# Cubic Regularization for Differentiable Games

**Shuang Li, Youye Xie, Qiuwei Li, and Gongguo Tang**
Department of Electrical Engineering
Colorado School of Mines
Golden, CO 80401
{shuangli,youyexie,qiuli,gtang}@mines.edu

## Abstract

Due to the non-symmetric nature of the game Jacobian, many first-order methods used to find Nash equilibria for differentiable games, such as vanilla gradient descent and its variants, often perform poorly and exhibit relatively-slow convergence, oscillation around equilibria, or even divergence. Inspired by the close connection between differentiable games and classical nonlinear optimization, the latter of which has an array of well-established principles and methods in algorithmic design, this paper develops a new algorithm for smooth games based on the cubic-regularization method. The algorithm approximates each player's cost function as a quadratic function regularized by a cubic term, an idea originally proposed by Nesterov and Polyak as a theoretically guaranteed extension of Newton's method. The fixed points of the proposed algorithm are shown to coincide with second-order Nash equilibria of the given game. For two-player zero-sum games, these fixed points are also stable under the gradient flow dynamic. The theoretical findings are supported by numerical experiments.

## 1 Introduction

*Differentiable games* have been widely used and studied in signal processing [2] and wireless communication networks [9] as a powerful modeling framework. Recent literature has also reported several significant successes in training machine learning models with artificially constructed differentiable games, such as generative adversarial networks (GAN) [8], reinforcement learning [20], adversarial training [15], intrinsic curiosity [21], imaginative agents [22], and so on. Differentiable games are closely related to classical nonlinear optimization problems. The presence of multiple cost functions in games is reminiscent of multiple objective functions in vector optimization [6], while the partial control of the strategy profile shares similarity with distributed optimization [4]. When all players share the same cost function and control all variables, a differentiable game reduces to a centralized nonlinear optimization [3, 19]. Motived by this close connection, we focus on the investigation of differentiable games from an optimization perspective and generalize the methods used in nonlinear optimization to differentiable games.

The wide application of nonlinear optimization in science and engineering owes in part to the plethora of numerical algorithms developed by the optimization community [19]. Even simple local search algorithms such as gradient descent and its varaints have superior practical performance and solid theoretical ground [11, 12, 13, 14]. This is in sharp contrast to algorithms used in game optimization. For example, vanilla gradient descent and many of its variants can very often lead to limit oscillatory behavior, rather than converge to an equilibrium [16, 17]. This is a consequence of the non-symmetric nature of the game Jacobian and the complex optimization geometry it entails. Nonetheless, optimization literature provides an array of principles, ideas, and methods in the design of numerical procedures. Some of the most effective algorithms for classical optimization compute next iterate by minimizing a suitable model of the objective function. The model is usually formed

Abstract for: Smooth Games Optimization and Machine Learning Workshop (NeurIPS 2019), Vancouver, Canada.

using function and derivative information at the current iterate. Exemplar algorithms obtained in this manner include gradient descent, the Frank-Wolfe algorithm [7], (quasi-)Newton methods [19], and Nesterov-Polyak's cubic regularization [18].

**Contribution** Inspired by the Nesterov-Polyak's cubic regularization used in nonlinear programs, we develop a game optimization algorithm based on the cubic regularization to resolve the previously mentioned oscillatory and non-convergent issues of some existing algorithms in this work. In particular, in order to find Nash equilibria, we solve a sequence of iteratively constructed simple optimization or game subproblems. We demonstrate that the fixed points of the proposed algorithm coincide with second-order Nash equilibria of the given game. For two-player zero-sum games, these fixed points are also stable under the gradient flow dynamic.

## 2 Preliminaries

In this section, we present some background on differentiable games, which are games with differentiable cost functions and continuous decision variables.

