LEAD: Least-Action Dynamics for Min-Max Optimization

author names withheld

Under Review for OPT 2020

Abstract

Adversarial formulations in machine learning have rekindled interest in differentiable games. Specifically, the development of efficient optimization methods for two-player min-max games is an active area of research with a timely impact on emerging adversarial techniques, including generative adversarial networks (GANs). Existing methods for this type of problem typically employ intuitive, carefully hand-designed mechanisms for controlling the problematic rotational dynamics commonly encountered during optimization. In this work, we take a more principled approach to address this issue by casting min-max optimization as a physical system. We propose LEAD (Least-Action Dynamics), a novel optimizer that uses the principle of least-action from physics to discover an efficient optimizer for min-max games. We subsequently provide convergence analysis of this novel second-order optimizer in quadratic min-max games using the Lyapunov theory. Finally, we empirically test our method on synthetic problems and GANs to demonstrate improvements over baseline methods.

1. Introduction

Much of the advances in traditional machine learning can be attributed to the success of gradient-based methods. Modern machine learning formulations such as Generative Adversarial Networks (GANs) [12], multi-task learning, and multi-agent settings [30] in reinforcement learning [5] require joint optimization of two or more objectives. In these *game* settings, best practices and methods developed for single-objective optimization perform poorly [2, 10]. Notably they exhibit problematic rotational dynamics about the *Nash Equilibria* [25], which can slow down convergence. Recent work in game optimization [1, 2, 21, 23, 25, 32] demonstrates that one way of addressing these rotational dynamics is to use second-order information. These works intuitively introduce additional second-order terms in the optimization algorithm to suppress these rotations, thus improving convergence.

Instead, in this work, we attempt to tackle these dynamics by taking a principled, *physical* approach into the problem. By likening the gradient-based optimization of two-player (zero-sum) games to a physical system's dynamics, we introduce additional "counter-rotational" forces to curb the problematic rotations. We then employ the principle of least action from physics, a variational principle, to discover a natural and efficient (continuous-time) dynamics of this modeled system. Next, by discretizing these continuous-time dynamics using implicit and symplectic Euler methods, we derive two novel second-order game optimization algorithms: implicit Least Action Dynamics (iLEAD) and symplectic Least Action Dynamics (sLEAD).

We then employ tools from dynamical systems theory, namely Lyapunov functions, to prove linear convergence of our optimizer in both continuous and discrete-time, in the quadratic min-max game setting. We then empirically demonstrate that sLEAD improves the performance of GANs on

tasks such as 8-Gaussians and CIFAR-10 while comparing the it performance against other first and second-order methods in these settings.

2. Background

Problem Setting In this work, we study the optimization problem of two-player zero-sum games $\min_X \max_Y f(X,Y)$ where $f : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$, and is assumed to be continuous and twice differentiable w.r.t. $X, Y \in \mathbb{R}$. In developing our framework below, we assume X, Y to be scalars. We later demonstrate both analytically (Appendix G, H) and empirically, that our results hold for the more general case of vectorial X and Y. Additionally, we assume $f(X,Y)$ to be convex-concave, with the Nash Equilibrium of the game occurring at $X = 0, Y = 0$.

Mechanics and Optimization In our attempt to find an efficient update scheme or trajectory to optimize the min-max objective $f(X,Y)$, we note from classical physics the following: under the influence of a net force F, the trajectory of motion of a physical object of mass m, is determined by Newton's 2nd Law, $m\ddot{X} = F$, where we have abused notation to express the object's coordinate as $X_t \equiv X$. As stated by the *Principle of Least Action*[1] [18], nature "selects" the particular trajectory of motion obtained from solving $m\ddot{X} = F$, over other possibilities as a quantity called *Action* is extremized along it. Hence, the ability to model our game optimization task in terms of an object moving under a force allows for a natural discovery of an efficient optimization path through the least action principle, according to $m\ddot{X} = F$.

As detailed in Appendix A, we argue that the physical dynamics of a charged particle experiencing vortex ([4]), magnetic and frictional forces as evaluated through Newton's 2nd Law, represent a "naturally efficient" optimization trajectory for minimax games in the form:

$$\begin{aligned} m\ddot{X} &= -\mu\dot{X} - \nabla_X f - 2q\left(\nabla_{XY} f\right)\dot{Y} \\ m\ddot{Y} &= -\mu\dot{Y} + \nabla_Y f + 2q\left(\nabla_{XY} f\right)\dot{X} \end{aligned} \tag{1}$$

As discussed in Appendix A, the vortex force in our physical system mimics the typical "rotations" of minimax optimization, while the magnetic and frictional forces serve as effective counter-rotations and momentum (in the optimization sense) respectively.

2.1. Optimizers on Discretization

In order to discretize the continuous-time trajectory of Eq.(1), we make use of Euler's implicit and explicit discretization schemes, (δ: discretization step-size, k: iteration step) through $\dot{X} = V_X$, to obtain two discretized algorithms, namely implicit Least Action Dynamics (iLEAD) and symplectic Least Action Dynamics (sLEAD). To state, here *symplectic*[2] discretization corresponds to using a combination of explicit and implicit steps [31].

Proposition 1 *The continuous-time EOMs (1) can be discretized in an implicit way as,*

$$\begin{aligned} x_{k+1} &= x_k + \beta_{imp}\Delta x_k - \eta_{imp}\nabla_x f(x_{k+1}, y_{k+1}) - \alpha_{imp}\nabla_{xy} f(x_{k+1}, y_{k+1})\Delta y_{k+1} \\ y_{k+1} &= y_k + \beta_{imp}\Delta y_k + \eta_{imp}\nabla_y f(x_{k+1}, y_{k+1}) + \alpha_{imp}\nabla_{yx} f(x_{k+1}, y_{k+1})\Delta x_{k+1} \end{aligned} \tag{2}$$

1. Also referred to as the Principle of Stationary Action.
2. sLEAD does *not* preserve the symplectic structure of Eq.(1), and hence is not symplectic in the strict sense.

where $\alpha_{\text{imp}}, \beta_{\text{imp}}$ and η_{imp} are hyper-parameters dependent on μ, q and δ (Proof in Appendix C). We refer to these discrete update rules as implicit Least Action Dynamics (iLEAD).

Proposition 2 *The continuous-time EOMs (1) can be discretized in a symplectic way as,*

$$\begin{aligned} x_{k+1} &= x_k + \beta_{sym}\Delta x_k - \eta_{sym}\nabla_x f\left(x_k, y_k\right) - \alpha_{sym}\nabla_{xy} f\left(x_k, y_k\right)\Delta y_k \\ y_{k+1} &= y_k + \beta_{sym}\Delta y_k + \eta_{sym}\nabla_y f\left(x_k, y_k\right) + \alpha_{sym}\nabla_{yx} f\left(x_k, y_k\right)\Delta x_k \end{aligned} \tag{3}$$

where, as above, $\alpha_{\text{sym}}, \beta_{\text{sym}}, \eta_{\text{sym}}$ are hyper-parameters dependent on μ, q and δ (Proof in Appendix D). We refer to these discrete update rules as symplectic Least Action Dynamics (sLEAD).

In the following, we will be using iLEAD to establish convergence guarantees of our method in discrete-time, while our experiments are instead performed using sLEAD. This is distinction is due to the fact that implicit methods are not practical to implement. Despite this distinction, we demonstrate that sLEAD still improves convergence in min-max optimization.

3. Convergence Analysis

3.1. Lyapunov Stability of Quadratic min-max Games

We now study the behavior of our method iLEAD on the quadratic min-max game,

$$f\left(X, Y\right) = \frac{h}{2}X^2 - \frac{h}{2}Y^2 + XY, \tag{4}$$

where h is the strong monotonicity. (Refer to Appendix G and H, for an extension of the following results to the general bilinear game $f\left(\mathbf{X}, \mathbf{Y}\right) = \mathbf{X}^T\mathbb{A}\mathbf{Y}$.)

3.2. Continuous Time Analysis

A general way to prove the stability of a dynamical system is to use a Lyapunov function [14, 22]. The scalar function $\mathcal{E}_t : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$, is a Lyapunov function of the continuous-time dynamics of Eq.(1), if $\forall\, t\ (i)\ \ \mathcal{E}_t(X, Y) \geq 0$ and, $(ii)\ \ \dot{\mathcal{E}}_t(X, Y) \leq 0$. The Lyapunov function $\mathcal{E}_t$ can be perceived to be a generalization of the total energy of the system and the requirement (ii) ensures that this generalized energy decreases along the trajectory of evolution, leading the system to convergence as we show next.

