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)

WP6: Experimental Evaluation and Further Refinements

Deliverable D.6.2: Technical report on the experimental evaluation and final conclusions

Prepared by the team members:

Michail Fasoulakis, Evangelos Markakis, Michail Samaris

Deliverable D.6.2: Technical Report on the experimental evaluation and final conclusions

1 Introduction

This is the last deliverable of WP6, and it was completed by Month M24, at the end of the project, in accordance to the timetable of ALGOLINC. The goal of the deliverable is to provide an overview of the experimental evaluation we have performed for the algorithms that we have analyzed mathematically within WP4. Namely, we have proven theoretical bounds on the convergence speed of two algorithms within our work in WP4 and the purpose here is to also provide experimental evidence and comparisons against other methods. The issues we discuss here concern the running time and the number of iterations of each algorithms and related properties. The results of the experimental evaluation are also included in the two articles that we have completed on these topics, which we discuss further at the end of this report.

For the sake of completeness, we briefly recall the algorithmic problem we are interested in, regarding zero-sum games, and also recall the experimental setup and the data we used, as described in Deliverable D.6.1.

2 Basic Mathematical Model: Bilinear Zero-sum Games

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 and machine learning, as e.g., they can model the process underlying the training of GANs.

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.

Definition 2.1 (Nash equilibrium [7, 8]). *A strategy profile $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ is a Nash equilibrium in the game $(R, -R)$, if and only if, for any i, j,*

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

[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.

As we are interested in algorithms that converge to equilibria, we also need to utilize some notion of approximating Nash equilibria. We use the following standard approximation 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 particular here, $f(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^\top R\boldsymbol{y}$, which is a bilinear and non-convex function.

3 Synthetic Data Generation

During the project, we decided to evaluate our algorithms based on data synthetically generated. We describe in more detail how we generated our datasets in Deliverable D.6.1. Here we recall briefly the types of games that we used.

- **Random games, sampling payoffs from a probability distribution.** We generated random $n \times n$ games, where each entry of the payoff matrix was filled by sampling from the uniform distribution. I.e, each number R_{ij} was a random number in the interval $[0, 1]$ derived according to the uniform distribution. In parallel to this, we also produced analogous games according to the Gaussian distribution, so as to observe whether the choice of the probability distribution plays a role in algorithmic performance. Finally, we used various values for the dimension of the game, n, which ranged from as low as 50, all the way to 15000.

- **Low rank zero-sum games.** We experimented with a class of games where the payoff entries are not all drawn independently from each other, which translates to a lower than full rank payoff matrix. Having a payoff matrix with relatively low rank provides more structure and this class of zero-sum games has been studied in the literature. To generate a matrix $\boldsymbol{R}$ of size n and fixed rank r, we uniformly sample matrices $\boldsymbol{U}, \boldsymbol{V}$ of dimensions $n \times r$. Obtaining $\boldsymbol{R}$ via $\boldsymbol{R} = \boldsymbol{U} \cdot \boldsymbol{V}^\top$ gives us a matrix of desired rank r.

- **Symmetric Zero-sum games.** we also tested a class of symmetric zero-sum games, which again is more structured than random games. Symmetric games are games where the payoff matrix satisfies $R = R^T$ and they have played a prominent role in understanding the algorithmic complexity of games in general. In order to construct such families, we used the following formula for filling in the entries of the payoff matrix, where P_{ij}^n is the entry of P at (i, j) when P is $n \times n$. Here symmetry is enforced, given the dependence on $i + j$.

$$P_{ij}^n = \frac{1}{n}(i + j - 2) \bmod n \tag{1}$$

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 $(\boldsymbol{x}, \boldsymbol{y})$
while $V(\boldsymbol{x}, \boldsymbol{y}) > \delta$ **do**
  $(\boldsymbol{x}', \boldsymbol{y}') = \text{FindDirection}(\boldsymbol{x}, \boldsymbol{y}, \rho)$
  $(\boldsymbol{x}, \boldsymbol{y}) = (1 - \varepsilon) \cdot (\boldsymbol{x}, \boldsymbol{y}) + \varepsilon \cdot (\boldsymbol{x}', \boldsymbol{y}')$
return $(\boldsymbol{x}, \boldsymbol{y})$. =0

- **Generalized Rock-Paper-Scissors Games.** We also used a family of games that originates from the classic Rock Paper Scissors (RPS) game. The RPS game is a small game of dimension 3×3. There is an easy way to define games of higher dimension that follow the structure of RPS. These are referred to as the generalized Rock Paper Scissors games and have been considered as candidate games in experimental evaluation of learning algorithms in past works.

- **Beyond bilinear: Convex-concave games** Finally, we also tried to investigate (but not exhaustively, given the time duration of the project) if our methods can be adapted and continue to have convergent behavior when we move away from the family of bilinear games. To that end, we used convex-concave games, where the Nash equilibrium corresponds to solving for the min-max objective $\min_{\boldsymbol{x}} \max_{\boldsymbol{y}} f(\boldsymbol{x}, \boldsymbol{y})$, for functions that are convex in the $\boldsymbol{x}$ variables and concave in $\boldsymbol{y}$. One prominent example that is used often is the payoff function $f(\boldsymbol{x}, \boldsymbol{y}) = ||\boldsymbol{x} - \boldsymbol{y}||^2 = \sum_{i \in [n]} (x_i - y_i)^2$.

4 Algorithm 1 and its Variants: An Optimization-based Approach for Zero-Sum Games

We recall our first algorithm, and its variants, from our IJCAI 2025 paper, which is part of Deliverable D.4.3. 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 requires ρ and ϵ to be correlated.

To define properly the algorithm, we also need to define approximate best response strategy sets. 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.

Our main algorithm is presented as Algorithm 1, where $V(\boldsymbol{x}, \boldsymbol{y})$ within the algorithm's pseudocode is the standard duality gap function.

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

We also present a different variant of our main approach. The idea is to gradually decay δ, which results in an improved mathematical analysis. This is presented as Algorithm 3.

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. $(\boldsymbol{x}', \boldsymbol{y}')$ and γ):
  minimize γ
    s.t. $\gamma \geq (\boldsymbol{e}_i)^\top \boldsymbol{R}\boldsymbol{y}' - (\boldsymbol{x}')^\top \boldsymbol{R}\boldsymbol{e}_j$,
    for any $i \in BR_r^\rho(\boldsymbol{y})$, for any $j \in BR_c^\rho(\boldsymbol{x})$,
    and with $\boldsymbol{x}', \boldsymbol{y}' \in \Delta^{n-1}$.
return $(\boldsymbol{x}', \boldsymbol{y}')$. =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 $\delta_i \leq \delta$ **then** break
return $(\boldsymbol{x}, \boldsymbol{y})$. =0

4.1 From Theory to Implementation

We discuss now certain issues regarding the selection of the parameters that the algorithms depend on.

Choice of $\boldsymbol{\varepsilon}$. We have established in the theoretical part of our paper 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. 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 $\boldsymbol{\rho}$ (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 algorithm 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*. We used $k=100$ for our experiments and point to our full version in [6] 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 discuss first our comparisons between Algorithm 3 with $\rho = 0.001$, henceforth called the *Constant ρ Variant*, then with $\rho_i = 0.01\sqrt{\delta_i}$, which we refer to as the *Adaptive ρ Variant* and our Fixed Support Variant discussed in Subsection 4.1. To compare our variants, before going into comparisons against other methods, we generated games of size $n \times n$, where n ranged from 500 to 5000 pure strategies with a step of 500. The games were generated according to the discussion in Section 3, and for each size we generated 30 games and solved them to an accuracy of 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 1, 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 1 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 Solvers and First-Order 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. Other commercial LP solvers had a similar behavior. 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 methods for solving zero-sum games [3]. We also compared our algorithms against another popular