Define $[K] \doteq \{1, \ldots, K\}$ as the set of positive integers up to $K$. For a $K$-player differentiable, non-cooperative game, we assume that each player $k$ has a strategy set $X_k \subset \mathbb{R}^{p_k}$ and a differentiable cost function $f_k : X_1 \times X_2 \times \cdots \times X_K \to \mathbb{R}$. Denote $X = X_1 \times X_2 \times \cdots \times X_K \subset \mathbb{R}^p$ with $p = \sum_{k=1}^{K} p_k$ and $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_K) \in X$ as the direct product of the players' strategy sets and the strategy profile, respectively. Here, each $\mathbf{x}_k \in X_k$ denotes the strategy or decision of player $k$. Define $\mathbf{x}_{-k} \doteq (\mathbf{x}_1, \ldots, \mathbf{x}_{k-1}, \mathbf{x}_{k+1}, \ldots, \mathbf{x}_k)$ as a set that contains all players' strategies except that of player $k$. We frequently abuse notation to write $\mathbf{x} = (\mathbf{x}_k, \mathbf{x}_{-k})$ and $f_k(\mathbf{x}) = f_k(\mathbf{x}_k, \mathbf{x}_{-k})$. We also assume that all the $K$ players are rational, that is, each player aims at minimizing its own cost function $f_k(\mathbf{x}_k, \mathbf{x}_{-k})$ by controlling its own optimization variable $\mathbf{x}_k$. In addition, each player explicitly knows that its own cost function $f_k(\mathbf{x}_k, \mathbf{x}_{-k})$ is impacted by the other players' strategies $\mathbf{x}_{-k}$. A $K$-player differentiable game reduces to a two-player zero-sum game when $K = 2$ and $f_1(\mathbf{x}_1, \mathbf{x}_2) = -f_2(\mathbf{x}_1, \mathbf{x}_2) = f(\mathbf{x}_1, \mathbf{x}_2)$.

A *Nash equilibrium* is a strategy profile $\mathbf{x}^\star = (\mathbf{x}_k^\star, \mathbf{x}_{-k}^\star) \in X$ such that

$$f_k(\mathbf{x}_k^\star, \mathbf{x}_{-k}^\star) \le f_k(\mathbf{x}_k, \mathbf{x}_{-k}^\star),\ \forall\, \mathbf{x}_k \in X_k \text{ for all } k \in [K],$$

or equivalently, $\mathbf{x}_k^\star$ solves the following optimization problem

$$\underset{\mathbf{x}_k \in X_k}{\text{minimize}}\ f_k(\mathbf{x}_k, \mathbf{x}_{-k}^\star) \text{ for all } k \in [K].$$

Nash equilibria are therefore pure strategy profiles of the game from which no player can do better by unilaterally changing its strategy. A strategy profile $\mathbf{x}^\star = (\mathbf{x}_k^\star, \mathbf{x}_{-k}^\star) \in X$ is a *local Nash equilibrium* [23, 10] if there exist open sets $W_k \subset X_k$ containing $\mathbf{x}_k$ and for each $k \in [K]$, one has

$$f_k(\mathbf{x}_k^\star, \mathbf{x}_{-k}^\star) \le f_k(\mathbf{x}_k, \mathbf{x}_{-k}^\star),\ \text{for all } \mathbf{x}_k \in W_k.$$

Denote $\nabla_k f_k(\mathbf{x})$ as the gradient of player $k$'s cost function $f_k(\mathbf{x}_k, \mathbf{x}_{-k})$ with respect to $\mathbf{x}_k$. The *game gradient* is then defined as

$$\mathbf{g}(\mathbf{x}) \doteq \left[ (\nabla_1 f_1(\mathbf{x}))^\top\ (\nabla_2 f_2(\mathbf{x}))^\top\ \cdots\ (\nabla_K f_K(\mathbf{x}))^\top \right]^\top \in \mathbb{R}^p.$$

Denote $\mathbf{J}_{kl}(\mathbf{x}) \doteq \nabla_{kl}^2 f_k(\mathbf{x})$ as the $(p_k \times p_l)$-block submatrix of the Hessian of $f_k(\cdot)$. The *game Jacobian* is then defined as

$$\mathbf{J}(\mathbf{x}) \doteq [\nabla_{kl}^2 f_k(\mathbf{x})]_{k,l=1}^{K} \in \mathbb{R}^{p \times p},$$

which is generally non-symmetric.

