n}$ is a constant positive-definite matrix, the continuous-time EOMs of Eq.(10) become:

$$\begin{aligned} \ddot{\boldsymbol{X}} &= -\mu\dot{\boldsymbol{X}} - h\boldsymbol{X} - \mathbb{A}\boldsymbol{Y} - q\mathbb{A}\dot{\boldsymbol{Y}} \\ \ddot{\boldsymbol{Y}} &= -\mu\dot{\boldsymbol{Y}} - h\boldsymbol{Y} + \mathbb{A}^T\boldsymbol{X} + q\mathbb{A}^T\dot{\boldsymbol{X}} \end{aligned} \tag{28}$$

We next define our continuous-time Lyapunov function in this case to be,

$$\begin{aligned} \mathcal{E}_t = &\frac{1}{2}\left(\dot{\boldsymbol{X}} + \mu\boldsymbol{X} + \mu\mathbb{A}\boldsymbol{Y}\right)^T\left(\dot{\boldsymbol{X}} + \mu\boldsymbol{X} + \mu\mathbb{A}\boldsymbol{Y}\right) \\ &+ \frac{1}{2}\left(\dot{\boldsymbol{Y}} + \mu\boldsymbol{Y} - \mu\mathbb{A}^T\boldsymbol{X}\right)^T\left(\dot{\boldsymbol{X}} + \mu\boldsymbol{Y} - \mu\mathbb{A}^T\boldsymbol{X}\right) \\ &+ \frac{1}{2}\left(\dot{\boldsymbol{X}}^T\dot{\boldsymbol{X}} + \dot{\boldsymbol{Y}}^T\dot{\boldsymbol{Y}}\right) + \boldsymbol{X}^T(h + \mathbb{A}\mathbb{A}^T)\boldsymbol{X} + \boldsymbol{Y}^T(h + \mathbb{A}^T\mathbb{A})\boldsymbol{Y} \\ \geq &\ 0\ \forall\ t \end{aligned} \tag{29}$$

The time-derivative of $\mathcal{E}_t$ is then given by,

$$\begin{aligned}
\dot{\mathcal{E}}_t &= \left(\dot{\boldsymbol{X}} + \mu\boldsymbol{X} + \mu\mathbb{A}\boldsymbol{Y}\right)^T \left(\ddot{\boldsymbol{X}} + \mu\dot{\boldsymbol{X}} + \mu\mathbb{A}\dot{\boldsymbol{Y}}\right) + \left(\dot{\boldsymbol{Y}} + \mu\boldsymbol{Y} - \mu\mathbb{A}^T\boldsymbol{X}\right)^T \left(\ddot{\boldsymbol{Y}} + \mu\dot{\boldsymbol{Y}} - \mu\mathbb{A}^T\dot{\boldsymbol{X}}\right) \\
&\quad + \left(\dot{\boldsymbol{X}}^T\ddot{\boldsymbol{X}} + \dot{\boldsymbol{Y}}^T\ddot{\boldsymbol{Y}}\right) + 2\left(\boldsymbol{X}^T(h + \mathbb{A}\mathbb{A}^T)\dot{\boldsymbol{X}} + \boldsymbol{Y}^T(h + \mathbb{A}^T\mathbb{A})\dot{\boldsymbol{Y}}\right) \\
&= \left(\dot{\boldsymbol{X}}^T + \mu\boldsymbol{X}^T + \mu\boldsymbol{Y}^T\mathbb{A}^T\right)\left((-q + \mu)\,\mathbb{A}\dot{\boldsymbol{Y}} - \mathbb{A}\boldsymbol{Y}\right) + \dot{\boldsymbol{X}}^T\left(-q\mathbb{A}\dot{\boldsymbol{Y}} - \mu\dot{\boldsymbol{X}} - \mathbb{A}\boldsymbol{Y}\right) \\
&\quad + \left(\dot{\boldsymbol{Y}}^T + \mu\boldsymbol{Y}^T - \mu\boldsymbol{X}^T\mathbb{A}\right)\left((q - \mu)\,\mathbb{A}^T\dot{\boldsymbol{X}} + \mathbb{A}^T\boldsymbol{X}\right) + \dot{\boldsymbol{Y}}^T\left(q\mathbb{A}^T\dot{\boldsymbol{X}} - \mu\dot{\boldsymbol{Y}} + \mathbb{A}^T\boldsymbol{X}\right) \\
&\quad + 2\left(\boldsymbol{X}^T(h + \mathbb{A}\mathbb{A}^T)\dot{\boldsymbol{X}} + \boldsymbol{Y}^T(h + \mathbb{A}^T\mathbb{A})\dot{\boldsymbol{Y}}\right) \\
&= (\mu\,(q - \mu) - 2)\left(\boldsymbol{Y}^T\mathbb{A}^T\dot{\boldsymbol{X}} - \boldsymbol{X}^T\mathbb{A}\dot{\boldsymbol{Y}}\right) - (\mu\,(q - \mu) - 2)\left(\boldsymbol{X}^T\mathbb{A}\mathbb{A}^T\dot{\boldsymbol{X}} + \boldsymbol{Y}^T\mathbb{A}^T\mathbb{A}\dot{\boldsymbol{Y}}\right) \\
&\quad - \mu\left(\boldsymbol{X}^T(h + \mathbb{A}\mathbb{A}^T)\boldsymbol{X} + \boldsymbol{Y}^T(h + \mathbb{A}^T\mathbb{A})\boldsymbol{Y}\right) - \mu\left(\dot{\boldsymbol{X}}^T\dot{\boldsymbol{X}} + \dot{\boldsymbol{Y}}^T\dot{\boldsymbol{Y}}\right)
\end{aligned} \tag{30}$$

where we have used the fact that $\boldsymbol{X}^T\mathbb{A}\boldsymbol{Y}$ being a scalar thus implying $\boldsymbol{X}^T\mathbb{A}\boldsymbol{Y} = \boldsymbol{Y}^T\mathbb{A}^T\boldsymbol{X}$. If we now set $q = (2/\mu) + \mu$ in the above, then that further leads to,

$$\begin{aligned}
\dot{\mathcal{E}}_t &= -\mu\left(\boldsymbol{X}^T(h + \mathbb{A}\mathbb{A}^T)\boldsymbol{X} + \boldsymbol{Y}^T(h + \mathbb{A}^T\mathbb{A})\boldsymbol{Y}\right) - \mu\left(\dot{\boldsymbol{X}}^T\dot{\boldsymbol{X}} + \dot{\boldsymbol{Y}}^T\dot{\boldsymbol{Y}}\right) \\
&= -\mu\left(h||\boldsymbol{X}||^2 + h||\boldsymbol{Y}||^2 + \left|\left|\mathbb{A}^T\boldsymbol{X}\right|\right|^2 + ||\mathbb{A}\boldsymbol{Y}||^2\right) - \mu\left(\left|\left|\dot{\boldsymbol{X}}\right|\right|^2 + \left|\left|\dot{\boldsymbol{Y}}\right|\right|^2\right) \leq 0\;\forall\, t
\end{aligned} \tag{31}$$

exhibiting that the Lyapunov function, Eq.(15) is *asymptotically stable* at all times t.