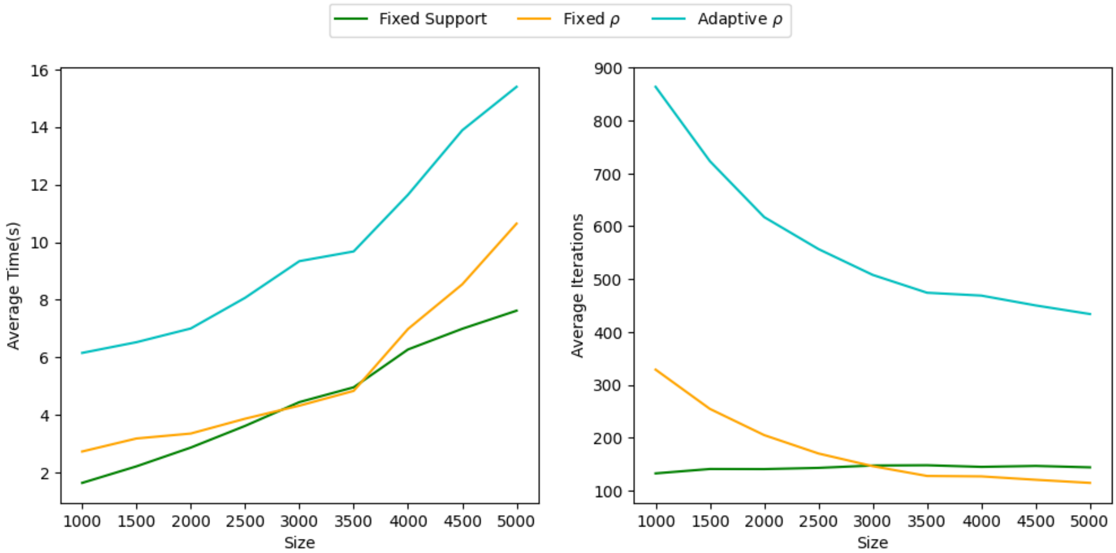

Figure 1: Average time and number of iterations for our variants.

method, which is the Optimistic Multiplicative Weights Update (OMWU) method. This however does not behave as well in practice, as explained in [2] (see also the next subsection for further elaboration on the performance of OMWU). For OGDA, we used a learning rate (step size) of $\eta = 0.01$.

We present here the comparisons for the uniformly random games, as described in Section 3. For the other classes of synthetically generated games, we refer to the full version of our paper in [6]. Qualitatively, the conclusions are quite similar for these other classes of games as well. As discussed in Subsection 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 2.

We summarize our main findings as follows:

- The LP solver was far slower, even for lower values of n, as shown in the left subplot of Figure 2, 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 2). 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).
- Our method behaves in a more smooth manner 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, which is a state of the art method for this class of games. On top of this, in several cases we even outperform OGDA, especially for games of larger dimension. One limitation of our current implementation is the choice of $\delta = 0.01$. For much lower accuracies, our method occasionally get stuck. We therefore feel that the overall approach deserves further exploration, especially on potential ways of accelerating its execution.

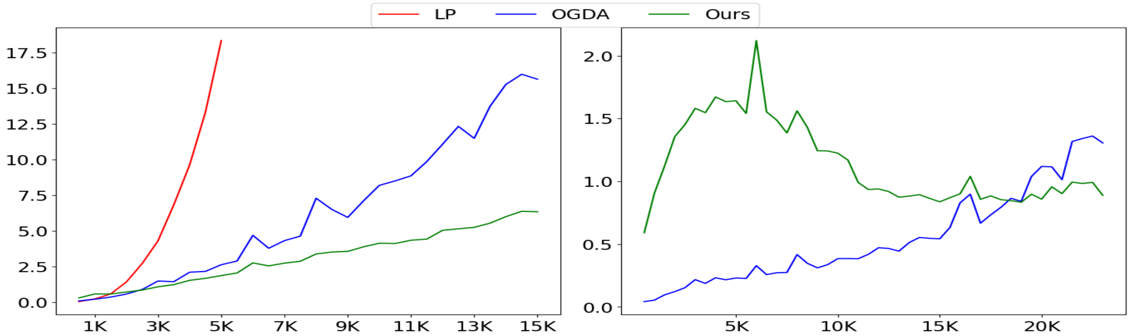

Figure 2: Time comparison between our Fixed Support Variant, LP solver and Optimistic Gradient Descent-Ascent

5 Algorithm 2: Learning Dynamics for Converging to Nash Equilibria

In parallel to the algorithms presented in Section 4, we have explored the convergence properties of learning algorithms with gradient feedback. In particular, we have evaluated experimentally the learning method that we studied in our second paper within WP4, in [5].

We briefly recall how this method works, whcih we refer to by the name of Forward Looking Best-Response Multiplicative Weights Update method (FLBR-MWU), as in [4]. 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).

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

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.

$$\begin{aligned} \boldsymbol{x}_i^t &= \boldsymbol{x}_i^{t-1} \cdot \frac{e^{\eta \boldsymbol{e}_i^\top R \hat{\boldsymbol{y}}^t}}{\sum_{j=1}^n \boldsymbol{x}_j^{t-1} e^{\eta \boldsymbol{e}_j^\top R \hat{\boldsymbol{y}}^t}}, \\ \boldsymbol{y}_j^t &= \boldsymbol{y}_j^{t-1} \cdot \frac{e^{-\eta \boldsymbol{e}_j^\top R^\top \hat{\boldsymbol{x}}^t}}{\sum_{i=1}^n \boldsymbol{y}_i^{t-1} e^{-\eta \boldsymbol{e}_i^\top R^\top \hat{\boldsymbol{x}}^t}}. \end{aligned} \tag{3}$$

5.1 Implementation Aspects

Setting η and ξ. In our evaluation, the main decision choices were how to pick the parameters η and ξ. Regarding η, this was a parameter for which we tried out a variety of different values, as reported in the next subsection. The parameter η is usually referred to as the learning rate and it should not generally be set to a very large quantity, as it could lead to divergence. As for ξ, driven by the theoretical analysis, it was also verified experimentally that it is consistently better to pick relatively large values for it. Therefore we settled to use a value of $\xi = 100$ or higher. Ideally, according to our theoretical analysis, the method works better when ξ goes to ∞. However, in practice, this can cause overflowing issues, given the calculations that the dynamics perform. Hence we had to settle with high enough but not too high values for ξ.

Initialization. Finally, another decision choice is the starting point of the dynamics. By this we mean the initial strategy profile for the players. This can potentially play a role in the convergence speed of learning algorithms, and there are 2 main choices in order to derive qualitative conclusions on whether initialization is important. In particular, we can either start with a pure strategy profile, where each player picks an arbitrary pure strategy, or we can start with the fully uniform profile, where every player selects the uniform distribution as her initial mixed strategy.

5.2 Comparisons Against Other First-Order Methods

As in Section 4, we compare FLBR against OMWU and against OGDA, with the latter being one the fastest and most well studied last-iterate method for bilinear games [3]. We do not compare against standard LP solvers, since already in Section 4, it was highlighted that LP solvers behave worse than OGDA. Hence, investigating whether FLBR is competitive against OGDA was what we were after.

We discuss here 2 types of comparisons to provide our qualitative conclusions. Further experiments, using the data sets discussed in Section 3, are provided in our completed paper at the end of this report. Firstly, we compare the three methods on random games, and more specifically when the matrices are populated from a standard Gaussian distribution. Given than OMWU performs quite poorly in these games, we then perform further comparisons only between FLBR and OGDA, complemented with more visualizations of different learning rates. For our secomd experiment reported here, we used the generalized Rock Paper Scissors (RPS) game of higher dimensions, as discussed in Section 3. In all our experiments, we use $\xi = 100$ (as a result of our tuning w.r.t. how to set ξ).

Our main findings were as follows:

- In Figure 3, we see the comparisons on 50×50 Gaussian random 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 and is better than OMWU in time elapsed. The performance of OGDA is explained by [1], via last iterate analysis under the celebrated framework of *smoothed analysis*.
- In Figures 4 and 5, we see the comparisons for generalized RPS, for dimensions 11 and 101, and for various values of η. Again the methods are comparable with a slight advantage for FLBR.
- Finally, apart from the number of iterations shown in the previous figures, we present some indicative time comparisons between FLBR and OGDA in Tables 1 and 2. Again the conclusion remains the same, that OGDA is better in random games and FLBR becomes better in RPS, and generally in more structured games.

