

Navigating the Landscape of Multiplayer Games

Shayegan Omidshafiei^{1,*†}, Karl Tuyls^{1,*}, Wojciech M. Czarnecki², Francisco C. Santos³,
 Mark Rowland², Jerome Connor², Daniel Hennes¹, Paul Muller¹, Julien Pérolut¹,
 Bart De Vylder¹, Audrunas Gruslys², and Rémi Munos¹

¹DeepMind, Paris, France

²DeepMind, London, UK

³INESC-ID and Instituto Superior Técnico, Universidade de Lisboa, Portugal

* Equal contributors.

†Corresponding author.

Abstract

Multiplayer games have long been used as testbeds in artificial intelligence research, aptly referred to as the Drosophila of artificial intelligence. Traditionally, researchers have focused on using well-known games to build strong agents. This progress, however, can be better informed by characterizing games and their topological landscape. Tackling this latter question can facilitate understanding of agents and help determine what game an agent should target next as part of its training. Here, we show how network measures applied to response graphs of large-scale games enable the creation of a landscape of games, quantifying relationships between games of varying sizes and characteristics. We illustrate our findings in domains ranging from canonical games to complex empirical games capturing the performance of trained agents pitted against one another. Our results culminate in a demonstration leveraging this information to generate new and interesting games, including mixtures of empirical games synthesized from real world games.

Introduction

Games have played a prominent role as platforms for the development of learning algorithms and the measurement of progress in artificial intelligence (AI)^{122,128,154,179}. Multiplayer games, in particular, have played a pivotal role in AI research and have been extensively investigated in machine learning, ranging from abstract benchmarks in game theory over popular board games such as Chess^{23,137} and Go¹³⁶ (referred to as the Drosophila of AI research⁹²), to realtime strategy games such as StarCraft II¹⁶³ and Dota 2¹⁰⁷. Overall, AI research has primarily placed emphasis on training of strong agents; we refer to this as the Policy Problem, which entails the search for super human-level AI performance. Despite this progress, the need for a task theory, a framework for taxonomizing, characterizing, and decomposing AI tasks has become increasingly important in recent years^{66,145}. Naturally, techniques for understanding the space of games are likely beneficial for the algorithmic development of future AI entities^{26,145}. Understanding and decomposing the characterizing features of games can be leveraged for downstream training of agents via curriculum learning⁸, which seeks to enable agents to learn increasingly-complex tasks.

A core challenge associated with designing such a task theory has been recently coined the Problem Problem, defined as “the engineering problem of generating large numbers of interesting adaptive environments to support research”⁸⁴. Research associated with the Problem Problem has a rich history spanning over 30 years, including the aforementioned work on task theory^{10,66,145}, procedurally-generated videogame features^{17,49,150}, generation of games and rule-sets for General Game Playing^{18,50,78,100,110,147,149}, and procedural content generation techniques^{28,29,72,99,117,133,139,148,167,168}; we refer readers to Supplementary Note 1 for detailed discussion of these and related works. An important question that underlies several of these interlinked fields is: what makes a game interesting enough for an AI agent to learn to play? Resolving this

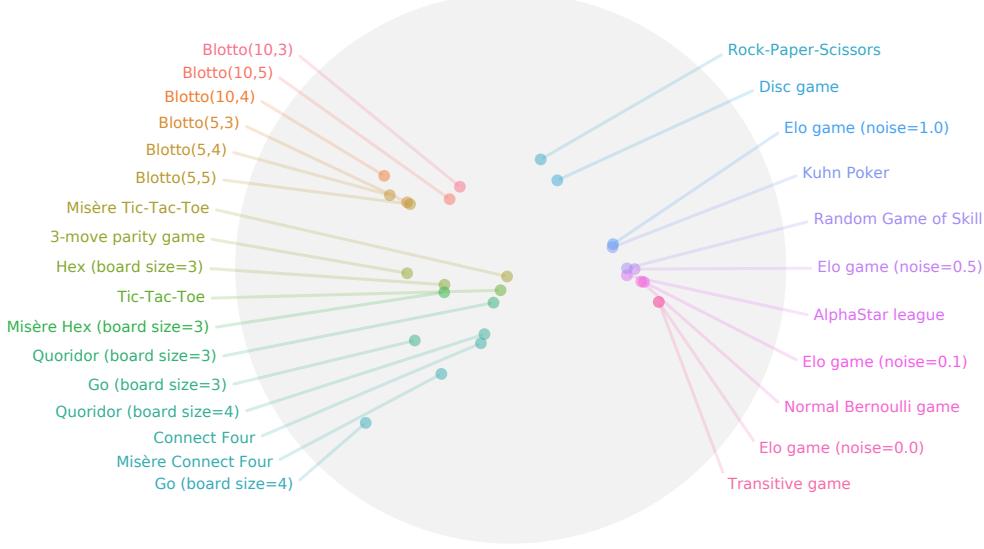


Figure 1: A landscape of games revealed by the proposed response graph-based workflow. This landscape is generated by collecting features associated with the response graph of each game, and plotting the top two principal components. At a high level, games whose response graphs are characteristically similar are situated close to one another in this landscape. Notably, variations of games with related rules are well-clustered together, indicating strong similarity despite their widely-varying raw sizes. Instances of Blotto cluster together, despite their payoff table sizes ranging from 20×20 for Blotto(5,3) to 1000×1000 for Blotto(10,5). Games with strong transitive components (e.g., variations of Elo games, AlphaStar League, Random Game of Skill, and Normal Bernoulli Game) can be observed to be well separated from strongly cyclical games (Rock–Paper–Scissors and the Disc game). Closely-related real-world games (i.e., games often played by humans in the real world, such as Hex, Tic-Tac-Toe, Connect Four and each of their respective Misère counterparts) are also clustered together.

requires techniques that can characterize the topological landscape of games, which is the topic of interest in this paper. We focus, in particular, on the characterization of multiplayer games (i.e., those involving interactions of multiple agents), and henceforth use the shorthand of games to refer to this class.

The objective of this paper is to establish tools that enable discovery of a topology over games, regardless of whether they are interesting or not; we do not seek to answer the interestingness question here, although such a toolkit can be useful for subsequently considering it. Naturally, many notions of what makes a game interesting exist, from the perspectives of human-centric game design, developmental learning, curriculum learning, AI training, and so on. Our later experiments link to the recent work of Czarnecki et al.³³, which investigated properties that make a game interesting specifically from an AI training perspective, as also considered here. We follow the interestingness characterization of Czarnecki et al.³³, which defines so-called Games of Skill that are engaging for agents due to: i) a notion of progress; ii) availability of diverse play styles that perform similarly well. We later show how clusters of games discovered by our approach align with this notion of interestingness. An important benefit of our approach is that it applies to adversarial and cooperative games alike. Moreover, while the procedural game structure generation results we later present target zero-sum games due to the payoff parameterization chosen in those particular experiments, they readily extend to general-sum games.