Next, consider the following expression,

$$\begin{aligned}
&-\rho\mathcal{E}_t - \frac{\rho\mu}{2}\left|\left|\boldsymbol{X} - \dot{\boldsymbol{X}}\right|\right|^2 - \frac{\rho\mu}{2}\left|\left|\boldsymbol{Y} - \dot{\boldsymbol{Y}}\right|\right|^2 - \frac{\rho\mu}{2}\left|\left|\dot{\boldsymbol{X}} - \mathbb{A}\boldsymbol{Y}\right|\right|^2 - \frac{\rho\mu}{2}\left|\left|\mathbb{A}^T\boldsymbol{X} + \dot{\boldsymbol{Y}}\right|\right|^2 \\
&\quad = -\rho\mathcal{E}_t - \frac{\rho\mu}{2}\left(||\boldsymbol{X}||^2 + ||\boldsymbol{Y}||^2\right) + \rho\mu\left(\boldsymbol{X}^T\dot{\boldsymbol{X}} + \boldsymbol{Y}^T\dot{\boldsymbol{Y}}\right) - \rho\mu\left(\left|\left|\dot{\boldsymbol{X}}\right|\right|^2 + ||\boldsymbol{Y}||^2\right) \\
&\qquad - \rho\mu\left(\boldsymbol{X}^T\mathbb{A}\dot{\boldsymbol{Y}} - \dot{\boldsymbol{X}}^T\mathbb{A}\boldsymbol{Y}\right) - \frac{\rho\mu}{2}\left(\left|\left|\mathbb{A}^T\boldsymbol{X}\right|\right|^2 + ||\mathbb{A}\boldsymbol{Y}||^2\right) \\
&\quad = -\rho\,(1 + \mu)\left(\left|\left|\dot{\boldsymbol{X}}\right|\right|^2 + \left|\left|\dot{\boldsymbol{Y}}\right|\right|^2\right) - \frac{\rho}{2}\left(\mu^2 + \mu + 2h\right)\left(||\boldsymbol{X}||^2 + ||\boldsymbol{Y}||^2\right) \\
&\qquad - \frac{\rho}{2}\left(\mu^2 + \mu + 2\right)\left(\left|\left|\mathbb{A}^T\boldsymbol{X}\right|\right|^2 + ||\mathbb{A}\boldsymbol{Y}||^2\right) \\
&\quad \leq -\rho\mathcal{E}_t
\end{aligned} \tag{32}$$

where ρ is some positive definite constant. This implies that the above expression is negative semi-definite by construction given $\mu \geq 0$. Now, for a general square matrix $\mathbb{A}$, we can perform a singular value decomposition (SVD) as $\mathbb{A} = \mathbb{V}^{\mathrm{T}}\mathbb{S}\mathbb{U}$. Here, $\mathbb{U}$ and $\mathbb{V}$ are the right and left unitaries of $\mathbb{A}$, while $\mathbb{S}$ is a diagonal matrix of singular values (σ_i) of $\mathbb{A}$. Using this

decomposition in Eq.(32), then allows us to write,

$$\begin{aligned}
&-\rho\left(1+\mu\right)\left(\left|\left|\dot{\boldsymbol{X}}\right|\right|^2+\left|\left|\dot{\boldsymbol{Y}}\right|\right|^2\right)-\frac{\rho}{2}\left(\mu^2+\mu+2h\right)\left(||\boldsymbol{X}||^2+||\boldsymbol{Y}||^2\right)\\
&\qquad-\frac{\rho}{2}\left(\mu^2+\mu+2\right)\left(\left|\left|\mathbb{A}^T\boldsymbol{X}\right|\right|^2+||\mathbb{A}\boldsymbol{y}||^2\right)\\
&=-\rho\left(1+\mu\right)\left(\left|\left|\mathbb{V}\dot{\boldsymbol{X}}\right|\right|^2+\left|\left|\mathbb{U}\dot{\boldsymbol{Y}}\right|\right|^2\right)-\frac{\rho}{2}\left(\mu^2+\mu+2h\right)\left(||\mathbb{V}\boldsymbol{X}||^2+||\mathbb{U}\boldsymbol{Y}||^2\right)\\
&\qquad-\frac{\rho}{2}\left(\mu^2+\mu+2\right)\left(||\mathbb{S}\mathbb{V}\boldsymbol{X}||^2+||\mathbb{S}\mathbb{U}\boldsymbol{Y}||^2\right)\\
&=-\rho\left(1+\mu\right)\left(\left|\left|\dot{\boldsymbol{\mathcal{X}}}\right|\right|^2+\left|\left|\dot{\boldsymbol{\mathcal{Y}}}\right|\right|^2\right)-\frac{\rho}{2}\left(\mu^2+\mu+2h\right)\left(||\boldsymbol{\mathcal{X}}||^2+||\boldsymbol{\mathcal{Y}}||^2\right)\\
&\qquad-\frac{\rho}{2}\left(\mu^2+\mu+2\right)\left(||\mathbb{S}\boldsymbol{\mathcal{X}}||^2+||\mathbb{S}\boldsymbol{\mathcal{Y}}||^2\right)\\
&=-\sum_{j=1}^{n}\rho\left(1+\mu\right)\left(\left|\left|\dot{\mathcal{X}}^j\right|\right|^2+\left|\left|\dot{\mathcal{Y}}^j\right|\right|^2\right)\\
&\qquad-\sum_{j=1}^{n}\frac{\rho}{2}\left(\left(1+\sigma_j^2+2h\right)\left(\mu^2+\mu\right)+2\sigma_j^2\right)\left(\left|\left|\mathcal{X}^j\right|\right|^2+\left|\left|\mathcal{Y}^j\right|\right|^2\right)
\end{aligned}\tag{33}$$

where we have made use of the relations $\mathbb{U}^{\mathrm{T}}\mathbb{U}=\mathbb{U}\mathbb{U}^{\mathrm{T}}=\mathbb{I}_n=\mathbb{V}^{\mathrm{T}}\mathbb{V}=\mathbb{V}\mathbb{V}^{\mathrm{T}}$, and additionally performed a basis change, as $\boldsymbol{\mathcal{X}}=\mathbb{V}\boldsymbol{X}$ and $\boldsymbol{\mathcal{Y}}=\mathbb{U}\boldsymbol{Y}$. Now, we know from Eq.(31) that,

$$\begin{aligned}
\dot{\mathcal{E}}_t&=-\mu\left(h||\boldsymbol{X}||^2+h||\boldsymbol{Y}||^2+\left|\left|\mathbb{A}^T\boldsymbol{X}\right|\right|^2+||\mathbb{A}\boldsymbol{Y}||^2\right)-\mu\left(\left|\left|\dot{\boldsymbol{X}}\right|\right|^2+\left|\left|\dot{\boldsymbol{Y}}\right|\right|^2\right)\\
&=-\mu\left(h||\boldsymbol{X}||^2+h||\boldsymbol{Y}||^2+\left|\left|\mathbb{U}^{\mathrm{T}}\mathbb{S}\mathbb{V}\boldsymbol{X}\right|\right|^2+\left|\left|\mathbb{V}^{\mathrm{T}}\mathbb{S}\mathbb{U}\boldsymbol{Y}\right|\right|^2\right)-\mu\left(\left|\left|\mathbb{V}\dot{\boldsymbol{X}}\right|\right|^2+\left|\left|\mathbb{U}\dot{\boldsymbol{Y}}\right|\right|^2\right)\\
&=-\mu\left(h||\boldsymbol{\mathcal{X}}||^2+h||\boldsymbol{\mathcal{Y}}||^2+||\mathbb{S}\boldsymbol{\mathcal{X}}||^2+||\mathbb{S}\boldsymbol{\mathcal{Y}}||^2\right)-\mu\left(\left|\left|\dot{\boldsymbol{\mathcal{X}}}\right|\right|^2+\left|\left|\dot{\boldsymbol{\mathcal{Y}}}\right|\right|^2\right)\\
&=-\sum_{j=1}^{n}\mu\left(\sigma_j^2+h\right)\left(\left|\left|\mathcal{X}^j\right|\right|^2+\left|\left|\mathcal{Y}^j\right|\right|^2\right)-\sum_{j=1}^{n}\mu\left(\left|\left|\dot{\mathcal{X}}^j\right|\right|^2+\left|\left|\dot{\mathcal{Y}}^j\right|\right|^2\right)
\end{aligned}\tag{34}$$