For an unconstrained game, a point $\mathbf{x}^\star$ satisfies the first-order optimality condition $\mathbf{g}(\mathbf{x}^\star) = \mathbf{0}$ is called a *first-order Nash critical point* of the game, and a point $\mathbf{x}^\star$ satisfies the additional second-order optimality condition $\mathbf{J}_{kk}(\mathbf{x}^\star) \succeq \mathbf{0},\ k \in [K]$ is then called a *second-order Nash equilibrium* [23]. In addition, if a second-order Nash equilibrium $\mathbf{x}^\star$ further satisfies that $\mathbf{J}_{kk}(\mathbf{x}^\star) \succ \mathbf{0},\ k \in [K]$, it is also a local Nash equilibrium.

## 3 Cubic-Regularization Based Methods

First-order methods relying on the game gradient, while being favored in large-scale machine learning applications, also exhibit many deficiencies in game context, including but not limited to relatively-slow convergence, oscillation around equilibria or even divergence, sensitivity to hyper-parameters, and stagnation at high training errors [24]. We develop a new algorithm based on Nesterov-Polyak's cubic regularization [18] that builds local approximate models using the second-order derivative information encoded in the game Jacobian and show that its fixed points coincide with the second-order Nash equilibria. For two-player zero-sum games, these fixed points are also stable under the gradient flow dynamic.

### 3.1 The proposed algorithm

Inspired by the Nesterov-Polyak's cubic regularization [18], we approximate the cost function $f_k(\mathbf{x}^t)$ for the $k$th player around iterate $\mathbf{x}^t$ as

$$f_k^{\mathrm{CR}}(\boldsymbol{\delta}_k;\mathbf{x}^t) \doteq f_k(\mathbf{x}^t) + \boldsymbol{\delta}_k^\top \nabla_k f_k(\mathbf{x}^t) + \frac{1}{2}\boldsymbol{\delta}_k^\top \nabla_{kk}^2 f_k(\mathbf{x}^t)\boldsymbol{\delta}_k + \frac{\rho_k}{3}\|\boldsymbol{\delta}_k\|_2^3$$

for some $\rho_k > 0$. Here, the superscript $^t$ denotes quantities evaluated at the current iterate $\mathbf{x}^t$. Note that $f_k^{\mathrm{CR}}(\boldsymbol{\delta}_k;\mathbf{x}^t)$ provides an upper bound on $f_k(\mathbf{x}_k^t+\boldsymbol{\delta}_k,\mathbf{x}_{-k}^t)$ as a function of $\boldsymbol{\delta}_k$ when $\rho_k$ is greater than the Lipschitz constant of the (partial) Hessian $\mathbf{J}_{kk}$ of the cost function $f_k(\cdot)$. After finding the minimizer $\boldsymbol{\delta}_k^t = T_k^{\mathrm{CR}}(\mathbf{x}^t)$ for each approximation $f_k^{\mathrm{CR}}(\boldsymbol{\delta}_k;\mathbf{x}^t)$, one then performs the update

$$\mathbf{x}_k^{t+1} = \mathbf{x}_k^t + \boldsymbol{\delta}_k^t = \mathbf{x}_k^t + T_k^{\mathrm{CR}}(\mathbf{x}^t), \quad k \in [K]. \tag{1}$$

### 3.2 Stability analysis

Next, we analyze the stability of our proposed algorithm.

**Theorem 3.1.** $\mathbf{x}^\star$ *is a fixed point of the proposed method* (1) *if and only if it is a second-order Nash equilibrium satisfying the following optimality conditions*

$$\mathbf{g}(\mathbf{x}^\star) = \mathbf{0}, \quad \textit{and} \quad \mathbf{J}_{kk}(\mathbf{x}^\star) \succeq \mathbf{0},\ k \in [K]. \tag{2}$$

