ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ

ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS

PROJECT: ALGOLINC – Algorithmic Learning Theory and Incentives: Synergies in Optimization and Mechanism Design

Project ID: 15877

Greece 2.0 NATIONAL RECOVERY AND RESILIENCE PLAN “BASIC RESEARCH FINANCING” (Horizontal support for all Sciences)

WP4: Games, Optimization and Online Learning

Deliverable D.4.3: Final technical report for WP4 (including research submitted to relevant conferences or journals)

Prepared by the team members:

Michail Fasoulakis, Evangelos Markakis

ΕΛ.Ι.Δ.Ε.Κ.
Ελληνικό Ίδρυμα Έρευνας & Καινοτομίας

Deliverable D.4.3: Final technical report for WP4 (including research submitted to relevant conferences or journals)

1 Overview of Publications

This deliverable concerns WP4, and was completed by Month M18 according to the timeline of the project. The goal of this report is to provide an overview of our progress regarding the research questions that formed the focus of WP4.

The output of our research within WP4 has been 2 papers, one accepted for publication at IJCAI 2025 and another one under submission, as follows:

- Michail Fasoulakis, Evangelos Markakis, Georgios Roussakis, Christodoulos Santorinaios. A Descent-based Method on the Duality Gap for Solving Zero-sum Games. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, IJCAI 2025, Montreal, Canada, August 16-22, pages 3839-3847, 2025.
- Michail Fasoulakis, Evangelos Markakis, Georgios Roussakis, Christodoulos Santorinaios. Improved Last Iterate Convergence Properties for the FLBR Dynamics. Under submission, 2025.

Both of these papers can be found at the Appendix of this deliverable.

2 Preliminaries and Relevant Definitions

We briefly recall the topic under consideration. WP4 is centered around the problem of finding Nash equilibria in bilinear zero-sum games. Zero-sum games have played a fundamental role in both game theory, being among the first classes of games formally studied, and in optimization.

A bilinear zero-sum game can be denoted as $(R, -R)$, where R is the payoff matrix of the row player. We assume $R \in [0, 1]^{n\times n}$ without loss of generality[1]. We consider mixed strategies $\boldsymbol{x} \in \Delta^{n-1}$ as a probability distribution (column vector) on the pure strategies of a player, with Δ^{n-1} be the $(n-1)$-dimensional simplex. We also denote by $\boldsymbol{e}_i$ the distribution corresponding to a pure strategy i, with 1 in the index i and zero elsewhere. A strategy profile is a pair $(\boldsymbol{x}, \boldsymbol{y})$, where $\boldsymbol{x}$ is the strategy of the row player and $\boldsymbol{y}$ is the strategy of the column player. Under a profile $(\boldsymbol{x}, \boldsymbol{y})$, the expected payoff of the row player is $\boldsymbol{x}^\top R\boldsymbol{y}$ and the expected payoff of the column player is $-\boldsymbol{x}^\top R\boldsymbol{y}$.

The most well-studied solution concept in games is that of a Nash equilibrium, defined as follows.

[1] We can easily see that we can do scaling for any $R \in \mathbb{R}^{n\times n}$ s.t. $R \in [0, 1]^{n\times n}$ keeping exactly the same Nash equilibria.

Definition 2.1 (Nash equilibrium [5, 6]). *A strategy profile* $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ *is a Nash equilibrium in the game* $(R, -R)$*, if and only if, for any* i, j*,*

$$\boldsymbol{x}^{*\top} R \boldsymbol{y}^* \geq \boldsymbol{e}_i^\top R \boldsymbol{y}^*, \text{ and, } \boldsymbol{x}^{*\top} R \boldsymbol{y}^* \leq \boldsymbol{x}^{*\top} R \boldsymbol{e}_j,$$

As we are interested in algorithms that converge to equilibria, we also need to utilize some notion of approximating Nash equilibria. We ue the following standard aproximation metric.

Definition 2.2 (δ-Nash equilibrium). *A strategy profile* $(\boldsymbol{x}, \boldsymbol{y})$ *is a* δ*-Nash equilibrium (in short,* δ*-NE) in the game* $(R, -R)$*, with* $\delta \in [0, 1]$*, if and only if, for any* i, j*,*

$$\boldsymbol{x}^\top R \boldsymbol{y} + \delta \geq \boldsymbol{e}_i^\top R \boldsymbol{y}, \text{ and, } \boldsymbol{x}^\top R \boldsymbol{y} - \delta \leq \boldsymbol{x}^\top R \boldsymbol{e}_j.$$

Furthermore, it can be easily seen that the equilibrium solutions of a zero-sum game correspond to solving a min-max optimization problem of the form:

$$\min_{\boldsymbol{x}} \max_{\boldsymbol{y}} f(\boldsymbol{x}, \boldsymbol{y})$$

where $f(\boldsymbol{x}, \boldsymbol{y})$ is the payoff function of the first player.

In the next 2 sections we summarize the progress we made on finding approximate equilibria. The first one is an optimization approach, whereas the second one concerns the analysis of a learning algorithm with gradient feedback.

3 An alternative descent-based method for finding approximate Nash equilibria.

It is well known that solving zero-sum games is in fact equivalent to solving linear programs. Despite the fact that a single linear program suffices to find a Nash equilibrium, there has been a surge of interest in recent years for faster algorithms, motivated in part by applications in machine learning. One reason for this is that we may have very large games to solve, corresponding to LPs with thousands of variables and constraints. A second reason could be that, e.g., in learning environments, the players may be using iterative algorithms that can only observe limited information, hence it would be impossible to run a single linear program for the entire game.

In [4], we have developed a method based on performing descent steps on the *duality gap* function of the strategy profiles. In order to define the duality gap, we introduce first the regret functions of the players as follows.

Definition 3.1 (Regret of a player). *For a game* $(R, -R)$*, the regret function* $f_R : \Delta^{n-1} \times \Delta^{n-1} \to [0, 1]$ *of the row player under a strategy profile* $(\boldsymbol{x}, \boldsymbol{y})$ *is*

$$f_R(\boldsymbol{x}, \boldsymbol{y}) = \max_i \boldsymbol{e}_i^\top R \boldsymbol{y} - \boldsymbol{x}^\top R \boldsymbol{y}.$$

Similarly, for the column player the regret function is

$$\begin{aligned} f_{-R}(\boldsymbol{x}, \boldsymbol{y}) &= \max_j \boldsymbol{x}^\top (-R) \boldsymbol{e}_j + \boldsymbol{x}^\top R \boldsymbol{y} \\ &= -\min_j \boldsymbol{x}^\top R \boldsymbol{e}_j + \boldsymbol{x}^\top R \boldsymbol{y}. \end{aligned}$$

In the bibliography, the duality gap is simply the sum of the regrets, i.e., the function $V(\boldsymbol{x}, \boldsymbol{y}) = f_R(\boldsymbol{x}, \boldsymbol{y}) + f_{-R}(\boldsymbol{x}, \boldsymbol{y}) = \max_i \boldsymbol{e}_i^\top R\boldsymbol{y} - \min_j \boldsymbol{x}^\top R\boldsymbol{e}_j$. This is an important quantity for evaluating the performance or convergence of algorithms in several related papers.

What is interesting with our approach is that especially for zero-sum games, the duality gap function is convex. This provides the intuition that applying descent steps with respect to this function might be more promising than performing the more standard gradient descent variants on the players' utility functions (which are non-convex). Unfortunately it is not so straightforward to compute the gradient of the duality gap function, and for this we have to resort to approximating the directional derivative. Eventually this is done by resorting to linear programming, albeit the linear programs that we end up solving are rather small in size, and hence easily solvable.

Our contributions in [4] can be summarized as follows.

- We propose and analyze an optimization approach for finding approximate Nash equilibria in zero-sum games. Our algorithm is a descent-based method applied to the duality gap function, inspired by the convexity of the duality gap function. The method is applying a steepest descent approach, where we find in each step the direction that minimises the directional derivative of the duality gap and move towards that.

- Our main result is that we have established mathematically that this method achieves a geometric rate of decrease on the duality gap and thus we reach quite fast an approximate equilibrium. This implies that the algorithm terminates after at most $O\left(\frac{1}{\rho} \cdot \log\left(\frac{1}{\delta}\right)\right)$ iterations with a δ-approximate equilibrium, where ρ is a parameter, related to the computation of the directional derivative.

- We exhibit that the method can also be further customized and explore several variants of the initial method. We show for example that a particular variant with a decaying schedule for the values of δ also converges after $O(\frac{1}{\sqrt{\delta}})$ iterations.

- At the same time, we have provided an experimental evaluation that verifies the fast convergence of our algorithm and its competitive performance against state of the art methods for solving zero-sum games, such as Optimistic Gradient Descent-Ascent (OGDA). Our findings are promising and reveal that the running time is comparable to (and often outperforms) OGDA, even with thousands of strategies per player. We therefore conclude that the overall approach deserves further exploration, as there are also potential ways of accelerating its running time.

4 A first-order learning method for converging to approximate Nash equilibria.

In parallel to the above, we have explored the convergence properties of learning algorithms with gradient feedback. These are algorithms where the players update their strategy in every iteration after receiving information about the gradient of the utility function. After considering carefully the existing bibliography, our research ended up focusing on learning dynamics that form a variation of the extra gradient and mirror prox methods. Such algorithms tend to have better convergence behavior, in contrast to vanilla gradient descent methods. In fact the particular dynamics that we studied within WP4, have been proposed in an earlier work from our research team, in [2]. However, our previous work only established asymptotic convergence, i.e., that the method attains last-iterate convergence, but without any concrete rates on the speed of convergence.

We refer to this method by the name of Forward Looking Best-Response Multiplicative Weights Update method (FLBR-MWU), as in [2]. We provide here a short description of the main idea behind the dynamics. This is an adaptation of the extra gradient method but applied to Multiplicative Weights Updates, and each iteration has an intermediate and a final step. Suppose that starting from some initial profile, we reach the profile $(\boldsymbol{x}^{t-1}, \boldsymbol{y}^{t-1})$ by the end of iteration $t-1$. In the intermediate step of iteration t, we compute a strategy $\hat{\boldsymbol{x}}^t$ for the row player (resp. $\hat{\boldsymbol{y}}^t$ for the column player), which is an approximate best-response strategy to $\boldsymbol{y}^{t-1}$ (resp. to $\boldsymbol{x}^{t-1}$). This serves as a *look ahead* step of what would be the currently optimal choices. In the final step of iteration t, we compute the new mixed strategy $\boldsymbol{x}^t$ for the row player, by performing MWU updates, but after assuming that the opponent was playing $\hat{\boldsymbol{y}}^t$.

Formally, the first step of the dynamics is defined below, at iteration t, and for all $i, j \in [n]$, given a non-negative parameter $\xi \in \mathbb{R}^+$ (ξ is chosen sufficiently large).

$$\begin{aligned} \hat{\boldsymbol{x}}_i^t &= \boldsymbol{x}_i^{t-1} \cdot \frac{e^{\xi \boldsymbol{e}_i^\top R \boldsymbol{y}^{t-1}}}{\sum_{j=1}^n \boldsymbol{x}_j^{t-1} e^{\xi \boldsymbol{e}_j^\top R \boldsymbol{y}^{t-1}}}, \\ \hat{\boldsymbol{y}}_j^t &= \boldsymbol{y}_j^{t-1} \cdot \frac{e^{-\xi \boldsymbol{e}_j^\top R^\top \boldsymbol{x}^{t-1}}}{\sum_{i=1}^n \boldsymbol{y}_i^{t-1} e^{-\xi \boldsymbol{e}_i^\top R^\top \boldsymbol{x}^{t-1}}}. \end{aligned} \tag{1}$$

The second step, which updates the profile $(\boldsymbol{x}^{t-1}, \boldsymbol{y}^{t-1})$ to $(\boldsymbol{x}^t, \boldsymbol{y}^t)$ is below, given the learning rate parameter $\eta \in (0,1)$. We assume that we use the same fixed constants η and ξ in all iterations.

$$\begin{aligned} \boldsymbol{x}_i^t &= \boldsymbol{x}_i^{t-1} \cdot \frac{e^{\eta \boldsymbol{e}_i^\top R \hat{\boldsymbol{y}}^t}}{\sum_{j=1}^n \boldsymbol{x}_j^{t-1} e^{\eta \boldsymbol{e}_j^\top R \hat{\boldsymbol{y}}^t}}, \\ \boldsymbol{y}_j^t &= \boldsymbol{y}_j^{t-1} \cdot \frac{e^{-\eta \boldsymbol{e}_j^\top R^\top \hat{\boldsymbol{x}}^t}}{\sum_{i=1}^n \boldsymbol{y}_i^{t-1} e^{-\eta \boldsymbol{e}_i^\top R^\top \hat{\boldsymbol{x}}^t}}. \end{aligned} \tag{2}$$

The output of our work in [3] provides a more complete treatment on the power and limitations of the FLBR method for bilinear games. Namely, our main contributions can be summarized as follows:

- So far it was only known that the FLBR algorithm attains asymptotic last-iterate convergence, but without any explicit rate. We answer the open question from [2] by establishing concrete rates of convergence. Using the duality gap as our metric, we showed a geometric rate, of the form $O(c^t)$, till we reach an approximate Nash equilibrium, for an appropriate level of approximation. More precisely, the parameter c here ($c < 1$) is independent of the entries in the payoff matrix, and dependent on the dimension.

- For games with a unique Nash equilibrium, we further prove that once we reach an approximate equilibrium, the duality gap keeps getting decreased with a geometric rate, till the exact equilibrium solution, albeit with the caveat that there is a dependence on the Jacobian matrix evaluated at the equilibrium. Furthermore, our proof highlights connections to a neighboring field, as it utilizes ideas from the analysis of the Arimoto-Blahut algorithm (for computing the Shannon's capacity of a discrete memoryless channel).

- We then investigate further properties of FLBR. We prove that it is not a no-regret algorithm, which was not known before. At the same time, we explore aspects of *forgetfulness*, as introduced recently in [1]. We show that in contrast to OMWU, FLBR seems to exhibit forgetfulness, which serves as an indication for fast performance.

- Finally, we perform an experimental comparison of FLBR against Optimistic Gradient Descent-Ascent (OGDA), which is among the best known methods for solving zero-sum games, and against Optimistic Multiplicative Weights Update (OMWU). We mostly focus on the comparison against OGDA since OMWU is not as competitive in practice (observed also in other recent works). The results reveal that FLBR is generally competitive against OGDA. In particular, FLBR seems to perform better in more structured games, whereas OGDA performs better on randomly generated games.

Overall, we believe that our work highlights the potential of such dynamics and stimulates further research towards faster last-iterate convergence speed for algorithms with gradient-feedback.

References

[1] Yang Cai, Gabriele Farina, Julien Grand-Clément, Christian Kroer, Chung-Wei Lee, Haipeng Luo, and Weiqiang Zheng. Fast last-iterate convergence of learning in games requires forgetful algorithms. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024*, 2024.