Comparing the above expression with Eq.(33), we note that a choice of ρ as,

$$\rho\leq\min\left\{\frac{\mu}{1+\mu},\ \frac{2\mu(\sigma_{\min}^2+h)}{\left(1+\sigma_{\min}^2+2h\right)\left(\mu^2+\mu\right)+2\sigma_{\min}^2}\right\}\ \forall\ j\in[1,n]\tag{35}$$

implies,

$$\begin{aligned}
&\dot{\mathcal{E}}_t\leq-\rho\mathcal{E}\\
&\Rightarrow\mathcal{E}_t\leq\mathcal{E}_0\exp\left(-\rho t\right)\\
\Rightarrow\ &X^T\left(h+\mathbb{A}\mathbb{A}^T\right)X+Y^T\left(h+\mathbb{A}^T\mathbb{A}\right)Y\leq\mathcal{E}_0\exp\left(-\rho t\right)\\
&\Rightarrow\mathcal{X}^T(h+\mathbb{S}^2)\mathcal{X}+\mathcal{Y}^T(h+\mathbb{S}^2)\mathcal{Y}\leq\mathcal{E}_0\exp\left(-\rho t\right)\\
&\Rightarrow\sum_{j=1}^{n}(h+\sigma_j^2)\left(||\mathcal{X}^j||^2+||\mathcal{Y}^j||^2\right)\leq\mathcal{E}_0\exp\left(-\rho t\right)\\
&\Rightarrow\sum_{j=1}^{n}(h+\sigma_j^2)\left(||X^j||^2+||Y^j||^2\right)\leq\mathcal{E}_0\exp\left(-\rho t\right)\\
&\therefore\ ||\boldsymbol{X}||^2+||\boldsymbol{Y}||^2\leq\frac{\mathcal{E}_0}{h+\sigma_{\min}^2}\exp\left(-\rho t\right)\forall\ j
\end{aligned}\tag{36}$$

□

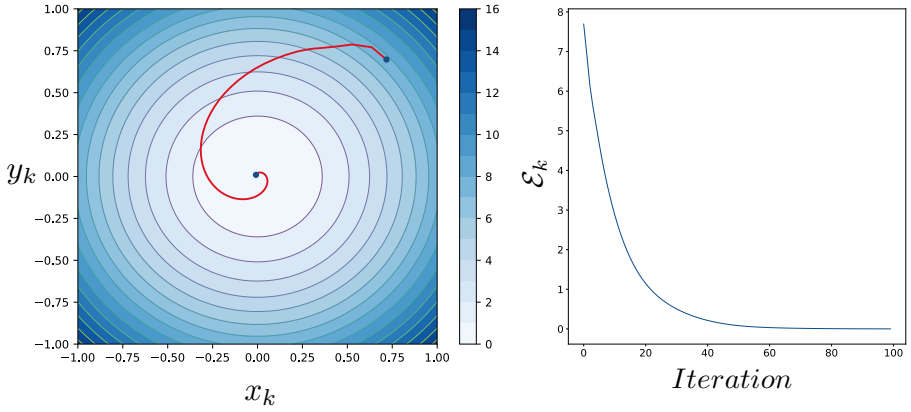

Figure 3: **Left:** Contours of the Lyapunov function $\mathcal{E}_k$, Eq. (29) (black), and convergence trajectory of LEAD (red) in the quadratic min-max game (Eq.(28)) to the Nash equilibrium $(0, 0)$. **Right**: The evolution of the discrete-time Lyapunov function of Eq. (29) over iteration, confirming $\mathcal{E}_k - \mathcal{E}_{k-1} \leq 0 \; \forall \; k \in \mathbf{N}$.

E. Proof of Theorem B.1

Theorem. The eigenvalues of $\nabla F_{\text{LEAD}}(\boldsymbol{\omega}^*)$ about the Nash equilibrium $\boldsymbol{\omega}^* = (x^*, y^*)$ of the quadratic min-max game are,

$$\mu_{\pm}(\alpha, \beta, \eta) = \frac{1 - (\eta + \alpha)\lambda + \beta - \eta h \pm \sqrt{\Delta}}{2} \tag{37}$$

where, $\Delta = (1 - (\eta + \alpha)\,\lambda + \beta - \eta h)^2 - 4\,(\beta - \alpha\lambda)$ and $\lambda \in \text{Sp}(\text{off-diag}[\nabla \boldsymbol{v}(\boldsymbol{\omega}^*)])$. Furthermore, for $h, \eta, |\alpha|, |\beta| << 1$, we have,

$$\begin{aligned}
\mu_{+}^{(i)}(\alpha, \beta, \eta) &\approx 1 - \eta h + \frac{(\eta + \alpha)^2 \lambda_i^2 + \eta^2 h^2 + \beta^2 - 2\eta h \beta}{4} \\
&\quad + \lambda_i \left(\frac{\eta + \alpha}{2} (\eta h - \beta) - \eta \right) \\
\mu_{-}^{(i)}(\alpha, \beta, \eta) &\approx \beta - \frac{(\eta + \alpha)^2 \lambda_i^2 + \eta^2 h^2 + \beta^2 - 2\eta h \beta}{4} \\
&\quad + \lambda_i \left(\frac{\eta + \alpha}{2} (\beta - \eta h) - \alpha \right)
\end{aligned} \tag{38}$$

Proof. For the quadratic game 27, the Jacobian of the vector field $\boldsymbol{v}$ is given by,

$$\nabla \boldsymbol{v} \equiv \nabla \begin{bmatrix} \nabla_x f(\boldsymbol{x}_t, \boldsymbol{y}_t) \\ -\nabla_y f(\boldsymbol{x}_t, \boldsymbol{y}_t) \end{bmatrix} = \begin{bmatrix} h\mathbb{I}_{2n} & \mathbb{A} \\ -\mathbb{A}^\top & h\mathbb{I}_{2n} \end{bmatrix} \in \mathbb{R}^{2n} \times \mathbb{R}^{2n}. \tag{39}$$

Let us next define a matrix $\mathbb{D}_q$ as,

$$\mathbb{D}_q = \begin{bmatrix} \nabla_{xy}^2 f(\boldsymbol{x}, \boldsymbol{y}) & 0 \\ 0 & -\nabla_{xy}^2 f(\boldsymbol{x}, \boldsymbol{y}) \end{bmatrix} = \begin{bmatrix} \mathbb{A} & 0 \\ 0 & -\mathbb{A}^\top \end{bmatrix} \in \mathbb{R}^{2n} \times \mathbb{R}^{2n} \tag{40}$$