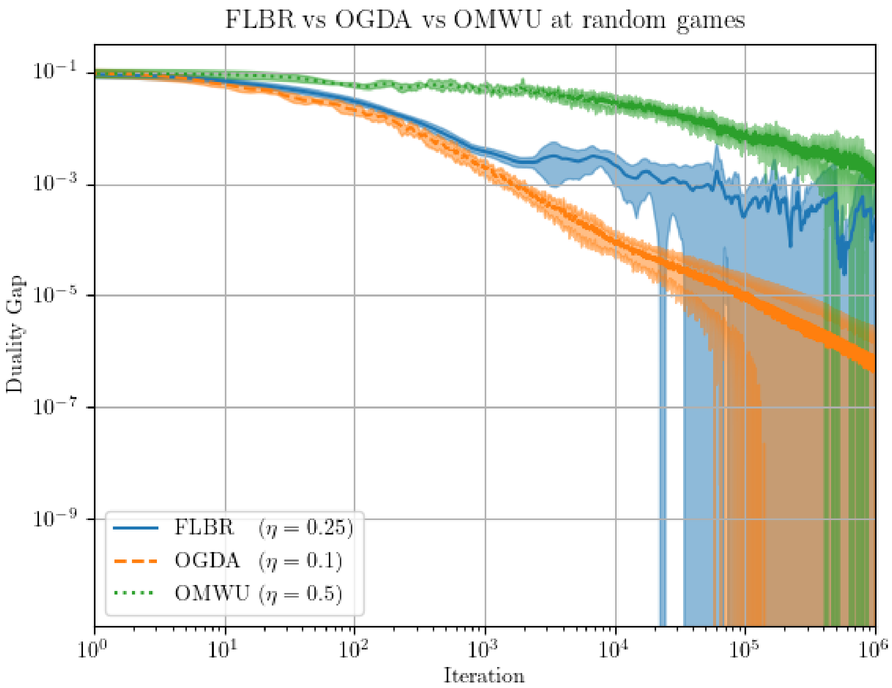

Figure 3: Comparison in Gaussian games

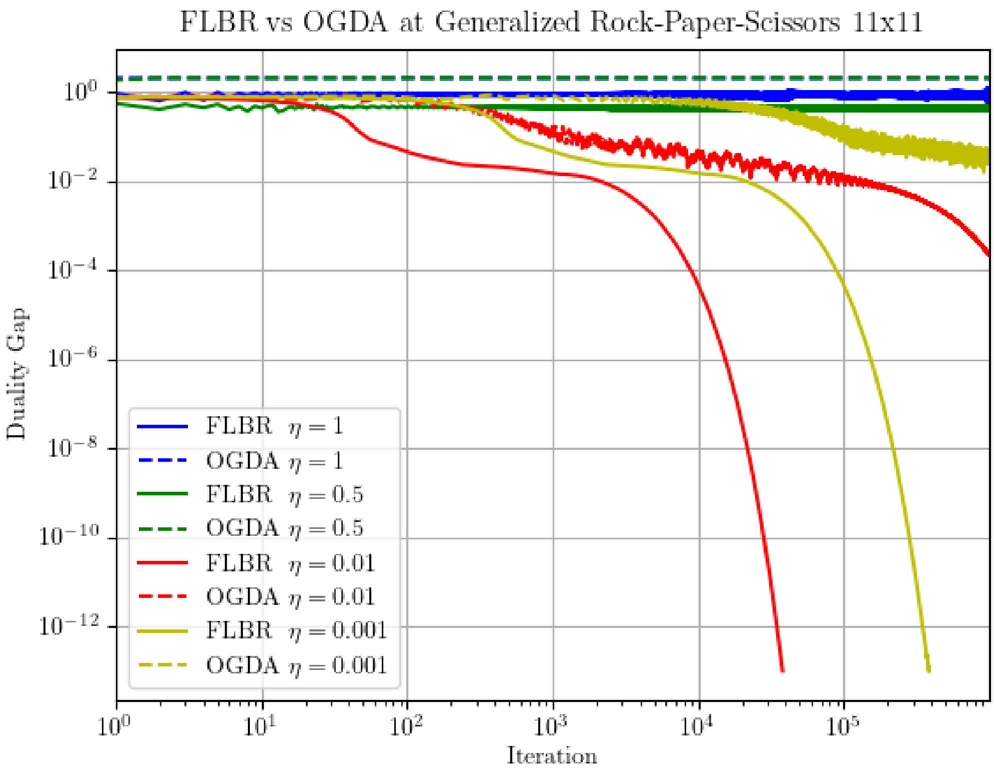

Figure 4: Comparisons over various values of η

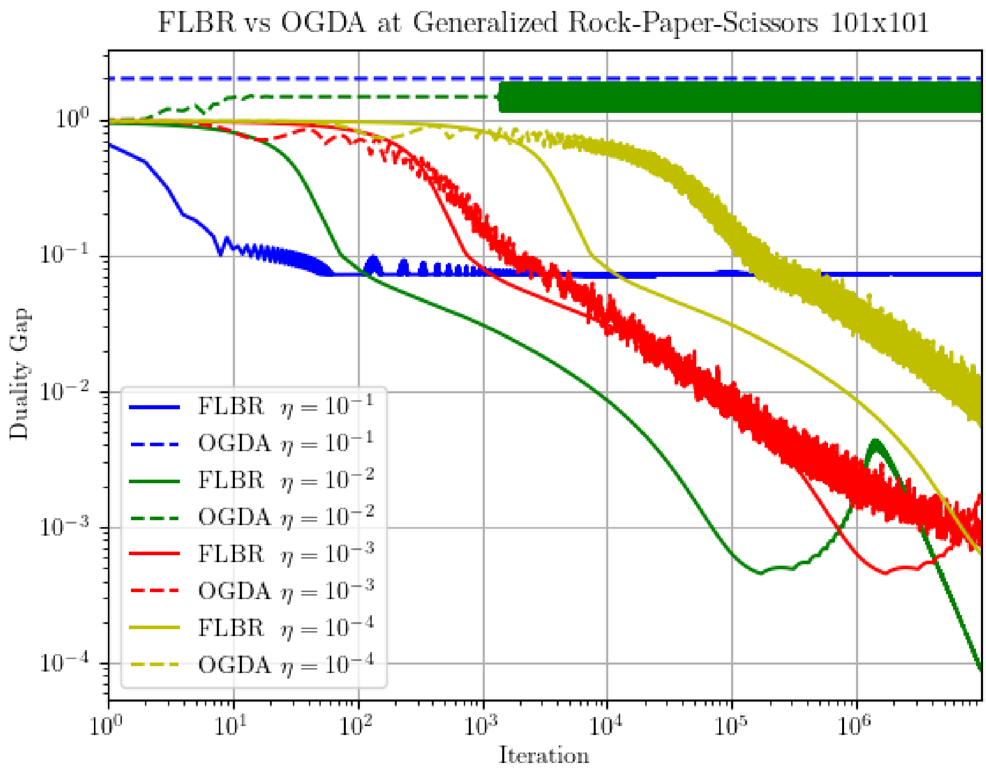

Figure 5: RPS games of higher dimension

One more conclusion that arises from the experiments (see 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.

Table 1: Comparison in Gaussian games

	Time (sec) to accuracy			
	10^{-2}	10^{-3}	10^{-4}	10^{-5}
OGDA	0.005	0.026	0.155	1.72
FLBR	0.005	0.14	0.8	3.87

Table 2: Comparison in RPS

	Time (sec) to accuracy			
	10^{-2}	10^{-3}	10^{-4}	10^{-5}
OGDA	4.73	14.45	24.28	34.00
FLBR	0.08	0.11	0.15	0.22

6 Conclusions

We have experimented with two algorithmic approaches for computing Nash equilibria in zero-sum games. The first method is an optimization-based approach, whereas the second one is a learning algorithm with gradient feedback. Overall, we can summarize our findings from our research as follows:

- Both of our algorithms and some of their appropriately fine-tuned variants exhibit competitive performance against state of the art methods, such as the Optimistic Gradient Descent Ascent (OGDA) method.

- The algorithm of Section 4 tends to outperform OGDA for games of higher dimensions, and especially when the number of pure strategies per player grows into the order of thousands.
- The algorithm of Section 5 exhibits a different behavior. It outperforms OGDA in more structured games, i.e., in symmetric games or in generalized Rock-Paper-Scissors games. For completely random games, where each payoff entry is sampled from the uniform or the Gaussian distribution, OGDA tends to perform better.
- Comparing the 2 algorithms of Section 4 and Section 5, the FLBR algorithm of Section 5 is faster.
- Our methods behave in a more smooth manner with less sharp jumps than OGDA when initialized with a uniform strategy profile, while the opposite holds for the uniform profile.
- Overall, we can conclude that none of the methods is always superior to the other ones. If we care for speed and accuracy, we would say that either OGDA or FLBR are appropriate. For random games with small or medium game sizes, OGDA seems to be the winner. On the other hand, for games with very large number of strategies, our method of Section 4 becomes the dominant choice. Finally, for games that exhibit more structure, the FLBR algorithm seems to be more appropriate.

