Consensus Multiplicative Weights Update: Learning to Learn using Projector-based Game Signatures

Nelson Vadori [1] Rahul Savani [2] Thomas Spooner [1] Sumitra Ganesh [1]

Abstract

Cheung and Piliouras (2020) recently showed that two variants of the Multiplicative Weights Update method - OMWU and MWU - display opposite convergence properties depending on whether the game is zero-sum or cooperative. Inspired by this work and the recent literature on learning to optimize for single functions, we introduce a new framework for learning last-iterate convergence to Nash Equilibria in games, where the update rule's coefficients (learning rates) along a trajectory are learnt by a reinforcement learning policy that is conditioned on the nature of the game: *the game signature*. We construct the latter using a new decomposition of two-player games into eight components corresponding to commutative projection operators, generalizing and unifying recent game concepts studied in the literature. We compare the performance of various update rules when their coefficients are learnt, and show that the RL policy is able to exploit the game signature across a wide range of game types. In doing so, we introduce CMWU, a new algorithm that extends consensus optimization to the constrained case, has local convergence guarantees for zero-sum bimatrix games, and show that it enjoys competitive performance on both zero-sum games with constant coefficients and across a spectrum of games when its coefficients are learnt.

1. Introduction

Motivated by applications in game theory, optimization, and Generative Adversarial Networks, the machine learning community has recently started using variants of the gradient method, such as optimistic gradient (Daskalakis et al., 2018) and consensus optimization (Mescheder et al., 2017). These methods enjoy linear convergence in cases like (unconstrained) bilinear games, where standard gradient descent fails (see also (Azizian et al., 2020; Gidel et al., 2019)). In the simplex-constrained setting where "strategy weights" are restricted to be non-negative and sum to one, deriving constant step-size [1] learning algorithms that display last-iterate convergence is difficult: to the best of our knowledge, existing algorithms include Optimistic Multiplicative Weights Update (OMWU), Optimistic Mirror Descent (OMD) and Optimistic Gradient Descent Ascent (OGDA). Such convergence guarantees with constant step-size were obtained for zero-sum bimatrix games in the case of OMWU (Daskalakis & Panageas, 2019), and then extended to the convex-concave zero-sum setting (Lei et al., 2021); for so-called "coherent" saddle point problems with a Lipschitz condition in the case of OMD, which includes the zero-sum bimatrix case (Mertikopoulos et al., 2019) (later generalized in (Hsieh et al., 2019)); for zero-sum games satisfying a so-called Saddle-Point Metric Subregularity condition in the case of OGDA, which includes the bimatrix and convex-concave setting (Wei et al., 2021). The difficulty is in part due to simplex constraints on the weights: tweaking one coordinate impacts all others, making it difficult for such an update rule to admit Nash Equilibria (NE) as fixed points. Last-iterate convergence is sometimes referred to as "pointwise", and has to be contrasted with convergence in the time-averaged sense.

In this work, we introduce **Consensus Multiplicative Weights Update (CMWU)**, a new algorithm that extends consensus optimization to the constrained case. It enjoys local last-iterate convergence to NE in the zero-sum bimatrix case, using constant learning rates. In contrast to OMWU, its update rule from t to $t+1$ only depends on players' strategies at time t. In contrast to OMD and OGDA, CMWU does not require the computation of mirror coordinates, although it requires a second order term. OMWU, CMWU and OMD all use softmax projection, but OGDA requires an orthogonal projection onto the unit simplex with respect to the L_2 norm, which requires vector sorting.

CMWU, like OMWU, OMD and OGDA, has convergence

[1] J.P. Morgan AI Research [2] Department of Computer Science, University of Liverpool. Correspondence to: Nelson Vadori <nelson.n.vadori@jpmorgan.com>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

[1] in the sense of time-independence.

guarantees, for bimatrix games, in the zero-sum case only. Recent work (Cheung & Piliouras, 2020) has shown that MWU and OMWU display opposite convergence properties depending on whether the game is zero-sum or cooperative, where players' payoffs are equal: OMWU (resp. MWU) converges (resp. diverges) in the zero-sum case, and vice-versa in the cooperative case. Since OMWU reduces to MWU with zero coefficient on the gradient difference term $\nabla_t - \nabla_{t-1}$, another way to see this opposite behavior is that if one chooses suitably the update rule's coefficients as a function of the nature of the game, one can hope to get convergence across a spectrum of games. The issue, however, is that a general two-player (2P) game is not purely zero-sum or cooperative, hence we need to answer the question: ***how to encode a 2P game?*** Recent literature has studied this question in specific settings: (Balduzzi et al., 2019) shows that every 2P zero-sum symmetric game can be split into the direct sum of a transitive and a cyclic component. (Cheung & Piliouras, 2020) introduces trivial games where the payoff is decoupled between the 2 players, and a related triviality metric that allows them to compute to which extent a game is trivial. In the game theory literature, it has been observed that commutative projectors can be used to construct (direct sum) game decompositions, for example (Hwang & Rey-Bellet, 2020) formulates decompositions into 2 and 3 components. We build on this work to formulate a new canonical decomposition of a 2P game into 2^n components, given n commutative projectors, and apply this result to the case $n = 3$, with projectors associated to zero-sum/cooperative, symmetric/antisymmetric and transitive/cyclic games. Importantly, this view generalizes the concept of transitivity in (Balduzzi et al., 2019) to any 2P game, and in addition we show that trivial games in (Cheung & Piliouras, 2020) in fact emanate from the same projector as transitive games. This unified view allows us to view trivial games in terms of their mirror, cyclic games, and gives an alternative to the triviality metric in (Cheung & Piliouras, 2020). Based on this decomposition, we introduce a new concept, the ***game signature***, which we use as a *feature* to learn the update rule's coefficients as a function of the game type, extending the learning to optimize (L2O) framework (Li & Malik, 2017) to games.

Contributions. **(1)** We introduce CMWU, a new constant step-size algorithm that enjoys local last-iterate convergence to NE in the constrained zero-sum bimatrix case, for which we show empirically that it enjoys competitive performance with the best known methods, see for example in figure 1 the case of rock-paper-scissors. **(2)** We introduce a new concept, the game signature, based on a new canonical decomposition of 2-player games into 2^n components, given n commutative projectors, and apply this result to the case $n = 3$, with projectors associated to zero-sum/cooperative, symmetric/antisymmetric and transitive/cyclic games. Importantly, this view generalizes the decomposition of (Balduzzi et al., 2019) to any 2P game, and unifies their transitive games with the trivial games of (Cheung & Piliouras, 2020). **(3)** Based on (1) and (2), we introduce a new framework for last-iterate convergence learning in games, where the algorithm update rule's coefficients along a trajectory are learnt by a reinforcement learning policy which is conditioned on the game signature. To our knowledge this is the first application of L2O to finding NE based on the nature of the game. We show empirically that our learned policy is able to exploit the game signature across a wide range of game types, that CMWU performs well in that framework, where we learn to exploit the gradient in transitive games and the Hessian in cyclic ones, and learn mirror behaviors across zero-sum and cooperative components.

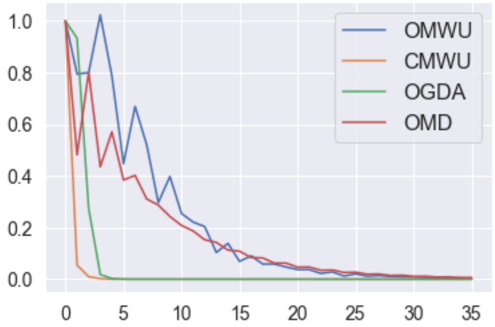

Figure 1. Rock-paper-scissors. Last-iterate convergence to Nash $\frac{\delta(x^t,y^t)}{\delta(x^0,y^0)}$. Best hyperparameters.

2. CMWU: Consensus Multiplicative Weights Update

Setting and notations. We consider a zero-sum bimatrix game where players receive payoffs $x^T Ay$ and $-x^T Ay$, $A \in \mathcal{M}_{n,m}$ is a matrix of size $n \times m$, $x \in \Delta_n$, $y \in \Delta_m$ the vectors of players' mixed strategies, and Δ_k the probability simplex in dimension k [2]. We define $v_A := x^{*T} Ay^*$ the value of the game, where (x^*, y^*) is a NE [3]. For $z \in \mathbb{R}^k$, we denote $D_z := \text{diag}(z)$ the diagonal matrix with z on the diagonal.

Key insights. We recall that in the unconstrained case, players' gradients are perturbed by the Hessian terms $AA^T x$, $A^T Ay$ to yield convergence, cf. details in appendix. There are two main insights in our extension of consensus optimization to the constrained setting. (1) what is the constrained equivalent of the unconstrained Hessian AA^T, and of its invertibility requirement guaranteeing its positive-definiteness so as to "bend" the eigenvalues of the game

[2] set of vectors with nonnegative coordinates summing to one.
[3] it is a classical result that at least one NE exists, and that v_A is the same across all possible NE.

Jacobian below one? **(2)** how can we design a Hessian-based update rule that admits any NE as a fixed point? For **(1)**, we introduce in definition 2.1 "renormalized" versions $H_x := A^T D_x A$, $H_y := A D_y A^T$ of the classical Hessian that we call **simplex-Hessians**. One way to see this renormalization is from the fact that, by commutativity of the spectrum of a matrix product, $sp(AB) = sp(BA)$, we can see the spectrum of H_x and H_y as scaled versions of that of AA^T. Related to invertibility, games like rock-paper-scissors do not have an invertible payoff matrix. Interestingly, our local convergence theorem 2.5 requires a **weak invertibility** condition (definition 2.4), where weak is both in the sense that it accommodates non-square matrices, and that it requires invertibility only on the subspace of vectors whose coordinates sum to zero. This is a relatively weak condition that explains why the algorithm works well for a large class of payoffs, including rock-paper-scissors. Finally, for **(2)**, a key observation is that equilibrium properties specifically make all NE be fixed points of the simplex-Hessian based CMWU update (proposition 2.2). Due to the simplex constraint, it is quite difficult to find update rules that admit NE as fixed points, since tweaking one coordinate impacts the others.

Definition 2.1. **(simplex-Hessians)** The simplex-Hessians $H_x \in \mathcal{M}_{m,m}$, $H_y \in \mathcal{M}_{n,n}$ associated to $x \in \Delta_n$, $y \in \Delta_m$ are defined as $H_x := A^T D_x A$, $H_y := A D_y A^T$.

The CMWU update. Simplex-Hessians are symmetric, square matrices by construction. We use them to extend consensus optimization to the constrained case. Let $h > 0$ be the learning rate and $\epsilon > 0$ the simplex-Hessian coefficient. We consider the multiplicative update $(x^{t+1}, y^{t+1}) = (\varphi_1(x^t, y^t), \varphi_2(x^t, y^t))$ at iteration stage t in (1), for $i \in [1, n]$ and $j \in [1, m]$. Note that, for $\epsilon = 0$, we recover the classical MWU. The key observation is proposition 2.2 that shows – using equilibrium properties only – that any NE is a fixed point of CMWU.

$$[\varphi_1(x,y)]_i := \frac{x_i \exp(h[Ay]_i - h\epsilon[H_y x]_i)}{\sum_{k=1}^n x_k \exp(h[Ay]_k - h\epsilon[H_y x]_k)}$$

$$[\varphi_2(x,y)]_j := \frac{y_j \exp(-h[A^T x]_j - h\epsilon[H_x y]_j)}{\sum_{k=1}^m y_k \exp(-h[A^T x]_k - h\epsilon[H_x y]_k)} \tag{1}$$

Proposition 2.2. ***(Fixed point property of CMWU)*** *Any NE (x^*, y^*) of the zero-sum bimatrix game A is a fixed point of the CMWU update (1), namely $\varphi_1(x^*, y^*) = x^*$, $\varphi_2(x^*, y^*) = y^*$.*

Proof. If $x_i^* = 0$, $[\varphi_1(x^*, y^*)]_i = x_i^* = 0$. By property of the NE (Daskalakis & Panageas, 2019), $x_i^* \neq 0 \Rightarrow [Ay^*]_i = v_A$, and $y_j^* \neq 0 \Rightarrow [A^T x^*]_j = v_A$. Hence if $x_i^* \neq 0$, $[H_{y^*} x^*]_i = [A D_{y^*} A^T x^*]_i = \sum_{j=1}^m A_{ij} y_j^* [A^T x^*]_j = \sum_{j: y_j^* \neq 0} A_{ij} y_j^* [A^T x^*]_j = [Ay^*]_i v_A = v_A^2$, and so $[\varphi_1(x^*, y^*)]_i = x_i^* \frac{\exp(hv_A - h\epsilon v_A^2)}{\exp(hv_A - h\epsilon v_A^2) \sum_{k=1}^n x_k} = x_i^*$. Similarly $\varphi_2(x^*, y^*) = y^*$ since for $y_j^* \neq 0$, $[H_{x^*} y^*]_j = v_A^2$. □

Definition 2.3. **(quasi-strict NE)** A NE (x^*, y^*) of the zero-sum game A is said to be quasi-strict if $[Ay^*]_i < v_A$ whenever $x_i^* = 0$, and $[A^T x^*]_j > v_A$ whenever $y_j^* = 0$.

Quasi-strict equilibria were introduced originally in (Harsanyi, 1973). Every interior NE [4] is by construction quasi-strict, and every bimatrix game has at least one quasi-strict equilibrium (Jansen, 1981; Norde, 1999; Brandt & Fischer, 2008). In general, bimatrix games can have multiple quasi-strict equilibria. The proof of local convergence for OMWU (Daskalakis & Panageas, 2019) requires uniqueness of the NE, however in our case we only need the weaker quasi-strict property to handle the non-interior equilibrium coordinates. We now introduce the concept of weak invertibility, which coincides with classical invertibility when $p = q$ and $\mathcal{V} = \mathbb{R}^q$.

Definition 2.4. **(weak invertibility)** Let $\mathcal{V} \subseteq \mathbb{R}^q$ a subspace. A possibly non-square $p \times q$ matrix M is said to be weakly $\mathcal{V}-$invertible if $Ker(M) \cap \mathcal{V} = \{0\}$, i.e. if $Mz \neq 0$ whenever $z \in \mathcal{V}$, $z \neq 0$.

We now discuss the key technical contribution of theorem 2.5. Let $\mathcal{Z}_n \subseteq \mathbb{R}^n$ the subspace of vectors whose coordinates sum to zero. $\mathcal{Z}_n$ is a space of dimension $n - 1$ as it is the eigenspace of the zero eigenvalue of the "all ones" J_n matrix. Regarding the eigenvalue analysis needed to show convergence in theorem 2.5, OMWU attacks the antidiagonal part of the Jacobian, in the sense that it aims at modifying the latter so as to get convergence. As stated in (Lei et al., 2021), which proves local convergence of OMWU in the convex-concave zero-sum setting: *"The key result of spectral analysis in (Daskalakis & Panageas, 2019) is in Lemma B.6 which states that an skew-symmetric matrix has imaginary eigenvalues.* In our CMWU case, we attack the diagonal part of the Jacobian, which does not exploit skew-symmetry and thus necessitates a different proof technique. Specifically, our technical contribution is to see the problem as the zero eigenvalue perturbation of the real symmetric matrices $J_{|\operatorname{supp}(x^*)|}$, $J_{|\operatorname{supp}(y^*)|}$ [5] which results in highlighting the interplay between the game equilibrium and the corresponding eigenspaces $\mathcal{Z}_{|\operatorname{supp}(x^*)|}$, $\mathcal{Z}_{|\operatorname{supp}(y^*)|}$ of vectors that sum to zero. We show that for small h, the perturbed eigenvalues have strictly positive real part, which makes the Jacobian of the update rule have spectral radius strictly less than one. We obtain a concise theorem condition

[4] i.e. one where both players play all of their available actions with positive probability.

[5] the support $\operatorname{supp}(z)$ of a vector z is a set which elements are the indexes of the non-zero entries of z.

at the intersection of the game payoff matrix and equilibrium, which explains why CMWU will work well on a large class of games which are not invertible in the classical sense, like rock-paper-scissors. Indeed, for the latter, $Az = 0$ for $z \neq 0$ only when z has all equal entries, which does not belong to $\mathcal{Z}_3$.

Theorem 2.5. ***(local convergence of CMWU)*** *Let (x^*, y^*) a quasi-strict NE of the zero-sum game A. Let A_* the payoff matrix whose rows have been restricted to $\operatorname{supp}(x^*)$ and columns to $\operatorname{supp}(y^*)$. Assume that A_* (resp. A_*^T) is weakly $\mathcal{Z}_{|\operatorname{supp}(y^*)|}$−invertible (resp. $\mathcal{Z}_{|\operatorname{supp}(x^*)|}$−invertible). Then, for sufficiently small h and $\epsilon = Ch^{-\delta}$ satisfying [6] $\epsilon < |v_A^{-1}|$ for some $\delta \in [0, 1)$, $C > 0$ not depending on h, there exists a neighborhood $U(h) \subset \Delta_n \times \Delta_m$ of (x^*, y^*) such that for all $(x^0, y^0) \in U(h)$, the CMWU iterates (1) converge to the NE, namely $\lim_{t\to\infty}(\varphi_1(x^t, y^t), \varphi_2(x^t, y^t)) = (x^*, y^*)$.*

3. Learning to reach Nash Equilibria using Game Signatures

We introduce ***game signatures*** in section 3.1 based on a new game decomposition in theorem 3.4, which we use as a feature to learn CMWU update's coefficients for general 2P games in section 3.2. This new decomposition of a game given a family of n commutative projectors can be seen as a way to *"play Lego"* with the game and is in our opinion an important observation of this work.