Theorem 1 *For the quadratic min-max game of Eq.(4), $\mathcal{E}_t$ as defined in Eq.(22) is a Lyapunov function for the dynamics of Eq.(1) under the choice $q = \left(2 + \mu^2\right)/\mu$, with $\dot{\mathcal{E}}_t \leq -\rho\mathcal{E}_t$ for some positive definite constant ρ dependent on μ and h. Hence, $X^2 + Y^2 \leq \frac{\mathcal{E}_0}{1+h}\exp\left(-\rho t\right)$*

Thus, the continuous-time dynamics of Eq.(1) for the quadratic game are convergent at a linear rate ρ, to the Nash equilibrium $(0, 0)$. (Proof in Appendix E).

3.3. Discrete Time Analysis

To perform Lyapunov analysis on our discrete-time dynamics of iLEAD (Eq.(2)), we note that a function $\mathcal{E}_k$ is a discrete Lyapunov function if $\forall\, k \in \mathbf{N}$, $(i)\, \mathcal{E}_k \geq 0$, and $(ii)\, \mathcal{E}_k - \mathcal{E}_{k-1} \leq 0$.

Theorem 2 *For the quadratic min-max game of Eq.(4), $\mathcal{E}_k$ as defined in Eq.(33), is a discrete Lyapunov function of the implicit dynamics of iLEAD (Eq.(2)) under the choice $q = \sqrt{5}\left(\frac{2+\mu^2}{\mu}\right)$ with $\mathcal{E}_k \leq \left(\frac{\mathcal{C}}{\mathcal{C}+\delta\mu}\right)\mathcal{E}_{k-1}$. This leads iLEAD to converge linearly to the Nash equilibrium $(0,0)$ as $x_k^2 + y_k^2 \leq \frac{\mu^2}{\mathcal{C}}\left(\frac{\mathcal{C}}{\mathcal{C}+\delta\mu}\right)^k \mathcal{E}_0$, where $\mathcal{C} = \mu^2\left(2\sqrt{5}+4h\right) + 4\sqrt{5}$, and δ is the discretization step-size (Proof in Appendix F).*

4. Experiments

4.1. Comparison of Computational Cost

We define the Jacobian of the gradient vector-field $\xi = (\nabla_x f(x,y), -\nabla_y f(x,y))$ as,

$$J = \begin{bmatrix} \nabla_x^2 f\,(x,y) & \nabla_{xy} f\,(x,y) \\ -\nabla_{yx} f\,(x,y) & -\nabla_y^2 f\,(x,y) \end{bmatrix}. \tag{5}$$

Now, consider player x. An sLEAD update for the same requires the computation of the term $\nabla_{xy} f\,(x_k, y_k)\,(y_k - y_{k-1})$, thereby involving only *one* out of the three distinct blocks of the full Jacobian J (5). On the other hand, original implementation of SGA [2] for example, requires the full computation of two Jacobian-vector products $J\xi, J^\top\xi$. Similarly, CGD [29] involves the computation of $\left(1 + \eta\nabla_{xy}^2 f(x_k,y_k)\nabla_{yx}^2 f(x_k,y_k)\right)^{-1}$ along with the Jacobian-vector product $\nabla_{xy}^2 f(x_k,y_k)\nabla_y f(x_k,y_k)$. While the inverse term in [29] is approximated using conjugate gradient method for implementation, this still involves the computation of ten Jacobian-vector products. See Fig.1 (Left) for a comparison with several first and second-order methods on the task of 8-Gaussians.

4.2. Generative Adversarial Networks

We test sLEAD+Adam on the task of CIFAR-10 [17] image generation with a non-zero-sum formulation (non-saturating) on a DCGAN architecture similar to [13] . As shown in Fig. 1 (Right), we observe that sLEAD outperforms all the other methods in terms of the Frechet Inception Distance (FID)[3] , reaching a score of 19.27. Furthermore, on a ResNet architecture similar to [11], sLEAD with Spectral Normalization (SN) [26] achieves an FID score 11.98. In comparison, on the same architecture vanilla SN achieves an FID score of 12.81. We give a detailed description of these experiments in Appendix I along with comparison of the Inception Score [28].

5. Related work

Authors in [25] provides a discussion on 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 large imaginary part results in oscillatory behavior. To mitigate this issue, they propose Consensus Optimization (CO). Authors in [2, 8, 20, 21] use the *Hamiltonian* of the gradient vector-field. To improve the convergence in games, Symplectic Gradient Adjustment (SGA) is proposed that disentangles the convergent parts of the dynamics from the rotational [2] . Another line of attack taken in [29] is to use second-order information as a regularizer of the dynamics and motivate the use

3. The FID [15] is a metric for evaluating the quality of generated samples using GANs. Lower FID corresponds to better sample quality.

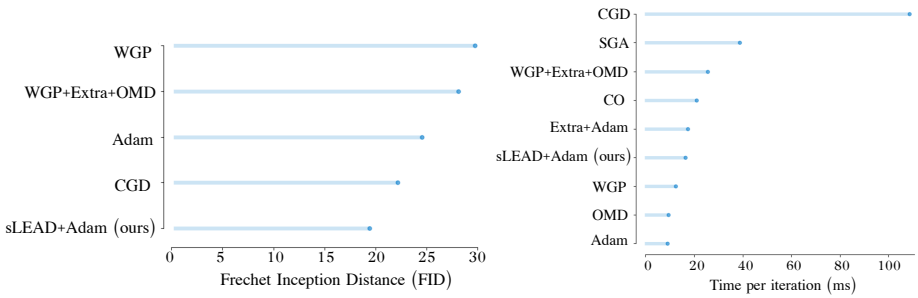

Figure 1: **Left:** Computational cost per iteration run of several well-known methods for GAN optimization. The numbers are reported on the 8-Gaussians generation task and averaged over 1k iterations. In the above, OMD: Optimistic Mirror Descent, WGP: WGAN + Gradient Penalty, EG: Extra-gradient, CO: Consensus Optimization, OMD: Optimistic Mirror Descent, SGA: Symplectic Gradient Adjustment and CGD: Competitive Gradient Descent. We observe that per-iteration time complexity of our method is very similar to extra-gradient and WGAN with Gradient Penalty and is much cheaper than other second order methods such as CGD and SGA. **Right:** Performance of several other methods reported after 100 epochs on a DCGAN architecture. FID is computed over 50k samples. Results on CGD (competative gradient descent) and WGP+Extra+OMD (WGAN with gradient penalty, extra-gradient and optimistic mirror descent) are reported from [29] and [24] respectively.

of Competitive Gradient Descent (CGD). In [32], 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. Another approach taken by [10], demonstrate how applying negative momentum over GDA can improve convergence in min-max games. Authors in [6] show that extrapolating the next value of the gradient using previous history, aids convergence.

6. Conclusion

In this paper, we leverage tools from physics and dynamical systems theory to propose a novel second-order optimization scheme for zero-sum games, to address the problematic rotational dynamics encountered in their training. By casting min-max game optimization as a physical system, we use the Principle of Least Action to "naturally" discover an effective optimization algorithm in this setting. Using Lyapunov theory, we prove our proposed optimizer to be convergent at a linear rate in the case of quadratic min-max games. We supplement our theoretical analysis with experiments in the quadratic as well as GAN settings, demonstrating improvements over baseline methods. Specifically for GAN training, we observe that our method outperforms other second-order methods, proposed to tackle the rotations, in terms of sample quality and computational efficiency. We believe our analysis helps bridge the gap between the usage of tools developed for physics and game optimization. Despite performing our analysis in a simplified setting, our promising empirical results on problems beyond this domain encourages us to believe our approach holds promise.

References

[1] Jacob Abernethy, Kevin A Lai, and Andre Wibisono. Last-iterate convergence rates for min-max optimization. *arXiv preprint arXiv:1906.02027*, 2019.

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

[3] Shane Barratt and Rishi Sharma. A note on the inception score. *arXiv preprint arXiv:1801.01973*, 2018.

[4] MV Berry and Pragya Shukla. Curl force dynamics: symmetries, chaos and constants of motion. *New Journal of Physics*, 18(6):063018, 2016.

[5] Lucian Bu, Robert Babu, Bart De Schutter, 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.

[6] Constantinos Daskalakis, Andrew Ilyas, Vasilis Syrgkanis, and Haoyang Zeng. Training gans with optimism. In *International Conference on Learning Representations*, 2018.

[7] Jakob Foerster, Richard Y Chen, Maruan Al-Shedivat, Shimon Whiteson, Pieter Abbeel, and Igor Mordatch. Learning with opponent-learning awareness. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 122–130. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

[8] Ian Gemp and Sridhar Mahadevan. Global convergence to the equilibrium of gans using variational inequalities. *arXiv preprint arXiv:1808.01531*, 2018.