*Proof.* Denote $\mathbf{J}_{kk}^\star = \mathbf{J}_{kk}(\mathbf{x}^\star)$ and $\mathbf{g}_k^\star = \mathbf{g}_k(\mathbf{x}^\star)$. Note the characterization in [18, 5]: $\boldsymbol{\delta}_k^\star$ is a global minimizer of $f_k^{\mathrm{CR}}(\boldsymbol{\delta}_k;\mathbf{x}^\star)$ if and only if

$$\nabla f_k^{\mathrm{CR}}(\boldsymbol{\delta}_k^\star;\mathbf{x}^t) = (\mathbf{J}_{kk}^\star + \rho_k\|\boldsymbol{\delta}_k^\star\|_2)\boldsymbol{\delta}_k^\star + \mathbf{g}_k^\star = \mathbf{0}, \quad \text{and} \quad \mathbf{J}_{kk}^\star + \rho_k\|\boldsymbol{\delta}_k^\star\|_2\mathbf{I} \succeq \mathbf{0},\ k \in [K]. \tag{3}$$

In addition, $\boldsymbol{\delta}_k^\star$ is unique whenever $\mathbf{J}_{kk}^\star + \rho_k\|\boldsymbol{\delta}_k^\star\|_2\mathbf{I} \succ \mathbf{0}$. On one hand, if $\mathbf{x}^\star$ satisfies (2), then clearly $\nabla f_k^{\mathrm{CR}}(\mathbf{0};\mathbf{x}^\star) = \mathbf{0}$ and $\mathbf{J}_{kk}^\star + \rho_k\|\mathbf{0}\|_2\mathbf{I} = \mathbf{J}_{kk}^\star \succeq \mathbf{0}$, so (3) is satisfied by the solution $\boldsymbol{\delta}_k^\star = T_k^{\mathrm{CR}}(\mathbf{x}^\star) = \mathbf{0}$, implying that $\mathbf{x}^\star$ is a fixed point of (1). On the other hand, if $\mathbf{x}^\star$ is a fixed point of (1), that is, $\boldsymbol{\delta}_k^\star = T_k^{\mathrm{CR}}(\mathbf{x}^\star) = \mathbf{0}$ is a solution of (3), by plugging in $\boldsymbol{\delta}_k^\star = \mathbf{0}$ into (3) we have $\mathbf{g}_k^\star = \mathbf{0}$ and $\mathbf{J}_{kk}^\star \succeq \mathbf{0}$ for $k \in [K]$, implying that $\mathbf{x}^\star$ is a second-order Nash equilibrium. □

**Theorem 3.2.** *For two-player zero-sum games, if* $\mathbf{x}^\star$ *is a fixed point of the proposed method* (1) *with* $\mathbf{J}_{kk}^\star \succ \mathbf{0}$*,* $k \in \{1,2\}$*, then it is also a locally stable equilibrium under the gradient flow dynamic, namely, all the eigenvalues of the game Jacobian* $\mathbf{J}(\mathbf{x}^\star)$ *have strictly positive real parts.*

*Proof.* By definition, it suffices to show that all the eigenvalues of the game Jacobian $\mathbf{J}^\star \doteq \mathbf{J}(\mathbf{x}^\star)$ have strictly positive real parts. For any $\lambda$ being an eigenvalue of $\mathbf{J}^\star$, let $\mathbf{v}$ be the corresponding eigenvector. We observe that

$$\mathbf{v}^H\mathbf{J}^\star\mathbf{v} = \lambda \iff (\mathbf{v}^H\mathbf{J}^\star\mathbf{v})^H = \lambda^H \iff \mathbf{v}^H\mathbf{J}^{\star H}\mathbf{v} = \overline{\lambda},$$

where $(\cdot)^H$ and $\overline{(\cdot)}$ denote the conjugate transpose and conjugate operations, respectively. Therefore,

$$\mathrm{Re}(\lambda) = \frac{\lambda+\overline{\lambda}}{2} = \mathbf{v}^H\left(\frac{\mathbf{J}^\star + \mathbf{J}^{\star H}}{2}\right)\mathbf{v} = \mathbf{v}^H\begin{bmatrix}\mathbf{J}_{11}^\star & \\ & \mathbf{J}_{22}^\star\end{bmatrix}\mathbf{v} > 0,$$

where we have used the special structure of the game Jacobian in two-player zero-sum games $\mathbf{J}^\star = \begin{bmatrix}\mathbf{J}_{11}^\star & \mathbf{J}_{12}^\star \\ \mathbf{J}_{21}^\star & \mathbf{J}_{22}^\star\end{bmatrix} \in \mathbb{R}^{p\times p}$ with $\mathbf{J}_{12}^\star = -\mathbf{J}_{21}^{\star H}$. This completes the proof. □