3.1. Projector-based Game Signatures

We let the pure strategy sets of the two players $\mathcal{X}_1$, $\mathcal{X}_2$ be probability spaces equipped with probability measures μ_1, μ_2. The space of 2P games $\mathcal{G}$ can be identified with the space of pairs of functions $f = (f_1, f_2)$ on $\mathcal{X}_1 \times \mathcal{X}_2 \to \mathbb{R}$, where player j receives $f_j(x, y)$ when players play the pure strategies x and y.

We seek to answer the question: ***how to encode a 2P game?*** In the recent work (Cheung & Piliouras, 2020), *trivial* games have been introduced in the bimatrix context: this class of games consists of games which payoff is decoupled. In order to quantify how trivial a game is, authors introduce a triviality metric which consists in computing the closest trivial game to it, by taking the minimum (matrix) distance over all possible trivial games. In a separate work (Balduzzi et al., 2019), the concept of *transitive* game was introduced in the case of 2P zero-sum symmetric games, and it was proved that every such game can be decomposed uniquely into a transitive and a cyclic component, i.e. that the space of 2P zero-sum symmetric games is equal to the direct sum of transitive and cyclic games. In the bimatrix case, a transitive game has payoff $A_{ij} = a_i - a_j$. We summarize game definitions from these works in definition 3.1.

Definition 3.1. (Zero-sum, cooperative, symmetric, antisymmetric, trivial, transitive and cyclic games.) A game $f = (f_1, f_2)$ is zero-sum if $f_1 = -f_2$, cooperative if $f_1 = f_2$. If $\mathcal{X}_1 = \mathcal{X}_2$ and $f^T(x, y) := f(y, x)$, a game is symmetric if $f_1 = f_2^T$, antisymmetric if $f_1 = -f_2^T$. A game is trivial (Cheung & Piliouras, 2020) if $f_j(x, y) = \phi_j(x) + \lambda_j(y)$ for some ϕ_j, λ_j. A zero-sum symmetric game is transitive (Balduzzi et al., 2019) if $f_1(x, y) = \phi(x) - \phi(y)$ for some ϕ, and cyclic if $\int f_1(x, y) d\mu_2(y) = 0$ for all $x \in \mathcal{X}_1$.

It is then natural to ask the following question: *How to extend the concepts of transitive and cyclic components of a zero-sum symmetric game to any 2P game? Is there a link between transitive games and trivial games?*

We provide answers to these questions in proposition 3.5 and corollary 3.6 by generalizing transitivity, unifying it with triviality, and providing a general game decomposition of 8 components valid for *any* 2P game [7], 2 of which being the zero-sum symmetric transitive and cyclic games in (Balduzzi et al., 2019), 4 of which being trivial games of (Cheung & Piliouras, 2020), while the 4 other ones constitute a generalization of the concept of cyclicity to any 2P game. Actually we go further: we provide a systematic way to obtain game decompositions of 2^n components in direct sum, given n commutative projectors. We build on the key observation in (Candogan et al., 2011; Hwang & Rey-Bellet, 2020) that *projection operators*, or **projectors**, yield game decompositions. In particular, our unified view shows that every 2P game contains a trivial component, which is easy to compute using ρ_T. This represents an alternative to the triviality metric in (Cheung & Piliouras, 2020): we can simply define the triviality of a game as the norm of its trivial component. Our toolkit of corollary 3.6 yields theorem 1 in (Balduzzi et al., 2019) as a corollary for free, without needing to resort to combinatorial Hodge theory. We begin with the definition of a projector, central to our analysis.

Definition 3.2. (projector) A projector $\rho : \mathcal{G} \to \mathcal{G}$ on $\mathcal{G}$ is a linear operator satisfying $\rho^2 = \rho$. In that case we have $\mathcal{G} = K_\rho \oplus R_\rho$ and $K_\rho = R_{\mathrm{Id}-\rho}$, where K_ρ, R_ρ are the kernel and range of ρ.

Here, $\oplus$ is the direct sum operator, indicating that every element of $\mathcal{G}$ can be decomposed uniquely into an element of K_ρ and an element of R_ρ. In that case, $K_\rho \cap R_\rho = \{0\}$. Our new decomposition in theorem 3.4 and corollary 3.6 is an extension to the case $n > 2$ of the well-known lemma 3.3.

Lemma 3.3. *If p_1, p_2 are commutative projectors on $\mathcal{G}$,*

[6] we use the convention $|v_A^{-1}| := +\infty$ if $v_A = 0$.

[7] this is valid under the assumption that players have the same pure strategy set, otherwise we get a decomposition into 4 components, as specified in corollary 3.6.

then $p_1 p_2$ is a projector and:

$$\mathcal{G} = (K_{p_1} \cap K_{p_2}) \oplus (K_{p_1} \cap R_{p_2}) \oplus (R_{p_1} \cap K_{p_2}) \oplus (R_{p_1} \cap R_{p_2}) \, .$$

Theorem 3.4. ***(Game Decomposition)*** *Let $(\rho_i)_{i \in [1,n]}$ a family of n commutative projectors on $\mathcal{G}$. Then we have the canonical direct sum decomposition into 2^n components:*

$$\mathcal{G} = \bigoplus_{\mathcal{C}_i \in \{K_{\rho_i}, R_{\rho_i}\}} \bigcap_{i=1}^{n} \mathcal{C}_i \, . \qquad (2)$$

Theorem 3.4 tells us informally that starting from a game, we can compute its components easily by n compositions of projectors of the form ρ_i or $\mathrm{Id} - \rho_j$, where the order doesn't matter because of commutativity.

Let $\mathcal{N}_2 f(x,y) := \int f(x,z) d\mu_2(z)$, $\mathcal{N}_1 f(x,y) := \int f(z,y) d\mu_1(z)$, $\widehat{\mathcal{N}} := \mathcal{N}_1 + \mathcal{N}_2 - \mathcal{N}_1 \mathcal{N}_2$. We define the operators $\boldsymbol{\rho_Z}$, $\boldsymbol{\rho_S}$, $\boldsymbol{\rho_T}$ from $\mathcal{G} \to \mathcal{G}$ associated to zero-sum, symmetric and transitive games as:

$$\begin{aligned} \boldsymbol{\rho_Z} &: (f_1, f_2) \to \frac{1}{2}(f_1 - f_2, f_2 - f_1) \\ \boldsymbol{\rho_S} &: (f_1, f_2) \to \frac{1}{2}(f_1 + f_2^T, f_2 + f_1^T) \\ \boldsymbol{\rho_T} &: (f_1, f_2) \to (\widehat{\mathcal{N}} f_1, \widehat{\mathcal{N}} f_2) \end{aligned} \qquad (3)$$

In the zero-sum symmetric case, ρ_T reduces to the form $\phi(x) - \phi(y)$ in (Balduzzi et al., 2019), but in general it is different (of the form $\phi(x) + \lambda(y)$) and allows to define the transitive/cyclic components of any 2P game given by ρ_T and $\mathrm{Id} - \rho_T$. Similarly, ρ_Z, $\mathrm{Id} - \rho_Z$, ρ_S, $\mathrm{Id} - \rho_S$ compute the zero-sum, cooperative, symmetric and antisymmetric components of a game. (Hwang & Rey-Bellet, 2020) observed that ρ_Z and ρ_T commute, and use it to obtain decompositions into 2 and 3 components. In our case, we use theorem 3.4 to obtain our new decompositions into 2^n components, which allows to generalize and unify (Balduzzi et al., 2019; Cheung & Piliouras, 2020). For example, the cooperative-symmetric-cyclic component of a game f is simply given by $(\mathrm{Id} - \rho_Z) \rho_S (\mathrm{Id} - \rho_T) f$, and every component is obtained by applying such compositions of projectors.

Proposition 3.5. ***(generalization of transitivity/cyclicity and unification with triviality)*** *If a 2P game f is zero-sum symmetric transitive (resp. cyclic) in the sense of (Balduzzi et al., 2019), then $f \in R_{\rho_T}$ (resp. $f \in K_{\rho_T} = R_{\mathrm{Id} - \rho_T}$). Further, a 2P game is trivial in the sense of (Cheung & Piliouras, 2020) if and only if it is transitive, i.e. the class of trivial games coincides with R_{ρ_T}.*

Corollary 3.6. ***(Canonical decomposition of 2P games)*** *ρ_Z, ρ_T are commutative projectors. Consequently, by theorem 3.4, every 2P game can be decomposed into the direct sum of 4 components:* ***(ZT)*** *zero-sum-transitive,* ***(ZCy)*** *zero-sum-cyclic,* ***(CT)*** *cooperative-transitive,* ***(CCy)*** *cooperative-cyclic. Further, if $\mathcal{X}_1 = \mathcal{X}_2$, then ρ_S is a well-defined projector and commutes with ρ_Z, ρ_T. Consequently, any such game can be decomposed into the direct sum of 8 components:* ***(ZST)*** *zero-sum-symmetric-transitive,* ***(ZSCy)*** *zero-sum-symmetric-cyclic,* ***(ZAT)*** *zero-sum-antisymmetric-transitive,* ***(ZACy)*** *zero-sum-antisymmetric-cyclic,* ***(CST)*** *cooperative-symmetric-transitive,* ***(CSCy)*** *cooperative-symmetric-cyclic,* ***(CAT)*** *cooperative-antisymmetric-transitive,* ***(CACy)*** *cooperative-antisymmetric-cyclic.*

Finally, we define the concept of game norm, which is used to construct the game signature based on theorem 3.4 and corollary 3.6.

Definition 3.7. **(Game norm)** Let $f = (f_1, f_2) \in \mathcal{G}$ a game, and $|| \cdot ||$ a norm on the space of functions $\mathcal{X}_1 \times \mathcal{X}_2 \to \mathbb{R}$. The game norm $||f||$ is defined as $||f|| := \frac{1}{2}(||f_1|| + ||f_2||)$.

Definition 3.8. **(Game signature)** Let $(\rho_i)_{i \in [1,n]}$ a family of n commutative projectors. The game signature is defined as the vector of size 2^n consisting of the norms of the 2^n game components associated to the decomposition of theorem 3.4.

Let us look at how to compute the game components and signature in the bimatrix case with payoff matrices (A, B). We have from (3), $\rho_Z(A,B) = \frac{1}{2}(A - B, B - A)$, $\rho_S(A,B) = \frac{1}{2}(A + B^T, B + A^T)$, $\rho_T(A,B) = (\widehat{A}^{(1)} + \widehat{A}^{(2)} - \widehat{A}, \widehat{B}^{(1)} + \widehat{B}^{(2)} - \widehat{B})$, where $\widehat{A}^{(1)}$, $\widehat{A}^{(2)}$, $\widehat{A}$ are the matrices with entries $\widehat{A}^{(1)}_{ij} := \frac{1}{m} \sum_{j=1}^m A_{ij}$, $\widehat{A}^{(2)}_{ij} := \frac{1}{n} \sum_{i=1}^n A_{ij}$, $\widehat{A}_{ij} := \frac{1}{mn} \sum_{i=1}^n \sum_{j=1}^m A_{ij}$. The projectors associated respectively to cooperative, antisymmetric and cyclic games are $\rho_C(A,B) := [\mathrm{Id} - \rho_Z](A,B)$, $\rho_A(A,B) := [\mathrm{Id} - \rho_S](A,B)$, $\rho_{Cy}(A,B) := [\mathrm{Id} - \rho_T](A,B)$. With these definitions, the 8 components of corollary 3.6 are simply computed by composition of above projectors, namely $\rho_Z \rho_S \rho_T$, $\rho_Z \rho_S \rho_{Cy}$, $\rho_Z \rho_A \rho_T$, $\rho_Z \rho_A \rho_{Cy}$, $\rho_C \rho_S \rho_T$, $\rho_C \rho_S \rho_{Cy}$, $\rho_C \rho_A \rho_T$, $\rho_C \rho_A \rho_{Cy}$. Following definitions 6 and 7, the game signature is taken to be the vector of size 8 containing the norms $\frac{1}{2}(||A_i|| + ||B_i||)$ of the components $(A_i, B_i)_{i \in [1,8]}$, where the matrix norm is chosen to be the L_2 norm. We further divide these 8 norms by their overall sum so as to interpret them as weights.

3.2. RL formulation of signature-based learning in games

Game signature-based learning. We use our previous analysis to extend to games the framework developed in (Li & Malik, 2017), where a RL policy π outputs the step Δx in the function optimization update $x^{t+1} \leftarrow x^t + \Delta x$ as a function of current and past "trajectory information" such as gradient and function value at previous points. Since finding one NE of a bimatrix game is PPAD-hard (Chen et al., 2009; Daskalakis et al., 2009), we cannot expect a universal learnt policy to be polynomial-time for unrestricted inputs. However, worst-case "PPAD-hard instances" are very pathological so there is every possibility for good performance over many reasonable distributions over games.

There are two fundamental differences between games and single functions: **(i)** the solution concept is not the same: in games we are not trying to minimize a function, but to find a NE [8]. **(ii)** the update rule should depend on the nature of the game, as discussed in section 1 on the mirror behavior of CMWU and MWU. **(i)** can be dealt with using the concept of **best response gaps** in definition 3.9, whose sum δ is also called "Nash convergence metric" in (Lanctot et al., 2017). Indeed, by definition, (x^*, y^*) is a NE if and only if $\delta_1(x^*, y^*) = \delta_2(x^*, y^*) = 0$. One of the main ideas in this work is to give an answer to **(ii)** using our concept of game signature in definition 3.8, which will allow to learn a range of behaviors across the spectrum of games. In contrast to classical optimization where a suite of convergent algorithms exist, there is no algorithm that displays last-iterate convergence to NE in the case of general 2P games. In the following we consider bimatrix games (A, B) of size $n \times m$, where $x \in \Delta_n$, $y \in \Delta_m$.

Definition 3.9. **(best response gap)** The best response gaps of players 1 and 2 are defined as:

$$\begin{aligned} \delta_1(x,y) &:= \max_{i\in[1,n]} [Ay]_i - x^T Ay \\ \delta_2(x,y) &:= \max_{j\in[1,m]} [x^T B]_j - x^T By \\ \delta(x,y) &:= \delta_1(x,y) + \delta_2(x,y) \end{aligned} \tag{4}$$

RL state, actions and rewards. We frame our learning problem in the standard single-agent RL paradigm. We consider the modified CMWU update (1) where each player $k \in \{1, 2\}$ is allowed to have its own gradient and Hessian coefficients h_k, ϵ_k, on which we relax the positivity constraint. Note that in the bimatrix case (A, B), simplex-Hessians become $H_x := A^T D_x A$, $H_y := BD_y B^T$, see e.g. (Letcher et al., 2019) for the unconstrained case. We allow $h_k = h_k(x^t, y^t)$, $\epsilon_k = \epsilon_k(x^t, y^t)$ to depend on current trajectory location, importantly we do not allow these parameters to depend explicitly on time t. At stage t, the **state** s_t of the RL policy consists of the game signature plus trajectory information $\delta_1(x^t, y^t)$, $\delta_2(x^t, y^t)$, Ay^t, $B^T x^t$, H_{x^t}, H_{y^t}, $x^{t,T} Ay^t$, $x^{t,T} By^t$. Its **action** $a_t \sim \pi(\cdot|s_t) \in \mathbb{R}^4$ consists of $a_t = (h_1, h_2, h_1\epsilon_1, h_2\epsilon_2)$. We optimize the following objective over a trajectory of length τ:

$$\max_{\pi} \mathbb{E}_\pi \sum_{t=0}^{\tau-1} R(s_t, a_t), \;\; R(s_t, a_t) := -\frac{\delta(x^{t+1}, y^{t+1})}{\delta(x^0, y^0)} \tag{5}$$

where the expectation is taken over the initial location (x^0, y^0), the payoff matrices A, B of the two players (randomly sampled at the beginning of the episode), and the stochasticity of π. We formulated our problem using RL, as RL policies tend to be more stable and overcome the problem of compounding errors, where errors accumulated on a trajectory could lead to divergence (Li & Malik, 2017). This allowed us in practice to train on long trajectories $\tau = 1000$. However, we believe that it would also be possible to use LSTM-based approaches (Andrychowicz et al., 2016; Chen et al., 2020), or (Shen et al., 2021) that introduces an architecture to solve unconstrained min-max problems, and possibly output directly the updates Δx, Δy instead of the update rule's coefficients. We leave this topic for future research. On the other hand, outputing the coefficients allows us to obtain nice relationships in section 4 such as high discrepancy between gradient and Hessian coefficients on cyclic games, as opposed to transitive games where the gradient coefficient is high, and build on the mirror relationship seen in (Cheung & Piliouras, 2020), which another way to reformulate is that the coefficients should be chosen as a function of the game type.