We view all the above as promising results, especially since the methods we have analyzed may admit even further acceleration by appropriate refinements. We therefore feel that the overall approach deserve further exploration in the future, and believe also that they can be applicable in classes of non-bilinear games as well.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] Michail Fasoulakis, Evangelos Markakis, Georgios Roussakis, and Christodoulos Santorinaios. Improved last-iterate convergence properties for the flbr dynamics. Under submission, 2025.

[6] 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.

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

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

A Appendix

In the remainder of this deliverable we comment on the papers that resulted from our work, relevant to this experimentation. Regarding our algorithm and its variants in Section 4, this led to a publication at IJCAI 2025. The full version of this work is available at `https://arxiv.org/abs/2501.19138`. Regarding the analysis of the algorithm in Section 5, this is a paper currently under submission and we attach a version of it at the end of this report. It also shows additional experiments on top of that we discussed in Section 5.

Improved Last-iterate Convergence Properties for the FLBR Dynamics

Anonymous authors
Paper under double-blind review

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 $c < 1$ is independent of the game parameters. 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 bilinear 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 (Freund & Schapire, 1999).

In the recent years, the attention of the relevant community has gradually shifted from convergence on average 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 & 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. The currently best rate that is applicable to all such games is $O(1/\sqrt{t})$ in terms of the duality gap (Cai et al., 2022; Gorbunov et al., 2022), where the hidden terms in the $O(\cdot)$ notation depend on the game dimension but not on the payoff matrix. In fact this also holds for the more general class of convex-concave min-max optimization problems. It is conceivable though that better rates could be achieved for bilinear games. The work of (Wei et al., 2021) establishes a geometric convergence rate of $O(c^t)$ $(c < 1)$ for the OMWU method, discussed further in the sequel, albeit with game-dependent parameters within the $O(\cdot)$ term. It remains an open problem whether we can have a geometric convergence rate where the dependence is only on the game dimension.

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 FLBR 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. 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. As mentioned earlier, an analogous result also holds for the OMWU method (Wei et al., 2021), but for the KL divergence, and with a different dependence on the game parameters. We view as advantages of our analysis that it yields a simpler and more intuitive proof compared to Wei et al. (2021), and it also establishes the fast (non game-dependent) convergence to an approximate equilibrium before going towards the exact solution. 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 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 of FLBR against OGDA, which is among the best known methods for solving zero-sum games, and against 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. 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 the FLBR method for bilinear games.

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 & 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 & 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 & 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 & 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. 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 even beyond convex-concave functions), 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+). These ideas have also been applied successfully in the stochastic setting, under noisy gradient feedback, (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 & 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 is a probability distribution $x = (x_1, \dots, x_n)^\top$ over the standard simplex, where the vector $e_i^\top$, with 1 in the index i and zero elsewhere, corresponds to the pure strategy i. The support of a mixed strategy x is the set of the pure strategies to which x assigns positive 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 $x^\top Ry$ (resp $-x^\top Ry$).

Definition 1 (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 $i, j \in [n]$,*

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

By setting $\varepsilon = 0$ we have an exact NE. Next we will define our progress measure.

Definition 2. *For zero-sum games, the duality gap function V is defined as*

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

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 a zero-sum game, if and only if it 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 x, y it holds that $\max_i e_i^\top Ry \leq ||y-y^*||_1+v$ and $\min_j x^\top Re_j \leq ||x-x^*||_1+v$, where v is the value of the zero-sum game.*

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 follows:

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

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

When $\xi = \eta$ in the above steps, this is referred to as Mirror-Prox in Nemirovski (2004). Contrary to the conventional wisdom of using rather small learning rates to ensure contraction, our approach is to have a large value for ξ (aggressive rate for the intermediate exploration 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., $x^0 = y^0 = (1/n, \dots, 1/n)$. Furthermore, we will use a fixed η, independent of t in all iterations.

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

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. First, 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 solution, we can maintain a geometric rate all the way to the equilibrium, 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 t, hence the ratios above are well-defined. For brevity, we write $D_{KL}((x^*, y^*)||(x^t, y^t))$ as D^t. 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 $\eta \leq 1/2$*

$$\eta \cdot \left((\hat{x}^t)^\top R y^{t-1} - (x^{t-1})^\top R \hat{y}^t\right) \leq D^{t-1} - D^t + 4\eta^2.$$

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 $\lim_{\xi \to \infty} [(\hat{x}^t)^\top R y^{t-1} - (x^{t-1})^\top R \hat{y}^t] = V(x^{t-1}, y^{t-1})$.*

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*

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

All missing proofs are presented in Appendix B. The next theorem is the main result of this section.

Theorem 1. *Under Assumption 1, and for sufficiently small η and large ξ, the rate of convergence for the KL divergence till we reach a 6η-Nash equilibrium is inverse exponential, in the form $O(\ln n \cdot c^t)$, where $c < 1$ is independent of t and dependent on n and η. Similarly, the convergence rate of the duality gap to reach a 6η-NE is inverse exponential, in the form $O\left(\frac{\ln n}{\eta} \cdot c^t\right)$.*

Proof. By following the proof of Theorem 2 in Fasoulakis et al. (2022) and substituting $\max\{\varepsilon_1, \varepsilon_2\}$ with 6η we obtain that while we have not reached a 6η-NE it holds that

$$D^t \leq D^{t-1} - 2\eta^2 = D^{t-1}\left(1 - \frac{2\eta^2}{D^{t-1}}\right).$$

Due to Assumption 1 and the fact that the KL divergence only decreases till we reach an approximate equilibrium (Fasoulakis et al. (2022)), we have that $D^{t-1} \leq D^0 \leq 2\ln(n)$. Thus we conclude that

$$D^t \leq D^{t-1}\left(1 - \frac{\eta^2}{\ln(n)}\right).$$

For $\eta \leq \sqrt{\ln(n)}$ we can unroll the above inequality for all time steps up to t to obtain

$$D^t \leq D^{t-1}\left(1 - \frac{\eta^2}{\ln(n)}\right)^t \leq 2\ln(n)\left(1 - \frac{\eta^2}{\ln(n)}\right)^t.$$

This means that the KL divergence at time t is bounded by $2\ln(n) \cdot c^t$, where $c < 1$ is independent of t and dependent on η and n. Coming now to the duality gap, we conclude by Corollary 1 that

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

Note also that since we have not yet reached a 6η-NE, it holds that $V(x^t, y^t) \geq 6\eta$. Combining this with the above upper bound implies that for any time step t, till we reach an approximate equilibrium, we have that $\eta \leq \frac{2\ln(n)}{\eta}\left(1 - \frac{\eta^2}{\ln(n)}\right)^t$. By plugging this in (1), we eventually get:

$$V(x^t, y^t) \leq \frac{12\ln(n)}{\eta}\left(1 - \frac{\eta^2}{\ln(n)}\right)^t. \qquad \square$$

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 $t_0 = O(\ln\ln(n)/\ln(\eta))$. 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.[2]

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 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.

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:

[2] 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.

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 & 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 Appendix C. 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[3] 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}):

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

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.

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 $x_1^{2t+1} = (1-\delta)[1 - \delta(1 - e^{2\eta(t+1)})]^{-1}$.*

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

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

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:

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

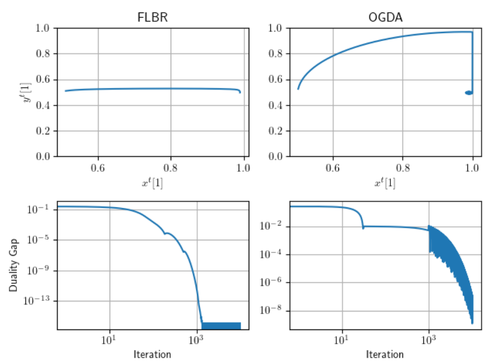

Figure 1: FLBR vs OGDA in game A_δ.

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.

5 Experimental evaluation

Experimentally, the method had seemed to be promising already in Fasoulakis et al. (2022). Here we start by comparing FLBR against OMWU and against OGDA, with the latter being 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 three methods on random games, and more specifically when the matrices are populated from a standard Gaussian distribution. Then we revisit the game A_δ discussed in Section 4.2. In both experiments we present one moderately finetuned choice of the learning rate η. Given than OMWU performs quite poorly both in the random games and in A_δ, we then perform further comparisons only between FLBR and OGDA, complemented with more visualizations of different learning rates. For our third experiment, and in order to get more meaningful comparisons, we have sought additional games that are simultaneously far from random and larger in size. To that end, we used the generalized Rock Paper Scissors (RPS) game of higher dimensions. In all our experiments, including the additional ones presented in Appendix D, we use $\xi = 100$ (as a result of our tuning w.r.t. how to set ξ).