Consequently, the update rule for LEAD can be written as:

$$\begin{aligned}
\begin{bmatrix} \boldsymbol{x}_{t+1} \\ \boldsymbol{y}_{t+1} \end{bmatrix} &= \begin{bmatrix} \boldsymbol{x}_t \\ \boldsymbol{y}_t \end{bmatrix} + \beta \begin{bmatrix} \boldsymbol{x}_t - \boldsymbol{x}_{t-1} \\ \boldsymbol{y}_t - \boldsymbol{y}_{t-1} \end{bmatrix} - \eta \begin{bmatrix} \nabla_x f(\boldsymbol{x}_t, \boldsymbol{y}_t) \\ -\nabla_y f(\boldsymbol{x}_t, \boldsymbol{y}_t) \end{bmatrix} - \alpha \begin{bmatrix} \nabla_{xy}^2 f(\boldsymbol{x}_t, \boldsymbol{y}_t) \Delta \boldsymbol{y}_t \\ -\nabla_{xy}^2 f(\boldsymbol{x}_t, \boldsymbol{y}_t) \Delta \boldsymbol{x}_t \end{bmatrix} \\
&= \begin{bmatrix} \boldsymbol{x}_t \\ \boldsymbol{y}_t \end{bmatrix} + \beta \begin{bmatrix} \boldsymbol{x}_t - \boldsymbol{x}_{t-1} \\ \boldsymbol{y}_t - \boldsymbol{y}_{t-1} \end{bmatrix} - \eta \boldsymbol{v} - \alpha \mathbb{D}_{\text{q}} \begin{bmatrix} \Delta \boldsymbol{y}_t \\ \Delta \boldsymbol{x}_t \end{bmatrix}
\end{aligned} \tag{41}$$

where $\Delta \boldsymbol{y}_t = \boldsymbol{y}_t - \boldsymbol{y}_{t-1}$ and $\Delta \boldsymbol{x}_t = \boldsymbol{x}_t - \boldsymbol{x}_{t-1}$.

Next, by making use of the permutation matrix $\mathbb{P}$,

$$\mathbb{P} := \begin{bmatrix} 0 & \mathbb{I}_n \\ \mathbb{I}_n & 0 \end{bmatrix} \in \mathbb{R}^{2n} \times \mathbb{R}^{2n}$$

we can re-express Eq. 41 as,

$$\begin{aligned} \begin{bmatrix} \boldsymbol{\omega}_{t+1} \\ \boldsymbol{\omega}_t \end{bmatrix} &= \begin{bmatrix} \mathbb{I}_{2n} & 0 \\ \mathbb{I}_{2n} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\omega}_t \\ \boldsymbol{\omega}_{t-1} \end{bmatrix} + \beta \begin{bmatrix} \mathbb{I}_{2n} & -\mathbb{I}_{2n} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\omega}_t \\ \boldsymbol{\omega}_{t-1} \end{bmatrix} - \eta \begin{bmatrix} \boldsymbol{v} \\ 0 \end{bmatrix} - \alpha \begin{bmatrix} \mathbb{D}_{\mathrm{q}} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbb{P} & -\mathbb{P} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\omega}_t \\ \boldsymbol{\omega}_{t-1} \end{bmatrix} \\ &= \begin{bmatrix} \mathbb{I}_{2n} & 0 \\ \mathbb{I}_{2n} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\omega}_t \\ \boldsymbol{\omega}_{t-1} \end{bmatrix} + \beta \begin{bmatrix} \mathbb{I}_{2n} & -\mathbb{I}_{2n} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\omega}_t \\ \boldsymbol{\omega}_{t-1} \end{bmatrix} - \eta \begin{bmatrix} \boldsymbol{v} \\ 0 \end{bmatrix} - \alpha \begin{bmatrix} \mathbb{D}_{\mathrm{q}}\mathbb{P} & -\mathbb{D}_{\mathrm{q}}\mathbb{P} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\omega}_t \\ \boldsymbol{\omega}_{t-1} \end{bmatrix} \end{aligned} \tag{42}$$

where $\boldsymbol{\omega}_t \equiv (\boldsymbol{x}_t, \boldsymbol{y}_t)$. Hence, the Jacobian of F_{LEAD} is then given by,

$$\begin{aligned} \nabla F_{\text{LEAD}} &= \begin{bmatrix} \mathbb{I}_{2n} & 0 \\ \mathbb{I}_{2n} & 0 \end{bmatrix} + \beta \begin{bmatrix} \mathbb{I}_{2n} & -\mathbb{I}_{2n} \\ 0 & 0 \end{bmatrix} - \eta \begin{bmatrix} \nabla \boldsymbol{v} & 0 \\ 0 & 0 \end{bmatrix} - \alpha \begin{bmatrix} \mathbb{D}_{\mathrm{q}}\mathbb{P} & -\mathbb{D}_{\mathrm{q}}\mathbb{P} \\ 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} (1+\beta)\,\mathbb{I}_{2n} - \eta\nabla\boldsymbol{v} - \alpha\mathbb{D}_{\mathrm{q}}\mathbb{P} & -\beta\mathbb{I}_{2n} + \alpha\mathbb{D}_{\mathrm{q}}\mathbb{P} \\ \mathbb{I}_{2n} & 0 \end{bmatrix} \end{aligned} \tag{43}$$

It is to be noted that, for games of the form of Eq. 27, we specifically have,

$$\nabla \boldsymbol{v} = \mathbb{D}_{\mathrm{q}}\mathbb{P} + h\mathbb{I}_{2n}$$

and,

$$\text{off-diag}[\nabla \boldsymbol{v}] = \mathbb{D}_{\mathrm{q}}\mathbb{P}$$

Therefore, Eq. 43 becomes,

$$\nabla F_{\text{LEAD}} = \begin{bmatrix} (1+\beta-\eta h)\,\mathbb{I}_{2n} - (\eta+\alpha)\,\mathbb{D}_{\mathrm{q}}\mathbb{P} & -\beta\mathbb{I}_{2n} + \alpha\mathbb{D}_{\mathrm{q}}\mathbb{P} \\ \mathbb{I}_{2n} & 0 \end{bmatrix} \tag{44}$$

We next proceed to study the eigenvalues of this matrix which will determine the convergence properties of LEAD around the Nash equilibrium. Using Lemma 1 of (Gidel et al., 2019), we can then write the characteristic polynomial of ∇F_{LEAD} as,

$$\begin{aligned} &\det\left(X\mathbb{I}_{4n} - \nabla F_{\text{LEAD}}\right) = 0 \\ \Rightarrow &\det\left(\begin{bmatrix} (X-1)\,\mathbb{I}_{2n} - (\beta-\eta h)\,\mathbb{I}_{2n} + (\eta+\alpha)\,\mathbb{D}_{\mathrm{q}}\mathbb{P} & \beta\mathbb{I}_{2n} - \alpha\mathbb{D}_{\mathrm{q}}\mathbb{P} \\ -\mathbb{I}_{2n} & X\mathbb{I}_{2n} \end{bmatrix}\right) = 0 \\ \Rightarrow &\det\left(\left[(X-1)(X-\beta)\,\mathbb{I}_{2n} + X\eta h\mathbb{I}_{2n} + (X\eta + X\alpha - \alpha)\,\mathbb{D}_{\mathrm{q}}\mathbb{P}\right]\right) = 0 \\ \Rightarrow &\det\left(\left[((X-1)(X-\beta) + X\eta h)\,\mathbf{U}\mathbf{U}^{-1} + (X\eta + X\alpha - \alpha)\,\mathbf{U}\Lambda\mathbf{U}^{-1}\right]\right) = 0 \\ \Rightarrow &\det\left(\left[((X-1)(X-\beta) + X\eta h)\,\mathbb{I}_{2n} + (X\eta + X\alpha - \alpha)\,\Lambda\right]\right) = 0 \\ \Rightarrow &\prod_{i=1}^{2n}\left[(X-1)(X-\beta) + X\eta h + (X\eta + \alpha(X-1))\,\lambda_i\right] = 0 \end{aligned} \tag{45}$$