[2] M. Fasoulakis, E. Markakis, Y. Pantazis, and C. Varsos. Forward looking best-response multiplicative weights update methods for bilinear zero-sum games. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS'22)*, pages 11096–11117, 2022.

[3] Michail Fasoulakis, Evangelos Markakis, Georgios Roussakis, and Christodoulos Santorinaios. Improved last-iterate convergence properties for the flbr dynamics. Submitted to NeurIPS 2025, 2025.

[4] Michail Fasoulakis, Evangelos Markakis, Giorgos Roussakis, and Christodoulos Santorinaios. A descent-based method on the duality gap for solving zero-sum games. arXiv:2501.19138, 2025.

[5] J. Nash. Non-cooperative games. *Annals of Mathematics*, 54 (2), 1951.

[6] J. Von Neumann. Zur theorie der gesellschaftsspiele. *Math. Ann.*, 100:295–320, 1928.

A Appendix

In the remainder of this deliverable we include the 2 articles mentioned in Section 1, namely [3] and [4].

A Descent-based Method on the Duality Gap for Solving Zero-sum Games

Michail Fasoulakis[1] , **Evangelos Markakis** [2,3,5] ,
Giorgos Roussakis[4] and **Christodoulos Santorinaios**[2,3]
[1]Royal Holloway, University of London, UK
[2]Athens University of Economics and Business, Greece
[3]Archimedes, Athena Research Center, Greece
[4]Foundation for Research and Technology–Hellas, Greece
[5]Input Output Global (IOG), Greece
Michail.Fasoulakis@rhul.ac.uk, {markakis, santgchr}@aueb.gr, roussakis@ics.forth.gr

Abstract

We focus on the design of algorithms for finding equilibria in 2-player zero-sum games. Although it is well known that such problems can be solved by a single linear program, there has been a surge of interest in recent years for simpler algorithms, motivated in part by applications in machine learning. Our work proposes such a method, inspired by the observation that the duality gap (a standard metric for evaluating convergence in min-max optimization problems) is a convex function for bilinear zero-sum games. To this end, we analyze a descent-based approach, variants of which have also been used as a subroutine in a series of algorithms for approximating Nash equilibria in general non-zero-sum games. In particular, we study a steepest descent approach, by finding the direction that minimises the directional derivative of the duality gap function. Our main theoretical result is that the derived algorithms achieve a geometric decrease in the duality gap until we reach an approximate equilibrium. Finally, we complement this with an experimental evaluation, which provides promising findings. Our algorithm is comparable with (and in some cases outperforms) some of the standard approaches for solving 0-sum games, such as OGDA (Optimistic Gradient Descent/Ascent), even with thousands of available strategies per player.

1 Introduction

Our work focuses on the design of algorithms for finding Nash equilibria in 2-player bilinear zero-sum games. Zero-sum games have played a fundamental role both in game theory, being among the first classes of games formally studied, and in optimization, as it is easily seen that their equilibrium solutions correspond to solving a min-max optimization problem. Even further, solving zero-sum games is in fact equivalent to solving linear programs, as properly demonstrated in [Adler, 2013].

Despite the fact that a single linear program (and its dual) suffices to find a Nash equilibrium, there has been a surge of interest in recent years, for faster algorithms, motivated in part by applications in machine learning. One reason for this is that we may have very large games to solve, corresponding to LPs with thousands of variables and constraints. A second reason could be that e.g., in learning environments, the players may be using iterative algorithms that can only observe limited information, hence it would be impossible to run a single LP for the entire game. As an additional motivation, finding new algorithms for such a fundamental problem can provide insights that could be of further value and interest.

The above considerations have led to a variety of approaches and algorithms, spanning already a few decades of research. Some of the earlier works on this domain have focused purely on an optimization viewpoint. In parallel to this, significant attention has been drawn to learning-oriented algorithms, such as first-order methods. The latter class of algorithms performs gradient descent or ascent on the utility functions of the two players, and some of the proposed variants have been very successful in practice, such as the optimistic gradient and the extra gradient methods [Korpelevich, 1976; Popov, 1980]. Several works have focused on theoretical guarantees for their performance, and a standard metric used in the analysis is the duality gap. This is simply the sum of the regrets of the two players in a given profile, and therefore the goal often amounts to proving appropriate rates of decrease for the duality gap over the iterations of an algorithm.

Our work is motivated by the observation that the duality gap is a convex function for zero-sum games. This naturally gives rise to the suggestion that instead of performing gradient descent on the utility function of a player, which is not a convex function, we could apply a descent procedure directly on the duality gap. It is not straightforward that this can indeed be useful as it is not a priori clear that we can perform a descent step fast (i.e., finding the direction to move to). Nevertheless, it can form the basis for investigating new approaches for zero-sum games.

1.1 Our Contributions

Motivated by the above discussion, we propose and analyze an optimization approach for finding approximate Nash equilibria in zero-sum games. Our algorithm is a descent-based method applied to the duality gap function, and is essentially an adaptation of a subroutine in the algorithms of [Tsaknakis and Spirakis, 2008; Deligkas *et al.*, 2017; Deligkas *et al.*, 2023] which are for general games, tailored to zero-sum games and with a different objective function. The method is applying a steepest descent approach, where we find in each step the direction that minimises the directional derivative of the duality gap function and move towards that. In Section 3 we provide the algorithm and our theoretical analysis. Our main result is that the derived algorithm achieves a geometric decrease in the duality gap until we reach an approximate equilibrium. This implies that the algorithm terminates after at most $O\left(\frac{1}{\rho} \cdot \log\left(\frac{1}{\delta}\right)\right)$ iterations with a δ-approximate equilibrium, where ρ is a parameter, related to the computation of the directional derivative. We exhibit that the method can also be further customized and show that a different variant also converges after $O(\frac{1}{\sqrt{\delta}})$ iterations.

In Section 4, we complement our theoretical analysis with an experimental evaluation. Even though the method does need to solve a linear program in each iteration to find the desirable direction, this turns out to be of much smaller size on average (in terms of the number of constraints) than solving the linear program of the entire game. We compare our method against standard LP solvers, but also against state-of-the-art procedures for zero-sum games, such as Optimistic Gradient Descent-Ascent (OGDA). Our findings are promising and reveal that the running time is comparable to (and often outperforms) OGDA, even with thousands of strategies per player. We therefore conclude that the overall approach deserves further exploration, as there are also potential ways of accelerating its running time, discussed in Section 4.

1.2 Related Work

As already mentioned, conceptually, the works most related to ours are [Tsaknakis and Spirakis, 2008; Deligkas *et al.*, 2017; Deligkas *et al.*, 2023]. Although these papers do not consider zero-sum games, they do utilize a descent-based part as a starting point. The main differences with our work is that first of all, their descent is performed with respect to the maximum regret among the two players, whereas we use the duality gap function. Furthermore the descent phase is only a subroutine of their algorithms, since it does not suffice to establish guarantees for general games. Hence their focus is less on the decent phase itself and more on utilizing further procedures to produce approximate equilibria.

There is a plethora of algorithms for linear programming and zero-sum games, which is impossible to list here, but we comment on what we feel are most relevant. When focusing on optimization algorithms for zero-sum games, [Hoda *et al.*, 2010] use Nesterov's first order smoothing techniques to achieve an ϵ-equilibrium in $O(1/\epsilon)$ iterations, with added benefits of simplicity and rather low computational cost per iteration. Following up on that work, [Gilpin *et al.*, 2012] propose an iterated version of Nesterov's smoothing technique, which runs within $O(\frac{||A||}{\delta(A)} \cdot \ln(1/\epsilon))$ iterations. However, while this is a significant improvement, the complexity depends on a condition measure $\delta(A)$, with A being the payoff matrix, not necessarily bounded by a constant. Another optimization approach that is relevant in spirit to ours is via the Nikaido-Isoda function [Nikaido and Isoda, 1955] and its variants. E.g., in [Raghunathan *et al.*, 2019] they run a descent method on the Gradient NI function, which is convex for zero-sum games. We are not aware though of any direct connection to the duality gap function that we use here.

Apart from the optimization viewpoint, there has been great interest in designing faster learning algorithms for zero-sum games. Although this direction started already several decades ago, e.g. with the fictitious play algorithm [Brown, 1951; Robinson, 1951], it has received significant attention more recently given the relevance to formulating GANs in deep learning [Goodfellow *et al.*, 2014] and also other applications in machine learning. Some of the earlier and standard results in this area concern convergence *on average*. That is, it has been known that by using no-regret algorithms, such as the Multiplicative Weights Update (MWU) methods [Arora *et al.*, 2012] the empirical average of the players' strategies over time converges to a Nash equilibrium in zero-sum games. Similarly, one could also utilize the so-called Gradient Descent/Ascent (GDA) algorithms.

Within the last decade, there has also been a great interest in algorithms attaining the more robust notion of *last-iterate convergence*. This means that the strategy profile (x_t, y_t), reached at iteration t, converges to the actual equilibrium as $t \to \infty$. Negative results in [Bailey and Piliouras, 2018] and [Mertikopoulos *et al.*, 2018] exhibit that several no-regret algorithms such as many MWU as well as GDA variants, do not satisfy last-iterate convergence. Motivated by this, there has been a series of works on obtaining algorithms with provable last iterate convergence. The positive results that have been obtained for zero-sum games is that improved versions of Gradient Descent such as the Extra Gradient method [Korpelevich, 1976] or the Optimistic Gradient method [Popov, 1980] attain last iterate convergence. In particular, [Daskalakis *et al.*, 2018] and [Liang and Stokes, 2019] show that the optimistic variant of GDA (referred to as OGDA) converges for zero-sum games. Analogously, OMWU (the optimistic version of MWU) also attains last iterate convergence, shown in [Daskalakis and Panageas, 2019] and further analyzed in [Wei *et al.*, 2021]. The rate of convergence of optimistic gradient methods in terms of the duality gap was studied in [Cai *et al.*, 2022; Gorbunov *et al.*, 2022], and was later improved to $O(1/t)$ in [Cai and Zheng, 2023]. Further approaches with convergence guarantees have also been proposed, based on variations of the Mirror-Prox method [Fasoulakis *et al.*, 2022] as well as primal-dual hybrid gradient methods [Lu and Yang, 2023].

Finally, several of the methods mentioned above are applicable beyond bilinear min-max problems (e.g., to convex-concave). For even more general games, [Diakonikolas *et al.*, 2021] obtain positive results for a class of non-convex and non-concave problems. Multi-player games are also studied, among others in [Golowich *et al.*, 2020], where Optimistic Gradient is analyzed. The picture however is overall more complex for general games with negative results also established in [Daskalakis *et al.*, 2021].

2 Preliminaries

We consider bilinear zero-sum games $(\boldsymbol{R}, -\boldsymbol{R})$, with n pure strategies per player, where $\boldsymbol{R}$ is the payoff matrix of the row player. We assume $\boldsymbol{R} \in [0,1]^{n\times n}$ without loss of generality[1]. We consider mixed strategies $\boldsymbol{x} \in \Delta^{n-1}$ as a probability distribution (column vector) on the pure strategies of a player, with Δ^{n-1} be the $(n-1)$-dimensional simplex. We also denote by $\boldsymbol{e}_i$ the distribution corresponding to a pure strategy i, with 1 in the index i and zero elsewhere. A strategy profile is a pair $(\boldsymbol{x}, \boldsymbol{y})$, where $\boldsymbol{x}$ is the strategy of the row player and $\boldsymbol{y}$ is the strategy of the column player. Under a profile $(\boldsymbol{x}, \boldsymbol{y})$, the expected payoff of the row player is $\boldsymbol{x}^\top \boldsymbol{R} \boldsymbol{y}$ and the expected payoff of the column player is $-\boldsymbol{x}^\top \boldsymbol{R} \boldsymbol{y}$.

A pure strategy i is a ρ-best-response strategy against $\boldsymbol{y}$ for the row player, for $\rho \in [0,1]$, if and only if, $\boldsymbol{e}_i^\top \boldsymbol{R}\boldsymbol{y} + \rho \geq \boldsymbol{e}_j^\top \boldsymbol{R}\boldsymbol{y}$, for any j. Similarly, a pure strategy j for the column player is a ρ-best-response strategy against some strategy $\boldsymbol{x}$ of the row player if and only if $\boldsymbol{x}^T \boldsymbol{R} \boldsymbol{e}_j \leq \boldsymbol{x}^T \boldsymbol{R}\boldsymbol{e}_i + \rho$, for any i. Having these, we define as $BR_r^\rho(\boldsymbol{y})$ the set of the ρ-best-response pure strategies of the row player against $\boldsymbol{y}$ and as $BR_c^\rho(\boldsymbol{x})$ the set of the ρ-best-response pure strategies of the column player against $\boldsymbol{x}$. For $\rho = 0$, we will use $BR_r(\boldsymbol{y})$ and $BR_c(\boldsymbol{x})$ for the best response sets.

Definition 1 (Nash equilibrium [Nash, 1951; Von Neumann, 1928])**.** *A strategy profile $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ is a Nash equilibrium in the game $(\boldsymbol{R}, -\boldsymbol{R})$, if and only if, for any i, j,*

$$v = {\boldsymbol{x}^*}^\top \boldsymbol{R}\boldsymbol{y}^* \geq \boldsymbol{e}_i^\top \boldsymbol{R}\boldsymbol{y}^*, \text{ and, } v = {\boldsymbol{x}^*}^\top \boldsymbol{R}\boldsymbol{y}^* \leq {\boldsymbol{x}^*}^\top \boldsymbol{R}\boldsymbol{e}_j,$$

where v is the value of the row player (value of the game).

Definition 2 (δ-Nash equilibrium)**.** *A strategy profile $(\boldsymbol{x}, \boldsymbol{y})$ is a δ-Nash equilibrium (in short, δ-NE) in the game $(\boldsymbol{R}, -\boldsymbol{R})$, with $\delta \in [0,1]$, if and only if, for any i, j,*

$$\boldsymbol{x}^\top \boldsymbol{R}\boldsymbol{y} + \delta \geq \boldsymbol{e}_i^\top \boldsymbol{R}\boldsymbol{y}, \text{ and, } \boldsymbol{x}^\top \boldsymbol{R}\boldsymbol{y} - \delta \leq \boldsymbol{x}^\top \boldsymbol{R}\boldsymbol{e}_j.$$

With these at hand, we can now define the regret functions of the players as follows.