Our main findings and conclusions are as follows:

- In Figure 2, we see the comparisons on 50×50 Gaussian random 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 and is better than OMWU in time elapsed. The performance of OGDA is explained by Anagnostides & Sandholm (2024), via last iterate analysis under the celebrated framework of *smoothed analysis* (Spielman & Teng, 2004).

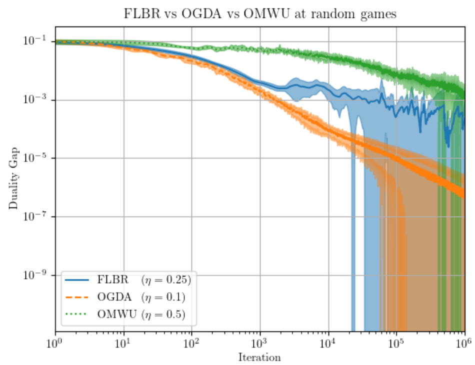

Figure 2: Comparison in Gaussian games

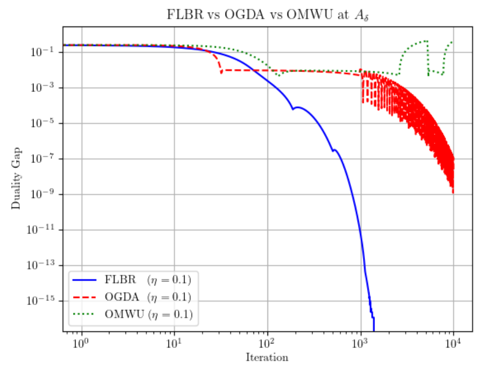

Figure 3: Further comparisons for game A_δ

- In Figure 3, we see the comparisons for the game A_δ. Here the conclusion reverses: the methods are comparable once again but now FLBR comes on top. And OMWU is quite far away.
- In Figures 4 and 5, we see the comparisons for generalized RPS, for dimensions 11 and 101, and for various values of η. Again the methods are comparable with a slight advantage for FLBR.
- Finally, apart from the number of iterations shown in the previous figures, we present some indicative time comparisons between FLBR and OGDA in Tables 1 and 2. Again the conclusion remains the same, that OGDA is better in random games and FLBR becomes better in RPS, and generally in more structured games (as also verified in our additional experiments in the Appendix).

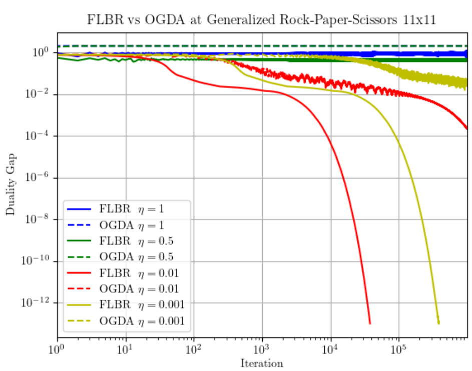

Figure 4: Comparisons over various values of η

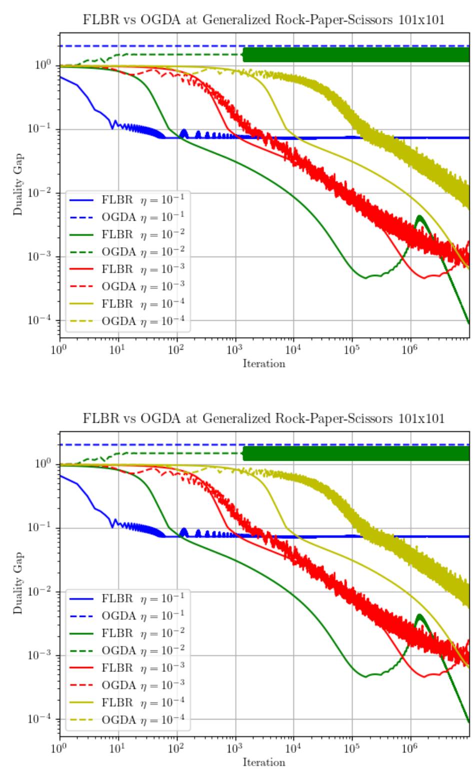

Figure 5: RPS games of higher dimension

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 (see 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. As a step towards further explorations for the performance of FLBR, it would be interesting to study if our results generalize beyond bilinear payoffs to classes of convex-concave functions. We have conducted some initial experimentation on this, presented in Section D.3.

Table 1: Comparison in Gaussian games

	Time (sec) to accuracy			
	10^{-2}	10^{-3}	10^{-4}	10^{-5}
OGDA	0.005	0.026	0.155	1.72
FLBR	0.005	0.14	0.8	3.87

Table 2: Comparison in RPS

	Time (sec) to accuracy			
	10^{-2}	10^{-3}	10^{-4}	10^{-5}
OGDA	4.73	14.45	24.28	34.00
FLBR	0.08	0.11	0.15	0.22

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*, pp. 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, pp. 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)*, pp. 321–338, 2018.

George W Brown. Iterative solution of games by fictitious play. *Act. Anal. Prod Allocation*, 13(1): 374, 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*, pp. 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*, pp. 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)*, pp. 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 (eds.), *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*, pp. 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*, pp. 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)*, pp. 11096–11117, 2022.

Yoav Freund and Robert E Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29(1-2):79–103, 1999.

Andrew Gilpin, Javier Pena, and Tuomas Sandholm. First-order algorithm with convergence forequilibrium 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)*, pp. 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*, pp. 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*, pp. 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*, pp. 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.

Arkadi Nemirovski. Prox-method with rate of convergence o (1/t) for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15(1):229–251, 2004.

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*, pp. 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)*, pp. 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.

A MISSING PROOFS FROM SECTION 2

Proof of Lemma 1. We have that for any i,

$$\begin{aligned} |e_i^\top Ry - e_i^\top Ry^*| &= |\sum_j R_{ij} \cdot y_j - \sum_j R_{ij} \cdot y_j^*| \\ &= |\sum_j R_{ij} \cdot (y_j - y_j^*)| \\ &\le \sum_j |R_{ij} \cdot (y_j - y_j^*)| \\ &= \sum_j R_{ij} \cdot |(y_j - y_j^*)| \\ &\le \sum_j |(y_j - y_j^*)| \\ &= ||y - y^*||_1. \end{aligned}$$

Thus, if $b = \arg\max_i e_i^\top Ry$, then $\max_i e_i^\top Ry = e_b^\top Ry \le ||y - y^*||_1 + e_b^\top Ry^* \le ||y - y^*||_1 + v$. The second part of the lemma follows in a similar manner. □

B MISSING PROOFS FROM SECTION 3

B.1 PROOF OF LEMMA 3

Proof. We first rewrite the KL terms, by using the definition of the dynamics.

$$\begin{aligned} &D_{KL}((x^*, y^*)||(x^{t-1}, y^{t-1})) - D_{KL}((x^*, y^*)||(x^t, y^t)) \\ &= \sum\nolimits_{i=1}^n x_i^* \cdot \ln(x_i^t / x_i^{t-1}) + \sum\nolimits_{j=1}^n y_j^* \cdot \ln(y_j^t / y_j^{t-1}) \\ &= \sum_{i=1}^n x_i^* \cdot \ln e^{\eta \cdot e_i^\top R\hat{y}^t} - \ln\left(\sum_{k=1}^n x_k^{t-1} \cdot e^{\eta \cdot e_k^\top R\hat{y}^t}\right) \\ &\quad + \sum_{j=1}^n y_j^* \cdot \ln e^{-\eta \cdot e_j^\top R^\top \hat{x}^t} - \ln\left(\sum_{k=1}^n y_k^{t-1} \cdot e^{-\eta \cdot e_k^\top R^\top \hat{x}^t}\right) \\ &= \eta \cdot (x^*)^T R\hat{y}^t - \eta \cdot (y^*)^T R^\top \hat{x}^t - \ln\left(\sum_{k=1}^n x_k^{t-1} \cdot e^{\eta \cdot e_k^\top R\hat{y}^t}\right) - \ln\left(\sum_{k=1}^n y_k^{t-1} \cdot e^{-\eta \cdot e_k^\top R^\top \hat{x}^t}\right). \end{aligned}$$