[9] Gauthier Gidel, Hugo Berard, Gaëtan Vignoud, Pascal Vincent, and Simon Lacoste-Julien. A variational inequality perspective on generative adversarial networks. In *International Conference on Learning Representations*, 2018.

[10] Gauthier Gidel, Reyhane Askari Hemmat, Mohammad Pezeshki, Rémi Le Priol, Gabriel Huang, Simon Lacoste-Julien, and Ioannis Mitliagkas. Negative momentum for improved game dynamics. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1802–1811, 2019.

[11] Xinyu Gong. *sngan.pytorch*, 2019. https://github.com/GongXinyuu/sngan.pytorch.

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[13] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pages 5767–5777, 2017.

[14] Wolfgang Hahn, Hans H Hosenthien, and H Lehnigk. *Theory and application of Liapunov's direct method*. Prentice-Hall Englewood Cliffs, NJ, 1963.

[15] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, pages 6626–6637, 2017.

[16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[17] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[18] LD Landau and EM Lifshitz. *Course of theoretical physics. vol. 1: Mechanics*. Oxford, 1960.

[19] Lev Davidovich Landau, JS Bell, MJ Kearsley, LP Pitaevskii, EM Lifshitz, and JB Sykes. *Electrodynamics of continuous media*, volume 8. elsevier, 2013.

[20] Alistair Letcher, David Balduzzi, Sébastien Racaniere, James Martens, Jakob N Foerster, Karl Tuyls, and Thore Graepel. Differentiable game mechanics. *Journal of Machine Learning Research*, 20(84):1–40, 2019.

[21] Nicolas Loizou, Hugo Berard, Alexia Jolicoeur-Martineau, Pascal Vincent, Simon Lacoste-Julien, and Ioannis Mitliagkas. Stochastic hamiltonian gradient methods for smooth games. *ICML*, 2020.

[22] Aleksandr Mikhailovich Lyapunov. The general problem of the stability of motion. *International journal of control*, 55(3):531–534, 1992.

[23] Eric V Mazumdar, Michael I Jordan, and S Shankar Sastry. On finding local nash equilibria (and only local nash equilibria) in zero-sum games. *arXiv preprint arXiv:1901.00838*, 2019.

[24] Panayotis Mertikopoulos, Bruno Lecouat, Houssam Zenati, Chuan-Sheng Foo, Vijay Chandrasekhar, and Georgios Piliouras. Optimistic mirror descent in saddle-point problems: Going the extra (gradient) mile. *arXiv preprint arXiv:1807.02629*, 2018.

[25] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. The numerics of gans. In *Advances in Neural Information Processing Systems*, pages 1825–1835, 2017.

[26] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

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

[28] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016.

[29] Florian Schäfer and Anima Anandkumar. Competitive gradient descent. In *Advances in Neural Information Processing Systems*, pages 7623–7633, 2019.

[30] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In *Advances in Neural Information Processing Systems*, pages 527–538, 2018.

[31] Bin Shi, Simon S Du, Weijie Su, and Michael I Jordan. Acceleration via symplectic discretization of high-resolution differential equations. In *Advances in Neural Information Processing Systems*, pages 5745–5753, 2019.

[32] Yuanhao Wang, Guodong Zhang, and Jimmy Ba. On solving minimax optimization locally: A follow-the-ridge approach. In *International Conference on Learning Representations*, 2019.

Appendix A. Description of the Physical System

In our attempt to cast game optimization as a physical system, we take inspiration from Polyak's heavy-ball momentum [27] method[4] in single objective minimization of an objective $f(x)$,

$$x_{k+1} = x_k + \beta (x_k - x_{k-1}) - \eta \nabla_x f(x_k), \tag{6}$$

which in continuous-time translates to (detailed derivation in Appendix B),

$$m\ddot{X} = -\nabla_X f(X). \tag{7}$$

Comparing Eqns.(7) and $m\ddot{X} = F$, we notice that in this case $F = -\nabla_X f(X)$, i.e. $f(X)$ acts as a *potential* function [18]. This thus lends the interpretation that Polyak's heavy-ball method Eq.(6) can be interpreted as an object (ball) of mass m rolling down under a potential $f(X)$ to reach the minimum.

Armed with this observation, we notice that, while a straightforward extension of Eq.(7) to two-dimensions is not of much use[5]; the following generalization closely resembles Gradient Descent-Ascent (in continuous-time) on our min-max objective $f(X, Y)$,

$$\begin{aligned} m\ddot{X} &= -\nabla_X f(X, Y) \\ m\ddot{Y} &= \nabla_Y f(X, Y). \end{aligned} \tag{8}$$

As it turns out, Eq.(8) corresponds to the equations of motion ($m\ddot{X} = F$) of an object moving under a *curl force* [4]: $\boldsymbol{F}_{\text{curl}} = (-\nabla_X f, \nabla_Y f)$ in the 2-dimensional XY plane. To investigate the nature of the above trajectory (8), we consider the prototypical min-max objective, $f(X, Y) = XY$. In this "bilinear" setting, the mass m object is found to spin away from the origin (Nash Equilibrium) over time.

While the system with a curl force (Eq.(8)) by itself does not seem to hold much promise in regards to providing an (efficient) optimization trajectory converging to the Nash Equilibrium, it nevertheless acts as a foundation. The physical nature of the formulation allows us to select from the extensive set of other physical forces to be added to the right-hand side of Eq.(8). This provides for a more physical and systematic way to "counter" the rotational effects of $\boldsymbol{F}_{\text{curl}}$, to exhibit desired convergent behavior possibly. To this end, we consider the simplest force known to produce rotatory motion on a particle of charge q: a magnetic force,

$$\boldsymbol{F}_{\text{mag}} = q\dot{\boldsymbol{X}} \times (\boldsymbol{\nabla} \times \boldsymbol{A}) \tag{9}$$

Here, $\boldsymbol{A}$ is the magnetic *vector potential* generating a magnetic field as, $\boldsymbol{B} = \nabla \times \boldsymbol{A}$ [19]. In order to make a prudent choice of $\boldsymbol{F}_{\text{mag}}$, we take note that desirable counter-rotations are observed if $\boldsymbol{A}$ is itself chosen to be a rotating vector field, like of the type $\boldsymbol{F}_{\text{curl}}$. Specifically, since we are attempting to counteract or negate the rotational effects of $\boldsymbol{F}_{\text{curl}}$, we *choose* $\boldsymbol{A} = -\boldsymbol{F}_{\text{curl}}$. This results in,

$$\boldsymbol{F}_{\text{mag}} = \left(-2q(\nabla_{XY} f)\dot{Y}, 2q(\nabla_{XY} f)\dot{X}\right) \tag{10}$$

The final ingredient we add to our system is a (velocity-dependent) frictional force $\boldsymbol{F}_{\text{fric}} = \left(\mu\dot{X}, \mu\dot{Y}\right)$, where μ is the friction coefficient. Eq.(8). The reason behind adding friction stems from

4. Arbitrary momentum coefficient results in incorporating friction in the equivalent physical system
5. Corresponds to *single*-objective minimization w.r.t. two variables

the fact that a) $\boldsymbol{F}_{\text{curl}}$ causes a particle's speed of motion to increase over time [4] while, b) $\boldsymbol{F}_{\text{mag}}$ does not cause any change in its speed of motion. Hence, under the influence of curl and magnetic forces, our mass m particle will keep increasing in speed over time, thus preventing convergence to a point. It is in this regard that a dissipative force such as friction comes of use.

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,

$$\begin{aligned} m\ddot{X} &= -\mu\dot{X} - \nabla_X f - 2q\left(\nabla_{XY} f\right)\dot{Y} \\ m\ddot{Y} &= -\mu\dot{Y} + \nabla_Y f + 2q\left(\nabla_{XY} f\right)\dot{X} \end{aligned} \tag{11}$$

Without loss of generality, we hereon assume our physical object to be of mass $m = 1$. By discretizing Eq.(1), we obtain our novel optimization algorithms for min-max games in Section section 2.1.