Definition 3 (Regret of a player)**.** *For a game $(\boldsymbol{R}, -\boldsymbol{R})$, the regret function $f_{\boldsymbol{R}} : \Delta^{n-1} \times \Delta^{n-1} \to [0,1]$ of the row player under a strategy profile $(\boldsymbol{x}, \boldsymbol{y})$ is*

$$f_{\boldsymbol{R}}(\boldsymbol{x}, \boldsymbol{y}) = \max_i \boldsymbol{e}_i^\top \boldsymbol{R}\boldsymbol{y} - \boldsymbol{x}^\top \boldsymbol{R}\boldsymbol{y}.$$

Similarly, for the column player the regret function is

$$\begin{aligned} f_{-\boldsymbol{R}}(\boldsymbol{x}, \boldsymbol{y}) &= \max_j \boldsymbol{x}^\top (-\boldsymbol{R}) \boldsymbol{e}_j + \boldsymbol{x}^\top \boldsymbol{R}\boldsymbol{y} \\ &= -\min_j \boldsymbol{x}^\top \boldsymbol{R}\boldsymbol{e}_j + \boldsymbol{x}^\top \boldsymbol{R}\boldsymbol{y}. \end{aligned}$$

An important quantity for evaluating the performance or convergence of algorithms is the sum of the regrets, i.e., the function $V(\boldsymbol{x}, \boldsymbol{y}) = f_{\boldsymbol{R}}(\boldsymbol{x}, \boldsymbol{y}) + f_{-\boldsymbol{R}}(\boldsymbol{x}, \boldsymbol{y}) = \max_i \boldsymbol{e}_i^\top \boldsymbol{R}\boldsymbol{y} - \min_j \boldsymbol{x}^\top \boldsymbol{R}\boldsymbol{e}_j$. This is referred to in the bibliography as the *duality gap* in the case of zero-sum games.

[1]We can easily see that we can do scaling for any $\boldsymbol{R} \in \mathbb{R}^{n\times n}$ s.t. $\boldsymbol{R} \in [0,1]^{n\times n}$ keeping exactly the same Nash equilibria.

2.1 Warmup: Duality Gap Properties

Next, we present some known results about the *duality gap* function $V(\boldsymbol{x}, \boldsymbol{y})$ and its connection to Nash equilibria.

Theorem 1. *The duality gap $V(\boldsymbol{x}, \boldsymbol{y})$ is convex in its domain.*

Theorem 2. *A strategy profile $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ is a Nash equilibrium of the game $(\boldsymbol{R}, -\boldsymbol{R})$, if and only if, it is a (global) minimum[2] of the function $V(\boldsymbol{x}, \boldsymbol{y})$.*

Similarly to the previous theorem, we also have the following.

Theorem 3. *Let $(\boldsymbol{x}, \boldsymbol{y})$ be a strategy profile in a zero-sum game. If $V(\boldsymbol{x}, \boldsymbol{y}) \leq \delta$, then $(\boldsymbol{x}, \boldsymbol{y})$ is a δ-NE.*

3 Descent-based Algorithms on the Duality Gap: Theoretical Analysis

In this section, we present our main algorithm along with some improved variants, based on a gradient-descent approach for the function $V(\boldsymbol{x}, \boldsymbol{y})$ in zero-sum games. The algorithm can be seen as an adaptation[3] of a descent procedure that forms the initial phase of algorithms proposed for general non-zero-sum games, in [Tsaknakis and Spirakis, 2008; Deligkas *et al.*, 2017; Deligkas *et al.*, 2023]. The main idea behind the algorithm is that since the global minimum of the duality gap function $V(\boldsymbol{x}, \boldsymbol{y})$ is a Nash equilibrium and the duality gap is a convex function for zero-sum bilinear games, we use a descent method based on the directional derivative of $V(\boldsymbol{x}, \boldsymbol{y})$. This differs substantially from applying the more common idea of gradient descent/ascent (GDA) on the utility functions of the players, which are not convex functions. To identify the direction that minimizes the directional derivative at every step we use linear programming (albeit solving much smaller linear programs on average than the program describing the zero-sum game itself). As a drawback of the method, we note that it requires the full knowledge of the payoff matrix instead of just gradient feedback in each iteration.

To begin with, we define first the directional derivative.

Definition 4. *The directional derivative of the duality gap at a point $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{y})$, with respect to a direction $\boldsymbol{z}' = (\boldsymbol{x}', \boldsymbol{y}') \in \Delta^{n-1} \times \Delta^{n-1}$ is the limit, if it exists,*

$$\nabla_{\boldsymbol{z}'} V(\boldsymbol{z}) = \lim_{\varepsilon \to 0} \frac{V\Big((1-\varepsilon)\cdot \boldsymbol{z} + \varepsilon \cdot \boldsymbol{z}'\Big) - V(\boldsymbol{z})}{\varepsilon}$$

We provide below a much more convenient form for the directional derivative that facilitates the remaining analysis.

Theorem 4. *The directional derivative of the duality gap V at a point $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{y})$ with respect to a direction $\boldsymbol{z}' = (\boldsymbol{x}', \boldsymbol{y}') \in \Delta^{n-1} \times \Delta^{n-1}$, is given by*

$$\nabla_{\boldsymbol{z}'} V(\boldsymbol{z}) = \max_{i \in BR_r(\boldsymbol{y})} \boldsymbol{e}_i^\top \boldsymbol{R}\boldsymbol{y}' - \min_{j \in BR_c(\boldsymbol{x})} (\boldsymbol{x}')^\top \boldsymbol{R}\boldsymbol{e}_j - V(\boldsymbol{z})$$

Furthermore, by the definition of directional derivative we have the following consequence.

[2]Note that the set of Nash equilibria in zero-sum games and the set of optimal solutions, minimizing the duality gap are convex and identical to each other.

[3]Here as the objective function we use the sum of the regrets instead of the maximum of the two regrets.

Lemma 1. *Given $\delta \in [0,1]$, let $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{y})$ be a strategy profile that is not a δ-Nash equilibrium. Then*

$$\nabla_{\boldsymbol{z}'} V(\boldsymbol{z}) < -\delta,$$

where $\boldsymbol{z}' = (\boldsymbol{x}', \boldsymbol{y}') \in \Delta^{n-1} \times \Delta^{n-1}$ is a direction that minimizes the directional derivative.

The proof of Lemma 1 follows by a more general result presented in Lemma 3 below (using also Lemma 2). In a similar manner to Definition 4, we define now an approximate version of the directional derivative. The reason we do that will become clear later on, in order to show that the duality gap decreases from one iteration of the algorithm to the next. The main idea in the definition below is to include approximate best responses in the maximization and minimization terms involved in the statement of Theorem 4. Namely, for $\rho > 0$, recall the definition of $BR_r^{\rho}(\boldsymbol{y})$ as the set of ρ-best response strategies of the row player against strategy $\boldsymbol{y}$ of the column player (and similarly for $BR_c^{\rho}(\boldsymbol{x})$).

Definition 5 (ρ-directional derivative)**.** *The ρ-directional derivative of the duality gap V at a point $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{y})$ with respect to a direction $\boldsymbol{z}' = (\boldsymbol{x}', \boldsymbol{y}') \in \Delta^{n-1} \times \Delta^{n-1}$ is*

$$\nabla_{\rho,\boldsymbol{z}'} V(\boldsymbol{z}) = \max_{i \in BR_r^{\rho}(\boldsymbol{y})} \boldsymbol{e}_i^\top \boldsymbol{R}\boldsymbol{y}' - \min_{j \in BR_c^{\rho}(\boldsymbol{x})} (\boldsymbol{x}')^\top \boldsymbol{R}\boldsymbol{e}_j - V(\boldsymbol{z}).$$

Lemma 2. *It holds that for any direction $\boldsymbol{z}' = (\boldsymbol{x}', \boldsymbol{y}') \in \Delta^{n-1} \times \Delta^{n-1}$, and for any $\rho > 0$,*

$$\nabla_{\boldsymbol{z}'} V(\boldsymbol{z}) \leq \nabla_{\rho,\boldsymbol{z}'} V(\boldsymbol{z}).$$

Lemma 3. *Given $\delta \in [0,1]$, let $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{y})$ be a strategy profile that is not a δ-Nash equilibrium. Then*

$$\nabla_{\rho,\boldsymbol{z}'} V(\boldsymbol{z}) < -\delta,$$

where $\boldsymbol{z}' = (\boldsymbol{x}', \boldsymbol{y}') \in \Delta^{n-1} \times \Delta^{n-1}$ is a direction that minimizes the ρ-directional derivative.

The proofs of these lemmas and any other missing proofs from this section are deferred to the full version of this work in [Fasoulakis *et al.*, 2025].

3.1 The Main Algorithm

We now present our algorithm. Algorithm 1 takes as input a game and 3 parameters, namely $\delta \in (0,1]$, which refers to the approximation guarantee that is desired, $\rho \in (0,1]$ which involves the approximation to the directional derivative, and ϵ, which refers to the size of the step taken in each iteration. Our theoretical analysis will require ρ and ϵ to be correlated.

Observation 1. *If $\rho = 1$, then Algorithm 2 returns an exact Nash equilibrium of the game $(\boldsymbol{R}, -\boldsymbol{R})$.*

We conclude the presentation of our main algorithm with the following remark.

Remark 1. *The choice of ρ demonstrates the trade off between global optimization (Linear Programming) and the descent-based approach. In the extreme case where $\rho = 1$, Observation 1 shows one iteration would suffice, solving the (large) linear program of the entire zero-sum game. On the other hand, when ρ is small, close to 0, then the method solves in each iteration rather small linear programs in Algorithm 2 (dependent on the sets $BR_c^{\rho}(\boldsymbol{x}), BR_r^{\rho}(\boldsymbol{y})$).*

Algorithm 1 The gradient descent-based algorithm

Input: A 0-sum game $(\boldsymbol{R}, -\boldsymbol{R})$, an approximation parameter $\delta \in (0,1]$, a constant $\rho \in (0,1]$, and a constant $\epsilon \in (0,1]$.
Output: A δ-NE strategy profile.

```
Pick an arbitrary strategy profile (x, y)
while V(x, y) > δ do
    (x', y') = FindDirection(x, y, ρ)
    (x, y) = (1 − ε) · (x, y) + ε · (x', y')
return (x, y).
```

Algorithm 2 FindDirection$(\boldsymbol{x}, \boldsymbol{y}, \rho)$

Input: A strategy profile $(\boldsymbol{x}, \boldsymbol{y})$ and parameter $\rho \in (0,1]$.
Output: The direction $(\boldsymbol{x}', \boldsymbol{y}')$ that minimizes the ρ-directional derivative.

```
Solve the linear program (w.r.t. (x', y') and γ):
    minimize γ
        s.t. γ ≥ (e_i)^T R y' − (x')^T R e_j,
        for any i ∈ BR_r^ρ(y), for any j ∈ BR_c^ρ(x),
        and with x', y' ∈ Δ^{n−1}.
return (x', y').
```

3.2 Proof of Correctness and Rate of Convergence

Our main result is the following theorem.

Theorem 5. *For any constants $\delta, \rho \in (0,1]$, and with $\epsilon = \rho/2$, Algorithm 1 returns a δ-Nash equilibrium in bilinear zero-sum games after at most $O(\frac{1}{\rho \cdot \delta} \log \frac{1}{\delta})$ iterations, and with a geometric rate of convergence for the duality gap.*

To prove Theorem 5, we will start with the following auxiliary lemma. The interpretation of the lemma is that when the column player moves from $\mathbf{y}$ to the strategy $(1-\epsilon)\mathbf{y} + \epsilon\mathbf{y}'$, it is still better for the row player to choose a strategy from the set $BR_{\rho}(\mathbf{y})$, as long as ρ is large enough.

Lemma 4. *If $\varepsilon \leq \frac{\rho}{2}$, then it holds that*

$$\max\Big\{0, \max_{i \in \overline{BR_r^{\rho}(\boldsymbol{y})}} \boldsymbol{e}_i^\top \boldsymbol{R}\Big((1-\varepsilon)\cdot \boldsymbol{y} + \varepsilon \cdot \boldsymbol{y}'\Big) - \max_{i \in BR_r^{\rho}(\boldsymbol{y})} \boldsymbol{e}_i^\top \boldsymbol{R}\Big((1-\varepsilon)\cdot \boldsymbol{y} + \varepsilon \cdot \boldsymbol{y}'\Big)\Big\} = 0.$$

Similarly, for the column player, it holds that

$$\max\Big\{0, -\min_{j \in \overline{BR_c^{\rho}(\boldsymbol{x})}} \Big((1-\varepsilon)\cdot \boldsymbol{x} + \varepsilon \cdot \boldsymbol{x}'\Big)^\top R\boldsymbol{e}_j + \min_{j \in BR_c^{\rho}(\boldsymbol{x})} \Big((1-\varepsilon)\cdot \boldsymbol{x} + \varepsilon \cdot \boldsymbol{x}'\Big)^\top R\boldsymbol{e}_j\Big\} = 0.$$

We can now establish that the duality gap decreases geometrically, as long as we have not yet found a δ-approximate equilibrium. We first show an additive decrease.

Lemma 5. *Let $\epsilon \leq \frac{\rho}{2}$ and suppose that after t iterations we are at a profile $(\boldsymbol{x}^t, \boldsymbol{y}^t)$, which is not a δ-Nash equilibrium. Then,*

$$V(\boldsymbol{x}^{t+1}, \boldsymbol{y}^{t+1}) \leq V(\boldsymbol{x}^t, \boldsymbol{y}^t) - \epsilon \cdot \delta$$

where $(\boldsymbol{x}^{t+1}, \boldsymbol{y}^{t+1})$ is the strategy profile at iteration $t+1$.