4. Experiments

Zero-sum bimatrix games. We first evaluate in table 1 the performance of CMWU against known algorithms on 200 randomly sampled bimatrix zero-sum games in dimension $n = m = 25, 50, 75, 100$, where each matrix entry, as well as the initial location x^0, y^0 are sampled independently $\mathcal{U}[-1, 1]$, where $\mathcal{U}$ is the uniform distribution [9]. To quantify speed of convergence to NE and treat different seeds equally with respect to their random starting point, we consider the convergence metric $\beta_\tau := 100 \cdot \tau^{-1} \sum_{t=1}^{\tau} \frac{\delta(x^t, y^t)}{\delta(x^0, y^0)}$ which quantifies the trajectory-averaged best response gap relative to the starting point, for a given iteration budget τ, in percentage points. OMWU, OGDA and OMD all have one parameter, the learning rate h, whereas CMWU has in addition the Hessian coefficient ϵ. We run experiments for a range of constant learning rates h, and pick the best for each algorithm and each game size. The complete set of results for all values of h and score standard deviations are reported in the appendix, together with background on these algorithms, see also (Daskalakis & Panageas, 2019), (Mertikopoulos et al., 2019), (Wei et al., 2021). We find that OGDA performs best, but CMWU performs competitively and does well on a wider range of learning rates. Note that OMWU, CMWU and OMD all use softmax projection, but OGDA requires an orthogonal projection onto the unit simplex with respect to the L_2 norm, which requires vector sorting.

[8] other solution concepts exist, such as correlated equilibria.

[9] we apply softmax to x^0, y^0.

Table 1. Zero-sum bimatrix games: average score β_{500} over 200 seeds, for various game sizes n. Best learning rate h for each algorithm. $\epsilon = 0.25 \cdot h^{-1}n$.

n	CMWU	OGDA	OMD	OMWU
25	8	**2**	25	26
50	7	**3**	23	24
75	8	**2**	21	24
100	8	**2**	17	24

Learning last-iterate convergence on general games. We consider square games of size $n = m = 10$. During training, for each RL episode of length $\tau = 1000$, we sample randomly A, B, (x^0, y^0) as previously discussed. We compute the 8 pure components $[(A_j, B_j)]_{j\in[1,8]}$ of corollary 3.6, and consider a mixture $(\sum_{k=1}^3 w_k A_{j_k}, \sum_{k=1}^3 w_k B_{j_k})$ of 3 such components j_k chosen at random (j_k's are not necessarily distinct), with respective weights w_k chosen at random in the probability simplex. We then apply the RL-based methodology in section 3.2 to learn the algorithm coefficients along trajectories.

We consider 3 versions of each algorithm: ***Base***, ***Partial-RL*** and ***Full-RL***, all trained with the methodology described above. The RL policy in the ***Base*** method learns the coefficients of the vanilla versions of the algorithms (h for all algorithms, ϵ in addition for CMWU), and has its state s_t masked, i.e. it doesn't know about trajectory information nor the game signature (cf. section 3.2), and therefore outputs constant coefficients (chosen however to optimize performance during training across the spectrum of games). The ***Partial-RL*** method is an improvement over the Base method in that the policy knows about trajectory information and the game signature. The ***Full-RL*** method learns 4 coefficients for each algorithm. For CMWU, we learn $(h_1, h_2, \epsilon_1, \epsilon_2)$ as mentioned in section 3.2. For OGDA and OMD, we learn separate coefficients for both players and both the mirror and regular steps. For OMWU, we learn different coefficients for the current and previous gradients ∇_t, ∇_{t-1}.

We train policies with PPO (Schulman et al., 2017). At test time, we evaluate on mixtures of $k = 1$ to 4 pure components as reported in table 2, where the acronyms used are defined in corollary 3.6, over 200 seeds (iteration budget τ is detailed in appendix). We construct these $k-$mixtures as previously discussed. Note that for clarity of presentation, we have grouped the symmetric and antisymmetric versions of the games on the same row, i.e. the row ZT will contain mixtures of ZST and ZAT, the row ZT + CCy mixtures of ZST, ZAT, CSCy, CACy. We find in table 2 that RL-based methods perform well across the spectrum of games compared to the base counterparts, indicating that the policy is able to exploit trajectory information as well as the game signature. CMWU displays slightly best performance across all k-mixtures, notably on the challenging mixture of zero-sum and cooperative cyclic components ZCy + CCy, where other methods fail. There, we learn to apply low, sometimes negative learning rate, as well as high Hessian coefficient in order to "tame" the cyclicity. In figure 2, we illustrate this phenomenon by displaying last-iterate convergence $\frac{\delta(x^t,y^t)}{\delta(x^0,y^0)}$ as a function of time t. To further confirm that the policy is able to exploit the signature, we conducted a small ablation study: we ran the Partial-RL CMWU method of table 2 by zeroing-out the game signature, so that the policy input contains all trajectory information except from the signature. We get results in line with the Base method, where no information is known, which shows that the signature is crucial: specifically, ZCy + CCy and ZT + ZCy + CCy respectively decrease from 8 to 96, 19 to 70, hence we learn how to converge in these games by exploiting the signature.

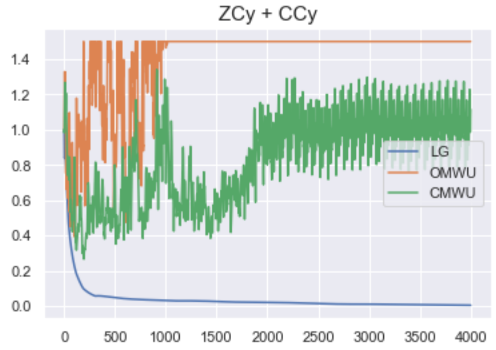

Figure 2. ZCy + CCy: last-iterate convergence $t \rightarrow \frac{\delta(x^t,y^t)}{\delta(x^0,y^0)}$. Full-RL CMWU learns to use negative gradient coefficient h_k and high Hessian coefficient $|h_k\epsilon_k|$. OMWU and CMWU diverge.

In figure 5 we plot for Full-RL CMWU the mean value and pairwise correlations among the learnt coefficients across the 8 pure game components [10]. We see that **(1)** for transitive games, our method learns to apply higher gradient coefficients h_k: this matches intuition since these games have decoupled payoff, hence players can act more intensely on their gradients. For cyclic games, it applies higher absolute Hessian to counter the cyclicity. **(2)** We see a **mirror effect** between zero-sum and cooperative games as suggested in (Cheung & Piliouras, 2020): Hessian coefficients have opposite sign, which also reflects on the correlations. **(3)** The performance and learned coefficients for the symmetric and antisymmetric parts of a given game type are almost identical, which seems to indicate that this feature has little importance. **(4)** The correlation $\mathrm{corr}(\delta_1 - \delta_2, h_1 - h_2)$ is always significantly positive, 74% in average. This hints at a possible "WoLF effect" learnt by the policy, similar

[10] we compute the correlation between the coefficients time-series for each seed, and average over seeds.

to (Bowling & Veloso, 2001): when the best response gap of player 1 is low compared to player 2 (player 1 "winning"), then player 1 becomes more careful. We leave this interesting observation for future work.

In figure 3 we compute, for the full-RL CMWU method, the average standardized per-episode-trajectory, i.e. "shape", of coefficients of gradient $G_k = h_k$ and Hessian $H_k = -h_k\epsilon_k$ of players $k = 1, 2$ across game types. We obtain these average trajectories as follows: for each seed, we consider the standardized time-series of coefficients [11], and then average the latter over seeds. We see that typically, there is a transient regime where the coefficients change significantly, followed by a stationary regime where they stabilize (neighborhood of the Nash). In addition to the mirror effect between zero-sum and cooperative games discussed earlier in figure 5, we see a another such mirror effect where in the zero-sum case, all coefficients are higher in the transient regime than in the stationary regime, whereas in the cooperative case, both Hessian coefficients adopt an opposite behavior where they are lower in the transient regime.

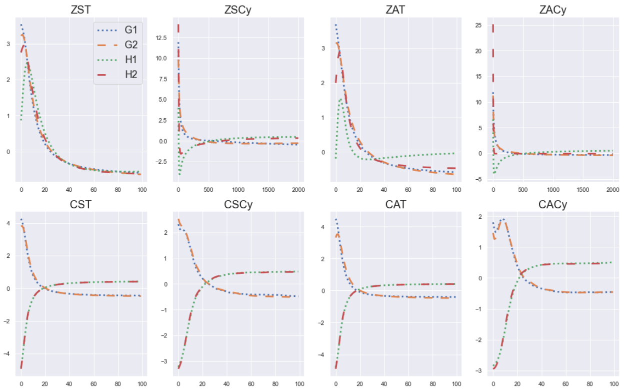

Figure 3. Average standardized per-episode-trajectory ("shape") of coefficients of gradient $G_k = h_k$ and Hessian $H_k = -h_k\epsilon_k$ of players $k = 1, 2$ across game types, as a function of time t.

In figure 4, we used the standard KernelExplainer tool associated to the SHAP algorithm (Lundberg & Lee, 2017) to look at the contributions of 4 input feature groups ("BR": best response gaps δ_k, "GRAD": gradient, "HESS": Hessian, "PAY": payoff value) to the gradient and Hessian coefficients h_k, $-h_k\epsilon_k$, which constitute the RL policy output. We see that in the transitive case, the gradient is the most important, but for cyclic games, we get a more varied behavior. For each of the 4 input groups, the percentage contribution displayed is the sum of the absolute SHAP contributions of its individual components: for example, for the best response input "BR", we sum the absolute contributions of the two player's best responses δ_1 and δ_2. We normalize SHAP contributions such that for each game type, they sum to one. Further details are provided in appendix, where we also display percentage contributions for the "signed" case where for each one of the 4 feature groups, we sum over the signed SHAP contributions of its components, rather than absolute.

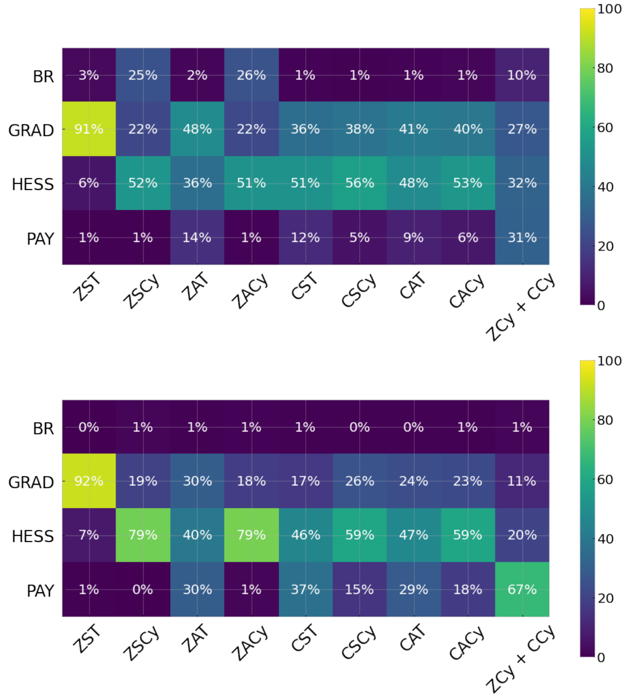

Figure 4. Percentage contribution towards absolute SHAP importance aggregated across absolute values with respect to **(top)** gradient learning rates $G_k = h_k$ and **(bottom)** Hessian learning rates $H_k = -h_k\epsilon_k$ for players k for the four input groups and eight unique game types. Columns are normalized to sum to one.

5. Conclusion and Future Research

We introduced a new algorithm, CMWU, together with a new framework for learning NE based on the nature of the game, materialized by our game signature and constructed using new game decompositions that generalize and unify game concepts studied in the recent works. We believe this work asks new research questions, among which we suggest **(1)** Extension of CMWU from zero-sum bimatrix to the convex-concave setting, as was done for OMWU (Lei et al., 2021), and investigation of global convergence properties of CMWU. **(2)** Allowing the neural network to output the update steps Δx, Δy directly as done in (Andrychowicz et al., 2016; Chen et al., 2020; Shen et al., 2021), as opposed to the update rule's coefficients. **(3)** Finding other relevant projectors in the space of 2P games. Can the game signature be learnt from experience? If yes, can the learnt features be mapped to new projectors?

[11] substract the mean, divide by standard deviation.

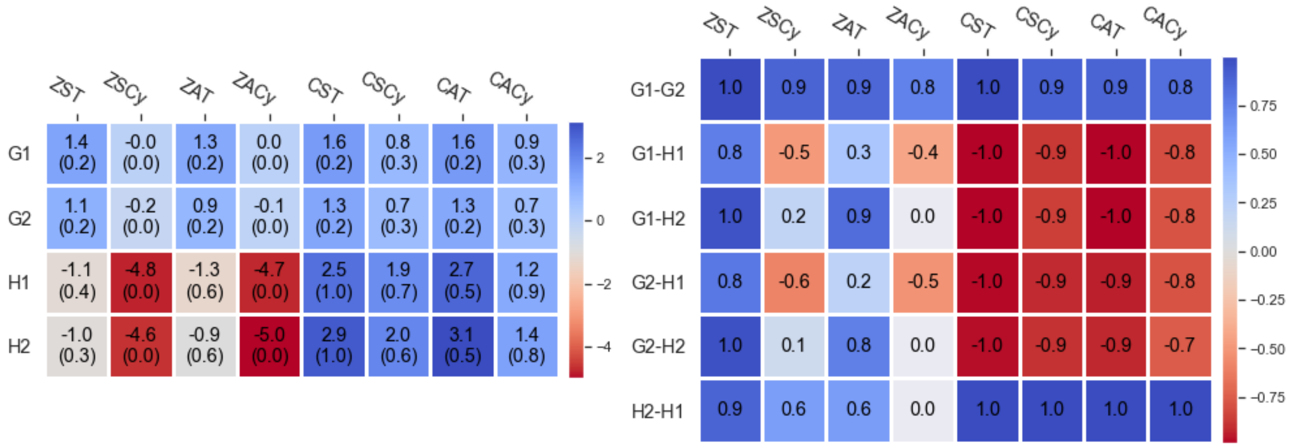

Figure 5. Full-RL CMWU coefficients of gradient $G_k = h_k$ and Hessian $H_k = -h_k \epsilon_k$ of players $k = 1, 2$ across 8 pure game components. **(left)** Mean value. **(right)** Pairwise correlations. StDev in brackets.

Table 2. Learning Games for various algorithms and methods, cf. details in text. OMWU (**O**), CMWU (**C**), OGDA (**G**), OMD (**M**). Full-RL (**F**), Partial-RL (**P**), Base (**B**). Average score β_τ over 200 seeds.

Game Type	C-F	G-F	M-F	O-F	C-P	G-P	M-P	O-P	C-B	G-B	M-B	O-B
ZT	6	93	5	5	8	3	4	21	7	**1**	3	4
ZCy	4	254	19	16	**3**	79	**3**	73	7	437	**3**	4
CT	4	119	5	5	5	**1**	2	14	9	**1**	2	4
CCy	4	791	4	4	6	1	3	11	21	**0**	3	3
Avg. Mixtures of 1	5	314	8	8	6	21	**3**	30	11	110	**3**	4
ZT + ZCy	**7**	89	10	17	**7**	16	42	40	29	122	53	47
ZT + CT	3	91	3	3	4	1	2	11	6	**0**	1	2
ZT + CCy	6	289	4	4	7	**1**	3	13	16	**1**	3	4
ZCy + CT	15	101	**10**	15	16	16	26	39	11	147	25	16
ZCy + CCy	10	322	30	58	**8**	25	120	104	116	137	123	119
CT + CCy	5	275	5	5	6	**1**	3	13	12	**1**	3	4
Avg. Mixtures of 2	**8**	195	10	17	**8**	10	33	37	32	68	35	32
ZT + ZCy + CT	12	39	9	18	13	**3**	38	35	22	40	39	32
ZT + ZCy + CCy	19	114	23	36	19	**13**	94	66	69	41	93	87
ZT + CT + CCy	5	104	5	5	7	**1**	2	14	11	**1**	2	3
ZCy + CT + CCy	20	122	18	32	28	**13**	84	61	58	33	78	72
ZT + ZCy + CT + CCy	15	49	12	18	18	**5**	40	32	36	9	38	34
Avg. Mixtures of 3-4	14	86	13	22	17	**7**	52	42	39	25	50	46
Avg. k-Mixtures	**9**	198	10	16	10	13	29	36	27	68	29	27

References

Andrychowicz, M., Denil, M., Gomez, S., Hoffman, M. W., Pfau, D., TomSchaul, Shillingford, B., and Freitas, N. D. Learning to learn by gradient descent by gradient descent. In *Advances in Neural Information Processing Systems*, 2016.