How does one topologically analyze games? One can consider characterizations of a game as quantified by measures such as the number of strategies available, players involved, whether the game is symmetric, and so on. One could also order the payouts to players to taxonomize games, as done in prior works exploring 2×2 games^{19,87,115}. For more complex games, however, such measures are crude, failing to disambiguate differences in similar games. One may also seek to classify games from the standpoint of computational complexity. However, a game that is computationally challenging to solve may not necessarily be interesting

to play. Overall, designation of a single measure characterizing games is a non-trivial task.

It seems useful, instead, to consider measures that characterize the possible strategic interactions in the game. A number of recent works have considered problems involving such interactions^{5,6,7,63,64,65,80,89,97,106,121}. Many of these works analyze agent populations, relying on game-theoretic models capturing pairwise agent relations. Related models have considered transitivity (or lack thereof) to study games from a dynamical systems perspective^{4,155}; here, a transitive game is one where strategies can be ordered in terms of strength, whereas an intransitive game may involve cyclical relationships between strategies (e.g., Rock–Paper–Scissors). Fundamentally, the topology exposed via pairwise agent interactions seems a key enabler of the powerful techniques introduced in the above works. In related literature, graph theory is well-established as a framework for topological analysis of large systems involving interacting entities^{12,38,160}. Complexity analysis via graph-theoretic techniques has been applied to social networks^{130,170}, the webgraph^{41,51}, biological systems^{13,85,109}, econometrics^{60,143}, and linguistics¹⁶⁴. Here we demonstrate that the combination of graph and game theory provides useful tools for analyzing the structure of general-sum, many-player games.

The primary contribution of this work is a graph-based toolkit for analysis and comparison of games. As detailed below, the nodes in our graphs are either strategies (in abstract games) or AI agents (in empirical games, where strategies correspond to learned or appropriately-sampled player policies). The interactions between these agents, as quantified by the game’s payoffs, constitute the structure of the graph under analysis. We show that this set of nodes and edges, also known as the α -Rank response graph^{97,106,121}, yields useful insights into the structure of individual games and can be used to generate a landscape over collections of games (as in Figure 1). We subsequently use the toolkit to analyze various games that are both played by humans or wherein AI agents have reached human-level performance, including Go, MuJoCo Soccer, and StarCraft II. Our overall analysis culminates in a demonstration of how the topological structure over games can be used to tackle the interestingness question of the Problem Problem, which seeks to automatically generate games with interesting characteristics for learning agents⁸⁴.

Results

Overview

We develop a foundational graph-theoretic toolkit that facilitates analysis of canonical and large-scale games, providing insights into their related topological structure in terms of their high-level strategic interactions. The prerequisite game theory background and technical details are provided in the Methods section, with full discussion of related works and additional details in Supplementary Note 1.

Our results are summarized as follows. We use our toolkit to characterize a number of games, first analyzing motivating examples and canonical games with well-defined structures, then extending to larger-scale empirical games datasets. For these larger games, we rely on empirical game-theoretic analysis^{166,174}, where we characterize an underlying game using a sample set of policies. While the empirical game-theoretic results are subject to the policies used to generate them, we rely on a sampling scheme designed to capture a diverse variety of interactions within each game, and subsequently conduct sensitivity analysis to validate the robustness of the results. We demonstrate correlation between the complexity of the graphs associated with games and the complexity of solving the game itself. In Supplementary Note 2, we evaluate our proposed method against baseline approaches for taxonomization of 2×2 games¹⁹. We finally demonstrate how this toolkit can be used to automatically generate interesting game structures that can, for example, subsequently be used to train AI agents.

Motivating example

Let us start with a motivating example to solidify intuitions and explain the workflow of our graph-theoretic toolkit, using classes of games with simple parametric structures in the player payoffs. Specifically, consider games of three broad classes (generated as detailed in the Supplementary Methods 1): games in which strategies have a clear transitive ordering (Figure 2a); games in which strategies have a cyclical structure wherein all but the final strategy are transitive with respect to one another (Figure 2d); and games with random (or no clear underlying) structure (Figure 2g). We shall see that the core characteristics of games with shared underlying structure is recovered via the proposed analysis.

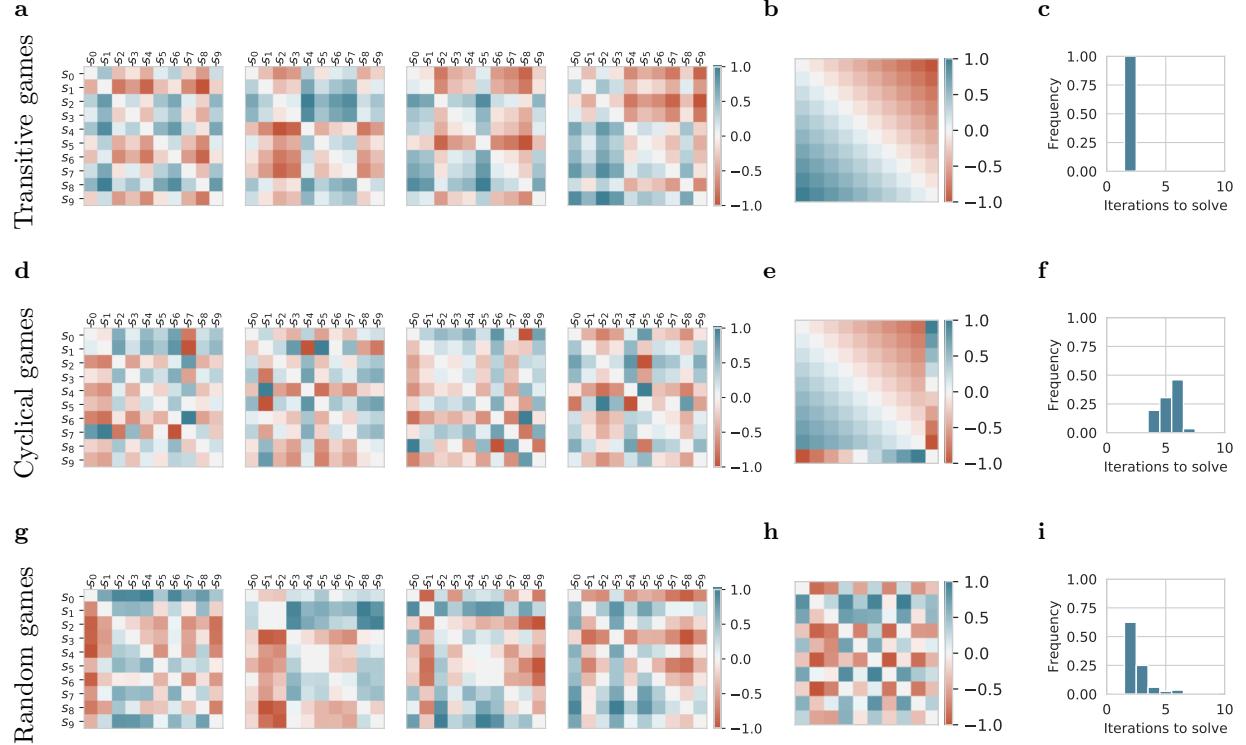


Figure 2: Motivating example of three classes of two-player, symmetric zero-sum games. **a**, **d**, and **g**, respectively, visualize payoffs for instances of games with transitive, cyclical, and random structure. Each exemplified game consists of two players with 10 strategies each (with payoff row and column labels, $\{s_0, \dots, s_9\}$, indicating the strategies). Despite the numerous payoff variations possible in each class of games illustrated, each shares the underlying payoff structure shown, respectively, in **b**, **e**, and **h**. Moreover, variations in payoffs can notably impact the difficulty of solving (i.e., finding the Nash equilibrium) of these games, as visualized in **c**, **f**, **i**.