Appendix B. Derivation of Eq. 6

Proof Polyak's heavy ball method with unit momentum in the minimization of a single objective $f(x)$ is given by,

$$x_{k+1} = x_k + (x_k - x_{k-1}) - \eta\nabla_x f(x_k), \tag{12}$$

where η is the learning rate. One can rewrite this equation as,

$$\frac{(x_{k+\delta} - x_k) - (x_k - x_{k-\delta})}{\Delta^2} = -\frac{\eta}{\delta^2}\nabla_x f(x_k), \tag{13}$$

where δ is the discretization step-size. In the limit $\delta, \eta \to 0$, Eq.(13) then becomes ($x_k \to X(t) \equiv X$),

$$m\ddot{X} = -\nabla_X f(X) \tag{14}$$

This is equivalent to Newton's 2nd Law of motion ($m\ddot{X} = F$) of a particle of mass $m = \delta^2/\eta$, if we identify $F = -\nabla_X f(X)$. ■

Appendix C. Proof of Proposition 1

Proof The EOMs of the quadratic game in continuous-time (Eq.(1)), can be discretized in a purely implicit way as ($m = 1$),

$$x_{k+1} - x_k = \delta v^x_{k+1} \tag{15a}$$
$$y_{k+1} - y_k = \delta v^y_{k+1} \tag{15b}$$
$$v^x_{k+1} - v^x_k = -q\delta\nabla_{xy} f(x_{k+1}, y_{k+1})\, v^y_{k+1} - \mu\delta v^x_{k+1} - \delta\nabla_x f(x_{k+1}, y_{k+1}) \tag{15c}$$
$$v^y_{k+1} - v^y_k = q\delta\nabla_{xy} f(x_{k+1}, y_{k+1})\, v^x_{k+1} - \mu\delta v^y_{k+1} + \delta\nabla_y f(x_{k+1}, y_{k+1}), \tag{15d}$$

where δ is the discretization step-size. Plugging in Eqns.(15b), (15d) in Eqns.(15a), (15c) then give us,

$$\begin{aligned} x_{k+1} &= x_k + \beta_{\text{imp}}\Delta x_k - \eta_{\text{imp}}\nabla_x f(x_{k+1}, y_{k+1}) - \alpha_{\text{imp}}\nabla_{xy} f(x_{k+1}, y_{k+1})\Delta y_{k+1} \\ y_{k+1} &= y_k + \beta_{\text{imp}}\Delta y_k + \eta_{\text{imp}}\nabla_y f(x_{k+1}, y_{k+1}) + \alpha_{\text{imp}}\nabla_{xy} f(x_{k+1}, y_{k+1})\Delta x_{k+1} \end{aligned} \tag{16}$$

where $\Delta x_{k+1} = x_{k+1} - x_k$, and,

$$\beta_{\text{imp}} = \frac{1}{1+\mu\delta},\ \eta_{\text{imp}} = \frac{\delta^2}{1+\mu\delta},\ \alpha_{\text{imp}} = \frac{q\delta}{1+\mu\delta}. \tag{17}$$

■

Appendix D. Proof of Proposition 2

Proof The EOMs of the quadratic game in continuous-time (Eq.(1)), can be discretized in a symplectic [31] way as,

$$x_{k+1} - x_k = \delta v^x_{k+1}, \tag{18a}$$
$$y_{k+1} - y_k = \delta v^y_{k+1}, \tag{18b}$$
$$v^x_{k+1} - v^x_k = -q\delta\nabla_{xy} f\left(x_k, y_k\right) v^y_k - \mu\delta\delta v^x_k - \delta\nabla_x f\left(x_k, y_k\right) \tag{18c}$$
$$v^y_{k+1} - v^y_k = q\delta\nabla_{xy} f\left(x_k, y_k\right) v^x_k - \mu\delta v^y_k + \delta\nabla_y f\left(x_k, y_k\right) \tag{18d}$$

where δ is the discretization step-size. Now, using Eqns.(18a) and (18b) above, we can further re-express Eqns.(18c), (18d) as,

$$\begin{aligned} x_{k+1} &= x_k + \beta_{\text{sym}}\Delta x_k - \eta_{\text{sym}}\nabla_x f\left(x_k, y_k\right) - \alpha_{\text{sym}}\nabla_{x,y} f\left(x_k, y_k\right)\Delta y_k \\ y_{k+1} &= y_k + \beta_{\text{sym}}\Delta y_k + \eta_{\text{sym}}\nabla_y f\left(x_k, y_k\right) + \alpha_{\text{sym}}\nabla_{x,y} f\left(x_k, y_k\right)\Delta x_k \end{aligned} \tag{19}$$

where $\Delta x_k = x_k - x_{k-1}$, and,

$$\beta_{\text{sym}} = 1 - \mu\delta,\ \eta_{\text{sym}} = \delta^2,\ \alpha_{\text{sym}} = q\delta \tag{20}$$

■

Appendix E. Proof of Theorem 1

Proof For the class of functions in (4), Eq.(1) translates to,

$$\begin{aligned} \ddot{X} &= -q\dot{Y} - \mu\dot{X} - hX - Y \\ \ddot{Y} &= q\dot{X} - \mu\dot{Y} - hY + X \end{aligned} \tag{21}$$

The Lyapunov function for the above EOMs is then defined to be,

$$\begin{aligned} \mathcal{E}_t = &\frac{1}{2}\left(\dot{X} + \mu X + \mu Y\right)^2 + \frac{1}{2}\left(\dot{Y} + \mu Y - \mu X\right)^2 + \frac{1}{2}\left(\dot{X}^2 + \dot{Y}^2\right) \\ &+ (1+h)\left(X^2 + Y^2\right) \\ \geq &\ 0\ \forall\ t \end{aligned} \tag{22}$$

Next, by Eq.(21), we can compute the time derivative of $\mathcal{E}_t$ as,

$$\begin{aligned}
\dot{\mathcal{E}}_t &= \dot{X}\left(\ddot{X}+\mu\dot{X}+\mu\dot{Y}\right)+\mu X\left(\ddot{X}+\mu\dot{X}+\mu\dot{Y}\right)+\mu Y\left(\ddot{X}+\mu\dot{X}+\mu\dot{Y}\right)\\
&\quad+\dot{Y}\left(\ddot{Y}+\mu\dot{Y}-\mu\dot{X}\right)+\mu Y\left(\ddot{Y}+\mu\dot{Y}-\mu\dot{X}\right)-\mu X\left(\ddot{Y}+\mu\dot{Y}-\mu\dot{X}\right)\\
&\quad+\dot{X}\ddot{X}+\dot{Y}\ddot{Y}+2\left(1+h\right)\left(\dot{X}X+\dot{Y}Y\right)\\
&=\dot{X}\left(-q\dot{Y}-hX-Y+\mu\dot{Y}\right)+\mu X\left(-q\dot{Y}-hX-Y+\mu\dot{Y}\right)\\
&\quad+\mu Y\left(-q\dot{Y}-hX-Y+\mu\dot{Y}\right)+\dot{Y}\left(q\dot{X}-hY+X-\mu\dot{X}\right)\\
&\quad+\mu Y\left(q\dot{X}-hY+X-\mu\dot{X}\right)-\mu X\left(q\dot{X}-hY+X-\mu\dot{X}\right)\\
&\quad+\dot{X}\left(-q\dot{Y}-\mu\dot{X}-hX-Y\right)+\dot{Y}\left(q\dot{X}-\mu\dot{Y}-hY+X\right)\\
&\quad+2\left(1+h\right)\left(\dot{X}X+\dot{Y}Y\right)\\
&=\left(2-\mu q+\mu^2\right)X\dot{Y}+\left(2-\mu q+\mu^2\right)\dot{Y}Y-\mu\left(1+h\right)Y^2-\mu\dot{X}^2\\
&\quad-\left(2-\mu q+\mu^2\right)\dot{X}Y+\left(2-\mu q+\mu^2\right)\dot{X}X-\mu\left(1+h\right)X^2-\mu\dot{Y}^2
\end{aligned} \tag{23}$$

Now, by choosing to set,

$$q=\frac{2+\mu^2}{\mu} \tag{24}$$

Eq.(23) reduces to,

$$\dot{\mathcal{E}}_t=-\mu\left(1+h\right)\left(X^2+Y^2\right)-\mu\left(\dot{X}^2+\dot{Y}^2\right)\leq 0\ \ \forall\, t \tag{25}$$

as both μ and $h>0$. Specifically, we observe that for $(X,Y)\neq(0,0)$,

$$\begin{aligned}
&\dot{\mathcal{E}}_t<0,\\
&\therefore\ \mathcal{E}_t\to 0,\ \text{as } t\to\infty,\\
&\text{hence},\ X^2+Y^2\leq\mathcal{E}_t\to 0,\ \text{as } t\to\infty
\end{aligned} \tag{26}$$

guaranteeing *asymptotic stability* of our algorithm, i.e. convergence to the Nash Equilibrium $(X,Y)=(0,0)$ as $t\to\infty$. With this result in hand, let us now consider the following expression,

$$-\rho\mathcal{E}_t-\frac{\rho\mu}{2}\left(X-\dot{X}\right)^2-\frac{\rho\mu}{2}\left(Y-\dot{Y}\right)^2-\frac{\rho\mu}{2}\left(\dot{X}-Y\right)^2-\frac{\rho\mu}{2}\left(X+\dot{Y}\right)^2 \tag{27}$$

where ρ is a constant and is determined as,

$$0\leq\rho\leq\min\left\{\frac{\mu\left(1+h\right)}{\mu^2+\mu+\left(1+h\right)},\ \frac{\mu}{1+\mu}\right\} \tag{28}$$