Azizian, W., Mitliagkas, I., Lacoste-Julien, S., and Gidel, G. A Tight and Unified Analysis of Gradient-Based Methods for a Whole Spectrum of Differentiable Games. *AISTATS*, 2020.

Balduzzi, D., Racaniere, S., Martens, J., Foerster, J., Tuyls, K., and Graepel, T. The Mechanics of n-Player Differentiable Games. *Proceedings of the 35th International Conference on Machine Learning*, 2018.

Balduzzi, D., Garnelo, M., Bachrach, Y., Czarnecki, W., Perolat, J., Jaderberg, M., and Graepel, T. Open-ended learning in symmetric zero-sum games. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 434–443, 2019.

Bhatia, R. *Matrix Analysis*. Springer-Verlag New York, 1997.

Bowling, M. and Veloso, M. Rational and Convergent Learning in Stochastic Games. *IJCAI*, 2001.

Brandt, F. and Fischer, F. On the hardness and existence of quasi-strict equilibria. In *Algorithmic Game Theory*, pp. 291–302. Springer Berlin Heidelberg, 2008.

Candogan, O., Menache, I., Ozdaglar, A., and Parrilo, P. A. Flows and Decompositions of Games: Harmonic and Potential Games. *Mathematics of Operations Research*, 36(3):474–503, 2011.

Chen, T., Zhang, W., Zhou, J., Chang, S., Liu, S., Amini, L., and Wang, Z. Training Stronger Baselines for Learning to Optimize. *NeurIPS*, 2020.

Chen, X., Deng, X., and Teng, S. Settling the complexity of computing two-player nash equilibria. *J. ACM*, 56(3): 14:1–14:57, 2009.

Cheung, Y. K. and Piliouras, G. Chaos, Extremism and Optimism: Volume Analysis of Learning in Games. *NeurIPS*, 2020.

Daskalakis, C. and Panageas, I. Last-Iterate Convergence: Zero-Sum Games and Constrained Min-Max Optimization. *10th Innovations in Theoretical Computer Science (ITCS) conference*, 2019.

Daskalakis, C., Goldberg, P. W., and Papadimitriou, C. H. The complexity of computing a nash equilibrium. *SIAM J. Comput.*, 39(1):195–259, 2009.

Daskalakis, C., Ilyas, A., Syrgkanis, V., and Zeng, H. Training GANs with Optimism. *ICLR*, 2018.

Gidel, G., Hemmat, R. A., Pezeshki, M., Lepriol, R., Huang, G., Lacoste-Julien, S., and Mitliagkas., I. Negative Momentum for Improved Game Dynamics. *AISTATS*, 2019.

Harsanyi, J. C. Games with Randomly Disturbed Payoffs: a New Rationale for Mixed-strategy Equilibrium points. *Internat. J. Game Theory*, 1973.

Hsieh, Y.-G., Iutzeler, F., Malick, J., and Mertikopoulos, P. On the convergence of single-call stochastic extra-gradient methods. *Advances in Neural Information Processing Systems 32:Annual Conference on Neural Information Processing Systems 2019*, 2019.

Hwang, S.-H. and Rey-Bellet, L. Strategic decompositions of normal form games: Zero-sum games and potential games. *Games and Economic Behavior*, 122:370–390, 2020. ISSN 0899-8256.

Jansen, M. J. M. Regularity and Stability of Equilibrium Points of Bimatrix Games. *Mathematics of Operations Research*, 1981.

Lanctot, M., Zambaldi, V., Gruslys, A., Lazaridou, A., Tuyls, K., Perolat, J., Silver, D., , and Graepel, T. A Unified Game Theoretic Approach to Multiagent Reinforcement Learning. *NeurIPS*, 2017.

Lei, Q., Nagarajan, S. G., Panageas, I., and Wang, X. Last iterate convergence in no-regret learning: constrained min-max optimization for convex-concave landscapes. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130, pp. 1441–1449, 2021.

Letcher, A., Balduzzi, D., Racaniere, S., Martens, J., Foerster, J., Tuyls, K., and Graepel, T. Differentiable Game Mechanics. *Journal of Machine Learning Research*, 2019.

Li, K. and Malik, J. Learning to Optimize. *ICLR*, 2017.

Liang, E., Liaw, R., Nishihara, R., Moritz, P., Fox, R., Goldberg, K., Gonzalez, J., Jordan, M., and Stoica, I. RLlib: Abstractions for distributed reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 3053–3062, 2018.

Lundberg, S. M. and Lee, S.-I. A unified approach to interpreting model predictions. *NeurIPS*, 2017.

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

ICLR '19: Proceedings of the 2019 International Conference on Learning Representations, 2019.

Mescheder, L., Nowozin, S., and Geiger, A. The Numerics of GANs. *NeurIPS*, 2017.

Norde, H. Bimatrix games have quasi-strict equilibria. *Mathematical Programming*, 1999.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal Policy Optimization Algorithms. *https://arxiv.org/abs/1707.06347*, 2017.

Shen, J., Chen, X., Heaton, H., Chen, T., Liu, J., Yin, W., and Wang, Z. Learning a Minimax Optimizer: A Pilot study. *ICLR*, 2021.

Wei, C.-Y., Lee, C.-W., Zhang, M., and Luo, H. Linear Last-iterate convergence in constrained saddle-point optimization. *ICLR*, 2021.

A. Background

Consensus optimization in the unconstrained case. In the unconstrained case where players' pure strategy sets are given by $\mathcal{X}_1 = \mathbb{R}^n$, $\mathcal{X}_2 = \mathbb{R}^m$, and their payoffs f_1, $f_2 : \mathcal{X}_1 \times \mathcal{X}_2 \to \mathbb{R}$, we define the game gradient and Jacobian as follows (Balduzzi et al., 2018):

$$v := \begin{pmatrix} \nabla_x f_1 \\ \nabla_y f_2 \end{pmatrix}, \qquad H := \begin{pmatrix} \nabla^2_x f_1 & \nabla^2_{x,y} f_1 \\ \nabla^2_{x,y} f_2 & \nabla^2_y f_2 \end{pmatrix}$$

The consensus optimization update rule takes the following form, for $x \in \mathcal{X}_1$, $y \in \mathcal{X}_2$ (Balduzzi et al., 2018; Mescheder et al., 2017):

$$\begin{pmatrix} x \\ y \end{pmatrix} \leftarrow \begin{pmatrix} x \\ y \end{pmatrix} + hv - h\epsilon H^T v$$

where h, ϵ are some parameters. In particular in the (unconstrained) bilinear case $f_1(x, y) = x^T Ay$, $f_2(x, y) = x^T By$ where A, B are some matrices of size $n \times m$, we get:

$$\begin{pmatrix} x \\ y \end{pmatrix} \leftarrow \begin{pmatrix} x \\ y \end{pmatrix} + h \begin{pmatrix} Ay \\ B^T x \end{pmatrix} - h\epsilon \begin{pmatrix} 0 & A \\ B^T & 0 \end{pmatrix}^T \begin{pmatrix} Ay \\ B^T x \end{pmatrix} = \begin{pmatrix} x + hAy - h\epsilon BB^T x \\ y + hB^T x - h\epsilon A^T Ay \end{pmatrix}$$

OMWU. For a constrained bimatrix game (A, B) associated to players payoffs $x^T Ay$, $x^T By$, where the mixed strategies in the probability simplex $x \in \Delta_n$, $y \in \Delta_m$, the OMWU update rule (Cheung & Piliouras, 2020; Daskalakis & Panageas, 2019) is given by, for $i \in [1, n]$, $j \in [1, m]$:

$$\begin{aligned} x_i^{t+1} &= \frac{x_i^t \exp(2h[Ay^t]_i - h[Ay^{t-1}]_i)}{\sum_{k=1}^n x_k^t \exp(2h[Ay^t]_k - h[Ay^{t-1}]_k)} \\ y_j^{t+1} &= \frac{y_j^t \exp(2h[B^T x^t]_j - h[B^T x^{t-1}]_j)}{\sum_{k=1}^m y_k^t \exp(2h[B^T x^t]_j - h[B^T x^{t-1}]_j)} . \end{aligned}$$

OMD. Using the same notations as OMWU, the OMD update is given by (Mertikopoulos et al., 2019):

$$\begin{aligned} \widehat{x}_i^t &= \frac{x_i^t \exp(h[Ay^t]_i)}{\sum_{k=1}^n x_k^t \exp(h[Ay^t]_k)}, \quad \widehat{y}_j^t = \frac{y_j^t \exp(h[B^T x^t]_j)}{\sum_{k=1}^m y_k^t \exp(h[B^T x^t]_j)} \\ x_i^{t+1} &= \frac{x_i^t \exp(h[A\widehat{y}^t]_i)}{\sum_{k=1}^n x_k^t \exp(h[A\widehat{y}^t]_k)}, \quad y_j^{t+1} = \frac{y_j^t \exp(h[B^T \widehat{x}^t]_j)}{\sum_{k=1}^m y_k^t \exp(h[B^T \widehat{x}^t]_j)} \end{aligned}$$

OGDA. Let $\widehat{x}^0 := x^0$, $\widehat{y}^0 := y^0$, and Π be the orthogonal projection onto the unit simplex with respect to the L_2 norm, i.e. $\Pi(x) = \arg\min_{y \in \Delta_n} ||x - y||_2$. Using the same notations as OMWU, the OGDA update is given by (Wei et al., 2021):

$$x^{t+1} = \Pi\left[\widehat{x}^t + hAy^t\right], \quad y^{t+1} = \Pi\left[\widehat{y}^t + hB^T x^t\right]$$

$$\widehat{x}^{t+1} = \Pi\left[\widehat{x}^t + hAy^{t+1}\right], \quad \widehat{y}^{t+1} = \Pi\left[\widehat{y}^t + hB^T x^{t+1}\right]$$

B. Proofs

Proposition B.1. *(Proposition 2.1 in (Daskalakis & Panageas, 2019)). Let $\phi : \Delta \to \Delta$, with $\Delta := \Delta_n \times \Delta_m$, and consider the recurrence relation $z^{t+1} = \phi(z^t)$. If the Jacobian of the update rule ϕ at a fixed point z^* (i.e. $\phi(z^*) = z^*$) has spectral radius strictly less than one, then there exists a neighborhood U around z^* such that for all $z \in U$, the dynamics starting from z converge to z^*, i.e. $\lim_{t\to\infty} \phi(z^t) = z^*$, $z^0 = z$.*

Lemma B.2. *Let S a real symmetric matrix for which 0 is an eigenvalue, and let z a unit L_2-norm eigenvector associated to it. Let λ_ϵ and $z + z_\epsilon$ the perturbed eigenpair (possibly complex) associated to $S + \epsilon M$, where M is a real matrix. Then we have, as $\epsilon \to 0$:*

$$\lambda_\epsilon = \epsilon z^T M z + o(\epsilon)$$

Proof. The result is known in matrix perturbation theory, we recap the proof here for convenience. We have by definition $(S + \epsilon M)(z + z_\epsilon) = \lambda_\epsilon(z + z_\epsilon)$, hence:

$$\epsilon M z + S z_\epsilon + \epsilon M z_\epsilon = \lambda_\epsilon z + \lambda_\epsilon z_\epsilon$$

We have $z_\epsilon = o(1)$ as $\epsilon \to 0$ by continuity of the eigenvectors and eigenvalues with respect to ϵ (Bhatia, 1997), hence $\epsilon M z_\epsilon = o(\epsilon)$. By symmetry of S we have $z^T S = (Sz)^T = 0$, so multiplying by z^T on the left yields:

$$\begin{aligned} \epsilon z^T M z &= \lambda_\epsilon + \lambda_\epsilon \langle z_\epsilon, z\rangle + o(\epsilon) \\ \Rightarrow \quad \frac{\lambda_\epsilon}{\epsilon} &= \frac{1}{1 + o(1)} z^T M z + o(1) \end{aligned}$$

Taking the limit yields $\lim_{\epsilon\to 0} \frac{\lambda_\epsilon}{\epsilon} = z^T M z$, hence $\lambda_\epsilon = \epsilon z^T M z + o(\epsilon)$. □

Theorem B.3. *(local convergence of CMWU) Let (x^*, y^*) a quasi-strict NE of the zero-sum game A. Let A_* the payoff matrix whose rows have been restricted to $\mathrm{supp}(x^*)$ and columns to $\mathrm{supp}(y^*)$. Assume that A_* (resp. A_*^T) is weakly $\mathcal{Z}_{|\,\mathrm{supp}(y^*)|}-$invertible (resp. $\mathcal{Z}_{|\,\mathrm{supp}(x^*)|}-$invertible). Then, for sufficiently small h and $\epsilon = Ch^{-\delta}$ satisfying* [12] *$\epsilon < |v_A^{-1}|$ for some $\delta \in [0, 1)$, $C > 0$ not depending on h, there exists a neighborhood $U(h) \subset \Delta_n \times \Delta_m$ of (x^*, y^*) such that for all $(x^0, y^0) \in U(h)$, the CMWU iterates (1) converge to the NE, namely* $\lim_{t\to\infty}(\varphi_1(x^t, y^t), \varphi_2(x^t, y^t)) = (x^*, y^*)$.

Proof. The proof aims at applying the well-know proposition B.1, as in the OMWU case (Daskalakis & Panageas, 2019). There are 2 key differences in the proof of CMWU vs. OMWU: (i) since the update rule is different, we need a different proof to ensure that any NE is a fixed point of the update rule: this is proved in proposition 1. (ii) regarding the eigenvalue analysis, OMWU attacks the antidiagonal part of the Jacobian. As stated in (Lei et al., 2021), which proves local convergence of OMWU in the convex-concave zero-sum setting: *"The key result of spectral analysis in (Daskalakis & Panageas, 2019) is in Lemma B.6 which states that an skew-symmetric matrix (i.e. antisymmetric $A^T = -A$) has imaginary eigenvalues.* In our CMWU case, we will attack the diagonal part of the Jacobian, which does not exploit skew-symmetry and thus necessitates a different proof technique. Specifically, we will see the problem as an eigenvalue perturbation problem, namely the perturbation of the zero eigenvalue of the "all ones" matrix J_n. The remainder of the proof aims at computing the Jacobian and analyzing its eigenvalues.