Each of these figures visualizes the payoffs corresponding to 4 instances of games of the respective class, with each game involving 10 strategies per player; more concretely, entry $M(s_i, s_j)$ of each matrix visualized in Figures 2a, 2d and 2g quantifies the payoff received by the first player if the players, respectively, use strategies s_i and s_j (corresponding, respectively, to the i -th and j -th row and column of each payoff table). Despite the variance in payoffs evident in the instances of games exemplified here, each essentially shares the payoff structure exposed by re-ordering their strategies, respectively, in Figures 2b, 2e and 2h. In other words, the visual representation of the payoffs in this latter set of figures succinctly characterizes the backbone of strategic interactions within these classes of games, despite not being immediately apparent in the individual instances visualized.

More importantly, the complexity of learning useful mixed strategies to play in each of these games is closely associated with this structural backbone. To exemplify this, consider the computational complexity of solving each of these games; for brevity, we henceforth refer to solving a game as synonymous with finding a Nash equilibrium (similar to prior works^{16,20,119,171}, wherein the Nash equilibrium is the solution concept of interest). Specifically, we visualize this computational complexity by using the Double Oracle algorithm⁹³, which has been well-established as a Nash solver in multiagent and game theory literature^{15,71,80,116}. At a high level, Double Oracle starts from a sub-game consisting of a single randomly-selected strategy, iteratively expands the strategy space via best responses (computed by an oracle), until discovery of the Nash equilibrium of the full underlying game.

Figures 2c, 2f and 2i visualize the distribution of Double Oracle iterations needed to solve the corresponding games, under random initializations. Note, in particular, that although the underlying payoff

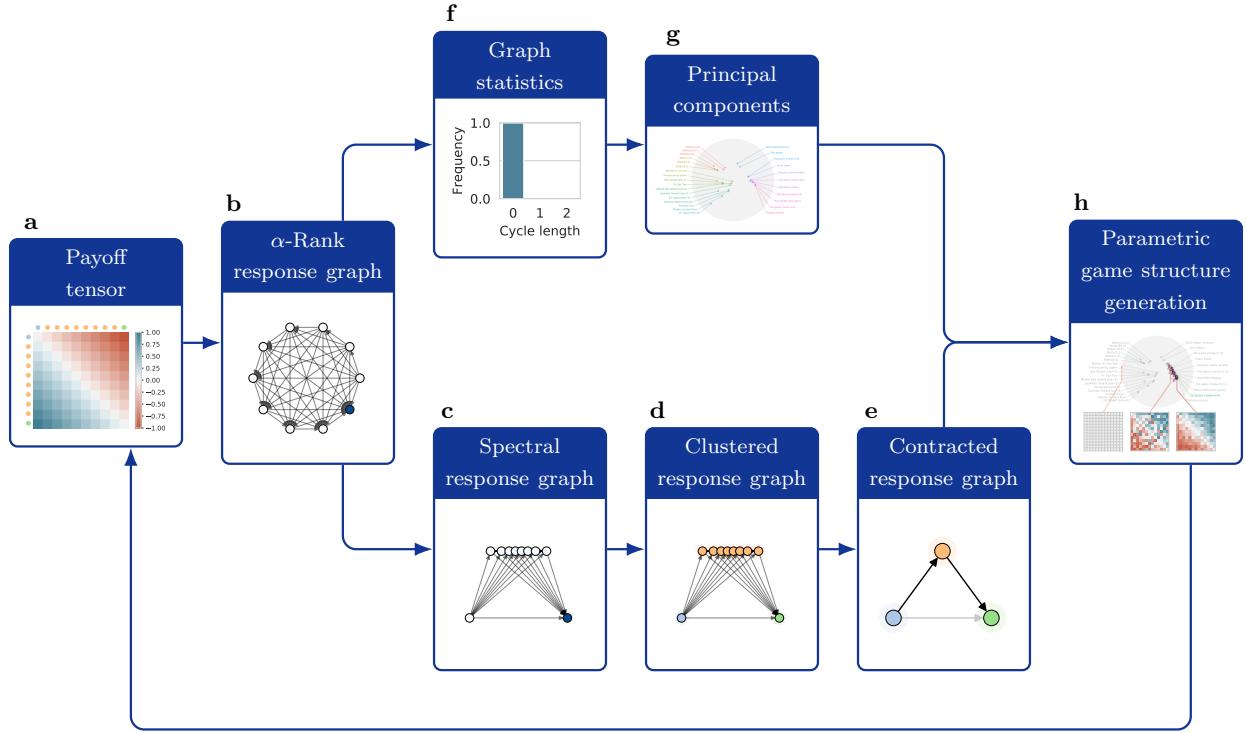


Figure 3: Method workflow, with accompanying transitive game results. Given the game payoffs **a**, the so-called α -Rank response graph of the game is visualized in **b**. In **c**, reprojection of the response graph by using the top eigenvectors of the graph Laplacian yields the spectral response graph, wherein similar strategies are placed close to one another. In **d**, taking this one step further, one can cluster the spectral response graph, yielding the clustered response graph, which exposes three classes of strategies in this particular example. In **e**, contracting the clustered graph by fusing nodes within each cluster yields the high-level characterization of transitive games. In **f**, the lack of cycles in the particular class of transitive games becomes evident. Finally, in **g** and **h**, one can extract the principal components of various response graph statistics and establish a feedback loop to a procedural game structure generation scheme to yield new games.

structure of the transitive and cyclical games respectively visualized in Figure 2a and Figure 2d is similar, the introduction of a cycle in the latter class of games has a substantial impact on the complexity of solving them (as evident in Figure 2f). In particular, whereas the former class of games are solved using a low (and deterministic) number of iterations, the latter class requires additional iterations due to the presence of cycles increasing the number of strategies in the support of the Nash equilibrium.