It can then be checked that on expansion, Eq.(27) becomes,

$$\Rightarrow-\rho\left(\mu^2+\mu+\left(1+h\right)\right)\left(X^2+Y^2\right)-\rho\left(1+\mu\right)\left(\dot{X}^2+\dot{Y}^2\right)\leq-\rho\mathcal{E}_t \tag{29}$$

Now, using Eq.(25) and the condition (28), we can then go on to write,

$$\begin{aligned}\dot{\mathcal{E}}_t &= -\mu\,(1+h)\left(X^2+Y^2\right) - \mu\left(\dot{X}^2+\dot{Y}^2\right)\\ &\leq -\rho\left(\mu^2+\mu+(1+h)\right)\left(X^2+Y^2\right) - \rho\,(1+\mu)\left(\dot{X}^2+\dot{Y}^2\right)\\ &\leq -\rho\mathcal{E}_t\end{aligned} \tag{30}$$

On integrating the above, one then gets,

$$\begin{aligned}\Rightarrow \frac{\mathrm{d}\mathcal{E}_t}{\mathcal{E}_t} &\leq -\rho \mathrm{d}t\\ \Rightarrow \mathcal{E}_t &\leq \mathcal{E}_0\exp\left(-\rho t\right)\\ \Rightarrow (1+h)\left(X^2+Y^2\right) \leq \mathcal{E}_t &\leq \mathcal{E}_0\exp\left(-\rho t\right)\\ \Rightarrow X^2+Y^2 &\leq \frac{\mathcal{E}_0}{1+h}\exp\left(-\rho t\right)\end{aligned} \tag{31}$$

This exhibits that our continuous-time optimizer in the quadratic min-max game (Eq.(21)), is convergent to the Nash equilibrium $(0,0)$ at a linear rate ρ, as determined from Eq.(28). ■

Appendix F. Proof of Theorem 2

Firstly, we recall that the continuous-time EOMs of the quadratic min-max game (4), can be discretized implicitly as:

$$\begin{aligned}x_k - x_{k-1} &= \delta v_k^x\\ y_k - y_{k-1} &= \delta v_k^y\\ v_k^x - v_{k-1}^x &= -q\delta v_k^y - \mu\delta v_k^x - h\delta x_k - \delta y_k\\ v_k^y - v_{k-1}^y &= q\delta v_k^x - \mu\delta v_k^y - h\delta y_k + \delta x_k\end{aligned} \tag{32}$$

We next define our discrete time Lyapunov function to be,

$$\begin{aligned}\mathcal{E}_k &= \frac{1}{2}\left(v_k^x + 2\sqrt{\frac{\sqrt{5}}{3}}\frac{x_k}{\mu} + 2\sqrt{\frac{\sqrt{5}}{3}}\frac{y_k}{\mu}\right)^2 + \frac{1}{2}\left(v_k^y + 2\sqrt{\frac{\sqrt{5}}{3}}\frac{y_k}{\mu} - 2\sqrt{\frac{\sqrt{5}}{3}}\frac{x_k}{\mu}\right)^2\\ &\quad + \frac{1}{2}\left(\left(v_k^x\right)^2 + \left(v_k^y\right)^2\right) + 2\sqrt{5}\left(1+\frac{2h}{\sqrt{5}}\right)\left(x_k^2+y_k^2\right)\\ &\leq \frac{3}{2}\left(\left(v_k^x\right)^2 + \frac{4\sqrt{5}}{3\mu^2}x_k^2 + \frac{4\sqrt{5}}{3\mu^2}y_k^2\right) + \frac{3}{2}\left(\left(v_k^y\right)^2 + \frac{4\sqrt{5}}{3\mu^2}y_k^2 + \frac{4\sqrt{5}}{3\mu^2}x_k^2\right)\\ &\quad + \frac{1}{2}\left(\left(v_k^x\right)^2 + \left(v_k^y\right)^2\right) + 2\sqrt{5}\left(1+\frac{2h}{\sqrt{5}}\right)\left(x_k^2+y_k^2\right)\\ &= 2\left(\left(v_k^x\right)^2 + \left(v_k^y\right)^2\right) + 2\sqrt{5}\left(\frac{2}{\mu^2}+1+\frac{2h}{\sqrt{5}}\right)\left(x_k^2+y_k^2\right)\end{aligned} \tag{33}$$

where we have used the Cauchy–Schwarz inequality (assuming u_j are scalars):

$$\left(\sum_j^n u_j\right)^2 \leq n\left(\sum_j^n u_j^2\right)$$

Therefore,

$$\begin{aligned}\mathcal{E}_k - \mathcal{E}_{k-1} \leq\ & 2\left(2\left(v_k^x - v_{k-1}^x\right)v_k^x - \left(v_k^x - v_{k-1}^x\right)^2\right) + 2\left(2\left(v_k^y - v_{k-1}^y\right)v_k^y - \left(v_k^y - v_{k-1}^y\right)^2\right) \\ & + \sqrt{5}\left(\frac{4}{\mu^2} + 2\left(1 + \frac{2h}{\sqrt{5}}\right)\right)\left(2\left(x_k - x_{k-1}\right)x_k - \left(x_k - x_{k-1}\right)^2\right) \\ & + \sqrt{5}\left(\frac{4}{\mu^2} + 2\left(1 + \frac{2h}{\sqrt{5}}\right)\right)\left(2\left(y_k - y_{k-1}\right)y_k - \left(y_k - y_{k-1}\right)^2\right)\end{aligned} \tag{34}$$

Plugging our implicitly discretized dynamics of Eq.(32) in the above, we then get,

$$\begin{aligned}\mathcal{E}_k - \mathcal{E}_{k-1} \leq\ & -4q\delta v_k^x v_k^y - 4\mu\delta\left(v_k^x\right)^2 - 4h\delta v_k^x x_k - 4\delta y_k v_k^x \\ & - 2\left(q\delta v_k^y + \mu\delta v_k^x + h\delta x_k + \delta y_k\right)^2 \\ & + 4q\delta v_k^x v_k^y - 4\mu\delta\left(v_k^y\right)^2 - 4h\delta v_k^y y_k + 4\delta x_k v_k^y \\ & - 2\left(q\delta v_k^x - \mu\delta v_k^y - h\delta y_k + \delta x_k\right)^2 \\ & + 2\sqrt{5}\left(\frac{4}{\mu^2} + 2\left(1 + \frac{2h}{\sqrt{5}}\right)\right)\delta x_k v_k^x - \sqrt{5}\left(\frac{4}{\mu^2} + 2\left(1 + \frac{2h}{\sqrt{5}}\right)\right)\left(\delta v_k^x\right)^2 \\ & + 2\sqrt{5}\left(\frac{4}{\mu^2} + 2\left(1 + \frac{2h}{\sqrt{5}}\right)\right)\delta y_k v_k^y - \sqrt{5}\left(\frac{4}{\mu^2} + 2\left(1 + \frac{2h}{\sqrt{5}}\right)\right)\left(\delta v_k^y\right)^2\end{aligned} \tag{35}$$

If we now impose the condition $\mu_r\delta \geq 1$, then that allows us to rewrite Eq.(35) as,

$$\begin{aligned}\mu\delta\left(\mathcal{E}_k - \mathcal{E}_{k-1}\right) \leq\ & -4q\mu\delta^2 v_k^x v_k^y - 4\mu^2\delta^2\left(v_k^x\right)^2 - 4h\mu\delta^2 x_k v_k^x - 4\mu\delta^2 y_k v_k^x \\ & - 2\left(q\delta v_k^y + \mu\delta v_k^x + h\delta x_k + \delta y_k\right)^2 \\ & + 4q\mu\delta^2 v_k^x v_k^y - 4\mu^2\delta^2\left(v_k^y\right)^2 - 4h\mu\delta^2 y_k v_k^y + 4\mu\delta^2 x_k v_k^y \\ & - 2\left(q\delta v_k^x - \mu\delta v_k^y - h\delta y_k + \delta x_k\right)^2 \\ & + \sqrt{5}\delta^2\left(\frac{4}{\mu^2} + 2\left(1 + \frac{2h}{\sqrt{5}}\right)\right)\left(2\mu x_k - v_k^x\right)v_k^x \\ & + \sqrt{5}\delta^2\left(\frac{4}{\mu^2} + 2\left(1 + \frac{2h}{\sqrt{5}}\right)\right)\left(2\mu y_k - v_k^y\right)v_k^y\end{aligned} \tag{36}$$