Step 1. By proposition 1, the NE is a fixed point of the update rule, it thus remains to show, according to proposition B.1, that the eigenvalues of the game Jacobian at equilibrium $H = H(x^*, y^*)$ have modulus strictly less then one, where H is defined as:

$$H := \begin{pmatrix} \frac{\partial^2 \varphi_1}{\partial x^2} & \frac{\partial^2 \varphi_1}{\partial x \partial y} \\ \frac{\partial^2 \varphi_2}{\partial x \partial y} & \frac{\partial^2 \varphi_2}{\partial y^2} \end{pmatrix}$$

Let us first compute H directly. We get, denoting δ_{ij} the Kroenecker delta ($\delta_{ij} = 1$ if $i = j$, $\delta_{ij} = 0$ otherwise):

$$\frac{\partial[\varphi_1]_i}{\partial x_j}(x^*, y^*) = \delta_{ij}\gamma_i + x_i^*\gamma_i \left[-h\epsilon H_{y^*}[i,j] - \gamma_j + h\epsilon \sum_{k=1}^n \gamma_k x_k^* H_{y^*}[k,j] \right], \quad i, j \in [1, n]$$

$$\begin{aligned} \frac{\partial[\varphi_1]_i}{\partial y_j}(x^*, y^*) =& x_i^*\gamma_i \left[hA_{ij} - h\epsilon A_{ij}[A^T x^*]_j \right. \\ & \left. + h\sum_{k=1}^n x_k^*\gamma_k(-A_{kj} + \epsilon A_{kj}[A^T x^*]_j) \right], \quad i \in [1, n], j \in [1, m] \end{aligned}$$

$$\gamma_i = \frac{\exp(h[Ay^*]_i - h\epsilon[H_{y^*}x^*]_i)}{\sum_{k=1}^n x_k^* \exp(h[Ay^*]_k - h\epsilon[H_{y^*}x^*]_k)}, \quad i \in [1, n]$$

[12] we use the convention $|v_A^{-1}| := +\infty$ if $v_A = 0$.

$$\frac{\partial[\varphi_2]_i}{\partial y_j}(x^*,y^*) = \delta_{ij}\eta_i + y_i^*\eta_i\left[-h\epsilon H_{x^*}[i,j] - \eta_j + h\epsilon\sum_{k=1}^{m}\eta_k y_k^* H_{x^*}[k,j]\right],\quad i,j\in[1,m]$$

$$\begin{aligned}\frac{\partial[\varphi_2]_i}{\partial x_j}(x^*,y^*) =& y_i^*\eta_i\,\Big[-hA_{ji} - h\epsilon A_{ji}[Ay^*]_j \\ &+h\sum_{k=1}^{n} y_k^*\eta_k(A_{jk}+\epsilon A_{jk}[Ay^*]_j)\Big],\quad i\in[1,m], j\in[1,n]\end{aligned}$$

$$\eta_i = \frac{\exp(-h[A^Tx^*]_i - h\epsilon[H_{x^*}y^*]_i)}{\sum_{k=1}^{m} y_k^*\exp(-h[A^Tx^*]_k - h\epsilon[H_{x^*}y^*]_k)},\quad i\in[1,m]$$

Step 2. Eigenspaces are invariant with respect to Jacobian row and column permutations, so we are free to rearrange H as desired. As in (Daskalakis & Panageas, 2019), we first group together equilibrium coordinates outside of the support, i.e. the indexes i and j such that $x_i^* = 0$ or $y_j^* = 0$. We denote $\mathrm{supp}(z)$ the support of a vector z, namely the indexes corresponding to its nonzero coordinates. Take $i \notin \mathrm{supp}(x^*)$, $j \notin \mathrm{supp}(y^*)$. In this case, using the partial derivatives computed in step 1, all partial derivatives are zero except the diagonal terms:

$$\frac{\partial[\varphi_1]_i}{\partial x_i}(x^*,y^*) = \gamma_i \qquad \frac{\partial[\varphi_2]_j}{\partial y_j}(x^*,y^*) = \eta_j$$

We use the NE property that $x_k^* \neq 0 \Rightarrow [Ay^*]_k = v_A$, and $y_k^* \neq 0 \Rightarrow [A^Tx^*]_k = v_A$ (Daskalakis & Panageas, 2019). This yields, since the sum in the denominator below is equal to the same sum restricted to $x_k^* \neq 0$ (as we multiply by x_k^*):

$$\gamma_i = \frac{\exp(h[Ay^*]_i - h\epsilon[H_{y^*}x^*]_i)}{\sum_{k=1}^{n} x_k^*\exp(hv_A - h\epsilon v_A^2)} = \exp(h[Ay^*]_i - hv_A + h\epsilon v_A^2 - h\epsilon[H_{y^*}x^*]_i)$$

where we have used that $[H_{y^*}x^*]_k = v_A^2$ when $x_k^* \neq 0$, cf. proof of proposition 1. Proof of proposition 1 also yields $[H_{y^*}x^*]_i = [Ay^*]_i v_A$ (remember $x_i^* = 0$). This yields:

$$\gamma_i = \exp(h[Ay^*]_i - hv_A + h\epsilon v_A^2 - v_A h\epsilon[Ay^*]_i) = \exp(-h(v_A - [Ay^*]_i)(1-\epsilon v_A))$$

and therefore $|\gamma_i| < 1$ when $(v_A - [Ay^*]_i)(1-\epsilon v_A) > 0$. Since the NE (x^*,y^*) is quasi-strict by assumption, $v_A - [Ay^*]_i > 0$, hence the condition is true when $1-\epsilon v_A > 0$, i.e. $\epsilon v_A < 1$. Similarly, we get:

$$\eta_j = \exp(-h([A^Tx^*]_j - v_A)(1+\epsilon v_A))$$

By quasi-strict property of the NE, we have $[A^Tx^*]_j - v_A > 0$, and so $|\eta_j| < 1$ when $1+\epsilon v_A > 0$, i.e. $\epsilon v_A > -1$. Both requirements $\epsilon v_A < 1$, $\epsilon v_A > -1$ yield $|\epsilon v_A| < 1$, which is true by assumption since we have assumed $0 < \epsilon < |v_A^{-1}|$ (with the convention $|v_A^{-1}| := +\infty$ if $v_A = 0$).

Step 3. We now consider the remaining coordinates, i.e. indexes i and j such that both $x_i^* \neq 0$ and $y_j^* \neq 0$. The partial derivatives of step 1 can be simplified: we use again the NE property that $x_i^* \neq 0 \Rightarrow [Ay^*]_i = v_A$, and $y_j^* \neq 0 \Rightarrow [A^Tx^*]_j = v_A$, which yield that all γ and η terms in these equations are equal to 1. Specifically for $i,j \in \mathrm{supp}(x^*)$:

$$\begin{aligned}\frac{\partial[\varphi_1]_i}{\partial x_j}(x^*,y^*) &= \delta_{ij} + x_i^*\left[-h\epsilon H_{y^*}[i,j] - 1 + h\epsilon\sum_{k=1}^{n} x_k^* H_{y^*}[k,j]\right] \\ &= \delta_{ij} + x_i^*\left[-h\epsilon H_{y^*}[i,j] - 1 + h\epsilon v_A^2\right]\end{aligned}$$

for $i \in \mathrm{supp}(x^*)$, $j \in \mathrm{supp}(y^*)$:

$$\begin{aligned}\frac{\partial[\varphi_1]_i}{\partial y_j}(x^*,y^*) &= x_i^*\left[hA_{ij} - h\epsilon A_{ij}v_A + h\sum_{k=1}^{n} x_k^*(-A_{kj} + \epsilon A_{kj}v_A)\right] \\ &= x_i^*\left[hA_{ij} - h\epsilon A_{ij}v_A - hv_A + h\epsilon v_A^2\right]\end{aligned}$$

for $i, j \in \operatorname{supp}(y^*)$

$$\begin{aligned}\frac{\partial[\varphi_2]_i}{\partial y_j}(x^*, y^*) &= \delta_{ij} + y_i^* \left[-h\epsilon H_{x^*}[i,j] - 1 + h\epsilon \sum_{k=1}^{m} y_k^* H_{x^*}[k,j] \right] \\ &= \delta_{ij} + y_i^* \left[-h\epsilon H_{x^*}[i,j] - 1 + h\epsilon v_A^2 \right]\end{aligned}$$

for $i \in \operatorname{supp}(y^*)$, $j \in \operatorname{supp}(x^*)$:

$$\begin{aligned}\frac{\partial[\varphi_2]_i}{\partial x_j}(x^*, y^*) =& y_i^* \left[-hA_{ji} - h\epsilon A_{ji} v_A + h \sum_{k=1}^{n} y_k^* (A_{jk} + \epsilon A_{jk} v_A) \right] \\ =& \, y_i^* \left[-hA_{ji} - h\epsilon A_{ji} v_A + h v_A + h\epsilon v_A^2 \right]\end{aligned}$$

The above expressions can be written in compact matrix forms. In the following, in order to avoid notational burden we will use the same notations for the original matrices and their support restricted counterparts, but in the expressions below, the upper-left, upper-right, lower-left and lower-right matrix blocks have sizes respectively: $|\operatorname{supp}(x^*)|^2$, $|\operatorname{supp}(x^*)| \times |\operatorname{supp}(y^*)|$, $|\operatorname{supp}(y^*)| \times |\operatorname{supp}(x^*)|$, $|\operatorname{supp}(y^*)|^2$. Denoting J as the all ones matrix, we get:

$$H = I - D\left[D_J + h\epsilon S - h\epsilon v_A^2 \widetilde{J} + hB\right], \quad \widetilde{J} := \begin{pmatrix} J & J \\ J & J \end{pmatrix}, \quad D := \begin{pmatrix} D_{x^*} & 0 \\ 0 & D_{y^*} \end{pmatrix}$$

$$S := \begin{pmatrix} H_{y^*} & v_A A_* \\ v_A A_*^T & H_{x^*} \end{pmatrix}, \quad D_J := \begin{pmatrix} J & 0 \\ 0 & J \end{pmatrix}, \quad B := \begin{pmatrix} 0 & -A_* + v_A J \\ A_*^T - v_A J & 0 \end{pmatrix}$$

Step 4. Remember that D has strictly positive diagonal elements due to the support restriction: D_{x^*}, D_{y^*} are of sizes $|\operatorname{supp}(x^*)|^2$, $|\operatorname{supp}(y^*)|^2$. Since the sum of x^* and y^* are both 1, eigenvalues of DD_J are 1 (multiplicity 2) and 0 (multiplicity $|\operatorname{supp}(x^*)| + |\operatorname{supp}(y^*)| - 2$). When $h = 0$, the zero eigenvalue of DD_J is causing H to have eigenvalue 1, so for small h, our task is to show that the perturbed eigenvalue λ_h away from zero has strictly positive real part, which will make the modulus of the corresponding eigenvalue of H strictly less than one (see corollary 9 in (Letcher et al., 2019)). Observe that the zero eigenvalue of DD_J has same eigenvectors as the zero eigenvalue of D_J since D is invertible. D_J is symmetric and bloc diagonal with J on the diagonal, so these eigenvectors are orthogonal and of the form $z := (z_1, z_2)$ of size $|\operatorname{supp}(x^*)| \times |\operatorname{supp}(y^*)|$ with one of z_1, z_2 an eigenvector associated to the zero eigenvalue of J, and the other 0. Note that the eigenspace $\mathcal{Z}_n$ associated to the zero eigenvalue of J_n is of dimension $n-1$ and consists of vectors whose sum is zero.

Step 5. Let λ_h a perturbation of the 0 eigenvalue of D_J for small h, associated to the perturbation $D_J + h\epsilon M + hB$, with $M := S - v_A^2 \widetilde{J}$. We use lemma B.2 (since D_J is symmetric) to get that $\lambda_h = h\epsilon z^T M z + h z^T B z + o(h\epsilon) + o(h)$. By assumption, $\epsilon = Ch^{-\delta}$ with $\delta \in [0, 1)$, so the term $o(h)$ is $o(h\epsilon)$ and $h\epsilon = o(1)$, hence $\lambda_h = h\epsilon z^T M z + h z^T B z + o(h\epsilon)$. Using the fact that one of z_1, z_2 is an eigenvector of J, and the other 0, we get $z^T \widetilde{J} z = 0$. Since B has zero diagonal blocs, its antidiagonal blocs will be multiplied by z_1 on one side and z_2 on the other side, which yield $z^T B z = 0$ since either $z_1 = 0$ or $z_2 = 0$. Finally we get $z^T S z = z_1^T H_{y^*} z_1 + z_2^T H_{x^*} z_2$, since its antidiagonal blocs will be multiplied by 0 for the same reason as for B.

Step 6. We have $z_1^T H_{y^*} z_1 = z_1^T A_* D_{y^*}^{\frac{1}{2}} D_{y^*}^{\frac{1}{2}} A_*^T z_1 = ||D_{y^*}^{\frac{1}{2}} A_*^T z_1||^2$. Notice that $||D_{y^*}^{\frac{1}{2}} A_*^T z_1|| = 0$ if and only if $A_*^T z_1 = 0$ as D_{y^*} is of size $|\operatorname{supp}(y^*)|^2$ has been restricted to rows and columns in the support of y^*. Similarly $||D_{x^*}^{\frac{1}{2}} A_* z_2|| = 0$ if and only if $A_* z_2 = 0$. By assumption, A_*^T is weakly $\mathcal{Z}_{|\operatorname{supp}(x^*)|}$−invertible, and A_* is weakly $\mathcal{Z}_{|\operatorname{supp}(y^*)|}$−invertible, hence we always have either $A_* z_2 \neq 0$ or $A_*^T z_1 \neq 0$ since one of z_1, z_2 is zero and the other one is in $\mathcal{Z}$ and non zero as an eigenvector associated to the zero eigenvalue of J. So overall $\lambda_h = h\epsilon\lambda + o(h\epsilon)$ for some $\lambda > 0$. Scaling on the left the matrix D_J by D doesn't change the sign of λ since we are working on the support of x^*, y^*. This concludes the proof that for h sufficiently small, the perturbed 0 eigenvalue of DD_J will have strictly positive real part.

□

Lemma B.4. *If p_1, p_2 are commutative projectors on $\mathcal{G}$, then $p_1 p_2$ is a projector and:*

$$\mathcal{G} = (K_{p_1} \cap K_{p_2}) \oplus (K_{p_1} \cap R_{p_2}) \oplus (R_{p_1} \cap K_{p_2}) \oplus (R_{p_1} \cap R_{p_2}).$$

Proof. The result is well-known in linear algebra. We recap the proof for convenience. **Step 1.** Let $u : \mathcal{G} \to \mathcal{G}$ and q a projector. Then u and q commute if and only if K_q and R_q are stable by u, i.e. $uK_q \subseteq K_q$, $uR_q \subseteq R_q$. Indeed, assume u and q commute and $x \in K_q$, $y \in R_q$. Then, $qux = uqx = 0$, hence $ux \in K_q$. By definition 5, $K_q = R_{\mathrm{Id}-q}$, but $\mathrm{Id}-q$ is a projector since it is linear and $(\mathrm{Id}-q)(\mathrm{Id}-q) = \mathrm{Id}-2q+q^2 = \mathrm{Id}-q$. So, $R_q = K_{\mathrm{Id}-q}$, and $(\mathrm{Id}-q)uy = u(\mathrm{Id}-q)y = 0$ since $y \in R_q = K_{\mathrm{Id}-q}$. Now, assume the reverse. Let $z \in \mathcal{G}$. We want to show $uqz = quz$. By definition 5, $z = x + y$ with $x \in K_q$, $y \in R_q$, so $uqz = uqy$ and $quz = qux + quy$. By assumption $ux \in K_q$, so $qux = 0$. So it remains to show $uqy = quy$. $y \in R_q$, so $y = qx_0$ for some $x_0 \in \mathcal{G}$, so $uqy = uqx_0$. Since $uy \in R_q$ by assumption, $quy = uy = uqx_0$, which shows $uqy = quy$.

Step 2. Let $\mathcal{K}$ a subspace of $\mathcal{G}$ and q a projector. We show that $\mathcal{K}$ is stable by q if and only if $\mathcal{K} = (\mathcal{K} \cap K_q) \oplus (\mathcal{K} \cap R_q)$. Indeed, assume $\mathcal{K} = (\mathcal{K} \cap K_q) \oplus (\mathcal{K} \cap R_q)$ and $z \in \mathcal{K}$. We want to show $qz \in \mathcal{K}$. By assumption $z = x + y$, with $x \in \mathcal{K} \cap K_q$, $y \in \mathcal{K} \cap R_q$. So $qz = qx + qy = qy = y \in \mathcal{K}$. Now assume the reverse, i.e. $\mathcal{K}$ is stable by q, and let $H := (\mathcal{K} \cap K_q) + (\mathcal{K} \cap R_q)$. Since q projector, by definition 5 and property of the direct sum we have $K_q \cap R_q = \{0\}$, so the sum is a direct sum: $H = (\mathcal{K} \cap K_q) \oplus (\mathcal{K} \cap R_q)$. The inclusion $H \subseteq \mathcal{K}$ is trivial, so it remains to show $\mathcal{K} \subseteq H$. Let $z \in \mathcal{K}$. Then $z = x + y$ with $x \in K_q$, $y \in R_q$, so $qz = qy = y$. Since $qz \in \mathcal{K}$ by assumption, then $y \in \mathcal{K}$, so $y \in \mathcal{K} \cap R_q$. But $x = z - y$, so also $x \in \mathcal{K}$, and so $x \in \mathcal{K} \cap K_q$, which proves that $z \in H$.