Workflow

Overall, the characterization of the topological structure of games is an important and nuanced problem. To address this problem, we use graph theory to build an analytical toolkit automatically summarizing the high-level strategic interactions within a game, and providing useful complexity measures thereof. Specifically, consider again our motivating transitive game, re-visualized using a collection of graph-based measures in Figure 3. Each of these measures provides a different viewpoint on the underlying game, collectively characterizing it. Specifically, given the game payoffs, Figure 3b visualizes the so-called α -Rank response graph of the game; here, each node corresponds to a strategy (for either player, as this particular game's payoffs are symmetric). Transition probabilities between nodes are informed by a precise evolutionary model (detailed in Methods and Omidshafiei et al.¹⁰⁶); roughly speaking, a directed edge from one strategy to another indicates the players having a higher preference for the latter strategy, in comparison to the former. The response graph, thus, visualizes all preferential interactions between strategies in the game. Moreover, the color intensity of each node indicates its so-called α -Rank, which measures the long-term preference of

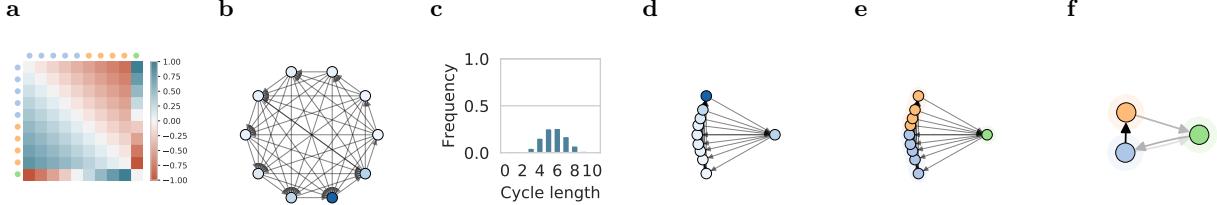


Figure 4: Cyclical game results. **a** game payoffs, **b** response graph, **c** cycles histogram, **d** spectral response graph, **e** clustered response graph, **f** contracted response graph.

the players for that particular strategy, as dictated by the transition model mentioned above; specifically, darker colors here indicate more preferable strategies.

This representation of a game as a graph enables a variety of useful insights into its underlying structure and complexity. For instance, consider the distribution of cycles in the graph, which play an important role in multiagent evaluation and training schemes^{5,106,138,155} and, as later shown, are correlated to the computational complexity of solving two-player zero-sum games (e.g., via Double Oracle). Figure 3f makes evident the lack of cycles in the particular class of transitive games; while this is clearly apparent in the underlying (fully ordered) payoff visualization of Figure 3a, it is less so in the unordered variants visualized in Figure 2a. Even so, the high-level relational structure between the strategies becomes more evident by conducting a spectral analysis of the underlying game response graph. Full technical details of this procedure are provided in the Methods section. At a high level, the so-called Laplacian spectrum (i.e., eigenvalues) of a graph, along with associated eigenvectors, captures important information regarding it (e.g., number of spanning trees, algebraic connectivity, and numerous related properties⁹⁶). Reprojecting the response graph by using the top eigenvectors yields the spectral response graph visualized in Figure 3c, wherein similar strategies are placed close to one another. Moreover, one can cluster the spectral response graph, yielding the clustered response graph, which exposes three classes of strategies in Figure 3d: a fully dominated strategy with only outgoing edges (a singleton cluster, on the bottom left of the graph), a transient cluster of strategies with both incoming and outgoing edges (top cluster), and a dominant strategy with all incoming edges (bottom right cluster). Finally, contracting the clustered graph by fusing nodes within each cluster yields the high-level characterization of transitive games shown in Figure 3e.

We can also conduct this analysis for instances of our other motivating games, such as the cyclical game visualized in Figure 4a. Note here the distinct differences with the earlier transitive game example; in the cyclical game, the α -Rank distribution in the response graph (Figure 4b) has higher entropy (indicating preference for many strategies, rather than one, due to the presence of cycles). Moreover, the spectral reprojection in Figure 4d reveals a clear set of transitive nodes (left side of visualization) and a singleton cluster of a cycle-inducing node (right side). Contracting this response graph reveals the fundamentally cyclical nature of this game (Figure 4f). Finally, we label each strategy (i.e., each row and column) of the original payoff table Figure 4a based on this clustering analysis. Specifically, the color-coded labels on the far left (respectively, top) of each row (respectively, column) in Figure 4a correspond to the clustered strategy colors in Figure 4d. This color-coding helps clearly identify the final strategy (i.e., bottom row of the payoff table) as the outlier enforcing the cyclical relationships in the game. Note that while there is no single graphical structure that summarizes the particular class of random games visualized earlier in Figure 2h, we include this analysis for several instances of such games in Supplementary Note 2.

Crucially, a key benefit of this analysis is that the game structure exposed is identical for all instances of the transitive and cyclical games visualized earlier in Figures 2a and 2d, making it significantly easier to characterize games with related structure, in contrast to analysis of raw payoffs. Our later case studies further exemplify this, exposing related underlying structures for several classes of more complex games.

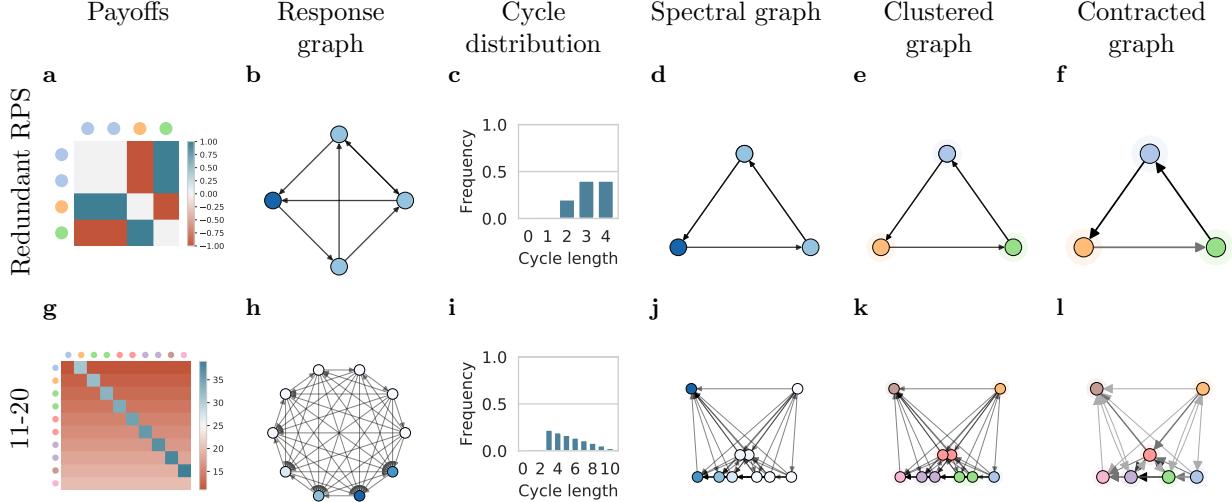


Figure 5: Results for Redundant Rock–Paper–Scissors (RPS) and 11–20 game. In Redundant RPS, the redundant copy of the first strategy (Rock) is clustered in the spectral response graph. In 11–20, seven clusters of strategies are revealed, exposing the cyclical nature of this game.