Where, in the above, we have performed an eigenvalue decomposition of $\mathbb{D}_{\mathrm{q}}\mathbb{P} = \mathbf{U}\Lambda\mathbf{U}^{-1}$. Therefore,

$$\begin{aligned} &X^2 - X\left(1 - (\eta+\alpha)\lambda_i + \beta - \eta h\right) + \beta - \alpha\lambda = 0,\ \lambda_i \in \text{Sp}(\mathbb{D}_{\mathrm{q}}\mathbb{P}) \\ &\Rightarrow X^{(i)} \equiv \mu_{\pm}^{(i)} = \frac{1 - (\eta+\alpha)\lambda_i + \beta - \eta h \pm \sqrt{\Delta}}{2} \end{aligned} \tag{46}$$

with,

$$\Delta = \left(1 - (\eta+\alpha)\,\lambda_i + \beta - \eta h\right)^2 - 4\left(\beta - \alpha\lambda_i\right) \tag{47}$$

Furthermore for $h, \eta, |\beta|, |\alpha| << 1$, we can approximate the above roots to be,

$$\begin{aligned} \mu_{+}^{(i)}(\alpha,\beta,\eta) &\approx 1 - \eta h + \frac{(\eta+\alpha)^2\lambda_i^2 + \eta^2h^2 + \beta^2 - 2\eta h\beta}{4} + \lambda_i\left(\frac{\eta+\alpha}{2}(\eta h - \beta) - \eta\right) \\ \mu_{-}^{(i)}(\alpha,\beta,\eta) &\approx \beta - \frac{(\eta+\alpha)^2\lambda_i^2 + \eta^2h^2 + \beta^2 - 2\eta h\beta}{4} + \lambda_i\left(\frac{\eta+\alpha}{2}(\beta - \eta h) - \alpha\right) \end{aligned} \tag{48}$$

□

F. Proof of Proposition 2

Proposition. For any $\lambda \in \text{Sp}(\text{off-diag}[\nabla \boldsymbol{v}(\omega^*)])$,

$$\nabla_\alpha \rho(\lambda)\big|_{\alpha=0} < 0 \Leftrightarrow \eta \in \left(0, \frac{2}{\text{Im}(\lambda_{\text{max}})}\right), \tag{49}$$

where $\text{Im}(\lambda_{\text{max}})$ is the imaginary component of the largest eigenvalue λ_{max}.

We observe from Proposition 2 above that for $h, \eta, |\alpha|, |\beta| << 1$,

$$\begin{aligned} \rho(\alpha, \eta, \beta) &:= \max\{|\mu_+^{(i)}|^2, |\mu_-^{(i)}|^2\} \; \forall \; i \\ &= \max\{\left|\mu_+^{(i)}\right|^2\} \; \forall \; i \end{aligned} \tag{50}$$

$$\begin{aligned} \therefore \nabla_\alpha \rho\big|_{\alpha=0} &\approx \max\left\{\frac{\eta^2 |\lambda_i|^2 - \eta^2 h^2 - \beta^2}{4} \eta |\lambda_i|^2 + \frac{\eta h \beta - (\eta h - \beta)^2}{2} \eta |\lambda_i|^2 \right. \\ &\quad \left. - (1+\beta)\eta |\lambda_i|^2 \right\} \forall \; i \\ &\approx \max\left\{\frac{\eta^3}{4} |\lambda_i|^4 - \left(1 + \beta + \frac{3\beta^2}{4}\right) \eta |\lambda_i|^2 \right\} \forall \; i \\ &< \max\left\{\left(\frac{\eta^2}{4} |\lambda_i|^2 - 1\right) \eta |\lambda_i|^2 \right\} \forall \; i \end{aligned} \tag{51}$$

where we have retained only terms up to cubic-order in $\eta, |\beta|$ and h. Hence, choosing $\eta \in \left(0, \frac{2}{\text{Im}(\lambda_{\text{max}})}\right)$, ensures:

$$\nabla_\alpha \rho\big|_{\alpha=0} < 0 \; \forall \; i, \tag{52}$$

We thus posit, that a choice of a positive α causes the norm of the limiting eigenvalue μ_+ of F_{LEAD} to decrease.

G. Proof of Theorem B.2

Theorem. Setting $\eta = \alpha = \frac{1}{2(\sigma_{\text{max}}(\mathbb{A}) + h)}$, then we have $\forall \; \epsilon > 0$,

$$\Delta_{t+1} \in \mathcal{O}\left(\left(1 - \frac{\sigma_{min}^2}{4(\sigma_{max}+h)^2} - \frac{(1+\beta/2)h}{\sigma_{max}+h} + \frac{3h^2}{8(\sigma_{max}+h)^2}\right)^t \Delta_0\right) \tag{53}$$

where $\sigma_{max}(\sigma_{min})$ is the largest (smallest) eigenvalue of $\mathbb{A}$, $\Delta_{t+1} := ||\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}^*||_2^2 + ||\boldsymbol{\omega}_t - \boldsymbol{\omega}^*||_2^2$.

Proof: From Eq. (46), we recall that the eigenvalues of $\nabla F_{\text{LEAD}}(\boldsymbol{\omega}^*)$ for the quadratic game are,

$$\mu_\pm^{(i)}(\alpha, \beta, \eta) = \frac{(1 - (\alpha+\eta)\lambda_i + \beta - \eta h)}{2} \left(1 \pm \sqrt{1 - \frac{4(\beta - \eta\lambda_i)}{(1-(\alpha+\eta)\lambda_i + \beta - \eta h)^2}}\right) \tag{54}$$

with $\lambda_i \in \text{Sp}(\text{off-diag}[\nabla \boldsymbol{v}(\omega^*)])$. Now, since in the quadratic-game setting considered, we have,

$$\text{off-diag}[\nabla \boldsymbol{v}(\omega^*)] = \mathbb{D}_{\text{q}}\mathbb{P} = \begin{bmatrix} 0 & \mathbb{A} \\ -\mathbb{A}^T & 0 \end{bmatrix} \tag{55}$$

hence, $\lambda_i = \pm i\sigma_i$ with σ_i being the singular values of $\mathbb{A}$. This, then allows us to write,

$$\mu_\pm^{(i)}(\alpha, \beta, \eta) = \frac{(1 - (\alpha+\eta)(\pm i\sigma_i) + \beta - \eta h)}{2} \left(1 \pm \sqrt{1 - \frac{4(\beta - \alpha(\pm i\sigma_i))}{(1-(\alpha+\eta)(\pm i\sigma_i) + \beta - \eta h)^2}}\right) \tag{56}$$

According to Proposition 1, the convergence behavior of LEAD is determined as, $\Delta_{t+1} \leq \mathcal{O}(\rho + \epsilon)\Delta_t \; \forall \; \epsilon > 0$, where (setting $\eta = \alpha$),

$$\begin{aligned} \rho &:= \max\{|\mu_+^{(i)}|^2, |\mu_-^{(i)}|^2\} \; \forall \; i \\ &= |\mu_+^{(i)}|^2 \; \forall \; i \end{aligned} \tag{57}$$