Step 3. We have $\mathcal{G} = K_{p_1} \oplus R_{p_1}$ by definition 5. By step 1, K_{p_1} and R_{p_1} are stable by p_2. By step 2, $K_{p_1} = (K_{p_1} \cap K_{p_2}) \oplus (K_{p_1} \cap R_{p_2})$ and $R_{p_1} = (R_{p_1} \cap K_{p_2}) \oplus (R_{p_1} \cap R_{p_2})$, which completes the proof. □

Theorem B.5. *(Game Decomposition) Let $(\rho_i)_{i\in[1,n]}$ a family of n commutative projectors on $\mathcal{G}$. Then we have the canonical direct sum decomposition:*

$$\mathcal{G} = \bigoplus_{\mathcal{C}_i \in \{K_{\rho_i}, R_{\rho_i}\}} \bigcap_{i=1}^{n} \mathcal{C}_i \ .$$

In particular, in the case $n = 3$ we have:

$$\begin{aligned}\mathcal{G} = &(K_{\rho_1} \cap K_{\rho_2} \cap K_{\rho_3}) \oplus (K_{\rho_1} \cap K_{\rho_2} \cap R_{\rho_3}) \oplus (K_{\rho_1} \cap R_{\rho_2} \cap K_{\rho_3}) \oplus (K_{\rho_1} \cap R_{\rho_2} \cap R_{\rho_3}) \\ &\oplus (R_{\rho_1} \cap K_{\rho_2} \cap K_{\rho_3}) \oplus (R_{\rho_1} \cap K_{\rho_2} \cap R_{\rho_3}) \oplus (R_{\rho_1} \cap R_{\rho_2} \cap K_{\rho_3}) \oplus (R_{\rho_1} \cap R_{\rho_2} \cap R_{\rho_3}) \ .\end{aligned}$$

Proof. Let us prove the result by induction. The result is true if $n = 1$ by definition 5: if p_1 is any projector, then $\mathcal{G} = K_{p_1} \oplus R_{p_1}$. The result is also true for $n = 2$ by lemma B.4. Now, assume the result is true at induction stage n, and consider a family of $n + 1$ commutative projectors $(\rho_i)_{i\in[1,n+1]}$. By the induction hypothesis applied to the family $(\rho_i)_{i\in[1,n]}$, we have:

$$\mathcal{G} = \bigoplus_{\mathcal{C}_i \in \{K_{\rho_i}, R_{\rho_i}\}} \bigcap_{i=1}^{n} \mathcal{C}_i$$

The goal is now to split each one of the subspaces $\mathcal{C}_i$ into 2 as follows: $\mathcal{C}_i = (\mathcal{C}_i \cap K_{\rho_{n+1}}) \oplus (\mathcal{C}_i \cap R_{\rho_{n+1}})$. If the latter is true, then the induction is true at stage $n+1$. By the step 2 of the proof of lemma B.4, we have $\mathcal{C}_i = (\mathcal{C}_i \cap K_{\rho_{n+1}}) \oplus (\mathcal{C}_i \cap R_{\rho_{n+1}})$ provided $\mathcal{C}_i$ is stable by ρ_{n+1}, i.e. $\rho_{n+1}\mathcal{C}_i \subseteq \mathcal{C}_i$. We now conclude the proof using a similar argument as step 1 of the proof of lemma B.4, since by assumption ρ_{n+1} commutes with all $(\rho_i)_{i\in[1,n]}$. Indeed, by definition of $\mathcal{C}_i$, it is an intersection of the following form, for some integers n_1, α_k, β_k:

$$\mathcal{C}_i = \cap_{k=1}^{n_1} K_{\rho_{\alpha_k}} \cap \cap_{k=1}^{n-n_1} R_{\rho_{\beta k}}$$

Take $z \in \mathcal{C}_i$. The goal is to show that $\rho_{n+1} z \in \mathcal{C}_i$. Since ρ_{n+1} commutes with every ρ_{α_k}, we have $\rho_{\alpha_k}\rho_{n+1} z = \rho_{n+1}\rho_{\alpha_k} z = 0$ since $z \in K_{\rho_{\alpha_k}}$, which shows that $\rho_{n+1} z \in \cap_{k=1}^{n_1} K_{\rho_{\alpha_k}}$. Similarly, since $R_{\rho_{\beta k}} = K_{\mathrm{Id}-\rho_{\beta k}}$, we have $(\mathrm{Id}-\rho_{\beta_k})\rho_{n+1} z = \rho_{n+1}(\mathrm{Id}-\rho_{\beta_k}) z = 0$ since $z \in R_{\rho_{\beta k}}$, which shows that $\rho_{n+1} z \in \cap_{k=1}^{n-n_1} R_{\rho_{\beta_k}}$, which concludes the proof. □

Proposition B.6. *(generalization of transitivity/cyclicity and unification with triviality) If a 2P game f is zero-sum symmetric transitive (resp. cyclic) in the sense of (Balduzzi et al., 2019), then $f \in R_{\rho_T}$ (resp. $f \in K_{\rho_T} = R_{\mathrm{Id}-\rho_T}$). Further, a 2P game is trivial in the sense of (Cheung & Piliouras, 2020) if and only if it is transitive, i.e. the class of trivial games coincides with R_{ρ_T}.*

Proof. Remember the following definitions. $\mathcal{N}_2 f(x,y) := \int f(x,z)d\mu_2(z), \mathcal{N}_1 f(x,y) := \int f(z,y)d\mu_1(z)$, and:

$$\rho_T : (f_1, f_2) \to (\widehat{\mathcal{N}} f_1, \widehat{\mathcal{N}} f_2), \;\; \widehat{\mathcal{N}} := \mathcal{N}_1 + \mathcal{N}_2 - \mathcal{N}_1\mathcal{N}_2 \,.$$

First, if a 2P game $f = (f_1, f_2)$ is zero-sum symmetric transitive in the sense of (Balduzzi et al., 2019), then $\mathcal{X}_1 = \mathcal{X}_2$, $\mu_1 = \mu_2 =: \mu$, $f_2 = -f_1$ and $f_1(x,y) = \phi(x) - \phi(y)$ for some function ϕ. We show that $\rho_T f = f$, namely that $f \in R_{\rho_T}$. For this, it suffices to show $\widehat{\mathcal{N}} f_1 = f_1$. Indeed, $f_2 = -f_1$ so $\widehat{\mathcal{N}} f_1 = f_1$ yields $\widehat{\mathcal{N}} f_2 = f_2$ by linearity. We have:

$$\begin{aligned}
\widehat{\mathcal{N}} f_1(x,y) &= \int f_1(x,z)d\mu(z) + \int f_1(z,y)d\mu(z) - \int\int f_1(z_1,z_2)d\mu(z_1)d\mu(z_2) \\
&= \phi(x) - \int \phi(z)d\mu(z) + \int \phi(z)d\mu(z) - \phi(y) - \left[\int \phi(z)d\mu(z) - \int \phi(z)d\mu(z)\right] \\
&= \phi(x) - \phi(y) = f_1(x,y)
\end{aligned}$$

If the game is zero-sum symmetric cyclic in the sense of (Balduzzi et al., 2019), then $f_2 = -f_1$, $f_1 = -f_1^T$, and $\int f_1(x,z)d\mu(z) = 0 \; \forall x \in \mathcal{X}_1$, so we have:

$$\begin{aligned}
\widehat{\mathcal{N}} f_1(x,y) &= \int f_1(x,z)d\mu(z) + \int f_1(z,y)d\mu(z) - \int\int f_1(z_1,z_2)d\mu(z_1)d\mu(z_2) \\
&= 0 - \int f_1(y,z)d\mu(z) + \int \left(\int f_1(z_2,z_1)d\mu(z_1)\right) d\mu(z_2) \\
&= 0 - 0 + \int 0 \; d\mu(z_2) = 0
\end{aligned}$$

hence indeed $f \in K_{\rho_T}$ ($f_2 = -f_1$ so $\widehat{\mathcal{N}} f_2 = -\widehat{\mathcal{N}} f_1 = 0$ by linearity).

Now, we show that (f_1, f_2) is trivial in the sense of (Cheung & Piliouras, 2020) if and only if it is transitive. A trivial game is of the form $f_1(x,y) = \alpha_1(x) + \beta_1(y)$, $f_2(x,y) = \alpha_2(x) + \beta_2(y)$ for some functions α_i, β_i. We have:

$$\widehat{\mathcal{N}} f_k(x,y) = \int f_k(x,z)d\mu_2(z) + \int f_k(z,y)d\mu_1(z) - \int\int f_k(z_1,z_2)d\mu_1(z_1)d\mu_2(z_2)$$

Therefore clearly, if a game belongs to the range R_{ρ_T} of ρ_T, it is trivial since $\widehat{\mathcal{N}} f_k$ is of the form $\alpha_k(x) + \beta_k(y)$ for $\alpha_k(x) = \int f_k(x,z)d\mu_2(z)$, $\beta_k(y) = \int f_k(z,y)d\mu_1(z) - \int\int f_k(z_1,z_2)d\mu_1(z_1)d\mu_2(z_2)$. It remains to show the opposite, i.e. that if a game is trivial, it belongs to R_{ρ_T}, i.e. that $\widehat{\mathcal{N}} f_1 = f_1$ and $\widehat{\mathcal{N}} f_2 = f_2$. We have:

$$\begin{aligned}
\widehat{\mathcal{N}} f_k(x,y) &= \int f_k(x,z)d\mu_2(z) + \int f_k(z,y)d\mu_1(z) - \int\int f_k(z_1,z_2)d\mu_1(z_1)d\mu_2(z_2) \\
&= \alpha_k(x) + \int \beta_k(z)d\mu_2(z) + \beta_k(y) + \int \alpha_k(z)d\mu_1(z) - \left[\int \alpha_k(z)d\mu_1(z) + \int \beta_k(z)d\mu_2(z)\right] \\
&= \alpha_k(x) + \beta_k(y) = f_k(x,y)
\end{aligned}$$

This concludes the proof. □

Corollary B.7. *(Canonical decomposition of 2P games) ρ_Z, ρ_T are commutative projectors. Consequently, by theorem B.5, every 2P game can be decomposed into the direct sum of 4 components: (ZT) zero-sum-transitive, (ZCy) zero-sum-cyclic, (CT) cooperative-transitive, (CCy) cooperative-cyclic. Further, if $\mathcal{X}_1 = \mathcal{X}_2$, then ρ_S is a well-defined projector and commutes with ρ_Z, ρ_T. Consequently, any such game can be decomposed into the direct sum of 8 components: (ZST) zero-sum-symmetric-transitive, (ZSCy) zero-sum-symmetric-cyclic, (ZAT) zero-sum-antisymmetric-transitive, (ZACy) zero-sum-antisymmetric-cyclic, (CST) cooperative-symmetric-transitive, (CSCy) cooperative-symmetric-cyclic, (CAT) cooperative-antisymmetric-transitive, (CACy) cooperative-antisymmetric-cyclic.*

Proof. Remember that we have the following definitions for the operators corresponding to zero-sum, symmetric and transitive games:

$$\begin{aligned}
&\rho_Z : (f_1, f_2) \to \frac{1}{2}(f_1 - f_2, f_2 - f_1), \;\; \rho_S : (f_1, f_2) \to \frac{1}{2}(f_1 + f_2^T, f_2 + f_1^T) \\
&\rho_T : (f_1, f_2) \to (\widehat{\mathcal{N}} f_1, \widehat{\mathcal{N}} f_2), \;\; \widehat{\mathcal{N}} := \mathcal{N}_1 + \mathcal{N}_2 - \mathcal{N}_1\mathcal{N}_2 \,.
\end{aligned}$$

where $\mathcal{N}_2 f(x,y) := \int f(x,z)d\mu_2(z)$, $\mathcal{N}_1 f(x,y) := \int f(z,y)d\mu_1(z)$. For convenience we write below the complementary (in the sense of direct sum) operators $\operatorname{Id} - \rho_Z$, $\operatorname{Id} - \rho_S$, $\operatorname{Id} - \rho_T$ corresponding respectively to cooperative, antisymmetric and cyclic games:

$$\begin{aligned}
&\operatorname{Id} - \rho_Z : (f_1, f_2) \to \frac{1}{2}(f_1 + f_2, f_1 + f_2), \quad \operatorname{Id} - \rho_S : (f_1, f_2) \to \frac{1}{2}(f_1 - f_2^T, f_2 - f_1^T) \\
&\operatorname{Id} - \rho_T : (f_1, f_2) \to (f_1 - \widehat{\mathcal{N}} f_1, f_2 - \widehat{\mathcal{N}} f_2)
\end{aligned}$$

We need to check that ρ_Z, ρ_T, ρ_S are commutative projectors. The fact that ρ_Z, ρ_T are commutative projectors has been observed in (Hwang & Rey-Bellet, 2020). First let's check that they are projectors. They are indeed linear, so it remains to show that they verify $\rho^2 = \rho$. We do so by direct computation:

$$\begin{aligned}
&\rho_Z^2(f_1, f_2) = \frac{1}{2}\rho_Z(f_1 - f_2, f_2 - f_1) = \frac{1}{4}(2f_1 - 2f_2, 2f_2 - 2f_1) \\
&= \frac{1}{2}(f_1 - f_2, f_2 - f_1) = \rho_Z(f_1, f_2) \\
&\rho_S^2(f_1, f_2) = \frac{1}{2}\rho_S(f_1 + f_2^T, f_2 + f_1^T) = \frac{1}{4}(f_1 + (f_1^T)^T + 2f_2^T, f_2 + (f_2^T)^T + 2f_1^T) \\
&= \frac{1}{2}(f_1 + f_2^T, f_2 + f_1^T) = \rho_S(f_1, f_2)
\end{aligned}$$

For ρ_T, we have $\rho_T^2(f_1, f_2) = (\widehat{\mathcal{N}}^2 f_1, \widehat{\mathcal{N}}^2 f_2)$. We show that $\widehat{\mathcal{N}}^2 f_k = \widehat{\mathcal{N}} f_k$. Denote $\gamma_k := \int\int f_k(z_1, z_2)d\mu_1(z_1)d\mu_2(z_2)$. We have:

$$\begin{aligned}
&\widehat{\mathcal{N}}^2 f_k(x,y) = \int f_k(x, z_2)d\mu_2(z_2) + \gamma_k - \gamma_k + \gamma_k + \int f_k(z_1, y)d\mu_1(z_1) - \gamma_k - (\gamma_k + \gamma_k - \gamma_k) \\
&= \int f_k(x, z_2)d\mu_2(z_2) + \int f_k(z_1, y)d\mu_1(z_1) - \gamma_k = \widehat{\mathcal{N}} f_k(x,y)
\end{aligned}$$

Finally, we check commutativity. We have:

$$\begin{aligned}
\rho_Z \rho_S(f_1, f_2) = \frac{1}{2}\rho_Z(f_1 + f_2^T, f_2 + f_1^T) = \frac{1}{4}(f_1 + f_2^T - f_2 - f_1^T, -f_1 - f_2^T + f_2 + f_1^T) \\
\rho_S \rho_Z(f_1, f_2) = \frac{1}{2}\rho_S(f_1 - f_2, f_2 - f_1) = \frac{1}{4}(f_1 + f_2^T - f_2 - f_1^T, -f_1 - f_2^T + f_2 + f_1^T)
\end{aligned}$$

Hence indeed ρ_Z and ρ_S commute. Then, using linearity of $\widehat{\mathcal{N}}$:

$$\rho_T \rho_Z(f_1, f_2) = \frac{1}{2}(\widehat{\mathcal{N}} f_1 - \widehat{\mathcal{N}} f_2, \widehat{\mathcal{N}} f_2 - \widehat{\mathcal{N}} f_1)$$

$$\rho_Z \rho_T(f_1, f_2) = \rho_Z(\widehat{\mathcal{N}} f_1, \widehat{\mathcal{N}} f_2) = \frac{1}{2}(\widehat{\mathcal{N}} f_1 - \widehat{\mathcal{N}} f_2, \widehat{\mathcal{N}} f_2 - \widehat{\mathcal{N}} f_1)$$

hence indeed ρ_Z and ρ_T commute. Finally:

$$\rho_T \rho_S(f_1, f_2) = \frac{1}{2}(\widehat{\mathcal{N}} f_1 + \widehat{\mathcal{N}} f_2^T, \widehat{\mathcal{N}} f_2 + \widehat{\mathcal{N}} f_1^T)$$

$$\rho_S \rho_T(f_1, f_2) = \rho_S(\widehat{\mathcal{N}} f_1, \widehat{\mathcal{N}} f_2) = \frac{1}{2}(\widehat{\mathcal{N}} f_1 + (\widehat{\mathcal{N}} f_2)^T, \widehat{\mathcal{N}} f_2 + (\widehat{\mathcal{N}} f_1)^T)$$

So it remains to show $(\widehat{\mathcal{N}} f_k)^T = \widehat{\mathcal{N}} f_k^T$. Remember that ρ_T is well-defined only when $\mathcal{X}_1 = \mathcal{X}_2 =: \mathcal{X}$, $\mu_1 = \mu_2 =: \mu$. In that case we have:

$$\begin{aligned}
\widehat{\mathcal{N}} f_k^T(x,y) &= \int f_k^T(x,z)d\mu(z) + \int f_k^T(z,y)d\mu(z) - \int\int f_k^T(z_1, z_2)d\mu(z_1)d\mu(z_2) \\
&= \int f_k(z,x)d\mu(z) + \int f_k(y,z)d\mu(z) - \int\int f_k(z_2, z_1)d\mu(z_1)d\mu(z_2) \\
&= \widehat{\mathcal{N}} f_k(y,x) = (\widehat{\mathcal{N}} f_k)^T(x,y)
\end{aligned}$$

This concludes the proof. □

C. Experiment details and additional findings

C.1. Zero-sum bimatrix games

We report in figure 6 the average performance and standard deviation of β_τ ($\tau = 500$) of all algorithms on 200 randomly sampled bimatrix zero-sum games in dimension $n = m = 25, 50, 75, 100$, for a wide range of learning rates h. We see that CMWU performs well across a wide band of learning rates.

C.2. Learning Games

Terminology. We use the terminology of corollary 2.1, namely **(ZT)** zero-sum-transitive, **(ZCy)** zero-sum-cyclic, **(CT)** cooperative-transitive, **(CCy)** cooperative-cyclic, **(ZST)** zero-sum-symmetric-transitive, **(ZSCy)** zero-sum-symmetric-cyclic, **(ZAT)** zero-sum-antisymmetric-transitive, **(ZACy)** zero-sum-antisymmetric-cyclic, **(CST)** cooperative-symmetric-transitive, **(CSCy)** cooperative-symmetric-cyclic, **(CAT)** cooperative-antisymmetric-transitive, **(CACy)** cooperative-antisymmetric-cyclic.