Proof. To shorten notation, let $\boldsymbol{x}^t = \boldsymbol{x}, \boldsymbol{y}^t = \boldsymbol{y}, \boldsymbol{x}'^t = \boldsymbol{x}'$, $\boldsymbol{y}'^t = \boldsymbol{y}', \boldsymbol{z}^t = (\boldsymbol{x}, \boldsymbol{y}), \boldsymbol{z}^{t+1} = (\boldsymbol{x}^{t+1}, \boldsymbol{y}^{t+1})$. Then we have

$$(\boldsymbol{x}^{t+1}, \boldsymbol{y}^{t+1}) = ((1-\varepsilon)\cdot \boldsymbol{x} + \varepsilon \cdot \boldsymbol{x}', (1-\varepsilon)\cdot \boldsymbol{y} + \varepsilon \cdot \boldsymbol{y}').$$

Similar to the arguments used for the proof of Theorem 4, we have that

$$\max_i \boldsymbol{e}_i^\top \boldsymbol{R}\boldsymbol{y}^{t+1} = \max_{i \in BR_r^\rho(\boldsymbol{y})} \boldsymbol{e}_i^\top \boldsymbol{R}\boldsymbol{y}^{t+1}$$
$$+ \max\Big\{0, \max_{i \in \overline{BR_r^\rho(\boldsymbol{y})}} \boldsymbol{e}_i^\top \boldsymbol{R}\boldsymbol{y}^{t+1} - \max_{i \in BR_r^\rho(\boldsymbol{y})} \boldsymbol{e}_i^\top \boldsymbol{R}\boldsymbol{y}^{t+1}\Big\}.$$

Note that since $\varepsilon \le \frac{\rho}{2}$, Lemma 4 applies and zeroes out the last term. Respectively, we obtain that

$$\min_j (\boldsymbol{x}^{t+1})^\top \boldsymbol{R}\boldsymbol{e}_j = \min_{j \in BR_c^\rho(\boldsymbol{x})} (\boldsymbol{x}^{t+1})^\top \boldsymbol{R}\boldsymbol{e}_j$$

Hence,

$$\begin{aligned}
V(\boldsymbol{z}^{t+1}) &= \max_i \boldsymbol{e}_i^\top \boldsymbol{R}\boldsymbol{y}^{t+1} - \min_j (\boldsymbol{x}^{t+1})^\top \boldsymbol{R}\boldsymbol{e}_j \\
&= \max_{i \in BR_r^\rho(\boldsymbol{y})} \boldsymbol{e}_i^\top \boldsymbol{R}\boldsymbol{y}^{t+1} - \min_{j \in BR_c^\rho(\boldsymbol{x})} (\boldsymbol{x}^{t+1})^\top \boldsymbol{R}\boldsymbol{e}_j \\
&= \max_{i \in BR_r^\rho(\boldsymbol{y})} \boldsymbol{e}_i^\top \boldsymbol{R}\Big((1-\varepsilon)\cdot \boldsymbol{y} + \varepsilon\cdot \boldsymbol{y}'\Big) \\
&\quad - \min_{j \in BR_c^\rho(\boldsymbol{x})} \Big((1-\varepsilon)\cdot \boldsymbol{x} + \varepsilon \cdot \boldsymbol{x}'\Big)^\top \boldsymbol{R}\boldsymbol{e}_j \\
&\le (1-\varepsilon)\max_i \boldsymbol{e}_i^\top \boldsymbol{R}\boldsymbol{y} + \varepsilon \max_{i \in BR_r^\rho(\boldsymbol{y})} \boldsymbol{e}_i^\top \boldsymbol{R}\boldsymbol{y}' \\
&\quad - (1-\varepsilon)\min_j \boldsymbol{x}^\top \boldsymbol{R}\boldsymbol{e}_j - \varepsilon \min_{j \in BR_c^\rho(\boldsymbol{x})} (\boldsymbol{x}')^\top \boldsymbol{R}\boldsymbol{e}_j \\
&= \max_i \boldsymbol{e}_i^\top \boldsymbol{R}\boldsymbol{y} - \min_j (\boldsymbol{x})^\top \boldsymbol{R}\boldsymbol{e}_j \\
&\quad + \varepsilon \cdot \Big(\max_{i \in BR_r^\rho(\boldsymbol{y})} \boldsymbol{e}_i^\top \boldsymbol{R}\boldsymbol{y}' - \min_{j \in BR_c^\rho(\boldsymbol{x})} (\boldsymbol{x}')^\top \boldsymbol{R}\boldsymbol{e}_j \\
&\qquad - \max_i \boldsymbol{e}_i^\top \boldsymbol{R}\boldsymbol{y} + \min_j (\boldsymbol{x})^\top \boldsymbol{R}\boldsymbol{e}_j \Big) \\
&= V(\boldsymbol{z}^t) + \varepsilon \cdot \nabla_{\rho, \boldsymbol{z}'} V(\boldsymbol{z}^t) < V(\boldsymbol{z}^t) - \varepsilon \cdot \delta,
\end{aligned}$$

where the last inequality follows from Lemma 3. □

The next step is to turn the additive decrease of Lemma 5 into a multiplicative decrease.

Corollary 1. *For $\epsilon = \rho/2$, we have that*

$$V(\boldsymbol{x}^{t+1}, \boldsymbol{y}^{t+1}) \le \Big(1 - \frac{\rho\cdot\delta}{4}\Big)\cdot V(\boldsymbol{x}^t, \boldsymbol{y}^t)$$

Proof. Using Lemma 5, we get that $V(\boldsymbol{z}^{t+1}) \le (1-c)\cdot V(\boldsymbol{z}^t)$ with $c = \frac{\varepsilon\cdot\delta}{V(\boldsymbol{z}^t)} \ge \frac{\rho\cdot\delta}{4}$, since $V(\boldsymbol{x}, \boldsymbol{y}) \le 2$ for any profile, and $\varepsilon = \frac{\rho}{2}$. □

Finally, we can complete the proof of our main theorem.

Proof of Theorem 5. We have already proved the geometric decrease of the duality gap, for constant ρ and δ. Hence, the algorithm eventually will satisfy that the duality gap is at most δ and will terminate with a δ-NE. It remains to bound the number of iterations that are needed. Suppose that the algorithm terminates after t iterations, with profile $(\boldsymbol{x}^t, \boldsymbol{y}^t)$. By repeatedly applying Corollary 1, we have that

$$V(\boldsymbol{x}^t, \boldsymbol{y}^t) \le (1-c)^t \cdot V(\boldsymbol{x}^0, \boldsymbol{y}^0)$$

Algorithm 3 Decaying Delta Speedup

Input: A 0-sum game $(\boldsymbol{R}, -\boldsymbol{R})$, an approximation parameter $\delta \in (0, 1]$ and a constant $\rho \in (0, 1]$.
Output: A δ-NE strategy profile.

Pick an arbitrary strategy profile $(\boldsymbol{x}, \boldsymbol{y})$
Set $i = 0$, $\delta_0 = 1$, $\varepsilon = \frac{\rho}{2}$.
while TRUE **do**
  $i = i + 1$, $\delta_i = \delta_{i-1}/2$.
  Update $(\boldsymbol{x}, \boldsymbol{y})$ via Algorithm 1 $\Big((\boldsymbol{R}, -\boldsymbol{R}), \delta_i, \rho, \varepsilon\Big)$.
  if $\delta_i \le \delta$ **then** break
return $(\boldsymbol{x}, \boldsymbol{y})$.

---

with $c = \frac{\rho\cdot\delta}{4}$. In order to ensure that $V(\boldsymbol{x}^t, \boldsymbol{y}^t) \le \delta$, it suffices to have that $2\cdot(1-c)^t \le \delta$, since $V(\boldsymbol{x}^0, \boldsymbol{y}^0) \le 2$.

$$2(1-c)^t \le \delta \implies t \ge \frac{\log\frac{2}{\delta}}{\log\frac{1}{1-c}} \implies t \ge \frac{1-c}{c}\log\frac{2}{\delta}$$

where the last inequality holds due to $\log x \le x - 1$, for $x \ge 1$. Since $\frac{1-c}{c} = O(\frac{1}{c})$, the proof is completed by substituting the value of c. □

Finally, we note that the worst-case complexity of each iteration occurs when Algorithm 2 has to solve LPs of size similar to the initial game. Empirically however, these LPs are of much smaller size as discussed in Section 4.

3.3 Decaying Schedule Speedups

In this section, we present a different implementation of our main approach, which results in an improved analysis. The idea is to gradually decay δ and use it to bound c, instead of the more coarse approximation of $V(\boldsymbol{x}, \boldsymbol{y}) \le 2$, that we used in the proof of Theorem 5. This is presented as Algorithm 3.

Theorem 6. *Algorithm 3 maintains a geometric decrease rate in the duality gap and reaches a δ-NE after at most $O\Big(\frac{1}{\rho}\cdot\log\Big(\frac{1}{\delta}\Big)\Big)$ iterations.*

Proof. We think of the iterations of the entire algorithm as divided into epochs, where each epoch corresponds to a new value for δ. Fix an epoch i, with $i > 0$. Within this epoch, Algorithm 1 is run with approximation parameter δ_i. Consider an arbitrary iteration of Algorithm 1 during this epoch, say at time $t+1$, starting with the profile $\boldsymbol{z}^t = (\boldsymbol{x}^t, \boldsymbol{y}^t)$ and ending at the profile $\boldsymbol{z}^{t+1} = (\boldsymbol{x}^{t+1}, \boldsymbol{y}^{t+1})$. By Lemma 5, we have that $V(\boldsymbol{z}^{t+1}) \le V(\boldsymbol{z}^t) - \epsilon\cdot\delta_i = (1-c_i)\cdot V(\boldsymbol{z}^t)$, where $c_i = \frac{\epsilon\cdot\delta_i}{V(\boldsymbol{z}^t)} = \frac{\rho\cdot\delta_i}{2\cdot V(\boldsymbol{z}^t)}$. Since we are at epoch i, we know that $V(\boldsymbol{z}^t) \le \delta_{i-1} = 2\cdot\delta_i$, because the duality gap was at most δ_{i-1} at the beginning of epoch i and within the epoch it only decreases further due to Lemma 5 (for epoch 1, it is even better, since $V(\boldsymbol{z}^t) \le V(\boldsymbol{z}^0) \le 2 = 2\delta_0 \le 4\delta_1$, where $\boldsymbol{z}^0$ is the initial profile). Therefore, $c_i \ge \frac{\rho\cdot\delta_i}{2\cdot\delta_{i-1}} = \frac{\rho}{4}$. Hence, we have established that in any iteration, regardless of the epoch:

$$V(\boldsymbol{z}^{t+1}) \le \Big(1 - \frac{\rho}{4}\Big)\cdot V(\boldsymbol{z}^t) \le \Big(1 - \frac{\rho}{4}\Big)^t \cdot V(\boldsymbol{z}^0).$$

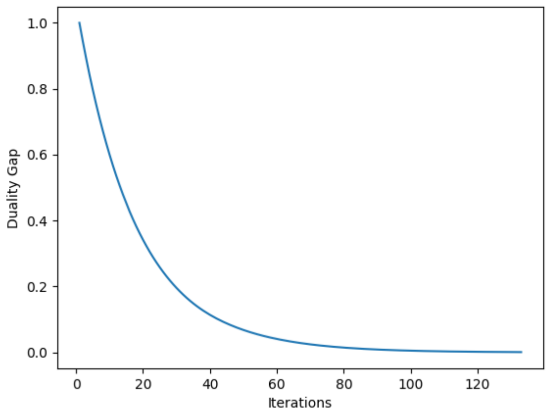

Figure 1: The decrease in the duality gap for a random game.

Since ρ is constant, we have a geometric decrease, and this proves the first part of the theorem.

To bound the total number of iterations, let t_i be the number of iterations of Algorithm 1 within epoch i, after which, the algorithm achieves a δ_i-NE. Then, similar to the proof of Theorem 5, and since in the beginning of epoch i, the duality gap is at most δ_{i-1}, we have that t_i should satisfy

$$(1-c_i)^{t_i} \cdot \delta_{i-1} \leq \delta_i \implies t_i \geq \frac{1}{\log \frac{1}{1-c_i}} \implies t_i \geq \frac{1-c_i}{c_i}$$

Thus, at epoch i, we need $t_i = O(\frac{1}{\rho})$ to reach a δ_i-NE. Next, note that if k is the total number of epochs required to achieve a δ-NE, when starting with δ_0, it holds that $\frac{\delta_0}{2^k} \leq \delta \implies k \geq \log \frac{\delta_0}{\delta}$. Since $\delta_0 = 1$, the number of required epochs is $O(\log \frac{1}{\delta})$. Therefore, the total number of iterations for the entire algorithm is $O\left(\frac{1}{\rho} \cdot \log \frac{1}{\delta}\right)$. □

To demonstrate the flexibility of our approach, we conclude the theoretical exploration with yet another variation, where we additionally use a decreasing schedule for the value of ρ. Specifically, this gives rise to the following scheme which we refer to as Algorithm 4.

- Use the same schedule for δ_i as Algorithm 3.
- At iteration i set $\rho_i = \sqrt{\delta_i}$, for Algorithm 1 (with $\varepsilon_i = \frac{\rho_i}{2}$).

Note that we have now eliminated the dependence on ρ but at the expense of making more expensive the dependence on δ. This new algorithm has the following performance.

Theorem 7. *Algorithm 4 reaches a δ-Nash equilibrium after at most $O\left(\frac{1}{\sqrt{\delta}}\right)$ iterations, for any constant δ.*

4 Experimental Evaluation

All our algorithms were implemented in Python 3.10.9, and were run on a Macbook M1 Pro(10 core) with 16GB RAM. Before proceeding to our main findings, we exhibit first that the geometric decrease in the duality gap can indeed be observed experimentally. Figure 1 shows a typical behavior of our algorithms, in terms of the duality gap. The figure here is for a random game of size $n = 1000$.

4.1 From Theory to Implementation

We deem useful to discuss first how to approach the selection of the parameters that the algorithms depend on. We have seen in Algorithm 1 and its variants two families of parameters: ρ_i and δ_i. A third parameter is the learning rate ε, which is the step size that we take in each iteration.

Choice of ε. We have established that as long as $\varepsilon \leq \rho/2$, the points along the line $(1-\varepsilon)\cdot(\boldsymbol{x},\boldsymbol{y})+\varepsilon\cdot(\boldsymbol{x}',\boldsymbol{y}')$ decrease the duality gap (Lemma 5). Note, though, that the problem of minimizing V along this set is a convex optimization problem. Hence, we can try to find the optimal ε_i at each iteration i, and there are a few possible approaches for this: line search, ternary search or even solving it exactly using dynamic programming. We decided to use the following heuristic: for large values of the duality gap, namely $V > 0.1$, we employ ternary search and as the duality gap decreases we use line search but only on a small part of the line. More specifically, once $V \leq 0.1$ we start with $\varepsilon = 0.2$ and decrease it by 10% across iterations. We decided upon this method since we noticed that experiments conform to theory for smaller values of V and ρ. Finally, a more ML-like approach would be to set a constant ε, similarly to a constant step size η in gradient methods. While this approach has merit, it did not show improved performance.

Choice of ρ (and a new algorithm). The most critical parameter regarding the running time of our algorithms is ρ, since it controls the size of the LPs in Algorithm 2, i.e., the number of constraints, via the sets of ρ-approximate best responses, $BR_r^\rho(\boldsymbol{y})$ and $BR_c^\rho(\boldsymbol{x})$. We need ρ to be large enough to avoid having only a single best response, in which case our algorithms reduces to Best Response Dynamics, while at the same time it should be small enough so that the LPs have small size and we can solve them fast. Our experimentation did not reveal any particular range of ρ with a consistently better performance. As a result, in addition to our existing algorithms, we developed one more approach, independent of ρ: we fix a number k (much smaller than n), and in every iteration, we include in the approximate best response set of each player its top k better responses. We refer to this approach as the *Fixed Support Variant* in the sequel. We used $k=100$ for our experiments and point to our full version in [Fasoulakis *et al.*, 2025] for justification.

Optimizing FindDirection. For this we used two implementation tricks. The first one is quite simple: it is easy to observe that the LP of Algorithm 2 is equivalent to solving two smaller LPs, one per player; it turns out that solving it this way is faster. The second trick revolves around ρ. Recall that the direction we find is itself an approximation. Hence, solving the LP approximately is meaningful, in the sense that it provides an even coarser approximate direction. It turns out that even a 0.1 approximate solution (which is achievable by setting an appropriate parameter in the LP solver) works for most cases, and results in significantly less running time.