Now assuming that η is small enough, such that, $\eta^3 \approx 0$ and $\beta^2 \approx 0$, we have,

$$\rho \approx 1 - \eta^2\sigma_i^2 + \frac{3}{2}\eta^2 h^2 - (2+\beta)\,\eta h \tag{58}$$

Furthermore, using a learning rate η as prescribed by Proposition 2, such as $\eta = \alpha = \frac{1}{2(\sigma_{\max}(\mathbb{A})+h)}$ we find,

$$r_{\text{LEAD}} = \frac{\sigma_{min}^2}{4(\sigma_{max}+h)^2} + \frac{(1+\beta/2)h}{\sigma_{max}+h} - \frac{3h^2}{8(\sigma_{max}+h)^2} \tag{59}$$

Therefore,

$$\begin{aligned}\Delta_{t+1} &\leq \mathcal{O}\left(\left(1 - r_{\text{LEAD}}\right)^t \Delta_0\right) \\ &= \mathcal{O}\left(\left(1 - \frac{\sigma_{min}^2}{4(\sigma_{max}+h)^2} - \frac{(1+\beta/2)h}{\sigma_{max}+h} + \frac{3h^2}{8(\sigma_{max}+h)^2}\right)^t \Delta_0\right)\end{aligned} \tag{60}$$

where $\Delta_{t+1} := ||\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}^*||_2^2 + ||\boldsymbol{\omega}_t - \boldsymbol{\omega}^*||_2^2$.

H. LEAD-Adam

Since Adam algorithm is commonly used in large-scale experiments, we extend LEAD to be used with the Adam algorithm.

Algorithm 1 Least Action Dynamics Adam (LEAD-Adam)

Input: learning rate η, momentum β, coupling coefficient α.
Initialize: $x_0 \leftarrow x_{init}$, $y_0 \leftarrow y_{init}$, $t \leftarrow 0$, $m_0^x \leftarrow 0$, $v_0^x \leftarrow 0$ $m_0^y \leftarrow 0$, $v_0^y \leftarrow 0$
while not converged **do**
  $t \leftarrow t + 1$
  $g_x \leftarrow \nabla_x f(x_t, y_t)$
  $g_{xy}\Delta y \leftarrow \nabla_y(g_x)(y_t - y_{t-1})$
  $g_t^x \leftarrow g_{xy}\Delta y + g_x$
  $m_t^x \leftarrow \beta_1.m_{t-1}^x + (1-\beta_1).g_t^x$
  $v_t^x \leftarrow \beta_2.v_{t-1}^x + (1-\beta_2).(g_t^x)^2$
  $\hat{m}_t \leftarrow m_t/(1-\beta_1^t)$
  $\hat{v}_t \leftarrow v_t/(1-\beta_2^t)$
  $x_{t+1} \leftarrow x_t - \eta\,\hat{m}_t/(\sqrt{\hat{v}_t} + \epsilon)$
  $g_y \leftarrow \nabla_y f(x_{t+1}, y_t)$
  $g_{xy}\Delta x \leftarrow \nabla_x(g_y)(x_{t+1} - x_t)$
  $g_t^y \leftarrow g_{xy}\Delta x + g_y$
  $m_t^y \leftarrow \beta_1.m_{t-1}^y + (1-\beta_1).g_t^y$
  $v_t^y \leftarrow \beta_2.v_{t-1}^y + (1-\beta_2).(g_t^y)^2$
  $\hat{m}_t^y \leftarrow m_t^y/(1-\beta_1^t)$
  $\hat{v}_t^y \leftarrow v_t^y/(1-\beta_2^t)$
  $y_{t+1} \leftarrow y_t + \eta\,\hat{m}_t^y/(\sqrt{\hat{v}_t^y} + \epsilon)$
end while
return (x, y) =0

I. 8-Gaussians Generation

We compare our method LEAD-Adam with vanilla-Adam (Kingma & Ba, 2014) on the generation task of a mixture of 8-Gaussians. Standard optimization algorithms such as vanilla-Adam suffer from mode collapse in this simple task, implying the generator cannot produce samples from one or several of the distributions present in the real data. Through Figure 4, we demonstrate that LEAD-Adam fully captures all the modes in the real data in both saturating and non-saturating losses.

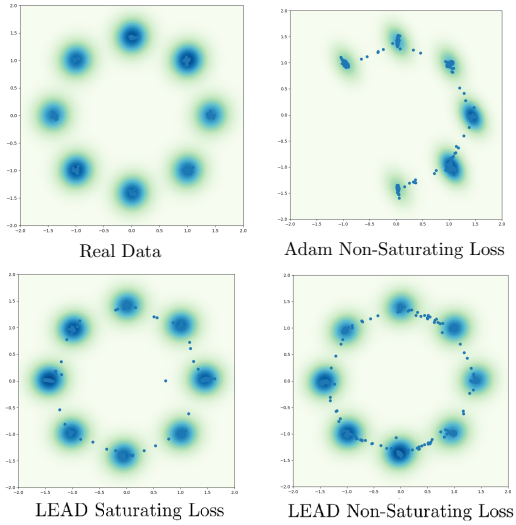

Figure 4: Performance of LEAD-Adam on the generation task of 8-Gaussians. All samples are shown after 10k iterations. Samples generated using Adam exhibit mode collapse, while LEAD-Adam does not suffer from this issue.

J. Experiments and Implementation Details

J.1. Mixture of Eight Gaussians

Dataset The real data is generated by 8-Gaussian distributions their mean are uniformly distributed around the unit circle and their variance is 0.05. The code to generate the data is included in the source code.

Architecture The architecture for Generator and Discriminator, each consists of four layers of affine transformation, followed by ReLU non-linearity. The weight initialization is default PyTorch's initialization scheme. See a schematic of the architecture in Table 3.

Table 3: Architecture used for the Mixture of Eight Gaussians.

Generator	Discriminator
Input: $z \in \mathbb{R}^{64} \sim \mathcal{N}(0, I)$	*Input:* $x \in \mathbb{R}^2$
Linear ($64 \to 2000$)	Linear ($2 \to 2000$)
ReLU	ReLU
Linear ($2000 \to 2000$)	Linear ($2000 \to 2000$)
ReLU	ReLU
Linear ($2000 \to 2000$)	Linear ($2000 \to 2000$)
ReLU	ReLU
Linear ($2000 \to 2$)	Linear ($2000 \to 1$)

Other Details We use the Adam (Kingma & Ba, 2014) optimizer on top of our algorithm in the reported results. Furthermore, we use batchsize of 128.

J.2. CIFAR 10 DCGAN

Dataset The CIFAR10 dataset is available for download at the following link; `https://www.cs.toronto.edu/~kriz/cifar.html`

Architecture The discriminator has four layers of convolution with LeakyReLU and batch normalization. Also, the generator has four layers of deconvolution with ReLU and batch normalization. See a schematic of the architecture in Table 4.

Other Details For the baseline we use Adam with β_1 set to 0.5 and β_2 set to 0.99. Generator's learning rate is 0.0002 and

Table 4: Architecture used for CIFAR-10 DCGAN.