We now further choose to set,

$$q = \sqrt{5}\left(\frac{2 + \mu^2}{\mu}\right) \tag{37}$$

leading the $x_k v_k^x$ and $y_k v_k^y$ terms in Eq.(36) to drop off to leave us with,

$$\begin{aligned}\mu\delta\left(\mathcal{E}_k-\mathcal{E}_{k-1}\right) &\leq \left(-6\mu^2\delta^2-2q^2\delta^2-\sqrt{5}\delta^2\left(\frac{4}{\mu^2}+2+\frac{4h}{\sqrt{5}}\right)\right)\left((v_k^x)^2+\left(v_k^y\right)^2\right)\\ &\quad+8\mu\delta^2\left(x_k v_k^y-y_k v_k^x\right)-2\delta^2\left(1+h^2\right)\left(x_k^2+y_k^2\right)\\ &\leq \left(-6\mu^2\delta^2-2q^2\delta^2-\sqrt{5}\delta^2\left(\frac{4}{\mu^2}+2+\frac{4h}{\sqrt{5}}\right)\right)\left((v_k^x)^2+\left(v_k^y\right)^2\right)\\ &\quad+8\mu\delta^2\left(x_k v_k^y-y_k v_k^x\right)-2\delta^2\left(x_k^2+y_k^2\right)\end{aligned} \tag{38}$$

Next, by adding to the R.H.S of the above, two positive semi-definite terms of the form,

$$\begin{aligned}\left(\delta x_k-4\mu\delta v_k^y\right)^2+\left(\delta y_k+4\mu\delta v_k^x\right)^2 &= \delta^2\left(x_k^2+y_k^2\right)-8\mu\delta^2\left(x_k v_k^y-y_k v_k^x\right)\\ &\quad+16\mu^2\delta^2\left(\left(v_k^y\right)^2+\left(v_k^x\right)^2\right)\end{aligned} \tag{39}$$

give us,

$$\begin{aligned}\mu\delta\left(\mathcal{E}_k-\mathcal{E}_{k-1}\right) &\leq \delta^2\left(10\mu^2-2q^2-\sqrt{5}\left(\frac{4}{\mu^2}+2+\frac{4h}{\sqrt{5}}\right)\right)\left((v_k^x)^2+\left(v_k^y\right)^2\right)\\ &\quad-\delta^2\left(x_k^2+y_k^2\right)\\ &\leq \delta^2\Bigg(10\mu^2-10\left(\frac{4}{\mu^2}+4+\mu^2\right)\\ &\quad-\sqrt{5}\left(\frac{4}{\mu^2}+2+\frac{4h}{\sqrt{5}}\right)\Bigg)\left((v_k^x)^2+\left(v_k^y\right)^2\right)-\delta^2\left(x_k^2+y_k^2\right)\\ &\leq \delta^2\left(-\frac{40}{\mu^2}-40-\frac{4\sqrt{5}}{\mu^2}-2\sqrt{5}-4h\right)\left((v_k^x)^2+\left(v_k^y\right)^2\right)\\ &\quad-\delta^2\left(x_k^2+y_k^2\right)\\ \Rightarrow \mathcal{E}_k-\mathcal{E}_{k-1} &\leq -\frac{\delta}{\mu}\left(4\sqrt{5}\left(\frac{2\sqrt{5}+1}{\mu^2}\right)+2\sqrt{5}\left(4\sqrt{5}+1\right)+4h\right)\left((v_k^x)^2+\left(v_k^y\right)^2\right)\\ &\quad-\frac{\delta}{\mu}\left(x_k^2+y_k^2\right)\\ &\leq -\frac{\delta}{\mu}\left[\left((v_k^x)^2+\left(v_k^y\right)^2\right)+\left(x_k^2+y_k^2\right)\right]\end{aligned} \tag{40}$$

Multiplying both sides of the above expression by,

$$\sqrt{5}\left(\frac{4}{\mu^2}+2+\frac{4h}{\sqrt{5}}\right) \tag{41}$$

we get,

$$\begin{aligned}\sqrt{5}\left(\frac{4}{\mu^2}+2+\frac{4h}{\sqrt{5}}\right)\left(\mathcal{E}_k-\mathcal{E}_{k-1}\right) &\leq -\frac{\sqrt{5}\delta}{\mu}\left(\frac{4}{\mu^2}+2+\frac{4h}{\sqrt{5}}\right)\left(\left(v_k^x\right)^2+\left(v_k^y\right)^2\right)\\ &\quad -\frac{\sqrt{5}\delta}{\mu}\left(\frac{4}{\mu^2}+2+\frac{4h}{\sqrt{5}}\right)\left(x_k^2+y_k^2\right)\\ &\leq -\frac{\delta}{\mu}\Bigg[2\left(\left(v_k^x\right)^2+\left(v_k^y\right)^2\right)\\ &\quad +\sqrt{5}\left(\frac{4}{\mu^2}+2+\frac{4h}{\sqrt{5}}\right)\left(x_k^2+y_k^2\right)\Bigg]\\ &\leq -\frac{\delta}{\mu}\mathcal{E}_k\end{aligned} \tag{42}$$

Therefore, we finally have from Eq.(42),

$$\mathcal{E}_k \leq \left(1-\frac{\delta\mu}{\mu^2(2\sqrt{5}+4h)+\delta\mu+4\sqrt{5}}\right)\mathcal{E}_{k-1} \tag{43}$$

Therefore, the rate of convergence of iLEAD in the quadratic min-max game is given by,

$$\begin{aligned}\mathcal{E}_k &\leq \left(1-\frac{\delta\mu}{\mu^2(2\sqrt{5}+4h)+\delta\mu+4\sqrt{5}}\right)^k\mathcal{E}_0\\ \Rightarrow x_k^2+y_k^2 &\leq \frac{1}{\sqrt{5}}\left(2+\frac{4h}{\sqrt{5}}\right)^{-1}\left(1-\frac{\delta\mu}{\mu^2(2\sqrt{5}+4h)+\delta\mu+4\sqrt{5}}\right)^k\mathcal{E}_0\\ &\leq \frac{\mu^2}{\mathcal{C}-4\sqrt{5}}\left(\frac{\mathcal{C}}{\mathcal{C}+\delta\mu}\right)^k\mathcal{E}_0\end{aligned} \tag{44}$$

where $\mathcal{C}=\mu^2\left(2\sqrt{5}+4h\right)+4\sqrt{5}$. This, completes our Proof on the convergence of iLEAD in the quadratic min-max setting, to the Nash equilibrium $(0,0)$ as $k\to\infty$.

Appendix G. Lyapunov stability of the general bilinear game (Continuous time)

Proof In the following, we consider a general bilinear game of the form, $f\left(\mathbf{X},\mathbf{Y}\right)=\mathbf{X}^T\mathbb{A}\mathbf{Y}$, where $\mathbf{X}\equiv\left(X^1,\cdots,X^n\right)$, $\mathbf{Y}\equiv\left(Y^1,\cdots,Y^n\right)$ are taken to be equi-dimensional parameter vectors, with $\mathbb{A}_{n\times n}$ being a constant positive-definite matrix. This is the vectorial generalization of Eq.(4) with h set to 0. For such a game, we can correspondingly generalize our continuous-time EOMs of Eq.(1) as,

$$\begin{aligned}\ddot{\mathbf{X}} &= -\mu\dot{\mathbf{X}}-\mathbb{A}\mathbf{Y}-q\mathbb{A}\dot{\mathbf{Y}}\\ \ddot{\mathbf{Y}} &= -\mu\dot{\mathbf{Y}}+\mathbb{A}^T\mathbf{X}+q\mathbb{A}^T\dot{\mathbf{X}}\end{aligned} \tag{45}$$