Computation of game components. Let (A, B) a bimatrix game. The operators ρ_Z, ρ_S, ρ_T of section 3 act as defined below in the bimatrix case (note that as mentioned in corollary 2.1, ρ_S is well defined only if $n = m$). As mentioned in the main text, our concepts of cyclic and transitive game (materialized by the operators $\mathrm{Id} - \rho_T$ and ρ_T) generalize that of (Balduzzi et al., 2019) which considered the zero-sum symmetric case: in the case where the game (A, B) is zero-sum symmetric, then $B = -A$, $A = -A^T$, and ρ_T, $\mathrm{Id} - \rho_T$ reduce to (Balduzzi et al., 2019), whereas in general they yield different expressions. The zero-sum symmetric transitive and cyclic components of the general game (A, B) are simply given by $\rho_Z \rho_S \rho_T(A, B)$ and $\rho_Z \rho_S(\mathrm{Id} - \rho_T)(A, B)$. This way, we can compute all 8 components of corollary 2.1 simply by applying these operators one after the other, where the order doesn't change as they commute.

$$\rho_Z(A, B) = \frac{1}{2}(A - B, B - A), \quad \rho_C(A, B) := [\mathrm{Id} - \rho_Z](A, B) = \frac{1}{2}(A + B, A + B)$$

$$\rho_S(A, B) = \frac{1}{2}(A + B^T, B + A^T), \quad \rho_A(A, B) := [\mathrm{Id} - \rho_S](A, B) = \frac{1}{2}(A - B^T, B - A^T)$$

$$\rho_T(A, B) = (\widehat{A}^{(1)} + \widehat{A}^{(2)} - \widehat{A}, \widehat{B}^{(1)} + \widehat{B}^{(2)} - \widehat{B}),$$

$$\rho_{Cy}(A, B) := [\mathrm{Id} - \rho_T](A, B) = (A - \widehat{A}^{(1)} - \widehat{A}^{(2)} + \widehat{A}, B - \widehat{B}^{(1)} - \widehat{B}^{(2)} + \widehat{B})$$

where $\widehat{A}^{(1)}$, $\widehat{A}^{(2)}$, $\widehat{A}$ are the matrices with entries $\widehat{A}^{(1)}_{ij} := \frac{1}{m}\sum_{j=1}^m A_{ij}$, $\widehat{A}^{(2)}_{ij} := \frac{1}{n}\sum_{i=1}^n A_{ij}$, $\widehat{A}_{ij} := \frac{1}{mn}\sum_{i=1}^n \sum_{j=1}^m A_{ij}$, and ρ_C, ρ_A, ρ_{Cy} are associated to cooperative, antisymmetric and Cyclic games.

With these definitions, the 8 components of corollary 2.1 are simply computed by composition of above projectors, namely: **(ZT)** $\rho_Z \rho_T(A, B)$, **(ZCy)** $\rho_Z(\mathrm{Id} - \rho_T)(A, B)$, **(CT)** $(\mathrm{Id} - \rho_Z)\rho_T(A, B)$, **(CCy)** $(\mathrm{Id} - \rho_Z)(\mathrm{Id} - \rho_T)(A, B)$, **(ZST)** $\rho_Z \rho_S \rho_T(A, B)$, **(ZSCy)** $\rho_Z \rho_S(\mathrm{Id} - \rho_T)(A, B)$, **(ZAT)** $\rho_Z(\mathrm{Id} - \rho_S)\rho_T(A, B)$, **(ZACy)** $\rho_Z(\mathrm{Id} - \rho_S)(\mathrm{Id} - \rho_T)(A, B)$, **(CST)** $(\mathrm{Id} - \rho_Z)\rho_S \rho_T(A, B)$, **(CSCy)** $(\mathrm{Id} - \rho_Z)\rho_S(\mathrm{Id} - \rho_T)(A, B)$, **(CAT)** $(\mathrm{Id} - \rho_Z)(\mathrm{Id} - \rho_S)\rho_T(A, B)$, **(CACy)** $(\mathrm{Id} - \rho_Z)(\mathrm{Id} - \rho_S)(\mathrm{Id} - \rho_T)(A, B)$.

Following definitions 6 and 7, the **game signature** is taken to be the vector of size 8 containing the norms $\frac{1}{2}(||A_i|| + ||B_i||)$ of the components $(A_i, B_i)_{i \in [1,8]}$, where the matrix norm is chosen to be the L_2 norm. We further divide these 8 norms by their overall sum so as to interpret them as weights.

Training details. The RL policy was trained in the RLlib framework (Liang et al., 2018), and run on AWS using a EC2 C5 24xlarge instance with 96 CPUs, using Proximal Policy Optimization (Schulman et al., 2017). We considered square games of size $n = m = 10$. We used configuration parameters in line with (Schulman et al., 2017), that is a clip parameter of 0.3, an adaptive KL penalty with a KL target of 0.01 and a learning rate of 10^{-6}. Episodes were taken of length $\tau = 1000$ time steps, using $B = 90$ parallel runs in between policy updates (12000 policy updates). As a result, each policy update was performed with a batch size of $1000 \cdot 90$ timesteps, together with 30 iterations of stochastic gradient descent with SGD minibatch size of 8192. We used a fully connected neural net with 2 hidden layers, 256 nodes per layer, and $\tanh$ activation. Since our action space is continuous, the outputs of the neural net are the mean and stDev of a standard normal distribution, which is then used to sample actions probabilistically (the covariance matrix across actions is chosen to be diagonal).

At test time, we use a different budget for each game type since they are more or less complex and hence take more or less time to converge). These are reported in table 3.

Table 3. Budget τ used at test time.

Game Type	Budget (10^2 iterations)
ZT	2
ZCy	40
CT	2
CCy	8
ZT + ZCy	40
ZT + CT	4
ZT + CCy	4
ZCy + CT	40
ZCy + CCy	40
CT + CCy	4
ZT + ZCy + CT	40
ZT + ZCy + CCy	40
ZT + CT + CCy	4
ZCy + CT + CCy	40
ZT + ZCy + CT + CCy	40

Additional results. In figures 7-10 we report the same metrics as in figures 3 and 5, but for all other game mixtures considered in table 2. In figures 5, 7, 9, for each seed, we compute the mean value and correlation between time-series of coefficients, which are then averaged over seeds.

C.3. SHAP Analysis

In addition to the results in Section C.2 and as presented in figure 4, we also performed a SHAP analysis of the learned policy in the case of Full-RL CMWU. This explainability method — as first introduced in (Lundberg & Lee, 2017) — aids in understanding the marginal contribution of each input feature to the output of a machine learning model. In this case, we were interested in two key questions:

1. Which features were most important on a global level?
2. How do the importances of each feature evolve as a function of the iteration during a given rollout?

The former is a common question to ask, and provides an overall perspective on the behaviour of the learnt policy. The latter is much more refined and, as we shall see (below), reveals that there is non-trivial use of features in the early stages of a trajectory. What's more, these importance processes appear to vary significantly between the 8 different game types.

In all cases presented below, the SHAP values were computed using the "KernelExplainer" provided by the SHAP package (Lundberg & Lee, 2017). For each unique game type we sampled 200 trajectories of 2000 steps as generated by the policy to construct a dataset. To simplify the analysis of the SHAP values, we also grouped features into

1. (**BR**) the best response gaps $\delta_1(x^t, y^t)$, $\delta_2(x^t, y^t)$;
2. (**GRAD**) the current gradients Ay^t, $B^T x^t$;
3. (**HESS**) the current Hessians $H_{x^t} = A^T D_{x^t} A$, $H_{y^t} = BD_{y^t} B^T$;
4. (**PAY**) the current payoffs $x^{t,T} Ay^t$, $x^{t,T} By^t$;

see Section 3.2 for more details. This aggregation was performed by *summing* over the contribution of each component of the feature groups in one of two ways: either sum over the absolute SHAP values as in figure 4, or take the absolute value after summing over the signed SHAP values as in figures 11-12. The latter can always be done while retaining consistency due the additivity property of Shapley values. It is less clear what the former encodes, since we may lose correlation interaction effects, but these values are informative nonetheless. As such, we include both sets of results.

Global importance. The global importance across feature groups and game types is summarised in Figures 4 and 11-12. Observe that while the contributions do differ between the two aggregation approaches, the distribution is broadly the same. We first note that ZST is unique amongst the eight game types in that almost all emphasis is placed on the gradient inputs; this is true for the gradient and Hessian outputs $G_k = h_k$ and $H_k = -h_k\epsilon_k$. In general, this is considered to be the "easiest" of the eight core types, so it is perhaps not so surprising. We also find that there is a tangible difference between the importance of inputs between the learnt gradient and Hessian coefficients. For example, more emphasis is put on the payoff terms with respect the Hessian, and, conversely, the best response gaps only appear significantly for the gradient outputs. There also appears to be quite different behaviour between the learnt coefficients for zero-sum versus co-operative games, while in other cases (such as CST and CAT compared to CSCY and CACY) where there appear to be emergent symmetries. It is of course hard to draw strict conclusions from these phenomena, but it is absolutely clear that the network has learnt tangibly different policies as a function of the game signature.

Iteration-level importance. The second type of analysis we performed focused on the evolution of feature importance as a function of intra-trajectory iteration. The results, which are summarised in Figure 13, suggest a number of intriguing properties of the learnt policy. Firstly, we find that in all cases there exists a transient regime in the first ~ 100 iterations. This is particularly noticeable in the ZST case in which the gradient importance grows very sharply from around 70% to nearly 100%. This rapid change in the early stages of learning is consistent across most games. However, it is clear that the zero-sum and cyclic variants have a much longer transient period. Indeed, for ZSCY it takes nearly 2000 steps to stabilise to some fixed limit point. This aligns closely with the intuition that zero-sum/cyclic games are generally harder to solve.

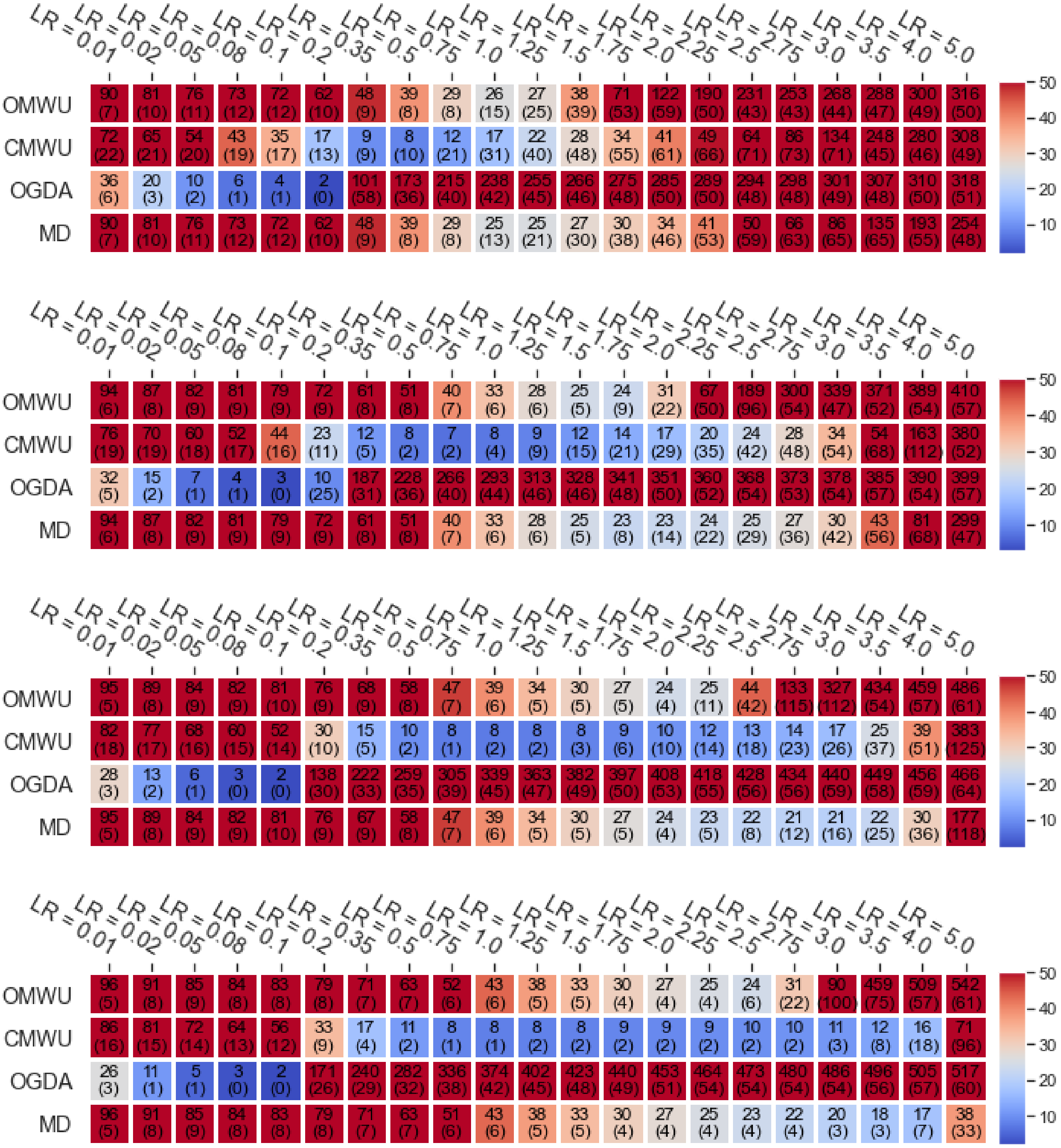

Figure 6. Zero-sum case: average score β_τ across various learning rates h and game sizes n. $\tau = 500$. $\epsilon = 0.25 \cdot h^{-1}n$. StDev in brackets. From **Top** to **Bottom**: $n = 25, 50, 75, 100$.

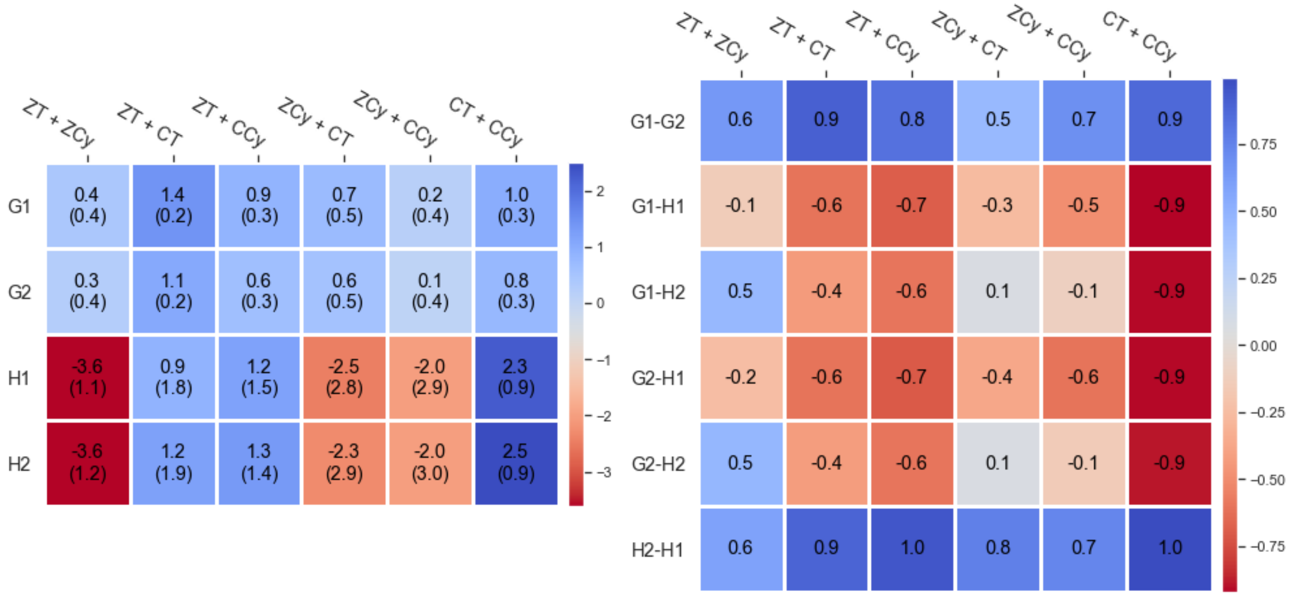

Figure 7. full-RL CMWU coefficients of gradient $G_k = h_k$ and Hessian $H_k = -h_k\epsilon_k$ of players $k = 1, 2$ across game types (8 pure game components - mixtures of 2). **(left)** Mean value. **(right)** Pairwise correlations. StDev in brackets.