4.2 Comparisons between Our Variants

We report first on our comparisons between Algorithm 3 with $\rho = 0.001$, henceforth called the *Constant ρ Variant*, Algorithm 4 with $\rho_i = 0.01\sqrt{\delta_i}$, which we refer to as the *Adap-*

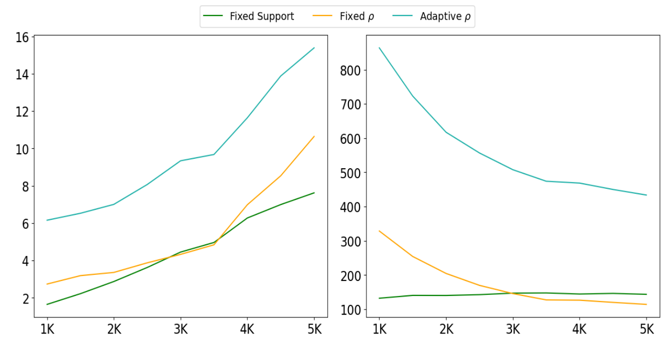

Figure 2: Average time and number of iterations for our variants

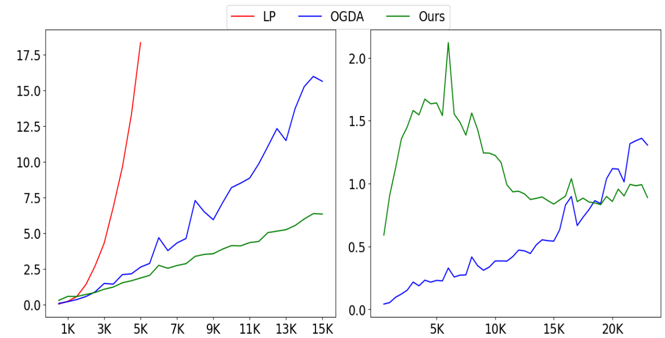

Figure 3: Time comparison between our Fixed Support Variant, LP solver and Optimistic Gradient Descent-Ascent

tive ρ Variant and our Fixed Support Variant discussed in Section 4.1. We note that for the variant with the adaptive value of ρ, we did not follow precisely the values presented by our theoretical analysis, of $\rho_i = \sqrt{\delta_i}$. Although theoretically equivalent, this change was only to avoid a blowup in the number of best response strategies used in Algorithm 2 during the first iterations, i.e. for δ_1 and δ_2 we would have $\rho > 0.7$, which is quite large and undesirable.

To test our algorithms we generated random games of size $n \times n$, where each entry is picked uniformly at random from $[0, 1]$. The size of the games range from 500 to 5000 pure strategies with a step of 500. For each size we generate 30 games and solve them to an accuracy of $\delta = 0.01$. We used two types of initialization in all methods, the fully uniform strategy profile and the profile $(\boldsymbol{e}_1, \boldsymbol{e}_1)$, i.e., first row, first column. The latter has the advantage of not being too close to a Nash equilibrium from the start, in almost all games, and reveals more clearly the exploration that the method performs. The averaged results are presented in Figure 2, where we show both the actual time and the number of iterations. In terms of actual time, our Fixed Support variant is the clear winner. Although Figure 2 reveals that as n grows, the Fixed ρ variant attains a lower number of iterations, this does not translate into improved running time. The intuition for this is that as n grows and ρ remains constant, we expect a larger number of strategies to be ρ-best responses. Consequently, the LP in Algorithm 2 is closer to the full LP and thus more informative, but at the same time more expensive to solve.

As a result of these comparisons, we select our Fixed Support variant as the variant to compare against other methods from the literature in the next subsection.

4.3 Comparisons with LP and Gradient Methods

We compared our Fixed Support variant against solving directly the full LP with a standard LP solver, and against a prominent first order method. Regarding the LP solver, we used the standard method of SciPy. We note that we used the same method for the smaller LPs that we solve in Algorithm 2 of our methods. To maintain an equal comparison with our algorithms, we used a tolerance of 0.01. As for first order methods, we compared against the last-iterate performance of Optimistic Gradient Descent Ascent (OGDA), which is among the fastest gradient based methods, with step size $\eta = 0.01$. Another popular method is Optimistic Multiplicative Weights Update (OMWU), which however does not behave as well in practice, as also explained in [Cai *et al.*, 2024].

For each value of n that we used, we generated 50 uniformly random games and 50 games using the Gaussian distribution. We also generated more structured but still random games, such as games with low rank. We present here the comparisons for the uniformly random games and we refer to the full version for the other classes of games. As in Section 4.2, we used two different initializations: starting from $(\boldsymbol{e}_1, \boldsymbol{e}_1)$ and starting from the uniform strategy profile: $(\frac{1}{n}, \dots, \frac{1}{n})$. The average running time can be seen in Figure 3. We summarize our findings as follows:

- The LP solver was far slower, even for lower values of n, as shown in the left subplot, and we dropped it from the experiments with larger games.
- When the initialization is (e_1, e_1) (or any pure strategy profile), the advantage of our method is more clear (see left subplot of Figure 3). When we start with the uniform profile, we observe that our method is slower for smaller games but becomes faster in very large games (right subplot).
- Another observation is that our method seems smoother with less sharp jumps than OGDA when starting from $(\boldsymbol{e}_1, \boldsymbol{e}_1)$ while the opposite holds for the uniform profile.

We view as the main takeaway of our experiments that our method is comparable to OGDA and in several cases even outperforms OGDA. One limitation of our current implementation is the choice of $\delta = 0.01$. For much lower accuracies, our methods occasionally get stuck. We therefore feel that the overall approach deserves further exploration, especially on potential ways of accelerating its execution.

5 Conclusions

We have analyzed a descent-based method for the duality gap in zero-sum games. Our goal has been to demonstrate the potential of such algorithms as a proof of concept. We expect that our method can be further optimized in practice and find this a promising direction for future work. In particular, one idea to explore is whether we can reuse the LP solutions we get in Algorithm 2 from one iteration to the next (since we only change the current solution slightly by a step of size ϵ). Exploring such *warm start* strategies (see e.g. [Yildirim and Wright, 2002]) could provide significant speedups.

Acknowledgments

This work was partially supported by the framework of the H.F.R.I call “Basic research Financing (Horizontal support of all Sciences)” under the National Recovery and Resilience Plan “Greece 2.0” funded by the European Union – NextGenerationEU (H.F.R.I. Project Number: 15877), by the research project ”Learning approaches for solution concepts in games (LASCON)” of the internal grants 2022 of the ICS-FORTH, and by the project MIS 5154714 of the National Recovery and Resilience Plan Greece 2.0 funded by the European Union under the NextGenerationEU Program.

References

[Adler, 2013] Ilan Adler. The equivalence of linear programs and zero-sum games. *Int. J. Game Theory*, 42(1):165–177, 2013.

[Arora *et al.*, 2012] Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory Comput.*, 8(1):121–164, 2012.

[Bailey and Piliouras, 2018] James P. Bailey and Georgios Piliouras. Multiplicative weights update in zero-sum games. In *Proceedings of the Conference on Economics and Computation (EC'18)*, pages 321–338, 2018.

[Brown, 1951] George W. Brown. Iterative solution of games by fictitious play. In T. C. Koopmans, editor, *Activity Analysis of Production and Allocation*. Wiley, New York, 1951.

[Cai and Zheng, 2023] Yang Cai and Weiqiang Zheng. Doubly optimal no-regret learning in monotone games. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 3507–3524. PMLR, 2023.

[Cai *et al.*, 2022] Y. Cai, A. Oikonomou, and W. Zheng. Finite-time last-iterate convergence for learning in multi-player games. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS'22)*, 2022.

[Cai *et al.*, 2024] Yang Cai, Gabriele Farina, Julien Grand-Clément, Christian Kroer, Chung-Wei Lee, Haipeng Luo, and Weiqiang Zheng. Fast last-iterate convergence of learning in games requires forgetful algorithms. *CoRR*, abs/2406.10631, 2024.

[Daskalakis and Panageas, 2019] Constantinos Daskalakis and Ioannis Panageas. Last-iterate convergence: Zero-sum games and constrained min-max optimization. In *Proceedings of the ITCS'19*, 2019.

[Daskalakis *et al.*, 2018] Constantinos Daskalakis, Andrew Ilyas, Vasilis Syrgkanis, and Haoyang Zeng. Training GANs with optimism. In *Proceedings of the International Conference on Learning Representations (ICLR'18)*, 2018.

[Daskalakis *et al.*, 2021] Constantinos Daskalakis, Stratis Skoulakis, and Manolis Zampetakis. The complexity of constrained min-max optimization. In *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 1466–1478. ACM, 2021.

[Deligkas *et al.*, 2017] A. Deligkas, J. Fearnley, R. Savani, and P. G. Spirakis. Computing approximate Nash equilibria in polymatrix games. *Algorithmica*, 77(2):487–514, 2017.

[Deligkas *et al.*, 2023] Argyrios Deligkas, Michail Fasoulakis, and Evangelos Markakis. A polynomial-time algorithm for 1/3-approximate Nash equilibria in bimatrix games. *ACM Trans. Algorithms*, 19(4):31:1–31:17, 2023.

[Diakonikolas *et al.*, 2021] Jelena Diakonikolas, Constantinos Daskalakis, and Michael I. Jordan. Efficient methods for structured nonconvex-nonconcave min-max optimization. In Arindam Banerjee and Kenji Fukumizu, editors, *The 24th International Conference on Artificial Intelligence and Statistics (AISTATS 2021)*, volume 130, pages 2746–2754, 2021.

[Fasoulakis *et al.*, 2022] M. Fasoulakis, E. Markakis, Y. Pantazis, and C. Varsos. Forward looking best-response multiplicative weights update methods for bilinear zero-sum games. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS'22)*, pages 11096–11117, 2022.

[Fasoulakis *et al.*, 2025] Michail Fasoulakis, Evangelos Markakis, Giorgos Roussakis, and Christodoulos Santorinaios. A descent-based method on the duality gap for solving zero-sum games. arXiv:2501.19138, 2025.

[Gilpin *et al.*, 2012] Andrew Gilpin, Javier Pena, and Tuomas Sandholm. First-order algorithm with convergence for-equilibrium in two-person zero-sum games. *Mathematical programming*, 133(1):279–298, 2012.

[Golowich *et al.*, 2020] Noah Golowich, Sarath Pattathil, and Constantinos Daskalakis. Tight last-iterate convergence rates for no-regret learning in multi-player games. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems (NeurIPS 2020)*, 2020.

[Goodfellow *et al.*, 2014] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In *Proceedings of Annual Conference on Neural Information Processing Systems (NIPS '14)*, pages 2672–2680, 2014.

[Gorbunov *et al.*, 2022] Eduard Gorbunov, Adrien B. Taylor, and Gauthier Gidel. Last-iterate convergence of optimistic gradient method for monotone variational inequalities. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems*, 2022.

[Hoda *et al.*, 2010] Samid Hoda, Andrew Gilpin, Javier Pena, and Tuomas Sandholm. Smoothing techniques for computing Nash equilibria of sequential games. *Mathematics of Operations Research*, 35(2):494–512, 2010.

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

[Liang and Stokes, 2019] Tengyuan Liang and James Stokes. Interaction matters: A note on non-asymptotic local convergence of generative adversarial networks. In *Proceedings of The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS'19*, pages 907–915, 2019.

[Lu and Yang, 2023] Haihao Lu and Jinwen Yang. On the infimal sub-differential size of primal-dual hybrid gradient method and beyond. *CoRR*, abs/2206.12061, 2023.

[Mertikopoulos *et al.*, 2018] Panayotis Mertikopoulos, Christos H. Papadimitriou, and Georgios Piliouras. Cycles in adversarial regularized learning. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018*, pages 2703–2717. SIAM, 2018.

[Nash, 1951] J. Nash. Non-cooperative games. *Annals of Mathematics*, 54 (2), 1951.

[Nikaido and Isoda, 1955] H. Nikaido and K. Isoda. Note on noncooperative convex games. *Pacific Journal of Mathematics*, 5(1):807815, 1955.

[Popov, 1980] L. Popov. A modification of the Arrow-Hurwicz method for search of saddle points. *Mathematical notes of the Academy of Sciences of the USSR*, 28:845–848, 1980.

[Raghunathan *et al.*, 2019] Arvind U. Raghunathan, Anoop Cherian, and Devesh K. Jha. Game theoretic optimization via gradient-based Nikaido-Isoda function. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019*, volume 97 of *Proceedings of Machine Learning Research*, pages 5291–5300. PMLR, 2019.

[Robinson, 1951] J. Robinson. An iterative method of solving a game. *Annals of Mathematics*, pages 296–301, 1951.

[Tsaknakis and Spirakis, 2008] H. Tsaknakis and P. G. Spirakis. An optimization approach for approximate Nash equilibria. *Internet Math.*, 5(4):365–382, 2008.

[Von Neumann, 1928] J. Von Neumann. Zur theorie der gesellschaftsspiele. *Math. Ann.*, 100:295–320, 1928.

[Wei *et al.*, 2021] Chen-Yu Wei, Chung-Wei Lee, Mengxiao Zhang, and Haipeng Luo. Linear last-iterate convergence in constrained saddle-point optimization. In *Proceedings of the 9th International Conference on Learning Representations ICLR '21*, 2021.

[Yildirim and Wright, 2002] E. Alper Yildirim and Stephen J. Wright. Warm-start strategies in interior-point methods for linear programming. *SIAM J. Optim.*, 12(3):782–810, 2002.

Improved Last-Iterate Convergence Rates for Bilinear Zero-Sum Games

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

The recent years have seen a surge of interest in algorithms with last-iterate convergence for 2-player games, motivated in part by applications in machine learning. Driven by this, we revisit a variant of Multiplicative Weights Update (MWU), defined recently by Fasoulakis et al. [2022], and denoted as Forward Looking Best Response MWU (FLBR-MWU). These dynamics are based on the approach of extra gradient methods, with the tweak of using a different learning rate in the intermediate step. So far, it has been proved that this algorithm attains asymptotic convergence but no explicit rate has been known. We answer the open question from Fasoulakis et al. by establishing a geometric convergence rate for the duality gap. In particular, we first show such a rate, of the form $O(c^t)$, till we reach an approximate Nash equilibrium, where c is independent of the game parameters (and $c < 1$). We then prove that from that point onwards, the duality gap keeps getting decreased with a geometric rate, albeit with a dependence on the maximum eigenvalue of the Jacobian matrix. Finally, we complement our theoretical analysis with an experimental comparison to OGDA, which ranks among the best last-iterate methods for solving 0-sum games. Although in practice it does not generally outperform OGDA, it is often comparable, with a similar average performance.

1 Introduction

Our work focuses on learning algorithms with convergence guarantees in 2-player zero-sum games. This is by now an extensively studied domain, spanning already a few decades of research progress. Given a game described by its payoff matrix, what we are after here is algorithms that eventually reach a Nash equilibrium, where no player has an incentive to deviate. Some of the earlier and standard results in this area concern convergence *on average*. I.e., it has long been known that by using no-regret algorithms, the empirical average of the players' strategies over time converges to a Nash equilibrium in zero-sum games and to more relaxed equilibrium notions (coarse correlated equilibria) for general games.