We now use the Taylor expansion of the exponential function in the arguments of the last two logarithms. For the first logarithmic term, this becomes:

$$\begin{aligned} \ln\left(\sum_{k=1}^n x_k^{t-1} \cdot e^{\eta \cdot e_k^\top R\hat{y}^t}\right) &= \ln\left(1 + \eta \cdot (x^{t-1})^\top R\hat{y}^t + \sum_{k=1}^n x_k^{t-1} \sum_{\ell \ge 2} \frac{(\eta \cdot e_k^\top R\hat{y}^t)^\ell}{\ell!}\right) \\ &\le \ln\left(1 + \eta \cdot (x^{t-1})^\top R\hat{y}^t + 2\eta^2\right). \end{aligned}$$

For the above we used the fact that $\sum_{\ell \ge 2} \frac{(\eta \cdot e_k^\top R\hat{y}^t)^\ell}{\ell!} \le \frac{\eta^2}{1-\eta} \le 2\eta^2$, since $\eta \le 1/2$. By exploiting now the inequality that $\ln(x) \le x - 1$, we finally obtain the bound

$$\ln\left(\sum_{k=1}^n x_k^{t-1} \cdot e^{\eta \cdot e_k^\top R\hat{y}^t}\right) \le \eta \cdot (x^{t-1})^\top R\hat{y}^t + 2\eta^2.$$

By carrying out similar calculations for the second logarithmic term, we will also get that

$$\ln\left(\sum_{k=1}^n y_k^{t-1} \cdot e^{-\eta \cdot e_k^\top R^\top \hat{x}^t}\right) \le -\eta \cdot (\hat{x}^t)^\top Ry^{t-1} + 2\eta^2.$$

This gives us:

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

By rearranging the terms, we obtain that

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

Note now that since (x^*,y^*) is a Nash equilibrium, and we are in a 0-sum game, then we know that $(x^*)^\top R\hat{y}^t \geq v$, where v is the value of the game. Similarly, $(\hat{x}^t)^\top R y^* \leq v$. Hence these terms cancel out in the above equation and the proof is complete. □

B.2 Proofs of Claim 1 and Corollary 1

Proof. Recalling Definition 2 we have that $V(x^{t-1},y^{t-1}) = \max_i e_i^\top R y^{t-1} - \min_j (x^{t-1})^\top R e_j$. But by Lemma 2, we have that $\hat{x}^t$ converges to a best response against y^{t-1}, and similarly for $\hat{y}^t$, which completes the proof. □

Proof. By Claim 1, we know that the quantity $(\hat{x}^t)^\top R y^{t-1} - (x^{t-1})^\top R\hat{y}^t$ converges to $V(x^{t-1},y^{t-1})$ as $\xi\to\infty$. This means that for any $\epsilon > 0$, there exists ξ_0 s.t. for every $\xi \geq \xi_0$ we have $|(\hat{x}^t)^\top R y^{t-1} - (x^{t-1})^\top R\hat{y}^t - V(x^{t-1},y^{t-1})| \leq \epsilon$. If we use $\epsilon = \eta$, there exists a large enough ξ_0, such that for any $\xi \geq \xi_0$, it holds that

$$V(x^{t-1},y^{t-1}) \leq (\hat{x}^t)^\top R y^{t-1} - (x^{t-1})^\top R\hat{y}^t + \eta.$$

By using now Lemma 3, we get the desired inequality. □

B.3 Proof of Lemma 4

First we show the following claim that we use in the proof of our Lemma.

Claim 3. $\dfrac{d\varphi(x(p),y(p))}{dp} = J(x(p),y(p))\cdot\Big(x^t - x^*, y^t - y^*\Big).$

In the equation above, the term $(x^t - x^*, y^t - y^*)$ is a vector of $2n$ coordinates, where for each $i\in[n]$ the i-th coordinate equals $x_i^t - x_i^*$, and the $(n+i)$-th coordinate equals $y_i^t - y_i^*$.

Proof. For the row player, we have that for any i,

$$\begin{aligned}\frac{d\varphi_{1,i}(x(p),y(p))}{dp} &= \sum_k \frac{dx_k(p)}{dp}\cdot\frac{d\varphi_{1,i}(x(p),y(p))}{dx_k(p)} + \sum_\ell \frac{dy_\ell(p)}{dp}\cdot\frac{d\varphi_{1,i}(x(p),y(p))}{dy_\ell(p)}\\ &= \sum_k \left(x_k^t - x_k^*\right)\cdot J(x(p),y(p))_{ik} + \sum_\ell \left(y_\ell^t - y_\ell^*\right)\cdot J(x(p),y(p))_{i,n+\ell}\end{aligned}$$

The above hold because $\frac{dx_k(p)}{dp} = x_k^t - x_k^*$ and $\frac{dy_\ell(p)}{dp} = y_\ell^t - y_\ell^*$. Analogous expressions hold for φ_2 as well, thus we conclude that

$$\frac{d\varphi(x(p),y(p))}{dp} = J(x(p),y(p))\cdot\Big(x^t - x^*, y^t - y^*\Big). \qquad \square$$

Proof. By the Mean Value Theorem (applied for our function $f^t = \varphi(x(p), y(p)) : \mathbb{R} \to \mathbb{R}^{2n}$), for each time t, there is a $p^t \in (0, 1)$ s.t.

$$\begin{aligned}
||(x^{t+1}, y^{t+1}) - (x^*, y^*)||_q &= ||\Big(\varphi_1(x^t, y^t), \varphi_2(x^t, y^t)\Big) - \Big(\varphi_1(x^*, y^*), \varphi_2(x^*, y^*)\Big)||_q \\
&= ||f^t(1) - f^t(0)||_q \\
&\leq ||\frac{df^t(p)}{dp}|_{p=p^t}||_q \cdot (1 - 0) \\
&= ||\Big((x^t, y^t) - (x^*, y^*)\Big) \cdot J(x(p^t), y(p^t))||_q \\
&\leq ||(x^t, y^t) - (x^*, y^*)||_q \cdot ||J(x(p^t), y(p^t))||_q
\end{aligned}$$

where the second inequality holds by the properties of the q-norm. □

B.4 Proof of Lemma 5

Proof. For the basis of the induction, consider $t = t_0$. Regarding the Jacobian, first note that

$$\begin{aligned}
||(x(p^{t_0}), y(p^{t_0})) - (x^*, y^*)||_q &= ||(1 - p^{t_0})(x^*, y^*) + p^{t_0}(x^{t_0}, y^{t_0}) - (x^*, y^*)||_q \\
&= ||p^{t_0}(x^{t_0}, y^{t_0}) - p^{t_0}(x^*, y^*)||_q \\
&\leq ||(x^{t_0}, y^{t_0}) - (x^*, y^*)||_q
\end{aligned}$$

Furthermore, by the continuity of the norm, for the given ε, there exists $\delta > 0$ s.t. if $||(x^*, y^*) - (x(p^{t_0}), y(p^{t_0}))||_q < \delta$, then $\Big| ||J(x(p^{t_0}), y(p^{t_0}))||_q - ||J(x^*, y^*)||_q \Big| < \varepsilon$. Therefore, if we use this value of δ, we get that if $||(x^{t_0}, y^{t_0}) - (x^*, y^*)||_q \leq \delta$, then $||(x(p^{t_0}), y(p^{t_0})) - (x^*, y^*)||_q < \delta$ (by the previous analysis), and consequently $||J(x(p^{t_0}), y(p^{t_0}))||_q < ||J(x^*, y^*)||_q + \varepsilon$. This establishes the basis.

For the induction step, assume that the condition holds for some $t \geq t_0$. We will establish it for $t + 1$.