## 4 Simulations

In this section, we test the proposed method and theory on the following two-player zero-sum game with

$$f_1(x, y) = -f_2(x, y) = f(x, y) = 2x^2 + \frac{1}{2}y^2 - 4xy + \frac{4}{3}y^3 - \frac{1}{4}y^4.$$

First of all, the first-order Nash critical points of this game are given by those points with zero game gradients, i.e.,

$$\mathbf{g}(x, y) = \begin{bmatrix} \nabla_x f_1(x, y) \\ \nabla_y f_2(x, y) \end{bmatrix} = \begin{bmatrix} 4x - 4y \\ -(y - 4x + 4y^2 - y^3) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

which gives that $\mathbf{z}_1 = (0, 0)$, $\mathbf{z}_2 = (1, 1)$, and $\mathbf{z}_3 = (3, 3)$. The game Jacobians evaluated at these points are then

$$\mathbf{J}(\mathbf{z}_1) = \begin{bmatrix} 4 & -4 \\ 4 & -1 \end{bmatrix}, \quad \mathbf{J}(\mathbf{z}_2) = \begin{bmatrix} 4 & -4 \\ 4 & -6 \end{bmatrix}, \text{ and } \quad \mathbf{J}(\mathbf{z}_3) = \begin{bmatrix} 4 & -4 \\ 4 & 2 \end{bmatrix}.$$

Note that $\mathbf{z}_1$ and $\mathbf{z}_3$ are locally stable equilibria since all their eigenvalues have strictly positive real parts, and $\mathbf{z}_3$ is the only second-order Nash equilibrium among these critical points. To show that the proposed cubic regularization based method is capable of locating the second-order Nash equilibrium, we apply it to this two-player zero-sum game and record its trajectory on the landscape of the objective function in Figure 1 (a) with initial point $(3, -1)$. We also implement the gradient descent and the CESP (curvature exploitation for the saddle point problem) method [1] for comparison, both with a step size $0.02$. The distance between the current step ($\mathbf{z}$) and the second-order Nash equilibrium ($\mathbf{z}_3$) is recorded in Figure 1 (b).

Note that gradient descent converges to a locally stable equilibrium $\mathbf{z}_1$ which is not a second-order Nash equilibrium. The CESP method is designed to avoid this case and converge to a second-order Nash equilibrium. Although both the CESP method and cubic regularization method converge to the second-order Nash equilibrium, the proposed cubic regularization method converges faster than the CESP method by a large margin and is much more robust to its hyperparameters.

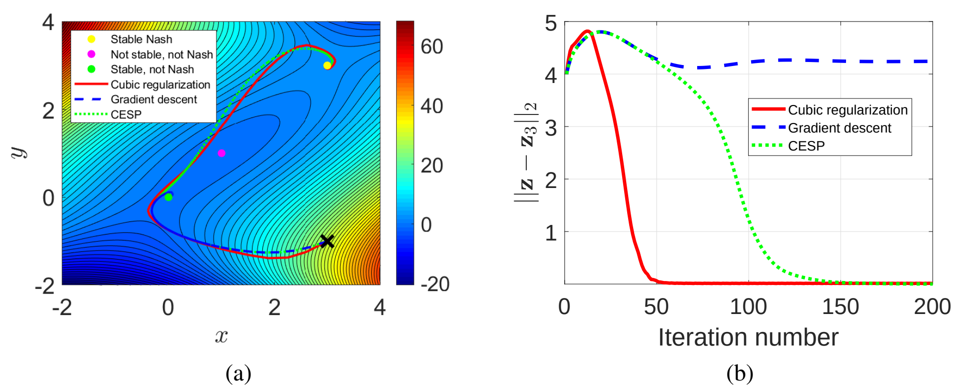


Figure 1: (a) The optimization trajectory of the proposed cubic regularization method, gradient descent, and the CESP method, when initialized at $(3, -1)$. (b) The distance between the current step and the second-order Nash equilibrium.