In the recent years, the attention of the relevant community has gradually shifted from convergence in expectation to the more robust notion of *last-iterate convergence*, a property most desirable from an application point of view. This means that the strategy profile (x^t, y^t), reached at iteration t of an iterative algorithm, converges to the actual equilibrium as $t \to \infty$. Unfortunately, many of the initially developed methods do not satisfy this property. No-regret algorithms, like the Multiplicative Weights Update (MWU) method, are known to converge only in an average sense. In fact, it was shown in [Bailey and Piliouras, 2018, Mertikopoulos et al., 2018] that several MWU variants do not satisfy last-iterate convergence.

Motivated by these considerations, there has been a series of works within the last decade studying last iterate convergence. The majority of these works has focused on the fundamental class of zero-sum games. Zero-sum games have played an important role in the development of game theory and optimization, and more recently, there has also been a renewed interest, given their relevance in formulating GANs in deep learning [Goodfellow et al., 2014]. The positive results that have been obtained for zero-sum games is that improved variants of Gradient Descent such as the Optimistic Gradient Descent/Ascent method (OGDA), or the Extra Gradient method (EG) attain last iterate convergence. Several other methods have also been obtained and compared to each other w.r.t. convergence rate. Overall, one can say that we have by now a much better understanding for the learning dynamics that converge in zero-sum games.

Despite the positive progress however, there are still several important questions that remain unanswered. First of all, it is often difficult to have tight results in analyzing such learning algorithms. And furthermore, even for bilinear, zero-sum games, the best attainable rate of convergence is not yet fully understood. For the case of convex-concave min-max optimization problems, the currently best rate is $O(1/\sqrt{t})$ in terms of the duality gap [Cai et al., 2022, Gorbunov et al., 2022]. But for the restriction to bilinear games, it is conceivable that better rates could be achieved.

1.1 Our Contributions

We focus on bilinear zero-sum games and we revisit a promising variant of MWU that was defined recently in [Fasoulakis et al., 2022], denoted as Forward Looking Best-Response Multiplicative updates (FLBR-MWU). The dynamics are based on the approach of extra gradient methods, with the tweak of using a different and more aggressive learning rate in the intermediate step. Our main contributions can be summarized as follows:

- So far it was only known that the algorithm attains asymptotic last-iterate convergence, but without any explicit rate. We answer the open question from [Fasoulakis et al., 2022] by establishing concrete rates of convergence. Using the duality gap as our metric, we first show a geometric rate, of the form $O(c^t)$, till we reach an approximate Nash equilibrium, for an appropriate level of approximation. Moreover, c is independent of the payoff matrix, except the dimension (and $c < 1$).
- For games with a unique Nash equilibrium, we further prove that once we reach the approximate equilibrium, the duality gap keeps getting decreased with a geometric rate, till the exact solution, albeit with the caveat that there is a dependence on the Jacobian matrix evaluated at the equilibrium. An analogous result holds for the OMWU method by [Wei et al., 2021], but for the KL divergence, and with different dependence on the game parameters.
- We then investigate further properties of FLBR. We prove that it is not a no-regret algorithm, which was not known before. At the same time, we explore aspects of *forgetfulness*, as introduced recently in [Cai et al., 2024]. We show that in contrast to OMWU, FLBR seems to exhibit forgetfulness, which serves as an indication for fast performance.
- Finally, we perform an experimental comparison against OGDA, which is among the best known methods for solving zero-sum games. The results reveal that FLBR is generally competitive against OGDA. While it does not outperform OGDA, it has a similar performance on average.

Overall, we believe our work provides a more complete treatment on the power and limitations of this method.

1.2 Related work

There is by now a vast literature on solving zero-sum games. Given the connection with linear programming, a variety of algorithms focus on optimization and LP-based methods for zero-sum games. Theoretically, the best guarantees for solving the corresponding linear program can be found in [Cohen et al., 2021] and [van den Brand et al., 2021]. Regarding other methods, Hoda et al. [2010] use Nesterov's first order smoothing techniques to achieve an ε-equilibrium in $O(1/\varepsilon)$ iterations, with added benefits of simplicity and rather low computational cost per iteration. Following up on that work, Gilpin et al. [2012] propose an iterated version of Nesterov's smoothing technique, which runs within $O(\frac{||A||}{\delta(A)} \cdot \ln(1/\varepsilon))$ iterations. This is a significant improvement, with the caveat that the complexity depends on a condition measure $\delta(A)$, with A being the payoff matrix.

In addition to the above, there has been great interest in designing faster learning algorithms for zero-sum games. Although this direction started already several decades ago, e.g. with the fictitious play algorithm [Brown, 1951, Robinson, 1951], it has received significant attention more recently given the relevance to formulating GANs in deep learning [Goodfellow et al., 2014] and also other applications in machine learning. Some of the earlier and standard results in this area concern convergence *on average*. That is, it has been known that by using no-regret algorithms, such as the Multiplicative Weights Update (MWU) methods [Arora et al., 2012] the empirical average of the players' strategies over time converges to a Nash equilibrium in zero-sum games. Similarly, one could also utilize the so-called Gradient Descent/Ascent (GDA) algorithms. Several other algorithms for zero-sum games are built within the framework of regret minimization both in theory [Carmon et al., 2019, 2024] and in applications [Farina et al., 2021].

Coming closer to our work, within the last decade, there has also been a great interest in algorithms attaining the more robust notion of *last-iterate convergence*. This means that the strategy profile (x_t, y_t), reached at iteration t, converges to the actual equilibrium as $t \to \infty$. Negative results in [Bailey and Piliouras, 2018] and [Mertikopoulos et al., 2018] exhibit that several no-regret algorithms such as many MWU as well as GDA variants, do not satisfy last-iterate convergence. Instead they may diverge or enter a limit cycle. Motivated by this, there has been a series of works on obtaining algorithms with provable last iterate convergence. The positive results that have been obtained for zero-sum games is that improved versions of Gradient Descent such as the Extra Gradient method [Korpelevich, 1976] or the Optimistic Gradient method [Popov, 1980] attain last iterate convergence. In particular, Daskalakis et al. [2018] and Liang and Stokes [2019] show that the optimistic variant of GDA (referred to as OGDA) converges for zero-sum games. Analogously, OMWU (the optimistic version of MWU) also attains last iterate convergence, shown in [Daskalakis and Panageas, 2019] and further analyzed in [Wei et al., 2021]. Further approaches with convergence guarantees have also been proposed, such as primal-dual hybrid gradient methods [Lu and Yang, 2023]. For the case of constrained bilinear zero-sum games, the best convergence rate for the duality gap achieved so far is by [Cai et al., 2022, Gorbunov et al., 2022], which is $O(1/\sqrt{t})$. We note that better rates are achievable for the case of unconstrained bilinear zero-sum games, as e.g., in [Mokhtari et al., 2020], but this is an easier problem from what we focus on here. We also note that for the metric of KL divergence, Wei et al. [2021] provide a geometric rate, which is dependent on game parameters.

The method we analyze here is inspired by the general approach of extra gradient methods, but with the tweak of using different learning rates in the intermediate and final step of each iteration. To our knowledge, the idea of using different rates in these two steps of each iteration has also been successful in other recent works as well. It has been used in [Azizian et al., 2020] for a model that concerns the unconstrained bilinear case. Again for the unconstrained case (but beyond bilinear and convex-concave), the work of Diakonikolas et al. [2021] showed how the use of different learning rates achieved convergence guarantees for their method (referred to as EG+). Finally these ideas have also been applied successfully for stochastic gradient methods [Hsieh et al., 2020].

Finally, several of these methods have also been studied beyond bilinear games, including among others [Golowich et al., 2020] and also [Diakonikolas et al., 2021] where positive results are shown for a class of non-convex and non-concave problems. There are also negative results however as e.g., established in [Daskalakis et al., 2021]. Going beyond min-max problems, the work of Patris and Panageas [2024] obtains last-iterate convergence rates in rank-1 games. Results for richer classes of games are provided in [Anagnostides et al., 2022], including potential and constant-sum polymatrix games. The landscape however is overall less clear.

2 Preliminaries

We consider 2-player, $n \times n$, zero-sum games $(R, -R)$. Without loss of generality, we consider that $R \in (0, 1]^{n \times n}$ is the payoff matrix of the row player, and $-R$ is the payoff matrix of the column player.[1] A (mixed) strategy profile for a player is a probability distribution $x = (x_1, \ldots, x_n)^T$, with e_i^T be vector the pure strategy i with 1 in the index i and zero elsewhere. The support of a mixed strategy x is the set of the pure strategies to which x assigns mass, i.e. $supp(x) = \{i : x_i > 0\}$.

A strategy profile is a tuple (x, y) where x (resp. y) is the strategy of the row (resp. column) player. Given a profile (x, y), the expected payoff of the row (resp. column) player is $x^T Ry$ (resp $-x^T Ry$).

[1]Any game can be transformed to a game with entries in the interval $(0, 1]$ with the same Nash equilibria.

Definition 1 (Nash equilibrium (NE))**.** *A strategy profile* (x^*, y^*) *is a Nash equilibrium of the game* $(R, -R)$ *if and only if, for any* i, j*,*

$$v = (x^*)^T Ry^* \geq e_i^T Ry^*, \text{ and } v = (x^*)^T Ry^* \leq x^{*T} Re_j.$$

where v is the value of the row player (commonly referred to as the value of the game).

Definition 2 (ε-Nash equilibrium (ε-NE))**.** *A strategy profile* (x, y) *is an* ε*-Nash equilibrium of the game* $(R, -R)$*, with* $R \in [0,1]^{n\times n}$*, for* $\varepsilon \in [0,1]$*, if and only if, for any* $i, j \in [n]$*,*

$$x^T Ry + \varepsilon \geq e_i^T Ry, \text{ and } x^T Ry - \varepsilon \leq x^T Re_j.$$

We can see that any 0-NE is an exact NE. Next we will define our progress measure.

Definition 3. *For zero-sum games, the duality gap function* V *is defined as*

$$V(x, y) = \max_i e_i^T Ry - \min_j x^T Re_j.$$

The duality gap is a central notion in game theory as it captures the combined loss of the players for not employing best responses and hence for deviating from a NE, as seen in the fact below.

Fact 1. *A strategy profile* (x^*, y^*) *is a Nash equilibrium of the zero-sum game* $(R, -R)$*, if and only if, is a (global) minimum of the function* $V(x, y)$*. Furthermore, if* $V(x, y) \leq \varepsilon$*, then* (x, y) *is an* ε*-NE.*

Before proceeding with the dynamics, we state a simple lemma that relates the L_1 norm with the duality gap function and defer its proof in Appendix A.

Lemma 1. *For any* y *it holds that* $\max_i e_i^T Ry \leq ||y - y^*||_1 + v$*. Similarly for the row player.*

2.1 FLBR-MWU dynamics

Here we restate the Forward Looking Best-Response Dynamics as introduced in [Fasoulakis et al., 2022]. These dynamics followed an extra gradient approach to find a Nash Equilibrium. Specifically, in each iteration there exists an intermediate step which is used as a prediction for the update step. The difference with other extra gradient-like approaches is that different learning rates are used in the intermediate and the final step, which appear crucial to the effectiveness of this approach.

Given an initial strategy profile (x^0, y^0), the two steps of the dynamics can be described as following:

$$\text{Step 1 (Intermediate): } \hat{x}_i^t = x_i^{t-1} \cdot \frac{e^{\xi \cdot e_i^\top Ry^{t-1}}}{\sum_j x_j^{t-1} \cdot e^{\xi \cdot e_j^\top Ry^{t-1}}}, \text{ and } \hat{y}_j^t = y_j^{t-1} \cdot \frac{e^{-\xi \cdot e_j^\top R^\top x^{t-1}}}{\sum_i y_i^{t-1} \cdot e^{-\xi \cdot e_i^\top R^\top x^{t-1}}},$$

$$\text{Step 2 (Update): } \quad x_i^t = x_i^{t-1} \cdot \frac{e^{\eta \cdot e_i^\top R\hat{y}^t}}{\sum_j x_j^{t-1} \cdot e^{\eta \cdot e_j^\top R\hat{y}^t}}, \quad \text{ and } y_j^t = y_j^{t-1} \cdot \frac{e^{-\eta \cdot e_j^\top R^\top \hat{x}^t}}{\sum_i y_i^{t-1} \cdot e^{-\eta \cdot e_i^\top R^\top \hat{x}^t}},$$

The rationale is that a large value for ξ is used (aggressive rate for the intermediate step) together with a small (conservative) learning rate $\eta \in (0,1)$ for the update step. Finally, we state an important property that we will use at various points in the sequel.

Lemma 2 ([Fasoulakis et al., 2022])**.** *For any* $t > 0$*, it holds that as* $\xi \to \infty$ *then* $\hat{x}^t$ *(resp.* $\hat{y}^t$*) converges to a best response strategy against* y^{t-1} *(resp. against* x^{t-1}*).*

Assumption 1: We will start the dynamics from the fully uniform distribution, i.e., $x^0 = y^0 = (1/n, \ldots, 1/n)$. Furthermore, we will use a fixed η, independent of t in all iterations.

3 Convergence analysis

In this section, we use the duality gap as a metric to study the rate of convergence for FLBR-MWU. This provides an answer to the question left open by [Fasoulakis et al., 2022]. Our analysis consists of two parts. Firstly, we obtain a geometric rate of convergence till an appropriate approximate equilibrium is reached, where the degree of approximation is dependent on η. Then, we show that if η is sufficiently small, so as to guarantee that we are close to the exact equilibrium, we can maintain a geometric rate all the way to it, at the cost of introducing a dependency on the game parameters.

3.1 Convergence to an approximate equilibrium

Let (x^*, y^*) be an arbitrary exact Nash equilibrium and let (x^t, y^t) be the strategy profile produced by the dynamics at the end of time step t. We stress that for the convergence to an approximate equilibrium, we do not need to assume uniqueness.

In our analysis, we will utilize the *Kullback-Leibler (KL)* divergence of a profile from (x^*, y^*), defined as follows.

$$D_{KL}((x^*, y^*)||(x^t, y^t)) = \sum\nolimits_{i=1}^n x_i^* \cdot \ln(x_i^*/x_i^t) + \sum\nolimits_{j=1}^n y_j^* \cdot \ln(y_j^*/y_j^t).$$

Note that by the definition of the dynamics, x_i^t and y_j^t are always positive for any i, j and any t, hence the ratios above are well-defined. The main technical property for the analysis of reaching an approximate equilibrium is the following lemma.

Lemma 3. *It holds that for any $t \geq 1$, and any $\eta \leq 1/2$*

$$\eta \cdot \left((\hat{x}^t)^T R y^{t-1} - (x^{t-1})^T R \hat{y}^t\right) \leq D_{KL}((x^*, y^*)||(x^{t-1}, y^{t-1})) - D_{KL}((x^*, y^*)||(x^t, y^t)) + 4\eta^2.$$

This lemma is crucial as it gives us a way to correlate the duality gap with the KL divergence. In particular, the left hand side of the formula is a proxy quantity for the duality gap, and converges to it should we choose a large enough ξ, as established in the following claim.