Since we have assumed that ε satisfies $||J(x^*, y^*)||_q + \varepsilon < 1$, the induction hypothesis yields that $||J(x(p^t), y(p^t))||_q < 1$. Using this and Lemma 4, we get that $||(x^{t+1}, y^{t+1}) - (x^*, y^*)||_q < ||(x^t, y^t) - (x^*, y^*)||_q$. This also implies that if $||(x^{t_0}, y^{t_0}) - (x^*, y^*)||_q \leq \delta$, this propagates throughout all the iterations for the same δ, so that $||(x^{t+1}, y^{t+1}) - (x^*, y^*)||_q < \delta$. And this in turn yields

$$\begin{aligned}
||(x(p^{t+1}), y(p^{t+1})) - (x^*, y^*)||_q &= ||(1 - p^{t+1})(x^*, y^*) + p^{t+1}(x^{t+1}, y^{t+1}) - (x^*, y^*)||_q \\
&= ||p^{t+1}(x^{t+1}, y^{t+1}) - p^{t+1}(x^*, y^*)||_q \\
&\leq ||(x^{t+1}, y^{t+1}) - (x^*, y^*)||_q \\
&< \delta
\end{aligned}$$

To finish the proof, we use the same argument as in the induction basis. Namely, by the continuity of the norm, for the given ε, and for the δ that was identified in the induction basis, we will have that $\Big| ||J(x(p^{t+1}), y(p^{t+1}))||_q - ||J(x^*, y^*)||_q \Big| < \varepsilon$, and thus

$$||J(x(p^{t+1}), y(p^{t+1}))||_q < ||J(x^*, y^*)||_q + \varepsilon < 1.$$

□

C Missing proofs from Section 4

C.1 Proof of Lemma 6

Proof. Recall that

$$x_1^{2t+1} = x_1^{2t-1} \cdot \frac{e^{\eta \cdot e_1^\top R y^{2t}}}{\sum_j x_j^{2t-1} \cdot e^{\eta \cdot e_j^\top R y^{2t}}} = x_1^{2t-1} \cdot \frac{e^{-\eta}}{\sum_j x_j^{2t-1} \cdot e^{\eta \cdot e_j^\top R y^{2t}}}$$
$$x_2^{2t+1} = x_2^{2t-1} \cdot \frac{e^{\eta \cdot e_2^\top R y^{2t}}}{\sum_j x_j^{2t-1} \cdot e^{\eta \cdot e_j^\top R y^{2t}}} = x_2^{2t-1} \cdot \frac{e^{\eta}}{\sum_j x_j^{2t-1} \cdot e^{\eta \cdot e_j^\top R y^{2t}}}$$

For brevity, let $x_1^{2t+1} = a^t$ and $x_2^{2t+1} = b^t$ we get that

$$a^t = a^{t-1} \cdot \frac{e^{-\eta}}{a^{t-1} e^{-\eta} + \beta^{t-1} e^{\eta}}$$
$$b^t = b^{t-1} \cdot \frac{e^{\eta}}{a^{t-1} e^{-\eta} + \beta^{t-1} e^{\eta}}$$

Note that $a^t + b^t = 1$ so we get

$$\begin{aligned}
a^t = a^{t-1} \cdot \frac{e^{-\eta}}{a^{t-1}e^{-\eta} + (1 - a^{t-1})e^{\eta}} = \frac{a^{t-1}e^{-\eta}}{a^{t-1}(e^{-\eta} - e^{\eta}) + e^{\eta}} &\implies \\
\frac{1}{a^t} = 1 - e^{2\eta} + e^{2\eta} \frac{1}{a^{t-1}} &\implies \\
\frac{1}{a^t} - 1 = e^{2\eta} \left(\frac{1}{a^{t-1}} - 1 \right) &\implies \\
\frac{1}{a^t} - 1 = e^{2\eta(t+1)} \left(\frac{1}{a^{-1}} - 1 \right) &
\end{aligned}$$

Recall that $a^{-1} = x_1^{-1} = 1 - \delta$ so we get that

$$\frac{1}{a^t} = 1 + e^{2\eta(t+1)} \frac{\delta}{1-\delta} \implies x_1^{2t+1} = \frac{1-\delta}{1 - \delta(1 - e^{2\eta(t+1)})}$$ □

C.2 Proof of Lemma 7

Proof. For a given T, we compute the total payoff of the row player for the first $2T$ iterations when both players use FLBR. Since at the even steps of this process, the strategy of both players is $(0, 1)$, we get:

$$\begin{aligned}
\sum_{i=0}^{2T} {x^i}^\top R y^i &= T \cdot (0,1)^\top R (0,1) + \sum_{i=0}^{T} {x^{2i+1}}^\top R y^{2i+1} \\
&= T + \sum_{i=1}^{T} (a^t, 1 - a^t)^\top R (1 - a^t, a^t) \\
&= T + \sum_{i=1}^{T} (a^t, 1 - a^t)^\top (1 - 2a^t, -1 + 2a^t) \\
&= T + \sum_{i=1}^{T} a^t - 2(a^t)^2 - 1 + 2a^t + a^t - 2(a^t)^2 \\
&= \sum_{i=1}^{T} 4a^t(1 - a^t)
\end{aligned}$$

where once again we set $x_1^{2t+1} = a^t$.

Next, we compute the payoff of the fixed strategy $x^* = (0,1)$ for the row player, against the column player playing in each iteration the FLBR strategy y^i as computed by the previous analysis. This is equal to:

$$\begin{aligned}\sum_{i=0}^{2T} {x^i}^\top R y^i &= T \cdot (0,1)^\top R(0,1) + \sum_{i=0}^{T} (0,1)^\top R y^{2i+1} \\ &= T + \sum_{i=0}^{T} (0,1)^\top (1-2a^t, -1+2a^t) \\ &= \sum_{i=0}^{T} 2a^t\end{aligned}$$

Hence, the regret for the row player when choosing her FLBR strategy against x^* is

$$\text{Reg}_{\text{FLBR}} \geq \sum_{i=0}^{T} 2a^t - \sum_{i=1}^{T} 4a^t(1-a^t) = \sum_{i=0}^{T} 2a^t(2a^t - 1)$$

To upper bound the expression we use that $a^t = 1/2$ hence we have that

$$\begin{aligned}\frac{1-\delta(1-e^{2\eta(T+1)})}{1-\delta} &= 2 \\ \delta e^{2\eta(T+1)} &= 1-\delta \\ 2\eta(T+1) &= \ln\left(\frac{1-\delta}{\delta}\right)\end{aligned}$$

Thus, up to time $\lceil \frac{T+1}{2} \rceil$ we have that

$$a^t \geq \frac{1-\delta}{1-\delta\left(1-\sqrt{\frac{1-\delta}{\delta}}\right)} = \frac{1-\delta}{1-\delta+\sqrt{\delta-\delta^2}}$$

For $\delta \to 0$ the expression tends to 1 so there is a sufficiently small δ such that $a^t \geq .95$ for $t \leq \lceil \frac{T+1}{2} \rceil$. Piecing everything together we get that

$$\begin{aligned}\text{Reg}_{\text{FLBR}} &\geq \sum_{i=0}^{T} 2a^t(2a^t-1) \\ &\geq \sum_{i=0}^{\lceil \frac{T+1}{2} \rceil} 2a^t(2a^t-1) \\ \text{Reg}_{\text{FLBR}} &\geq 0.855 \cdot T \quad \text{over } 2T \text{ rounds,}\end{aligned}$$

which completes the proof. □

D Additional Experiments

Our additional experiments follow a similar line of thought as the ones presented in the main paper. Namely, we start with random Gaussian games, where OGDA has a slight advantage over FLBR and then we present constructions of not so random games, with some inherent structure, which slow down OGDA but not FLBR.

Initializations As stated in Assumption 1, for the theoretical part of the paper we always initialize FLBR with the uniform distribution, i.e. $x_i = y_i = 1/n$. Here we deem useful to explore more options. Specifically, we test the following starting points:

- Uniform distribution.
- Almost pure strategy profile: $x_1 = y_1 = 1 - 1/n$ and $x_i = y_i = \frac{1}{n(n-1)}$
- Random: we sample x, y from $U(0, 1)$ and then rescale them
- Sequential: $x_i = y_i = \frac{2i}{n(n+1)}$

Assumptions on η, ξ In the theoretical part of the paper, we did not need any major assumption for η and ξ (apart from ξ being large enough) for reaching an approximate equilibrium. However, for the convergence to the exact solution, we needed to use $\eta\xi < 1$, to prove Theorem 2. In our experiments, we also tested combinations of values for these two parameters that violate this condition. What we observe experimentally is that the method can perform well even without this constraint (recall e.g., that in the main paper, we also used $\xi = 100$ and values of η for which $\eta\xi > 1$), but certainly not for any arbitrary combination.

D.1 Random games

In addition to the 1000×1000 Gaussian games presented in the main paper, we see in Figures 6 and 7 the comparisons between FLBR and OGDA for further Gaussian games of dimensions 50 and 500, where each entry of the payoff matrix is filled by sampling from the Gaussian distribution. What we observe is similar to the plots presented also in the main paper for Gaussian games, namely that OGDA performs better (as expected by the existing smoothed analysis for OGDA) and that FLBR is close but on average slower than OGDA.