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

$$\begin{aligned}\mathcal{E}_t &= \frac{1}{2}\left(\dot{\mathbf{X}} + \mu\mathbf{X} + \mu\mathbb{A}\mathbf{Y}\right)^T\left(\dot{\mathbf{X}} + \mu\mathbf{X} + \mu\mathbb{A}\mathbf{Y}\right) \\ &+ \frac{1}{2}\left(\dot{\mathbf{Y}} + \mu\mathbf{Y} - \mu\mathbb{A}^T\mathbf{X}\right)^T\left(\dot{\mathbf{X}} + \mu\mathbf{Y} - \mu\mathbb{A}^T\mathbf{X}\right) \\ &+ \frac{1}{2}\left(\dot{\mathbf{X}}^T\dot{\mathbf{X}} + \dot{\mathbf{Y}}^T\dot{\mathbf{Y}}\right) + \mathbf{X}^T\mathbb{A}\mathbb{A}^T\mathbf{X} + \mathbf{Y}^T\mathbb{A}^T\mathbb{A}\mathbf{Y} \\ &\geq 0\,\forall\, t\end{aligned} \tag{46}$$

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

$$\begin{aligned}\dot{\mathcal{E}}_t &= \left(\dot{\mathbf{X}} + \mu\mathbf{X} + \mu\mathbb{A}\mathbf{Y}\right)^T\left(\ddot{\mathbf{X}} + \mu\dot{\mathbf{X}} + \mu\mathbb{A}\dot{\mathbf{Y}}\right) \\ &\quad + \left(\dot{\mathbf{Y}} + \mu\mathbf{Y} - \mu\mathbb{A}^T\mathbf{X}\right)^T\left(\ddot{\mathbf{Y}} + \mu\dot{\mathbf{Y}} - \mu\mathbb{A}^T\dot{\mathbf{X}}\right) \\ &\quad + \left(\dot{\mathbf{X}}^T\ddot{\mathbf{X}} + \dot{\mathbf{Y}}^T\ddot{\mathbf{Y}}\right) + 2\left(\mathbf{X}^T\mathbb{A}\mathbb{A}^T\dot{\mathbf{X}} + \mathbf{Y}^T\mathbb{A}^T\mathbb{A}\dot{\mathbf{Y}}\right) \\ &= \left(\dot{\mathbf{X}}^T + \mu\mathbf{X}^T + \mu\mathbf{Y}^T\mathbb{A}^T\right)\left((-q+\mu)\,\mathbb{A}\dot{\mathbf{Y}} - \mathbb{A}\mathbf{Y}\right) \\ &\quad + \left(\dot{\mathbf{Y}}^T + \mu\mathbf{Y}^T - \mu\mathbf{X}^T\mathbb{A}\right)\left((q-\mu)\,\mathbb{A}^T\dot{\mathbf{X}} + \mathbb{A}^T\mathbf{X}\right) \\ &\quad + \dot{\mathbf{X}}^T\left(-q\mathbb{A}\dot{\mathbf{Y}} - \mu\dot{\mathbf{X}} - \mathbb{A}\mathbf{Y}\right) + \dot{\mathbf{Y}}^T\left(q\mathbb{A}^T\dot{\mathbf{X}} - \mu\dot{\mathbf{Y}} + \mathbb{A}^T\mathbf{X}\right) \\ &\quad + 2\left(\mathbf{X}^T\mathbb{A}\mathbb{A}^T\dot{\mathbf{X}} + \mathbf{Y}^T\mathbb{A}^T\mathbb{A}\dot{\mathbf{Y}}\right) \\ &= \left(\mu\,(q-\mu) - 2\right)\left(\mathbf{Y}^T\mathbb{A}^T\dot{\mathbf{X}} - \mathbf{X}^T\mathbb{A}\dot{\mathbf{Y}}\right) \\ &\quad - \left(\mu\,(q-\mu) - 2\right)\left(\mathbf{X}^T\mathbb{A}\mathbb{A}^T\dot{\mathbf{X}} + \mathbf{Y}^T\mathbb{A}^T\mathbb{A}\dot{\mathbf{Y}}\right) \\ &\quad - \mu\left(\mathbf{X}^T\mathbb{A}\mathbb{A}^T\mathbf{X} + \mathbf{Y}^T\mathbb{A}^T\mathbb{A}\mathbf{Y}\right) - \mu\left(\dot{\mathbf{X}}^T\dot{\mathbf{X}} + \dot{\mathbf{Y}}^T\dot{\mathbf{Y}}\right)\end{aligned} \tag{47}$$

where we have used the fact that $\mathbf{X}^T\mathbb{A}\mathbf{Y}$ being a scalar, implies $\mathbf{X}^T\mathbb{A}\mathbf{Y} = \mathbf{Y}^T\mathbb{A}^T\mathbf{X}$. Setting $q = (2/\mu) + \mu$ in the above, then leads to,

$$\begin{aligned}\dot{\mathcal{E}}_t &= -\mu\left(\mathbf{X}^T\mathbb{A}\mathbb{A}^T\mathbf{X} + \mathbf{Y}^T\mathbb{A}^T\mathbb{A}\mathbf{Y}\right) - \mu\left(\dot{\mathbf{X}}^T\dot{\mathbf{X}} + \dot{\mathbf{Y}}^T\dot{\mathbf{Y}}\right) \\ &= -\mu\left(\left|\left|\mathbb{A}^T\mathbf{X}\right|\right|^2 + \left|\left|\mathbb{A}\mathbf{Y}\right|\right|^2\right) - \mu\left(\left|\left|\dot{\mathbf{X}}\right|\right|^2 + \left|\left|\dot{\mathbf{Y}}\right|\right|^2\right) \leq 0\,\forall\, t\end{aligned} \tag{48}$$

exhibiting that the Lyapunov function, Eq.(46) is *asymptotically stable* at all times t. ■

Appendix H. Rate of convergence of the general bilinear game (Continuous time)

Proof Consider the following expression, where $\mathcal{E}_t$ is our general bilinear game Lyapunov function Eq.(46) :

$$\begin{aligned}
&-\rho\mathcal{E}_t - \frac{\rho\mu}{2}\left|\left|\mathbf{X}-\dot{\mathbf{X}}\right|\right|^2 - \frac{\rho\mu}{2}\left|\left|\mathbf{Y}-\dot{\mathbf{Y}}\right|\right|^2 - \frac{\rho\mu}{2}\left|\left|\dot{\mathbf{X}}-\mathbb{A}\mathbf{Y}\right|\right|^2 - \frac{\rho\mu}{2}\left|\left|\mathbb{A}^T\mathbf{X}+\dot{\mathbf{Y}}\right|\right|^2 \\
&\quad = -\rho\mathcal{E}_t - \frac{\rho\mu}{2}\left(||\mathbf{X}||^2+||\mathbf{Y}||^2\right) + \rho\mu\left(\mathbf{X}^T\dot{\mathbf{X}}+\mathbf{Y}^T\dot{\mathbf{Y}}\right) - \rho\mu\left(\left|\left|\dot{\mathbf{X}}\right|\right|^2+||\mathbf{Y}||^2\right) \\
&\qquad - \rho\mu\left(\mathbf{X}^T\mathbb{A}\dot{\mathbf{Y}}-\dot{\mathbf{X}}^T\mathbb{A}\mathbf{Y}\right) - \frac{\rho\mu}{2}\left(\left|\left|\mathbb{A}^T\mathbf{X}\right|\right|^2+||\mathbb{A}\mathbf{Y}||^2\right) \\
&\quad = -\rho\left(1+\mu\right)\left(\left|\left|\dot{\mathbf{X}}\right|\right|^2+\left|\left|\dot{\mathbf{Y}}\right|\right|^2\right) - \frac{\rho}{2}\left(\mu^2+\mu\right)\left(||\mathbf{X}||^2+||\mathbf{Y}||^2\right) \\
&\qquad - \frac{\rho}{2}\left(\mu^2+\mu+2\right)\left(\left|\left|\mathbb{A}^T\mathbf{X}\right|\right|^2+||\mathbb{A}\mathbf{Y}||^2\right) \\
&\quad \leq -\rho\mathcal{E}_t
\end{aligned} \tag{49}$$