Claim 1. *For any $t \geq 1$, it holds that $\lim_{\xi \to \infty}[(\hat{x}^t)^T R y^{t-1} - (x^{t-1})^T R \hat{y}^t] = V(x^{t-1}, y^{t-1})$.*

From this we have the following:

Corollary 1. *It holds that for any $t \geq 1$, for any $\eta \leq 1/2$, and for large enough ξ that*

$$V(x^{t-1}, y^{t-1}) \leq \frac{D_{KL}((x^*, y^*)||(x^{t-1}, y^{t-1})) - D_{KL}((x^*, y^*)||(x^t, y^t))}{\eta} + 5\eta.$$

The next theorem is the main result of Section 3.1.

Theorem 1. *Fix a constant $\rho > 1$. Under Assumption 1, and for sufficiently large ξ, the rate of convergence for the KL divergence till we reach an $O(\eta^{1/\rho})$-Nash equilibrium is inverse exponential, in the form $O\left(\frac{\ln n}{c^t}\right)$, where c is independent of t and dependent on n and η, defined in (1), with $c > 1$. Similarly, the convergence rate of the duality gap to reach an $O(\eta^{1/\rho})$-Nash equilibrium is inverse exponential, in the form $O\left(\frac{\ln n}{\eta \cdot c^t}\right)$, where c is as above.*

Proof. The first part of the proof, regarding the KL divergence, follows by exploiting further some properties established already in [Fasoulakis et al., 2022], whereas the second part exploits Lemma 3.

For the KL divergence, fix a constant $\gamma > 5$. By the proof of Theorem 2 in [Fasoulakis et al., 2022], we know that as long as we have not reached a $\gamma\eta^{1/\rho}$-equilibrium, the KL divergence decreases in each iteration by at least $\beta\eta^{1+1/\rho}$, where β is a constant.[2] Therefore, we have that for any time t till we reach such an equilibrium, it holds that

$$\begin{aligned} D_{KL}((x^*, y^*)||(x^t, y^t)) &\leq D_{KL}((x^*, y^*)||(x^{t-1}, y^{t-1})) - \beta\eta^{1+1/\rho} \\ &= D_{KL}((x^*, y^*)||(x^{t-1}, y^{t-1}))\left(1 - \frac{\beta\eta^{1+1/\rho}}{D_{KL}((x^*, y^*)||(x^{t-1}, y^{t-1}))}\right). \end{aligned}$$

Since we assumed that we start with the uniform profile, and since the KL divergence only decreases till we reach an approximate equilibrium, we have that $D_{KL}((x^*, y^*)||(x^{t-1}, y^{t-1})) \leq D_{KL}((x^*, y^*)||(x^0, y^0)) \leq 2\ln(n)$. Therefore we conclude that

$$D_{KL}((x^*, y^*)||(x^t, y^t)) \leq D_{KL}((x^*, y^*)||(x^{t-1}, y^{t-1}))\left(1 - \frac{\beta\eta^{1+1/\rho}}{2\ln(n)}\right).$$

[2] By following the calculations in [Fasoulakis et al., 2022], it can be derived that $\beta \geq \gamma - 5$.

Let $a = \frac{\beta\eta^{1+1/\rho}}{2\ln(n)}$. Since we have assumed that η is constant, independent of t, then a is also independent of t. Note also that for large enough n, $a < 1$. Therefore, by unrolling the above inequality for the previous time steps before t, in total we get

$$D_{KL}((x^*, y^*)||(x^t, y^t)) \leq D_{KL}((x^*, y^*)||(x^0, y^0))(1-a)^t = 2\ln(n)(1-a)^t.$$

This means that the KL divergence at time t is bounded by $\frac{2\ln(n)}{c^t}$, where c is independent of t and dependent on η and n, with $c > 1$. In particular:

$$c = \frac{1}{1-a} = \frac{2\ln(n)}{2\ln(n) - \beta\eta^{1+1/\rho}} \tag{1}$$

Coming now to the duality gap, we conclude by Corollary 1 that

$$V(x^t, y^t) \leq \frac{D_{KL}((x^*, y^*)||(x^t, y^t))}{\eta} + 5\eta \leq \frac{2\ln(n)(1-a)^t}{\eta} + 5\eta. \tag{2}$$

Note also that since we have not yet reached a $\gamma\eta^{1/\rho}$-equilibrium, with γ be a constant greater than 5, it holds that $V(x^t, y^t) \geq \gamma\eta^{1/\rho} \geq \gamma\eta$. Combining this with the above upper bound implies that for any time step t, till we reach an approximate equilibrium, we have that $\eta \leq \frac{2\ln(n)(1-a)^t}{(\gamma-5)\eta}$. By plugging this in (2), we eventually get:

$$V(x^t, y^t) \leq \frac{2\gamma\ln(n)}{(\gamma-5)\eta}(1-a)^t. \qquad \square$$

3.2 Convergence to an exact equilibrium under uniqueness

We proceed here to analyze the convergence till the method reaches an exact equilibrium. The technique here is based on a spectral analysis. and for this, we will need to further assume that the game has a unique Nash equilibrium (x^*, y^*). This is a rather common assumption in many related works, and we do not view this as a severe restriction, since the set of zero-sum games with non unique NE has Lebesgue measure equal to zero, [Van Damme, 1991].

Let t_0 be the time at which we reach the approximate equilibrium described in Section 3.1 and let (x^{t_0}, y^{t_0}) be the corresponding strategy profile. By Theorem 1, it can be extracted that $t_0 = O(\ln\ln(n)/\ln(\eta^{1+1/\rho}))$. The first step in the remaining analysis is to establish that this approximate equilibrium can be close to the actual Nash equilibrium. This is ensured if η is sufficiently small.

Corollary 2 (implied by Theorem 3 in [Fasoulakis et al., 2022]). *For any $\delta > 0$, and for any $q \geq 1$, there exists a sufficiently small η, such that $||(x^*, y^*) - (x^{t_0}, y^{t_0})||_q \leq \delta$.*

Using the above, the asymptotic last-iterate convergence of FLBR (but without a rate) was established in [Fasoulakis et al., 2022] by proving that the maximum eigenvalue of the Jacobian matrix at (x^*, y^*) is strictly less than 1. In order to obtain a rate of convergence, we give a more refined analysis, based on a technique utilized in [Nakagawa et al., 2021] (namely within the proof of their Theorem 5) for a fundamental problem in information theory.[3]

Theorem 2. *Let $(R, -R)$ be a zero-sum game with a unique NE (x^*, y^*). For a sufficiently small η and large enough ξ, such that $\eta\xi < 1$, the rate of convergence of the duality gap to the NE is inverse exponential for the FLBR dynamics, in the form A/b^t, where A and b are determined by the norm of the Jacobian matrix evaluated at (x^*, y^*).*

Proof. First, we recall some basic facts established in [Fasoulakis et al., 2022] that we use here, and for which uniqueness of equilibrium was needed. FLBR can be easily described as a discrete dynamical system, $\varphi(x, y) = (\varphi_1(x, y), \varphi_2(x, y))$, where $\varphi(x^t, y^t) = (x^{t+1}, y^{t+1})$, and where $\varphi_{1,i}(x, y)$ is the i-th coordinate of $\varphi_1(x, y)$ and similarly for $\varphi_{2,i}(x, y)$, for any $i \in [n]$. The Jacobian of this system is a $2n \times 2n$ matrix, determined by the partial derivatives of ϕ. Furthermore, when

[3] In particular, the problem tackled by [Nakagawa et al., 2021] was the convergence analysis of the Arimoto-Blahut algorithm for computing the Shannon's capacity of a discrete memoryless channel.

there exists a unique NE and $\eta\xi < 1$, [Fasoulakis et al., 2022] proved that there exists some $q \geq 1$, such that

$$\lambda_{\max} \leq ||J(x^*, y^*)||_q < 1,$$

where $\lambda_{\max}$ is the maximum eigenvalue of the Jacobian matrix at the profile (x^*, y^*).

For any $t \geq 0$, consider the strategy profile $(x(p), y(p)) = (1-p) \cdot (x^*, y^*) + p \cdot (x^t, y^t)$, with $p \in (0, 1)$, as a convex combination of the equilibrium and the profile (x^t, y^t). In our proof, we will eventually need to argue about the Jacobian matrix at such convex combinations (proof of Lemma 4).

Lemma 4. *For $t \geq t_0$: $||(x^{t+1}, y^{t+1}) - (x^*, y^*)||_q \leq ||(x^t, y^t) - (x^*, y^*)||_q \cdot ||J(x(p^t), y(p^t))||_q$.*

With the above lemma and the continuity of the norm, we can now prove by induction the following:

Lemma 5. *Given $\varepsilon > 0$, there exists a sufficiently small $\delta > 0$, such that if $||(x^{t_0}, y^{t_0}) - (x^*, y^*)||_q \leq \delta$, then for any $t \geq t_0$. $||J(x(p^t), y(p^t))||_q < ||J(x^*, y^*)||_q + \varepsilon$.*

Fix now a small $\varepsilon > 0$ and let $\lambda = ||J(x^*, y^*)||_q + \varepsilon$ so that $\lambda < 1$. By Lemma 5 and applying repeatedly Lemma 4, we have that, for any $t \geq t_0, ||(x^t, y^t) - (x^*, y^*)||_q < \lambda^{t-t_0} \cdot ||(x^{t_0}, y^{t_0}) - (x^*, y^*)||_q$. Therefore, given $\varepsilon > 0$, if we pick a sufficiently small η, we can ensure that there exists a small $\delta > 0$, such that Corollary 2 holds with this δ, i.e., $||(x^{t_0}, y^{t_0}) - (x^*, y^*)||_q < \delta$, and at the same time Lemma 5 holds with the chosen ε (and again for this δ). By the equivalence of the norms, all these yield that $||(x^t, y^t) - (x^*, y^*)||_1 < K \cdot \delta \cdot \lambda^{t-t_0}$, for some integer $K > 0$ independent of t, and dependent on q. This directly bounds the L_1 distances from the equilibrium strategies and by applying Lemma 1, we conclude that

$$V(x^t, y^t) \leq 2K \cdot \delta \cdot \lambda^{t-t_0} + v - v = O(K \cdot \delta \cdot \lambda^t). \qquad \square$$

4 Regret and forgetfulness

In this section, we focus on some previously unexplored aspects of the FLBR method.

4.1 Regret analysis

First and most importantly, a fundamental question is whether FLBR is a no-regret algorithm, for which we provide a negative answer. So far, in the literature of methods with last-iterate convergence, there exist both no-regret algorithms (such as Optimistic MWU [Daskalakis and Panageas, 2019]) and algorithms with regret (such as Extra Gradient). We note that the existence of regret by itself is not necessarily a negative indication for the performance of an algorithm. For example, OMWU is outperformed by algorithms that have regret, as discussed in [Cai et al., 2024].

Theorem 3. *FLBR is not a no-regret algorithm when ξ is sufficiently large.*

We provide a proof outline here, and defer the proofs of the lemmas that we use below to the Appendix. We first restate the FLBR dynamics, so that each iteration is replaced by two steps. We do this so as to explicitly view FLBR within the framework of online learning algorithms with gradient feedback. Hence in each step, each player observes the payoff of her pure strategies[4] and updates the mixed strategy accordingly. This gives the following formulation for the row player (and analogously for the column player). For technical convenience, we assume the initial profile is indexed as (x^{-1}, y^{-1}):

$$x_i^{2t} = x_i^{2t-1} \cdot \frac{e^{\xi \cdot e_i^\top R y^{2t-1}}}{\sum_j x_j^{2t-1} \cdot e^{\xi \cdot e_j^\top R y^{2t-1}}} \text{ and } x_i^{2t+1} = x_i^{2t-1} \cdot \frac{e^{\eta \cdot e_i^\top R y^{2t}}}{\sum_j x_j^{2t-1} \cdot e^{\eta \cdot e_j^\top R y^{2t}}}, \quad t \geq 0 \tag{3}$$

The example that we use for proving the theorem is the simple Matching-Pennies game:

$$R = \begin{bmatrix} +1 & -1 \\ -1 & +1 \end{bmatrix}$$

We use as the initialization $x^{-1} = (1-\delta, \delta)$ and $y^{-1} = (\delta, 1-\delta)$, for some small $\delta \in (0, 1/2)$. With this at hand, we can break down the proof of Theorem 3 in the lemmata that follow. For simplicity, we will carry out the proof here assuming $\xi \to \infty$. Under this, note that by Lemma 2, x^0 is a best response to y^{-1}, and hence we get that $x^0 = (0, 1)$. In fact we can inductively extend this argument.

[4] Note that this is precisely the gradient information, since e.g. $\frac{\partial (x^t)^T R y^t}{\partial x_i} = e_i^T R y^t$.

Claim 2. *For any $t \geq 0$, it holds that $x_1^{2t-1} > \frac{1}{2}$ and $y_1^{2t-1} < \frac{1}{2}$.*

Pairing this with Lemma 2, we get that $x^{2t} = (0,1)$, as a best response to y^{2t-1}, for any t, and symmetrically $y^{2t} = (0,1)$. Now we are in position to explicitly compute x_1^{2t-1}.

Lemma 6. *For sufficiently large ξ we get $x_1^{2t+1} = (1-\delta)(1-\delta(1-e^{2\eta(t+1)}))$.*

Clearly we also have $x_2^{2t+1} = 1 - x_1^{2t+1}$. Due to symmetry we obtain that $y_2^{2t+1} = x_1^{2t+1}$ and thus, we have obtained a closed form for the dynamics. The proof is then completed by the next lemma.

Lemma 7. *For δ sufficiently small, the regret of the algorithm, for the row player against the fixed strategy $x = (0,1)$, up until time T is $\Omega(T)$.*

4.2 Forgetfullness

In a very recent work, Cai et al. [2024] provided further insights on the performance of OMWU and related dynamics, as compared against OGDA. Their work was motivated by Panageas et al. [2023], where analogous intuitions were given for the fictitious play algorithm. In a nutshell, Cai et al. [2024] attributed the cause of relatively slow convergence of OMWU to a notion they term "forgetfullness". Although they did not provide a formal definition, intuitively, if a method is not forgetful, the produced strategies can get stuck to almost the same profile over many iterations, which slows down convergence. It was shown that this can occur under OMWU, whereas OGDA does not exhibit the same issues. Therefore, the main conclusion of their work is that forgetfulness seems to be a necessary condition for faster performance. Here we extend their experiment, comparing OGDA and FLBR-MWU. The hard game instance of [Cai et al., 2024] for OMWU, parameterized by $\delta \in (0,1)$, is the following:

$$A_\delta = \begin{bmatrix} \frac{1}{2}+\delta & \frac{1}{2} \\ 0 & 1 \end{bmatrix}$$

The game has a unique equilibrium (x^*, y^*) where $x_1^* = \frac{1}{1+\delta}$ and $y_1^* = \frac{1}{2(1+\delta)}$. In Figure 1, we highlight the behavior of FLBR and OGDA. In the upper subfigures, we show how the first coordinate of x^t and y^t vary over time, with the initialization $(x^0, y^0) = (1/2, 1/2)$. In the lower subfigures, we show the decrease in the duality gap over the iterations. Note that at the equilibrium, x_1^* is close to 1, whereas y_1^* is close to 1/2, and thus close to y_1^0. What we observe is that FLBR does behave similarly to OGDA in the sense that it forgets fast, regarding the coordinate x_1^t, and therefore avoiding slowdowns. But furthermore, FLBR does not overshoot y_1^t. It increases it marginally before reaching the actual equilibrium point, whereas OGDA overshoots before reaching the equilibrium. This fact justifies the much faster convergence time of FLBR against OGDA, seen in the lower subfigures.