Generator	Discriminator
Input: $z \in \mathbb{R}^{100} \sim \mathcal{N}(0, I)$	*Input:* $x \in \mathbb{R}^{3\times32\times32}$
conv. (ker: 4×4, $100 \to 1024$; stride: 1; pad: 0)	conv. (ker: 4×4, $3 \to 256$; stride: 2; pad: 1)
Batch Normalization	LeakyReLU
ReLU	conv. (ker: 4×4, $256 \to 512$; stride: 2; pad: 1)
conv. (ker: 4×4, $1024 \to 512$; stride: 2; pad: 1)	Batch Normalization
Batch Normalization	LeakyReLU
ReLU	conv. (ker: 4×4, $512 \to 1024$; stride: 2; pad: 1)
conv. (ker: 4×4, $512 \to 256$; stride: 2; pad: 1)	Batch Normalization
Batch Normalization	LeakyReLU
ReLU	conv. (ker: 4×4, $1024 \to 1$; stride: 1; pad: 0)
conv. (ker: 4×4, $256 \to 3$; stride: 2; pad: 1)	
Tanh	Sigmoid

discriminator's learning rate is 0.0001. The same learning rate and momentum were used to train LEAD model. We also add the mixed derivative term with $\alpha_d = 0.3$ and $\alpha_g = 0.0$.

The baseline is a DCGAN with the standard non-saturating loss (non-zero sum formulation). In our experiments, we compute the FID based on 50,000 samples generated from our model vs 50,000 real samples.

Samples

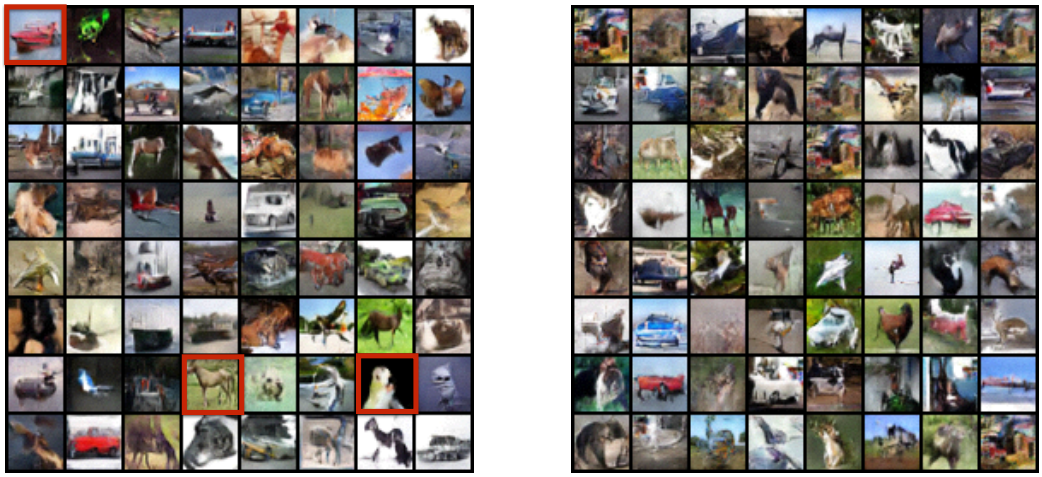

Figure 5: Performance of LEAD on CIFAR-10 image generation task on a DCGAN architecture. **Left**: LEAD achieves FID 19.27. **Right**: Vanilla Adam achieves FID 24.38. LEAD is able to generate better sample qualities from several classes such as ships, horses and birds (red). Best performance is reported after 100 epochs.

J.3. CIFAR 10 ResNet

Dataset The CIFAR10 dataset is available for download at the following link; `https://www.cs.toronto.edu/~kriz/cifar.html`

Architecture See Table 6 for a schematic of the architecture used for the CIFAR10 experiments with ResNet.

Other Details The baseline is a ResNet with non-saturating loss (non-zero sum formulation). Similar to (Miyato et al., 2018), for every time that the generator is updated, the discriminator is updated 5 times. For both the Baseline SNGAN and LEAD-Adam we use a β_1 of 0.0 and β_2 of 0.9 for Adam. Baseline SNGAN uses a learning rate of 0.0002 for both the generator and the discriminator. LEAD-Adam also uses a learning rate of 0.0002 for the generator but 0.0001 for the discriminator. LEAD-Adam uses an α of 0.5 and 0.01 for the generator and the discriminator respectively. Furthermore, we

Table 5: ResNet blocks used for the ResNet architectures (see Table 6).

Gen–Block
Shortcut:
Upsample($\times 2$)
Residual:
Batch Normalization
ReLU
Upsample($\times 2$)
conv. (ker: 3×3, $256 \to 256$; stride: 1; pad: 1)
Batch Normalization
ReLU
conv. (ker: 3×3, $256 \to 256$; stride: 1; pad: 1)

Dis–Block
Shortcut:
downsample
conv. (ker: 1×1, $3_{\ell=1}/128_{\ell\neq1} \to 128$; stride: 1)
Spectral Normalization
[AvgPool (ker:2×2, stride:2)], if $\ell \neq 1$
Residual:
[ReLU], if $\ell \neq 1$
conv. (ker: 3×3, $3_{\ell=1}/128_{\ell\neq1} \to 128$; stride: 1; pad: 1)
Spectral Normalization
ReLU
conv. (ker: 3×3, $128 \to 128$; stride: 1; pad: 1)
Spectral Normalization
AvgPool (ker:2×2)

Table 6: ResNet architectures used for experiments on CIFAR10.

Generator	**Discriminator**
Input: $z \in \mathbb{R}^{64} \sim \mathcal{N}(0, I)$	*Input:* $x \in \mathbb{R}^{3\times32\times32}$
Linear($64 \to 4096$)	D–ResBlock
G–ResBlock	D–ResBlock
G–ResBlock	D–ResBlock
G–ResBlock	D–ResBlock
Batch Normalization	ReLU
ReLU	AvgPool (ker:8×8)
conv. (ker: 3×3, $256 \to 3$; stride: 1; pad:1)	Linear($128 \to 1$)
$Tanh(\cdot)$	Spectral Normalization

evaluate both the baseline and our method on an exponential moving average of the generator's parameters.

In our experiments, we compute the FID based on 50,000 samples generated from our model vs 50,000 real samples and reported the mean and variance over 5 random runs. We have provided pre-trained models as well as the source code for both LEAD-Adam and Baseline SNGAN in our GitHub repository.

Samples

K. Comparison to other methods

In this section we compare our method with several other second order methods in the min-max setting.

The distinction of LEAD from SGA and LookAhead, can be understood by considering the 1st-order approximation of $x_{k+1} = x_k - \eta\nabla_x f\left(x_k, y_k + \eta\Delta y_k\right)$, where $\Delta y_k = \eta\nabla_y f\left(x_k + \eta\Delta x, y_k\right)$.

This gives rise to:

$$x_{k+1} = x_k - \eta\nabla_x f\left(x_k, y_k\right) - \eta^2\nabla^2_{xy} f\left(x_k, y_k\right)\Delta y, \tag{61}$$

$$y_{k+1} = y_k + \eta\nabla_y f\left(x_k, y_k\right) + \eta^2\nabla^2_{xy} f\left(x_k, y_k\right)\Delta x, \tag{62}$$

with $\Delta x, \Delta y$ corresponding to each player accounting for its opponent's potential next step. However, SGA and LookAhead

[2]For FtR, we provide the update for the second player given the first player performs gradient descent. Also note that in this table SGA is simplified for the two player zero-sum game. Non-zero sum formulation of SGA such as the one used for GANs require the computation of $\mathbb{J}\mathbf{v}, \mathbb{J}^\top\mathbf{v}$.