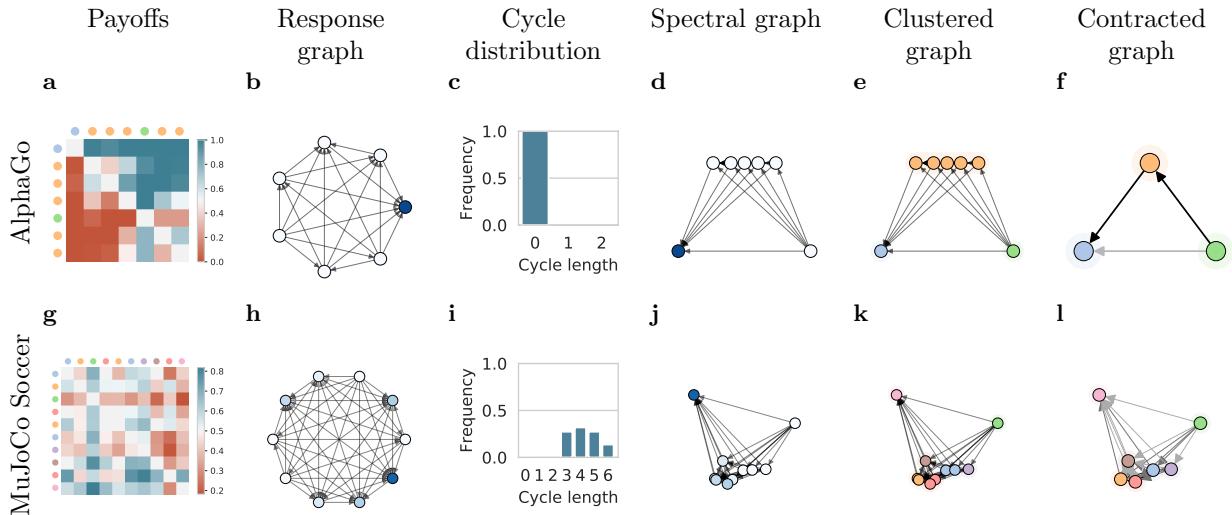


Figure 6: Results for empirical games of AlphaGo and MuJoCo soccer dataset. Note that as these are empirical games, strategies here correspond to trained AI agents. In AlphaGo, the strong transitive relationship between agents is revealed via our analysis. In MuJoCo soccer, more complex relations between similarly-performing agents are revealed in the clusters produced.

Analysis of Canonical and Real-World Games

The insights afforded by our graph-theoretic approach apply to both small canonical games and larger empirical games (where strategies are synonymous with trained AI agents).

Consider the canonical Rock–Paper–Scissors game, involving a cycle among the three strategies (wherein Rock loses to Paper, which loses to Scissors, which loses to Rock). Figure 5a visualizes a variant of this game involving a copy of the first strategy, Rock, which introduces a redundant cycle and thus affects the distribution of cycles in the game. Despite this, the spectral response graph (Figure 5d) reveals that the redundant game topologically remains the same as the original Rock–Paper–Scissors game, thus reducing to the original game under spectral clustering.

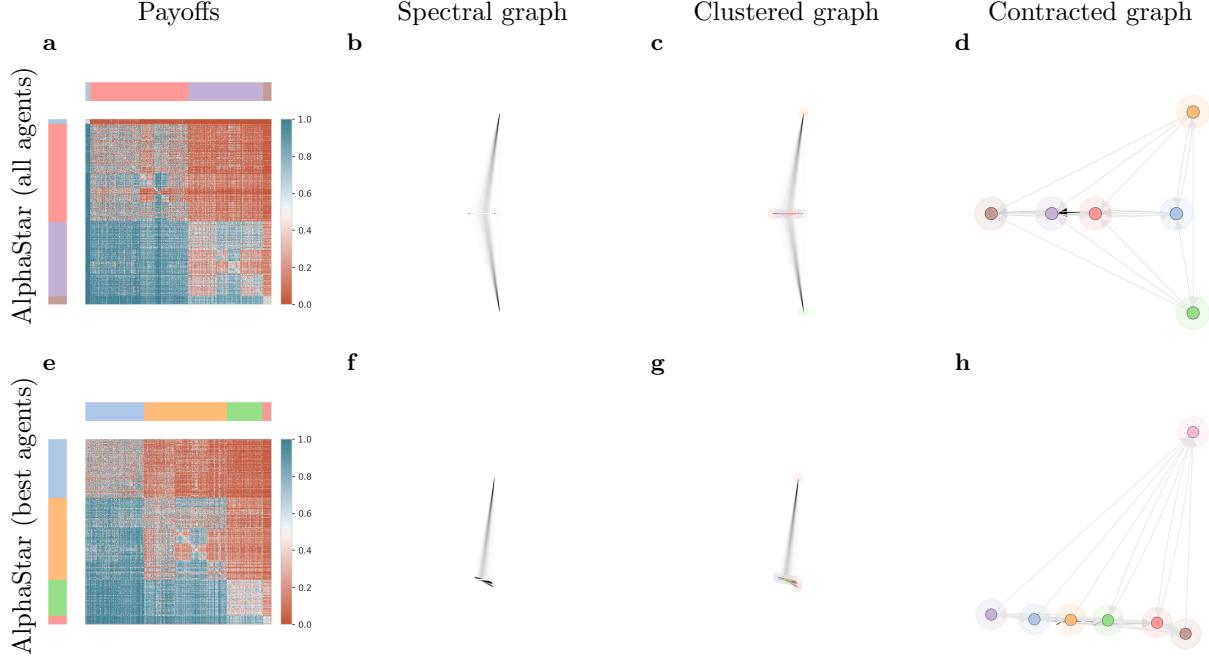


Figure 7: AlphaStar results, with both the full league and the league with best agents visualized. Spectral analysis of the empirical AlphaStar League game reveals that several key subsets of closely-performing agents, illustrated in **d**. Closer inspection of the agents used to construct this empirical payoff table reveals the following insights, with agent types corresponding to those detailed in Vinyals et al.¹⁶³: i) the blue, orange, and green clusters are composed of agents in the initial phases of training, which are generally weakest (as observed in **d**, and also visible as the narrow band of low payoffs in the top region of the payoff table **a**); ii) the red cluster consists primarily of various, specialized exploiter agents; iii) the purple and brown clusters are primarily composed of the league exploiters and main agents, with the latter being generally higher strength than the former.