Overall, even though this was only one example, it conveys the intuition that the intermediate step at FLBR, using large ξ has a particular effect in the dynamics: it makes the algorithm forgetful, and thus faster, albeit with the cost of adding regret, as shown in Section 4.1.

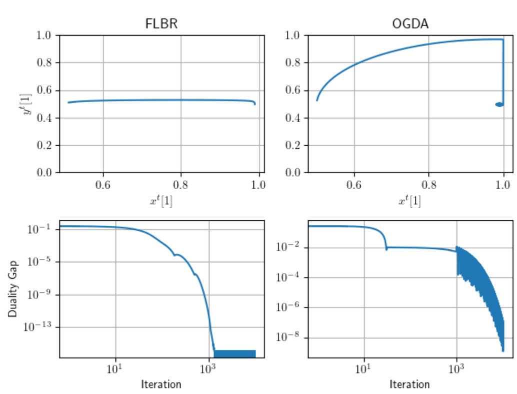

Figure 1: FLBR vs OGDA in game A_δ.

5 Experimental evaluation

Experimentally, the method had seemed to be promising already in [Fasoulakis et al., 2022], where it was shown to significantly outperform OMWU. Our experimental evaluation compares FLBR with OGDA, which is one the fastest and most well studied last-iterate method for bilinear games [Daskalakis et al., 2018].

We have performed 3 types of comparisons. Firstly, we compare the two methods on synthetic data, i.e., with random matrices populated from a standard Gaussian distribution. Then we revisit the game A_δ discussed in Section 4.2. To avoid any potential pitfalls from finetuning the step parameter, we repeat that experiment with different ηs. And third, since A_δ is only a 2×2 game, in order to get more meaningful comparisons we have selected additional games that are simultaneously far from random and larger in size. To that end, we used the generalized Rock Paper Scissors game of higher dimensions. Additional experiments are also presented in Appendix D.

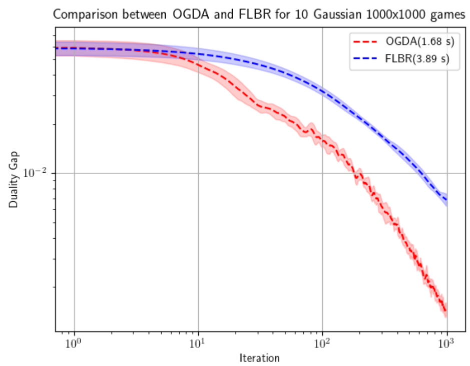

Figure 2: FLBR vs OGDA in Gaussian games

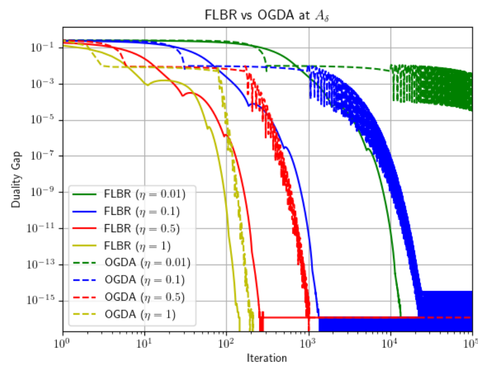

Figure 3: Further comparisons for game A_δ

Our main findings and conclusions are as follows:

- In terms of how to tune the parameters η and ξ, we use a $\xi = 100$ in every experiment involving FLBR and $\eta = 0.25$ in Figure 2. For the rest of the figures, ηs are specified in the legends.
- In Figure 2 we see the comparisons on Gaussian random games, on 1000×1000 games. The methods are comparable up to a point, with OGDA being better both in the number of iterations needed and the time elapsed per game. Nevertheless, FLBR is still close enough. The performance of OGDA can be explained by Anagnostides and Sandholm [2024], and their last iterate analysis on OGDA under the celebrated framework of *smoothed analysis*, [Spielman and Teng, 2004].
- In Figure 3, we see the comparisons for the game A_δ from [Cai et al., 2024]. Here the conclusion reverses: the methods are comparable once again but now FLBR comes on top. Moreover, the smaller η used the larger the performance gap.
- In Figures 4 and 5, we see the comparisons for generalized Rock Paper Scissors, for dimensions 11 and 101, and for various values of η. Again the methods are comparable with a slight advantage for FLBR.

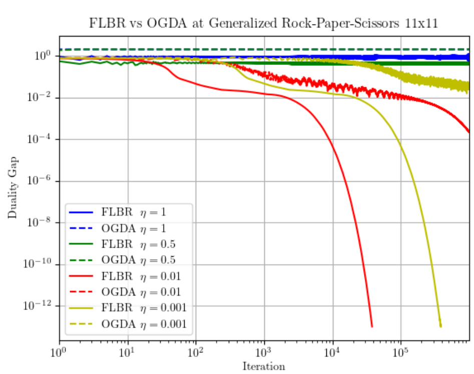

Figure 4: Comparisons over various values of η

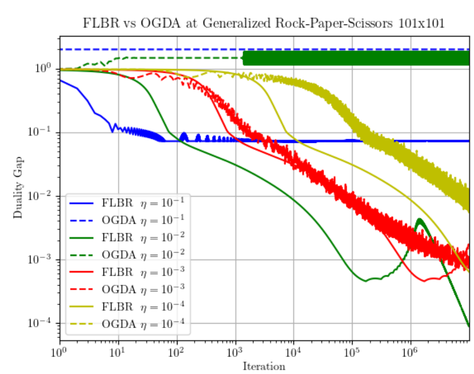

Figure 5: RPS games of higher dimension

Overall, even though the theoretical analysis of FLBR comes with the caveat of game-dependent parameters in its geometric convergence rate, the experiments reveal a competitive performance against OGDA. One more conclusion that arises from the experiments (esp. from Figures 4 and 5) is that FLBR seems to exhibit better robustness when varying η, unlike OGDA. We therefore conclude that the combination of different learning rate parameters, η and ξ, in FLBR can be viewed as a promising direction that could motivate further future works.

References

Ioannis Anagnostides and Tuomas Sandholm. Convergence of $\log(1/\epsilon)$ for gradient-based algorithms in zero-sum games without the condition number: A smoothed analysis. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024*, 2024.

Ioannis Anagnostides, Ioannis Panageas, Gabriele Farina, and Tuomas Sandholm. On last-iterate convergence beyond zero-sum games. In *International Conference on Machine Learning*, pages 536–581. PMLR, 2022.

Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory Comput.*, 8(1):121–164, 2012.

Waïss Azizian, Damien Scieur, Ioannis Mitliagkas, Simon Lacoste-Julien, and Gauthier Gidel. Accelerating smooth games by manipulating spectral shapes. In *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020*, volume 108, pages 1705–1715. PMLR, 2020.

James P. Bailey and Georgios Piliouras. Multiplicative weights update in zero-sum games. In *Proceedings of the Conference on Economics and Computation (EC'18)*, pages 321–338, 2018.

George W. Brown. Iterative solution of games by fictitious play. In *Activity Analysis of Production and Allocation*. 1951.

Yang Cai, Argyris Oikonomou, and Weiqiang Zheng. Finite-time last-iterate convergence for learning in multi-player games. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS'22)*, 2022.

Yang Cai, Gabriele Farina, Julien Grand-Clément, Christian Kroer, Chung-Wei Lee, Haipeng Luo, and Weiqiang Zheng. Fast last-iterate convergence of learning in games requires forgetful algorithms. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024*, 2024.

Yair Carmon, Yujia Jin, Aaron Sidford, and Kevin Tian. Variance reduction for matrix games. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems*, pages 11377–11388, 2019.

Yair Carmon, Arun Jambulapati, Yujia Jin, and Aaron Sidford. A whole new ball game: A primal accelerated method for matrix games and minimizing the maximum of smooth functions. In *Proceedings of the 2024 ACM-SIAM Symposium on Discrete Algorithms, SODA'24*, pages 3685–3723. SIAM, 2024.

Michael B. Cohen, Yin Tat Lee, and Zhao Song. Solving linear programs in the current matrix multiplication time. *J. ACM*, 68(1):3:1–3:39, 2021.

Constantinos Daskalakis and Ioannis Panageas. Last-iterate convergence: Zero-sum games and constrained min-max optimization. In *Proceedings of the ITCS'19*, 2019.

Constantinos Daskalakis, Andrew Ilyas, Vasilis Syrgkanis, and Haoyang Zeng. Training GANs with optimism. In *Proceedings of the International Conference on Learning Representations (ICLR'18)*, 2018.

Constantinos Daskalakis, Stratis Skoulakis, and Manolis Zampetakis. The complexity of constrained min-max optimization. In *53rd Annual ACM SIGACT Symposium on Theory of Computing (STOC '21)*, pages 1466–1478. ACM, 2021.

Jelena Diakonikolas, Constantinos Daskalakis, and Michael I. Jordan. Efficient methods for structured nonconvex-nonconcave min-max optimization. In Arindam Banerjee and Kenji Fukumizu, editors, *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event*, volume 130 of *Proceedings of Machine Learning Research*, pages 2746–2754. PMLR, 2021.

Gabriele Farina, Christian Kroer, and Tuomas Sandholm. Faster game solving via predictive blackwell approachability: Connecting regret matching and mirror descent. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI*, pages 5363–5371. AAAI Press, 2021.

Michail Fasoulakis, Evangelos Markakis, Yannis Pantazis, and Constantinos Varsos. Forward looking best-response multiplicative weights update methods for bilinear zero-sum games. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS'22)*, pages 11096–11117, 2022.

Andrew Gilpin, Javier Pena, and Tuomas Sandholm. First-order algorithm with convergence for-equilibrium in two-person zero-sum games. *Mathematical programming*, 133(1):279–298, 2012.

Noah Golowich, Sarath Pattathil, and Constantinos Daskalakis. Tight last-iterate convergence rates for no-regret learning in multi-player games. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems*, 2020.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In *Proceedings of Annual Conference on Neural Information Processing Systems (NIPS '14)*, pages 2672–2680, 2014.

Eduard Gorbunov, Adrien B. Taylor, and Gauthier Gidel. Last-iterate convergence of optimistic gradient method for monotone variational inequalities. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems*, 2022.

Samid Hoda, Andrew Gilpin, Javier Pena, and Tuomas Sandholm. Smoothing techniques for computing Nash equilibria of sequential games. *Mathematics of Operations Research*, 35(2): 494–512, 2010.

Yu-Guan Hsieh, Franck Iutzeler, Jérôme Malick, and Panayotis Mertikopoulos. Explore aggressively, update conservatively: Stochastic extragradient methods with variable stepsize scaling. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems*, 2020.

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

Tengyuan Liang and James Stokes. Interaction matters: A note on non-asymptotic local convergence of generative adversarial networks. In *Proceedings of The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS'19*, pages 907–915, 2019.

Haihao Lu and Jinwen Yang. On the infimal sub-differential size of primal-dual hybrid gradient method and beyond. *CoRR*, abs/2206.12061, 2023.

Panayotis Mertikopoulos, Christos H. Papadimitriou, and Georgios Piliouras. Cycles in adversarial regularized learning. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018*, pages 2703–2717. SIAM, 2018.

Aryan Mokhtari, Asuman Ozdaglar, and Sarath Pattathil. A unified analysis of extra-gradient and optimistic gradient methods for saddle point problems: Proximal point approach. In *International Conference on Artificial Intelligence and Statistics*, pages 1497–1507. PMLR, 2020.

Kenji Nakagawa, Yoshinori Takei, Shin-ichiro Hara, and Kohei Watabe. Analysis of the convergence speed of the Arimoto-Blahut algorithm by the second-order recurrence formula. *IEEE Transactions on Information Theory*, 67(10):6810–6831, 2021.

Ioannis Panageas, Nikolas Patris, Stratis Skoulakis, and Volkan Cevher. Exponential lower bounds for fictitious play in potential games. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023*, 2023.

Nikolas Patris and Ioannis Panageas. Learning Nash equilibria in rank-1 games. In *Proceedings of the Twelfth International Conference on Learning Representations (ICLR'24), To be appeared*, 2024.

Leonid Denisovich Popov. A modification of the Arrow-Hurwicz method for search of saddle points. *Mathematical notes of the Academy of Sciences of the USSR*, 28:845–848, 1980.

Julia Robinson. An iterative method of solving a game. *Annals of Mathematics*, pages 296–301, 1951.

Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *J. ACM*, 51(3):385–463, 2004. doi: 10.1145/990308.990310.

Eric Van Damme. *Stability and perfection of Nash equilibria*, volume 339. Springer, 1991.

Jan van den Brand, Yin Tat Lee, Yang P. Liu, Thatchaphol Saranurak, Aaron Sidford, Zhao Song, and Di Wang. Minimum cost flows, mdps, and l_1 regression in nearly linear time for dense instances. In *53rd Annual ACM SIGACT Symposium on Theory of Computing (STOC '21)*, pages 859–869. ACM, 2021.

Chen-Yu Wei, Chung-Wei Lee, Mengxiao Zhang, and Haipeng Luo. Linear last-iterate convergence in constrained saddle-point optimization. In *Proceedings of the 9th International Conference on Learning Representations ICLR '21*, 2021.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: All claims are established in Section 3, Section 4 and in the Appendix.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of our theoretical results in the Introduction and in Section 3 (regarding the game-dependent parameters on the formula expressing the convergence rate). Regarding our experimental comparisons, we discuss limitations in Section 5 (when comparing with the OGDA method).

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: All assumptions needed for our theoretical results are listed in the relevant sections and all proofs are provided in Section 3, Section 4 and in the Appendix.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Implementing the dynamics that we study in this paper can be done with a few lines of code (and the same is true for the other algorithm we compare against). Also in Section 5 we describe in detail all the experiments that we performed to compare the algorithms and these datasets can be very easily reproduced.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No] .

Justification: Our code is very simple, as explained in the previous question. Also, all our data were either randomly generated matrices sampled from the Gaussian distribution or matrices that are very easily reproduced (e.g., corresponding to generalized rock-paper-scissors games). All our datasets are also available upon request.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes] .

Justification: All details are explained in Section 5.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes] .

Justification: The only set of experiments that involves randomness is the one concerning random games sampled from the Gaussian distribution. For these games, we report the standard deviation.

Guidelines:

- The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes] .

   Justification: All such details are provided in Section 5.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes] .

   Justification: Our research does not raise any ethical issues.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No] .

    Our research is focused on theoretical foundations and therefore is not tied to any deployment or immediate application that could have direct impact.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA] .

    Justification: The paper does not use existing assets.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: the core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.