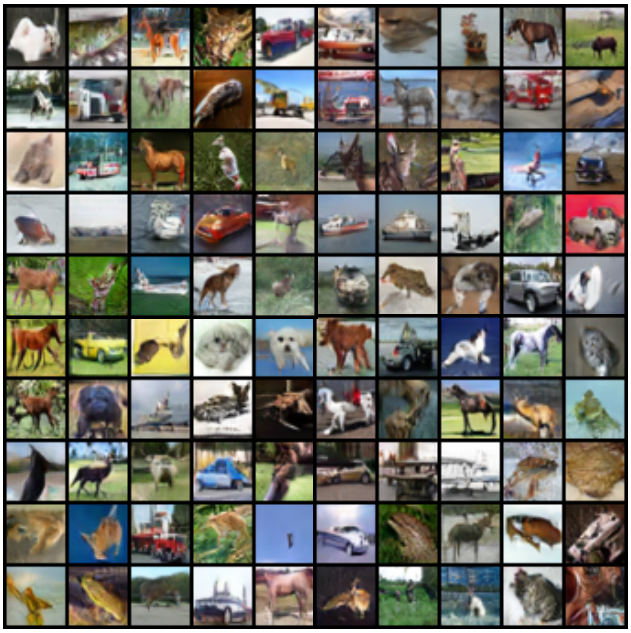

Figure 6: Generated sample of LEAD-Adam on CIFAR-10 after 50k iterations on a ResNet architecture. We achieve an FID score of 10.49 using learning rate $2e-4$ for the generator and the discriminator, α for the generator is set to 0.01 and for the discriminator is set to 0.5.

Table 7: Comparison of several second-order methods in min-max optimization. Each update rule, corresponding to a particular row, can be constructed by adding cells in that row from Columns 4 to 7 and then multiplying that by the value in Column 1. Furthermore, $\Delta x_{k+1} = x_{k+1} - x_k$, while $\mathcal{C} = \left(\mathrm{I} + \eta^2 \nabla^2_{xy} f \nabla^2_{yx} f\right)$. We compare the update rules of the first player[2] for the following methods: Gradient Descent-Ascent (GDA), Least Action Dynamics (LEAD, ours), Symplectic Gradient Adjustment (SGA), Competitive Gradient Descent (CGD), Consensus Optimization (CO), Follow-the-Ridge (FtR) and Learning with Opponent Learning Awareness (LOLA), in a zero-sum game.

		Coefficient	Momentum	Gradient	Interaction-xy	Interaction-xx
GDA	$\Delta x_{k+1} =$	1	0	$-\eta \nabla_x f$	$-\eta \nabla_x f$	0
LEAD	$\mathbf{\Delta x_{k+1}} =$	**1**	$\beta \mathbf{\Delta x_k}$	$-\eta \nabla_{\mathbf{x}} \mathbf{f}$	$-\alpha \nabla^{\mathbf{2}}_{\mathbf{xy}} \mathbf{f} \mathbf{\Delta y_k}$	0
SGA[4]	$\Delta x_{k+1} =$	1	0	$-\eta \nabla_x f$	$-\eta\gamma \nabla^2_{xy} f \nabla_y f$	0
CGD[38]	$\Delta x_{k+1} =$	$\mathcal{C}^{-1}$	0	$-\eta \nabla_x f$	$-\eta^2 \nabla^2_{xy} f \nabla_y f$	0
CO[28]	$\Delta x_{k+1} =$	1	0	$-\eta \nabla_x f$	$-\eta\gamma \nabla^2_{xy} f \nabla_y f$	$-\eta\gamma \nabla^2_{xx} f \nabla_x f$
FtR[41]	$\Delta y_{k+1} =$	1	0	$\eta_y \nabla_y f$	$\eta_x \left(\nabla^2_{yy} f\right)^{-1} \nabla^2_{yx} f \nabla_x f$	0
LOLA[11]	$\Delta x_{k+1} =$	1	0	$-\eta \nabla_x f$	$-2\eta\alpha \nabla_{xy} f \nabla_y f$	0

additonally *model* their opponent as *naive* learners i.e. $\Delta x = -\nabla_x f(x_k, y_k)$, $\Delta y = \nabla_y f(x_k, y_k)$. On the contrary, our method does away with such specific assumptions, instead modeling the opponent based on its most recent move.

Furthermore, there is a resemblance between LEAD and OGDA that we would like to address. The 1st order Taylor expansion of the difference in gradients term of OGDA yields the update (for x):

$$x_{k+1} = x_k - \eta \nabla_x f - \eta^2 \nabla^2_{xy} f \nabla_y f + \eta^2 \nabla^2_{xx} f \nabla_x f, \tag{63}$$

which contains an extra 2nd order term $\nabla^2_{xx} f$ compared to ours. As noted in (Schäfer & Anandkumar, 2019), the $\nabla^2_{xx} f$ term does not systematically aid in curbing the min-max rotations, rather causing convergence to non-Nash points in some settings. For e.g., let us consider the simple game $f(x, y) = \gamma(x^2 - y^2)$, where x, y, γ are all scalars, with the Nash equilibrium of this game located at $(x^* = 0, y^* = 0)$. For a choice of $\gamma \geq 6$, OGDA fails to converge for any learning rate while methods like LEAD, Gradient Descent Ascent (GDA) and CGD ((Schäfer & Anandkumar, 2019)) that do not contain the $\nabla_{xx} f(\nabla_{yy} f)$ term do exhibit convergence. See Figure 7 and (Schäfer & Anandkumar, 2019) for more discussion.

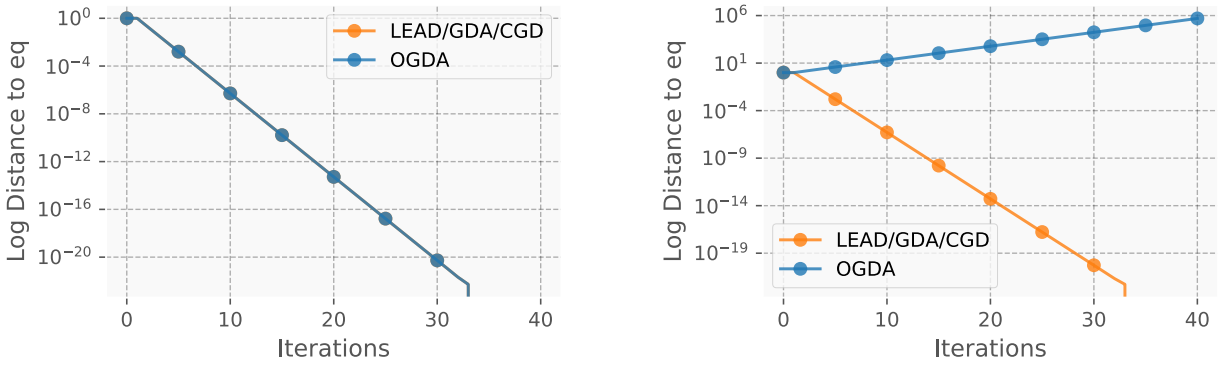

Figure 7: Figure depicting the convergence/divergence of several algorithms on the game of $f(x, y) = \gamma(x^2 - y^2)$ (Nash equilibrium at $x^* = 0, y^* = 0$). **Left**: For $\gamma = 1$, OGDA and LEAD/GDA/CGD (overlaying) are found to converge to the Nash eq. **Right**: For $\gamma = 6$, we find that OGDA fails to converge while LEAD/GDA/CGD (overlaying) converge. We conjecture that the reason behind this observation is the existence of $\nabla^2_{xx} f$ term in the optimization algorithm of OGDA.