This graph-based analysis also extends to general-sum games. As an example, consider the slightly more complex game of 11-20, wherein two players each request an integer amount of money between 11 to 20 units (inclusive). Each player receives the amount requested, though a bonus of 20 units is allotted to one player if they request exactly 1 unit less than the other player. The payoffs and response graph of this game are visualized, respectively, in Figures 5g and 5h, where strategies, from top-to-bottom and left-to-right in the payoff table, correspond to increasing units of money being requested. This game, first introduced by Arad and Rubinstein², is structurally designed to analyze so-called k -level reasoning, wherein a level-0 player is naive (i.e., here simply requests 20 units), and any level- k player responds to an assumed level- $(k - 1)$ opponent; e.g., here a level-1 player best responds to an assumed level-0 opponent, thus requesting 19 units to ensure receiving the bonus units.

The spectral response graph here (Figure 5j) indicates a more complex mix of transitive and intransitive relations between strategies. Notably, the contracted response graph (Figure 5l) reveals 7 clusters of strategies. Referring back to the rows of payoffs in Figure 5g, relabeled to match cluster colors, demonstrates that our technique effectively pinpoints the sets of strategies that define the rules of the game: weak strategies (11 or 12 units, first two rows of the payoff table, and evident in the far-right of the clustered response graph), followed by a set of intermediate strategies with higher payoffs (clustered pairwise, near the lower-center of the clustered response graph), and finally the two key strategies that establish the cyclical relationship within the game through k -level reasoning (19 and 20 units, corresponding to level-0 and level-1 players, in the far-left of the clustered response graph).

This analysis extends to more complex instances of empirical games, which involve trained AI agents, as next exemplified. Consider first the game of Go, as played by 7 AlphaGo variants: $AG(r)$, $AG(p)$, $AG(v)$,

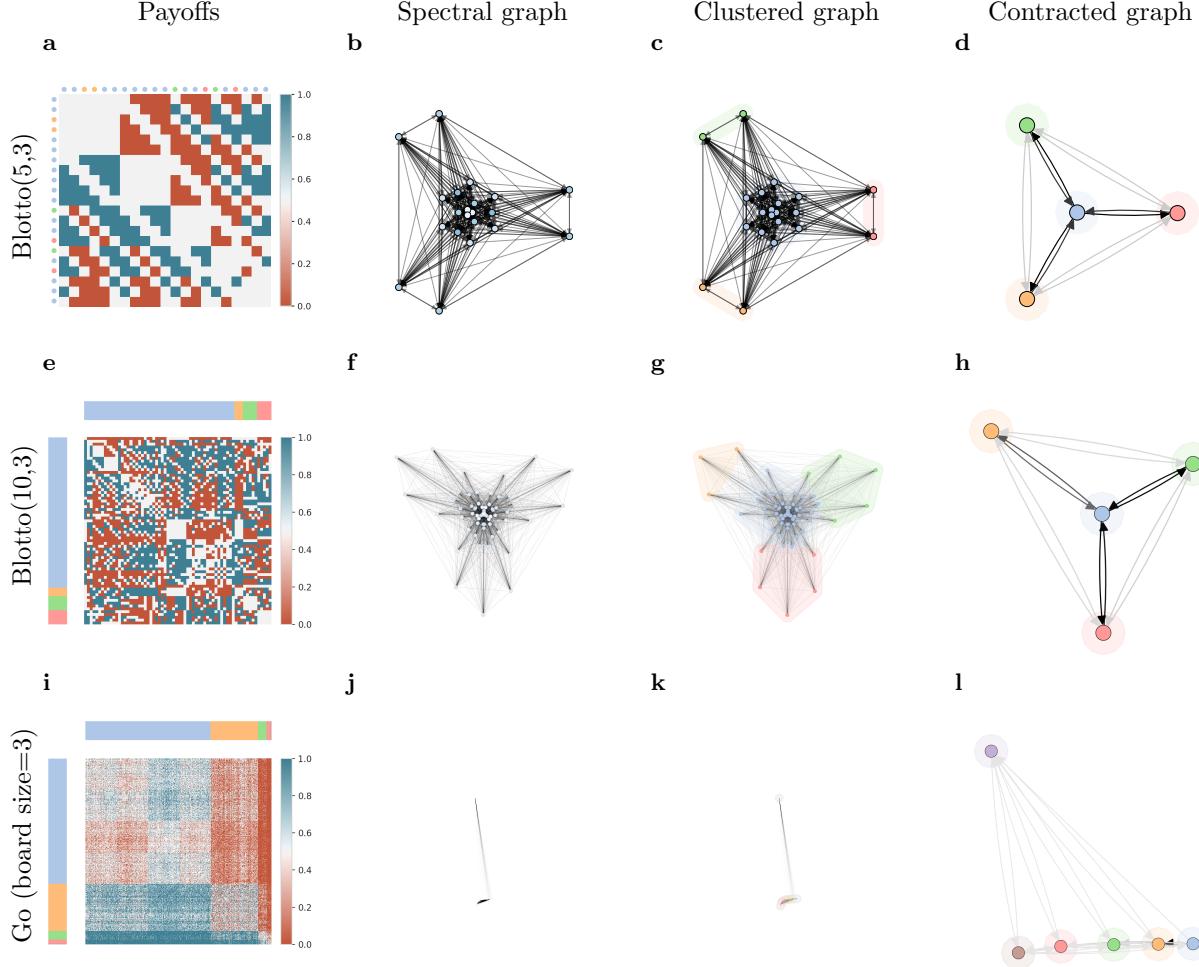


Figure 8: Results for Blotto(5,3), Blotto(10,3), and Go (board size=3). Despite the significant difference in sizes, both instances of Blotto yield a remarkably similar contracted response graph. Moreover, the contracted response graph for Go is notably different from AlphaGo results, due to the latter being an empirical game constructed from trained AI agents rather than a representative set of sampled policies.

$AG(rv)$, $AG(rp)$, $AG(vp)$, and $AG(rvp)$, where each variant uses the specified combination of rollouts r , value networks v , and/or policy networks p . We analyze the empirical game where each strategy corresponds to one of these agents, and payoffs (Figure 6a) correspond to the win rates of these agents when paired against each other (as detailed by Silver et al.¹³⁶, Table 9). The α -Rank distribution indicated by the node (i.e., strategy) color intensities in Figure 6b reveals $AG(rvp)$ as a dominant strategy, and the cycle distribution graph Figure 6c reveals a lack of cycles here. The spectral response graph, however, goes further, revealing a fully transitive structure (Figures 6d and 6e), as in the motivating transitive games discussed earlier. The spectral analysis on this particular empirical game, therefore, reveals its simple underlying transitive structure (Figure 6f).

Consider a more interesting empirical game, wherein agents are trained to play soccer in the continuous control domain of MuJoCo, exemplified in Figure 6 (second row). Each agent in this empirical game is generated using a distinct set of training parameters (e.g., feedforward vs. recurrent policies, reward shaping enabled and disabled, etc.), with full agent specifications and payoffs detailed by Liu et al.⁸⁸. The spectral response graph (Figure 6k) reveals two outlier agents: a strictly dominated agent (node in the top-right), and a strong (yet not strictly dominant) agent (node in the top-left). Several agents here are clustered pairwise, revealing their closely-related interactions with respect to the other agents; such information could,

for example, be used to discard or fuse such redundant agents during training to save computational costs.