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 (s_i) of $\mathbb{A}$. Using this decomposition in Eq.(49), allows us to write,

$$\begin{aligned}
&-\rho\left(1+\mu\right)\left(\left|\left|\dot{\mathbf{X}}\right|\right|^2+\left|\left|\dot{\mathbf{Y}}\right|\right|^2\right) - \frac{\rho}{2}\left(\mu^2+\mu\right)\left(||\mathbf{X}||^2+||\mathbf{Y}||^2\right) \\
&\qquad - \frac{\rho}{2}\left(\mu^2+\mu+2\right)\left(\left|\left|\mathbb{A}^T\mathbf{X}\right|\right|^2+||\mathbb{A}\mathbf{Y}||^2\right) \\
&\quad = -\rho\left(1+\mu\right)\left(\left|\left|\mathbb{V}\dot{\mathbf{X}}\right|\right|^2+\left|\left|\mathbb{U}\dot{\mathbf{Y}}\right|\right|^2\right) - \frac{\rho}{2}\left(\mu^2+\mu\right)\left(||\mathbb{V}\mathbf{X}||^2+||\mathbb{U}\mathbf{Y}||^2\right) \\
&\qquad - \frac{\rho}{2}\left(\mu^2+\mu+2\right)\left(||\mathbb{S}\mathbb{V}\mathbf{X}||^2+||\mathbb{S}\mathbb{U}\mathbf{Y}||^2\right) \\
&\quad = -\rho\left(1+\mu\right)\left(\left|\left|\dot{\mathcal{X}}\right|\right|^2+\left|\left|\dot{\mathcal{Y}}\right|\right|^2\right) - \frac{\rho}{2}\left(\mu^2+\mu\right)\left(||\mathcal{X}||^2+||\mathcal{Y}||^2\right) \\
&\qquad - \frac{\rho}{2}\left(\mu^2+\mu+2\right)\left(||\mathbb{S}\mathcal{X}||^2+||\mathbb{S}\mathcal{Y}||^2\right) \\
&\quad = -\sum_{j=1}^{n}\rho\left(1+\mu\right)\left(\left|\left|\dot{\mathcal{X}}^j\right|\right|^2+\left|\left|\dot{\mathcal{Y}}^j\right|\right|^2\right) \\
&\qquad - \sum_{j=1}^{n}\frac{\rho}{2}\left(\left(1+s_j^2\right)\left(\mu^2+\mu\right)+2s_j^2\right)\left(\left|\left|\mathcal{X}^j\right|\right|^2+\left|\left|\mathcal{Y}^j\right|\right|^2\right)
\end{aligned} \tag{50}$$

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 $\mathcal{X} = \mathbb{V}\mathbf{X}$ and $\mathcal{Y} = \mathbb{U}\mathbf{Y}$. Now, we know from Eq.(48) that,

$$\begin{aligned}
\dot{\mathcal{E}}_t &= -\mu\left(\left|\left|\mathbb{A}^T X\right|\right|^2 + ||\mathbb{A}Y||^2\right) - \mu\left(\left|\left|\dot{X}\right|\right|^2 + \left|\left|\dot{Y}\right|\right|^2\right) \\
&= -\mu\left(\left|\left|\mathbb{U}^{\mathrm{T}}\mathbb{S}\mathbb{V}X\right|\right|^2 + ||\mathbb{V}^{\mathrm{T}}\mathbb{S}\mathbb{U}Y||^2\right) - \mu\left(\left|\left|\mathbb{V}\dot{X}\right|\right|^2 + \left|\left|\mathbb{U}\dot{Y}\right|\right|^2\right) \\
&= -\mu\left(||\mathbb{S}\mathcal{X}||^2 + ||\mathbb{S}\mathcal{Y}||^2\right) - \mu\left(\left|\left|\dot{\mathcal{X}}\right|\right|^2 + \left|\left|\dot{\mathcal{Y}}\right|\right|^2\right) \\
&= -\sum_{j=1}^{n} \mu s_j^2\left(\left|\left|\mathcal{X}^j\right|\right|^2 + \left|\left|\mathcal{Y}^j\right|\right|^2\right) - \sum_{j=1}^{n} \mu_r\left(\left|\left|\dot{\mathcal{X}}^j\right|\right|^2 + \left|\left|\dot{\mathcal{Y}}^j\right|\right|^2\right)
\end{aligned} \tag{51}$$

Comparing the above expression with Eq.(50), we note that a choice of ρ as,

$$\rho \leq \min\left\{\frac{\mu}{1+\mu},\ \frac{2\mu s_j^2}{\left(1+s_j^2\right)\left(\mu^2+\mu\right)+2s_j^2}\right\}\ \forall\, j \in [1,n] \tag{52}$$

implies,

$$\begin{aligned}
\dot{\mathcal{E}}_t &\leq -\rho\mathcal{E} \\
\Rightarrow \mathcal{E}_t &\leq \mathcal{E}_0 \exp\left(-\rho t\right) \\
\Rightarrow X^T\mathbb{A}\mathbb{A}^T X + Y^T\mathbb{A}^T\mathbb{A}Y &\leq \mathcal{E}_0 \exp\left(-\rho t\right) \\
\Rightarrow \mathcal{X}^T\mathbb{S}^2\mathcal{X} + \mathcal{Y}^T\mathbb{S}^2\mathcal{Y} &\leq \mathcal{E}_0 \exp\left(-\rho t\right) \\
\Rightarrow \sum_{j=1}^{n} s_j^2\left(\mathcal{X}^j + \mathcal{Y}^j\right) &\leq \mathcal{E}_0 \exp\left(-\rho t\right) \\
\therefore\ \mathcal{X}^j + \mathcal{Y}^j &\leq \frac{\mathcal{E}_0}{s_j^2}\exp\left(-\rho\right)\forall\, j
\end{aligned} \tag{53}$$

■

This completes our Proof of convergence (continuous-time) of the generalized iLEAD (Eq.(45)) in the general bilinear game $f\left(\mathbf{X},\mathbf{Y}\right) = \mathbf{X}^T\mathbb{A}\mathbf{Y}$. It is to be noted, that the (continuous-time) fastest rate of convergence ρ, in the case of the above general bilinear game $f\left(\mathbf{X},\mathbf{Y}\right) = \mathbf{X}^T\mathbb{A}\mathbf{Y}$, is determined by the smallest eigenvalue s_j of $\mathbb{A}$, (Eq.(52)).

Appendix I. Experiments Details and implementation

In this Section, we refer to the reproducible code for all the plots that are available in the paper.

I.1. Mixture of Eight Gaussians Experiment

Dataset The real data is generated by 8-Gaussian distributions that are uniformly distributed around the unit circle. The code to generate the data is included in the source code related to Figure 1 (Left).

	Adam	**sLEAD**	WGP	OMD	WGP+Ex+OMD	CO	Extra+Adam	CGD
IS	6.24	**7.1**	6.5	5.74	7.3	7.1	6.38	7.2

Table 1: Comparison of several first order and second order methods in terms of the Inception Score (IS). We report published results on Optimistic Mirror Descent (reported from [9], Table 1), WGAN+GP+Extra-gradient+OMD (reported from [24]), Consensus Optimization (reported from [25]), Extra(gradient)+Adam (from [9]) and Competative Gradient Descent (from [29]).

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.

Other Details We use the Adam [16] optimizer on top of our algorithm in the reported results. Furthermore, we use batchsize of 128.

I.2. CIFAR 10

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.

Other Details For the baseline we use Adam with the first moment set to 0.5 and second moment set to 0.99. Generator's learning rate is 0.0002 and discriminator's learning rate is 0.0001. The same learning rate and momentum were used to train sLEAD model. We also add the mixed derivative term with $\alpha_d = 0.3$ and $\alpha_g = 0.0$.

The baseline is a DCGAN with non-saturating loss (non-zero sum formulation). In our experiments, we calculate the inception score using TensorFlow's inception model on 8000 generated samples and reported the mean and the variance of the inception score over those samples.

Inception Score The performance of generative models is often computed and compared using the Inception Score[28] and the FID. However, it has been shown that inception score is not a reliable metric to evaluate the performance of generative models [3]. Despite that, we provide a comparison of the inception score in table I.2 for the DCGAN architecture described above.

7. For FtR, we have provided the update for the second player given the first player performs gradient descent on f.

		Coefficient	Momentum	Gradient	Interaction-xy	Interaction-xx
GDA	$\Delta x_{k+1} =$	1	0	$-\eta\nabla_x f$	$-\eta\nabla_x f$	0
sLEAD	$\mathbf{\Delta x_{k+1}} =$	**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[2]	$\Delta x_{k+1} =$	1	0	$-\eta\nabla_x f$	$-\eta\gamma\nabla^2_{xy} f\nabla_y f$	0
CGD[29]	$\Delta x_{k+1} =$	$\mathcal{C}^{-1}$	0	$-\eta\nabla_x f$	$-\eta^2\nabla^2_{xy} f\nabla_y f$	0
CO[25]	$\Delta x_{k+1} =$	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[32]	$\Delta y_{k+1} =$	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[7]	$\Delta x_{k+1} =$	1	0	$-\eta\nabla_x f$	$-2\eta\alpha\nabla_{xy} f\nabla_y f$	0

Table 2: Comparison of several second-order methods in game 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, $\Delta x_k = x_k - x_{k-1}$, 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[7] for the following methods: Gradient Descent-Ascent (GDA), symplectic Least Action Dynamics (sLEAD, 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.