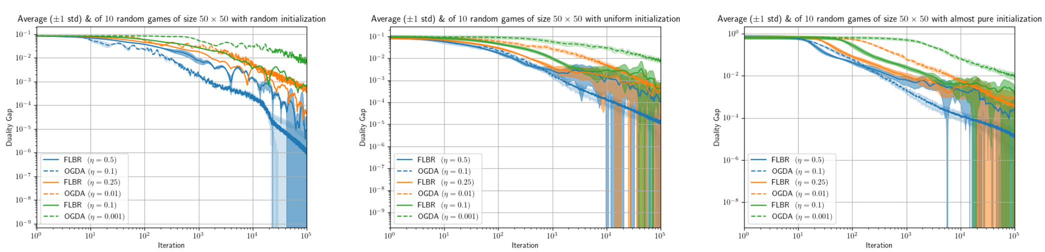

Figure 6: Random Gaussian 50×50 games with various initializations.

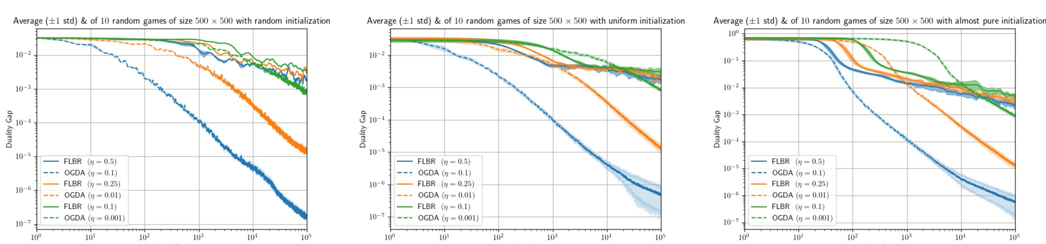

Figure 7: Random Gaussian 500×500 games with various initializations.

D.2 Structured games

We have already presented in the main paper our results on the Generalized Rock-Papers-Scissors game, which is arguably among the most famous zero-sum game. Here we also present comparisons using two more classes of more structured games.

First, we performed comparisons for games where the payoff matrix R is of low rank. Such games differ from random games, where with high probability the matrix has full rank. We constructed matrices, where the rank is approximately 5-10% of the dimension.

Interestingly, what we observe in Figures 8 and 9, is that FLBR is performing better than OGDA. The figures depict the comparisons for 50×50 games where the rank is 5 and for 500×500 games with rank equal to 25. An additional observation is that FLBR seems more robust against the various initializations that were used. For example OGDA, under the random and the uniform initialization does not converge for some choices of η.

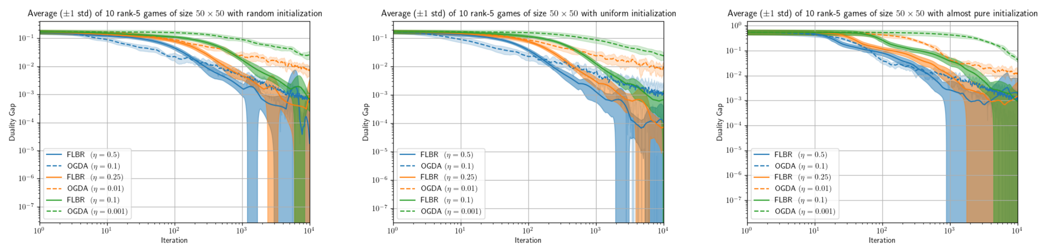

Figure 8: Games with low rank payoff matrix of size 50×50 with various initializations.

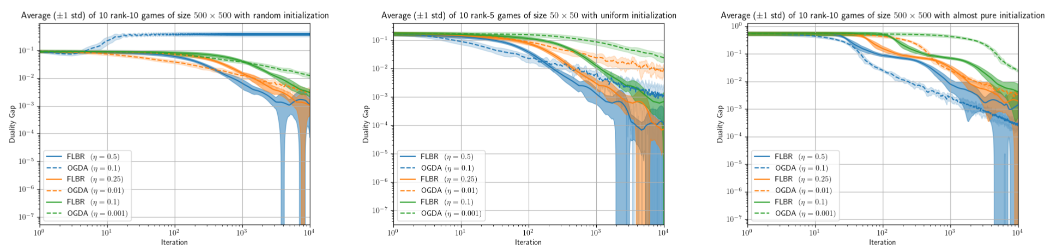

Figure 9: Games with low rank payoff matrix of size 500×500 with various initializations.

Moving on, we also tested a class of symmetric zero-sum games, which again is more structured than random games. In order to construct such families, we used the following formula for filling in the entries of the payoff matrix, where P^n_{ij} is the entry of P at (i, j) when P is $n \times n$. Here symmetry is enforced, given the dependence on $i + j$.

$$P^n_{ij} = \frac{1}{n}(i + j - 2) \bmod n \tag{3}$$

We note that for this class, we did not use the uniform initialization as this is an equilibrium of the game. What we observe in Figures 10 and 11, is that FLBR is having an advantage over OGDA for smaller dimensions, while OGDA becomes just slightly better, for the sequential and the almost pure initialization. The two methods have a very similar performance under the random initialization. Again, we observe a better robustness of FLBR with respect to the various initializations and the values of η. For example, we see that OGDA does not manage to converge for some of the choices used for η.

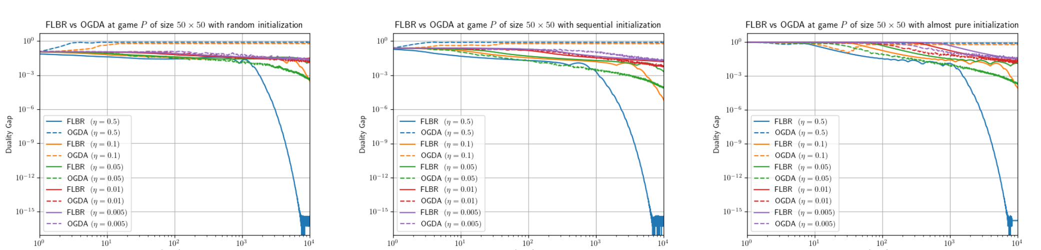

Figure 10: Structured games defined by Equation (3), of size 50×50 with various initializations.

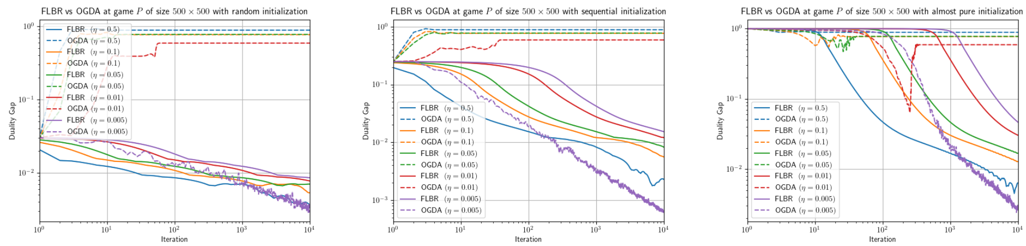

Figure 11: Structured games defined by Equation (3), of size 500×500 with various initializations.

Overall, a general conclusion that can be extracted from our experiments is that the two methods are of comparable performance, with OGDA doing better for randomly generated games, where FLBR gains an advantage for more structured games.

D.3 Experimentation beyond Bilinear Games

Finally, in our last set of experiments, we also tried to investigate if our method is convergent when we move away from bilinear games. To that end, we implemented the method as is for the min-max objective $f(x, y) = ||x - y||^2 = \sum_{i\in[n]}(x_i - y_i)^2$. The results are shown in Figure 12 for vectors of size 5. The equilibrium here is that both players get a zero payoff, and as we see in Figure 12, FLBR does not manage to converge. This is still far from conclusive, and it remains an interesting direction for future work to investigate under what families of convex-concave functions we could have convergence of FLBR or if the method needs adaptation to extend to more general domains.

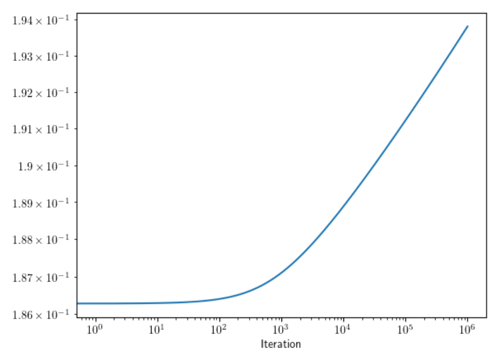

Figure 12: FLBR in a convex-concave setting with the payoff function $||x - y||^2$.