Consider next a significantly larger-scale empirical game, consisting of 888 StarCraft II agents from the AlphaStar Final league of Vinyals et al.¹⁶³. StarCraft II is a notable example, involving a choice of 3 races per player and realtime gameplay, making a wide array of behaviors possible in the game itself. The empirical game considered is visualized in Figure 7a, and is representative of a large number of agents with varying skill levels. Despite its size, spectral analysis of this empirical game reveals that several key subsets of closely-performing agents exist here, illustrated in Figure 7d. Closer inspection of the agents used to construct this empirical payoff table reveals the following insights, with agent types corresponding to those detailed in Vinyals et al.¹⁶³: i) the blue, orange, and green clusters are composed of agents in the initial phases of training, which are generally weakest (as observed in Figure 7d, and also visible as the narrow band of low payoffs in the top of Figure 7a); ii) the red cluster consists primarily of various, specialized exploiter agents; iii) the purple and brown clusters are primarily composed of the league exploiters and main agents, with the latter being generally higher strength than the former. To further ascertain the relationships between only the strongest agents, we remove the three clusters corresponding to the weakest agents, repeating the analysis in Figure 7 (bottom row). Here, we observe the presence of a series of progressively stronger agents (bottom nodes in Figure 7h), as well as a single outlier agent which quite clearly bests several of these clusters (top node of Figure 7h).

An important caveat, as this stage, is that the agents in AlphaGo, MuJoCo soccer, and AlphaStar above were trained to maximize performance, rather than to explicitly reveal insights into their respective underlying games of Go, soccer, and StarCraft II. Thus, this analysis focused on characterizing relationships between the agents from the Policy Problem perspective, rather than the underlying games themselves, which provide insights into the interestingness of the game (Problem Problem). This latter investigation would require a significantly larger population of agents, which cover the policy space of the underlying game effectively, as exemplified next.

Naturally, characterization of the underlying game can be achieved in games small enough where all possible policies can be explicitly compared against one another. For instance, consider Blotto(τ, ρ), a zero-sum two-player game wherein each player has τ tokens that they can distribute amongst ρ regions¹⁴. In each region, each player with the most tokens wins (see Tuyls et al.¹⁵⁵ for additional details). In the variant we analyze here, each player receives a payoff of +1, 0, and -1 per region respectively won, drawn, and lost. The permutations of each player's allocated tokens, in turn, induce strong cyclical relations between the possible policies in the game. While the strategy space for this game is of size $\binom{\tau+\rho-1}{\rho-1}$, payoffs matrices can be fully specified for small instances, as shown for Blotto(5,3) and Blotto(10,3) in Figure 8 (first and second row, respectively). Despite the differences in strategy space sizes in these particular instances of Blotto, the contracted response graphs in Figures 8d and 8h capture the cyclical relations underlying both instances, revealing a remarkably similar structure.

For larger games, the cardinality of the pure policy space typically makes it infeasible to fully enumerate policies and construct a complete empirical payoff table, despite the pure policy space being finite in size. For example, even in games such as Tic-Tac-Toe, while the number of unique board configurations is reasonably small ($9! = 362880$), the number of pure, behaviorally unique policies is enormous ($\approx 10^{567}$, see Czarnecki et al.³³, Section J for details). Thus, coming up with a principled definition of a scheme for sampling a relevant set of policies summarizing the strategic interactions possible within large games remains an important open problem. In these instances, we rely on sampling policies in a manner that captures a set of representative policies, i.e., a set of policies of varying skill levels, which approximately capture variations of strategic interactions possible in the underlying game. The policy sampling approach we use is motivated with the above discussions and open question in mind, in that it samples a set of policies with varying skill levels, leading to a diverse set of potential transitive and intransitive interactions between them.

Specifically, we use the policy sampling procedure proposed by Czarnecki et al.³³, which also seeks a set of representative policies for a given game. The details of this procedure are provided in Supplementary Methods 1, and at a high level involve three phases: i) using a combination of tree search algorithms, Alpha-Beta¹⁰¹ and Monte Carlo Tree Search³¹, with varying tree depth limits for the former and varying number of simulations allotted to the latter, thus yielding policies of varying transitive strengths; ii) using a range of random seeds in each instantiation of the above algorithms, thus producing a range of policies for each level of transitive strength; iii) repeating the same procedure with negated game payoffs, thus also covering the space of policies that actively seek to lose the original game. While this sampling procedure is a heuristic,

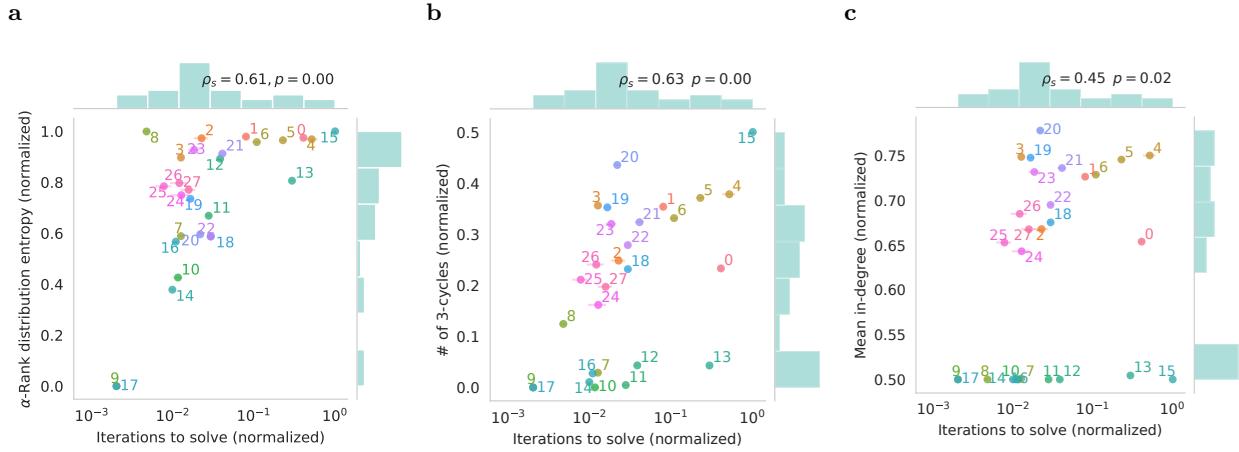
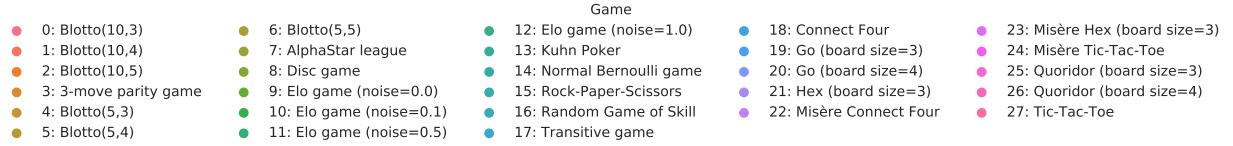


Figure 9: Response graph complexity vs. computational complexity of solving associated games. Each figure plots a respective measure of graph complexity against the normalized number of iterations needed to solve the associated game via the Double Oracle algorithm (with normalization done with respect to the total number of strategies in each underlying game). The Spearman correlation coefficient, ρ_s , is shown in each figure (with the reported two-sided p-value rounded to two decimals).

it produces a representative set of policies with varying degrees of transitive and intransitive relations, and thus provides an approximation of the underlying game that can be feasibly analyzed.