## 5 Conclusions

In this work, we investigated the computation of local Nash equilibria for differentiable games from an optimization perspective and generalized the methods used in nonlinear optimization to game settings. In particular, we developed a new algorithm based on Nesterov-Polyak's cubic regularization for differentiable games by building local approximations to each player's cost function using the second-order derivatives. We verified both theoretically and numerically that the fixed points of the proposed algorithm coincide second-order Nash equilibria. Additionally, for two-player zero-sum game, a fixed point with positive-definite partial Jacobian is also a locally stable equilibrium.

## References


[1] L. Adolphs, H. Daneshmand, A. Lucchi, and T. Hofmann. Local saddle point optimization: A curvature exploitation approach. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 486–495, 2019.

[2] G. Bacci, S. Lasaulce, W. Saad, and L. Sanguinetti. Game theory for networks: A tutorial on game-theoretic tools for emerging signal processing applications. *IEEE Signal Processing Magazine*, 33(1):94–119, 2015.

[3] D. P. Bertsekas. Nonlinear programming. *Journal of the Operational Research Society*, 48(3):334–334, 1997.

[4] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[5] Y. Carmon and J. C. Duchi. Gradient descent efficiently finds the cubic-regularized non-convex newton step. *arXiv preprint arXiv:1612.00547*, 2016.

[6] K. Deb. *Multi-objective optimization using evolutionary algorithms*, volume 16. John Wiley & Sons, 2001.

[7] M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1-2):95–110, 1956.

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

[9] Z. Han, D. Niyato, W. Saad, T. Basar, and A. Hjorungnes. *Game theory in wireless and communication networks: theory, models, and applications*. Cambridge University Press, 2012.

[10] C. Jin, P. Netrapalli, and M. I. Jordan. What is local optimality in nonconvex-nonconcave minimax optimization? *arXiv preprint arXiv:1902.00618*, 2019.

[11] J. D. Lee, I. Panageas, G. Piliouras, M. Simchowitz, M. I. Jordan, and B. Recht. First-order methods almost always avoid strict saddle points. *Mathematical Programming*, pages 1–27, 2019.

[12] J. D. Lee, M. Simchowitz, M. I. Jordan, and B. Recht. Gradient descent only converges to minimizers. In *Conference on Learning Theory*, pages 1246–1257, 2016.

[13] Q. Li, Z. Zhu, and G. Tang. Alternating minimizations converge to second-order optimal solutions. In *International Conference on Machine Learning*, pages 3935–3943, 2019.

[14] Q. Li, Z. Zhu, G. Tang, and M. B. Wakin. Provable bregman-divergence based methods for nonconvex and non-lipschitz problems. *arXiv preprint arXiv:1904.09712*, 2019.

[15] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

[16] P. Mertikopoulos, B. Lecouat, H. Zenati, C.-S. Foo, V. Chandrasekhar, and G. Piliouras. Optimistic mirror descent in saddle-point problems: Going the extra (gradient) mile. 2019.

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

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

[19] J. Nocedal and S. Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[20] S. Omidshafiei, J. Pazis, C. Amato, J. P. How, and J. Vian. Deep decentralized multi-task multi-agent reinforcement learning under partial observability. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2681–2690. JMLR. org, 2017.

[21] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 16–17, 2017.

[22] S. Racanière, T. Weber, D. Reichert, L. Buesing, A. Guez, D. J. Rezende, A. P. Badia, O. Vinyals, N. Heess, Y. Li, et al. Imagination-augmented agents for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 5690–5701, 2017.

[23] L. J. Ratliff, S. A. Burden, and S. S. Sastry. On the characterization of local nash equilibria in continuous games. *IEEE Transactions on Automatic Control*, 61(8):2301–2307, 2016.

[24] P. Xu, F. Roosta-Khorasani, and M. W. Mahoney. Second-order optimization for non-convex machine learning: An empirical study. *arXiv preprint arXiv:1708.07827*, 2017.