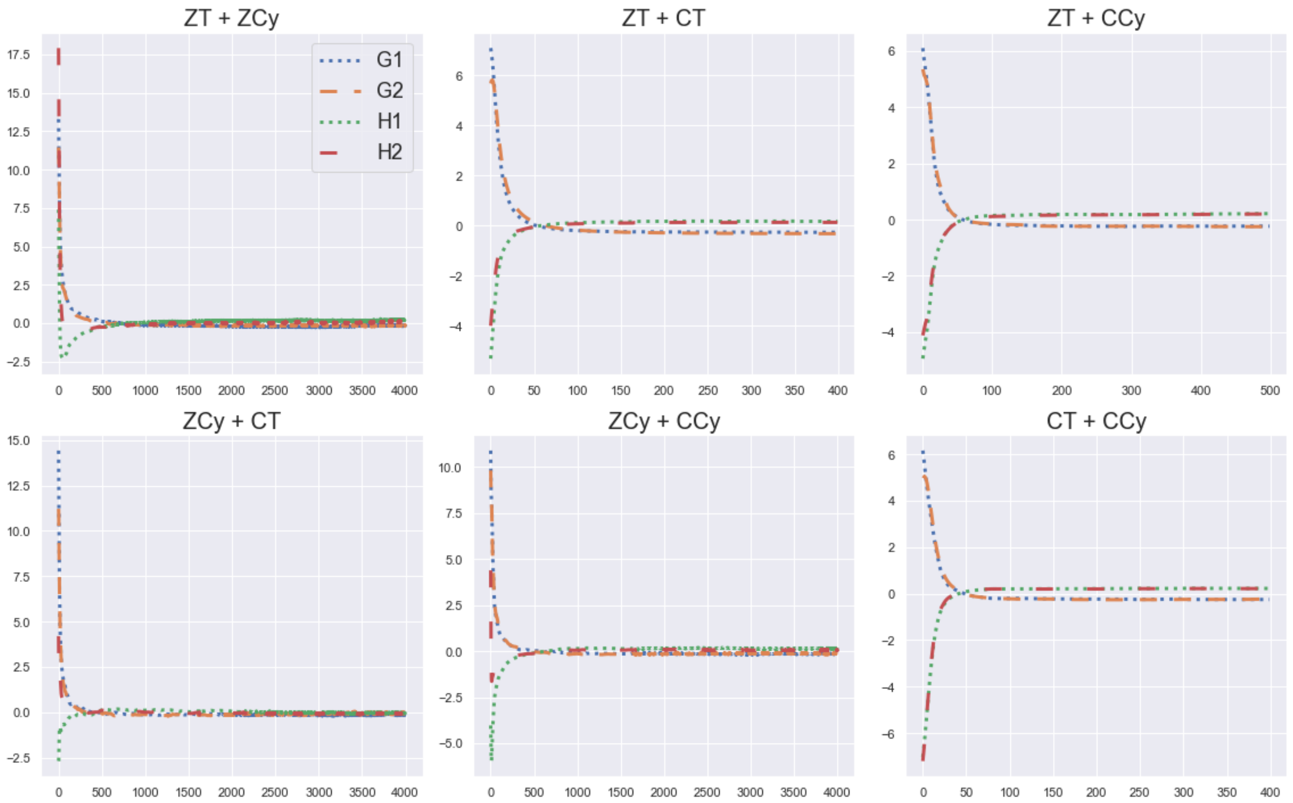

Figure 8. Average standardized per-episode-trajectory ("shape") of coefficients of gradient $G_k = h_k$ and Hessian $H_k = -h_k\epsilon_k$ of players $k = 1, 2$ across game types (8 pure game components - mixtures of 2), as a function of time t.

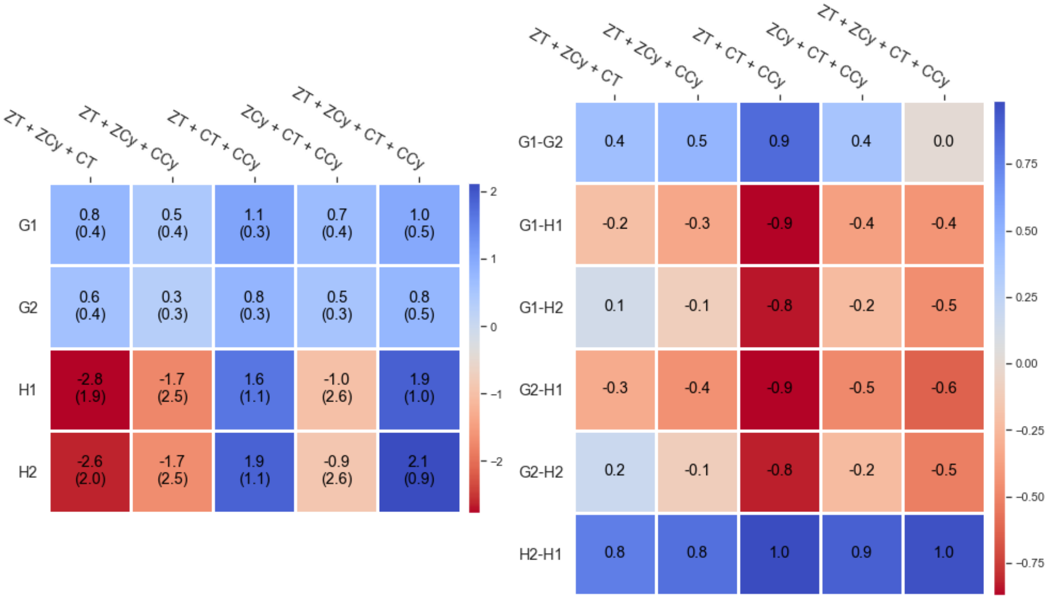

Figure 9. full-RL CMWU coefficients of gradient $G_k = h_k$ and Hessian $H_k = -h_k \epsilon_k$ of players $k = 1, 2$ across game types (8 pure game components - mixtures of 3-4). **(left)** Mean value. **(right)** Pairwise correlations. StDev in brackets.

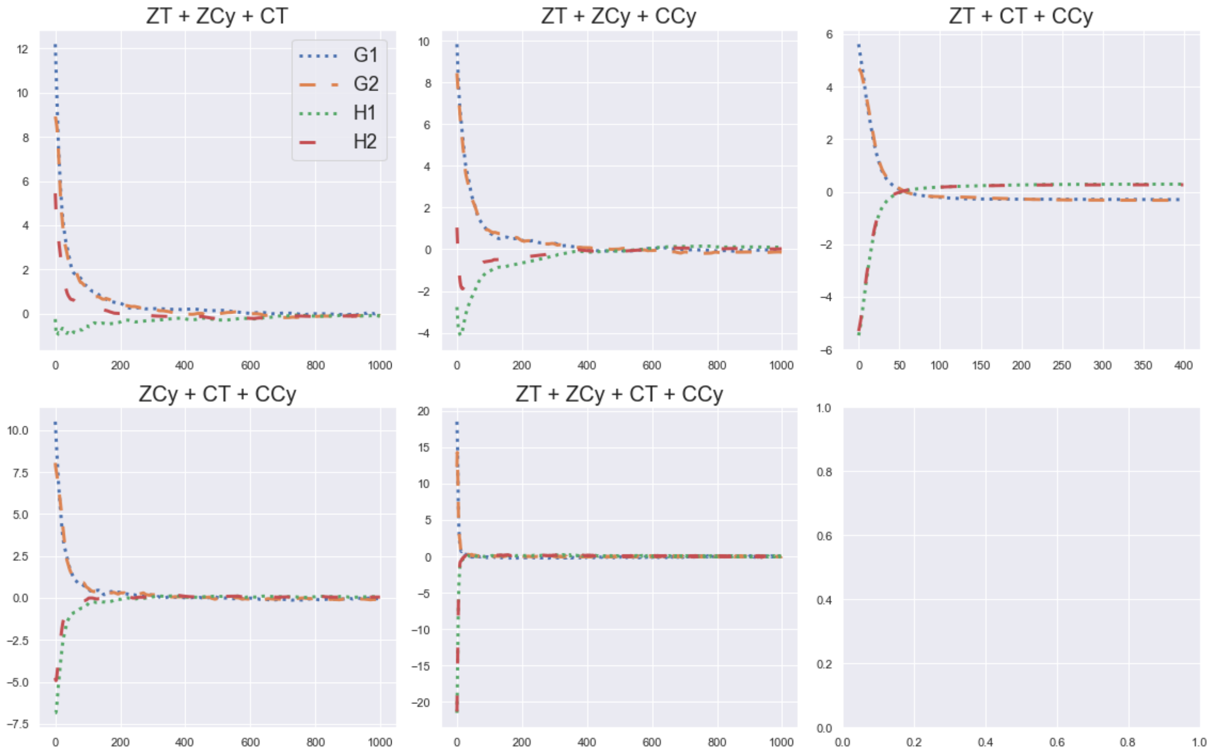

Figure 10. Average standardized per-episode-trajectory ("shape") of coefficients of gradient $G_k = h_k$ and Hessian $H_k = -h_k \epsilon_k$ of players $k = 1, 2$ across game types (8 pure game components - mixtures of 3-4), as a function of time t.

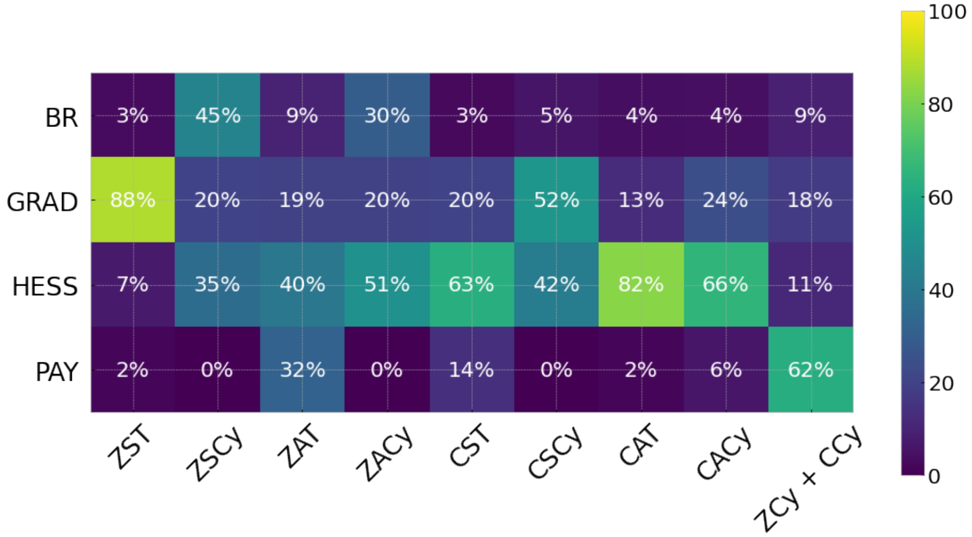

Figure 11. Percentage contribution towards absolute SHAP importance aggregated across signed values with respect to *gradient learning rates* $G_k = h_k$ for players $k = 1, 2$ (partial policy outputs) for the four input groups and eight unique game types. These values were computed for the data points at iteration 50 of each trajectory. Columns are normalised to unity with percentages illustrated explicitly in each box as well as by colour.

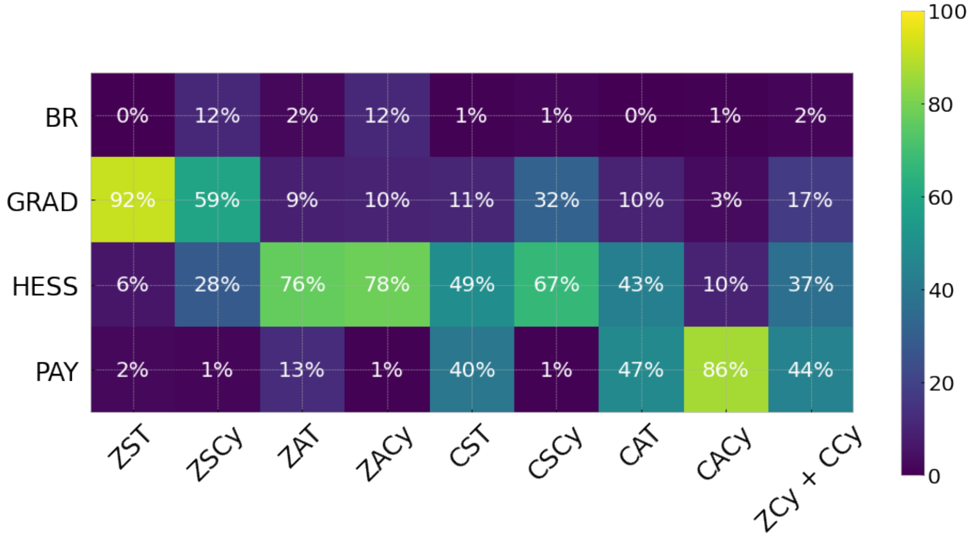

Figure 12. Percentage contribution towards absolute SHAP importance aggregated across signed values with respect to *Hessian learning rates* $H_k = -h_k\epsilon_k$ (partial policy outputs) for the four input groups and eight unique game types. These values were computed for the data points at iteration 50 of each trajectory. Columns are normalised to unity with percentages illustrated explicitly in each box as well as by colour.

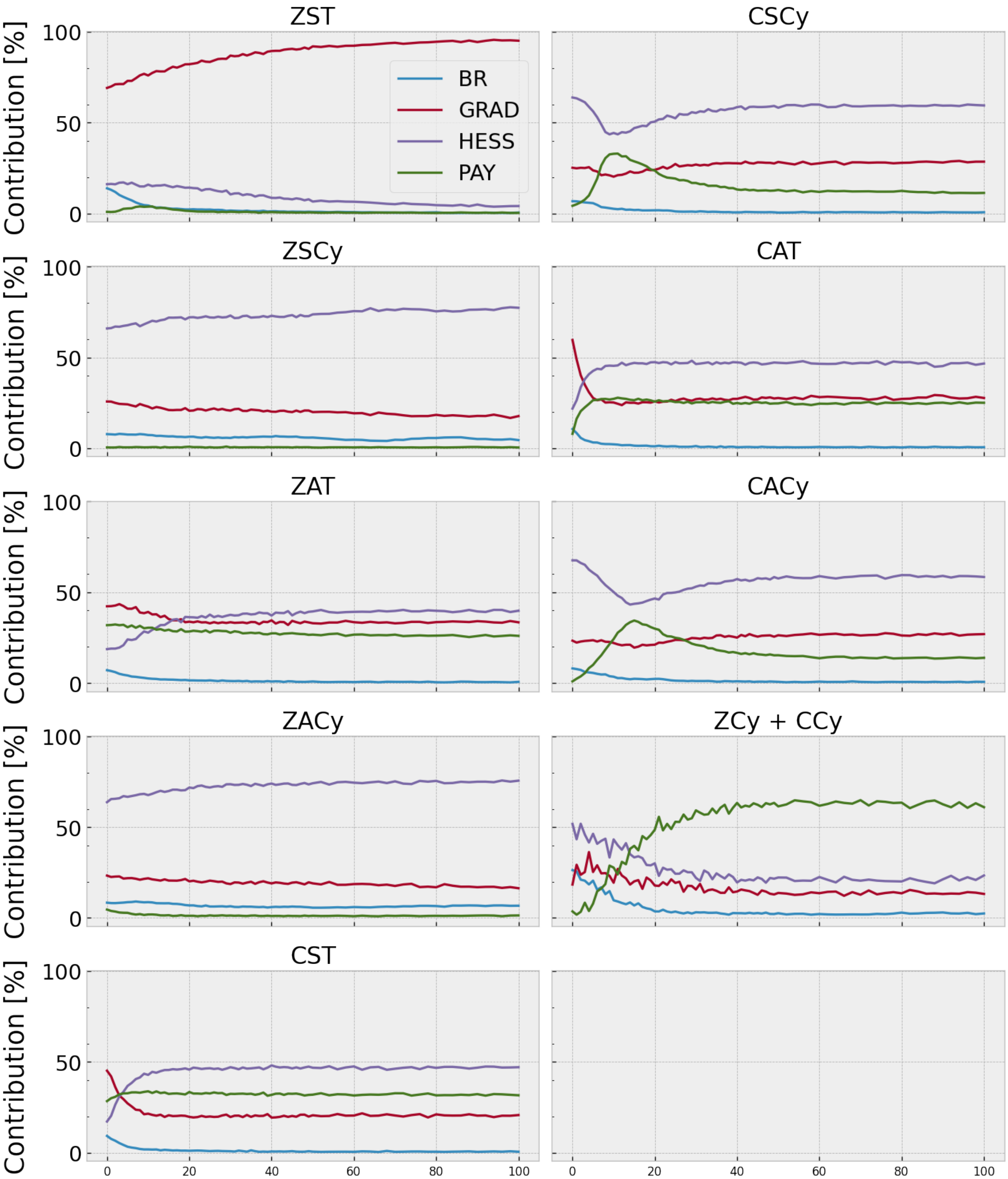

Figure 13. Percentage contribution towards absolute SHAP importance aggregated across the absolute values with respect to *all learning rates* for the four input groups and eight unique game types over time. In each plot, the curves are normalised to unity at any given iteration.