Let us revisit the example of Go, constructing our empirical game using the above policy sampling scheme, rather than the AlphaGo agents used earlier. We analyze a variant of the game with board size 3×3 , as shown in Figure 8 (third row). Notably, the contracted response graph (Figure 8l) reveals the presence of a strongly-cyclical structure in the underlying game, in contrast to the AlphaGo empirical game (Figure 6). Moreover, the presence of a reasonably strong agent (visible in the top of the contracted response graph) becomes evident here, though this agent also shares cyclical relations with several sets of other agents. Overall, this analysis exemplifies the distinction between analyzing an underlying game (e.g., Go) versus analyzing the agent training process (e.g., AlphaGo). Investigation of links between these two lines of analysis, we believe, makes for an interesting avenue for future work.

Linking response graph and computational complexity

A question that naturally arises is whether certain measures over response graphs are correlated with the computational complexity of solving their associated games. We investigate these potential correlations here, while noting that these results are not intended to propose that a specific definition of computational complexity (e.g., with respect to Nash) is explicitly useful for defining a topology / classification over games. In Figure 9, we compare several response graph complexity measures against the number of iterations needed to solve a large collection of games using the Double Oracle algorithm⁹³. The results here consider specifically the α -Rank entropy, number of 3-cycles, and mean in-degree (with details in Methods, and results for additional measures included in Supplementary Note 2). As in earlier experiments, solution of small-scale games is computed using payoffs over full enumeration of pure policies, whereas that of larger games is done using the empirical games over sampled policies. Each graph complexity measure reported is normalized with respect to the maximum measure possible in a graph of the same size, and the number of iterations to

solve is normalized with respect to the number of strategies in the respective game. Thus, for each game, the normalized number of iterations to solve provides a measure of its relative computational complexity compared to games with the same strategy space size (for completeness, we include experiments testing the effects of normalization on these results in Supplementary Note 2).

Several trends can be observed in these results. First, the entropy of the α -Rank distribution associated with each game correlates well with its computational complexity (see Spearman's correlation coefficient ρ_s in the top-right of Figure 9a). This matches intuition, as higher entropy α -Rank distributions indicate a larger support over the strategy space (i.e., strong strategies, with non-zero α -Rank mass), thus requiring additional iterations to solve. Moreover, the number of 3-cycles in the response graph also correlates well with computational complexity, again matching intuition as the intransitivities introduced by cycles typically make it more difficult to traverse the strategy space⁶. Finally, the mean in-degree over all response graph nodes correlates less so with computational complexity (though degree-based measures still serve a useful role in characterizing and distinguishing graphs of differing sizes⁹). Overall, these results indicate that response graph complexity provides a useful means of quantifying the computational complexity of games.

The Landscape of Games

The results, thus far, have demonstrated that graph-theoretic analysis can simplify games (via spectral clustering), uncover their topological structure (e.g., transitive structure of the AlphaGo empirical game), and yield measures correlated to the computational complexity of solving these games. Overall, it is evident that the perspective offered by graph theory yields a useful characterization of games across multiple fronts. Given this insight, we next consider whether this characterization can be used to compare a widely-diverse set of games.

To achieve this, we construct empirical payoff tables for a suite of games, using the policy sampling scheme described earlier for the larger instances (also see Supplementary Methods 1 for full details, including description of the games considered and analysis of the sensitivity of these results to the choice of empirical policies and mixtures thereof). For each game, we compute the response graphs and several associated local and global complexity measures (e.g., α -Rank distribution entropy, number of 3-cycles, node-wise in- and out-degree statistics, and several other measures detailed in the Methods section), which constitute a feature vector capturing properties of interest. Finally, a principal component analysis of these features yields the low-dimensional visualization of the landscape of games considered, shown in Figure 1.

We make several key insights given this empirical landscape of games. Notably, variations of games with related rules are well-clustered together, indicating strong similarity despite the widely-varying sizes of their policy spaces and empirical games used to construct them; specifically, all considered instances of Blotto cluster together, with empirical game sizes ranging from 20×20 for Blotto(5,3) to 1000×1000 for Blotto(10,5). Moreover, games with strong transitive components (e.g., variations of Elo games, AlphaStar League, Random Game of Skill, and Normal Bernoulli Game) are also notably separated from strongly cyclical games (Rock–Paper–Scissors, Disc game, and Blotto variations). Closely-related real-world games (i.e., games often played by humans in the real world, such as Hex, Tic-Tac-Toe, Connect Four, and each of their respective Misère counterparts wherein players seek to lose) are also well-clustered. Crucially, the strong alignment of this analysis with intuitions of the similarity of certain classes of games serves as an important validation of the graph-based analysis technique proposed in this work. Additionally, the analysis and corresponding landscape of games make clear that several games of interest for AI seem well-clustered together, which also holds for less interesting games (e.g., Transitive and Elo games).

We note that the overall idea of generating such a landscape over games ties closely with prior works on taxonomization of multiplayer games^{19,118}. Moreover, 2D visualization of the expressivity (i.e., style and diversity) and the overall space of procedurally-generated games features have been also investigated in closely related work^{131,140}. A recent line of related inquiry also investigates the automatic identification, and subsequent visualization of core mechanics in single-player games²⁴. Overall, we believe this type of investigation can be considered a method to taxonomize which future multiplayer games may be interesting, and which ones less so to train AI agents on.

While the primary focus of this paper is to establish a means of topologically studying games (and their similarities), a natural artifact of such a methodology is that it can enable investigation of interesting and non-interesting classes of games. There exist numerous perspectives on what may make a game interesting,

which varies as a function of the field of study or problem being solved. These include (overlapping) paradigms that consider interestingness from: a human-centric perspective (e.g., level of social interactivity, cognitive learning and problem solving, enjoyment, adrenaline, inherent challenge, aesthetics, story-telling in the game, etc.)^{39,57,76,82,114,165}; a curriculum learning perspective (e.g., games or tasks that provide enough learning signal to the human or artificial learner)³; a procedural content generation or optimization perspective (in some instances with a focus on General Game Playing)⁵⁰ that use a variety of fitness measures to generate new game instances (e.g., either direct measures related to the game structure, or indirect measures such as player win-rates)^{18,102,117,148,167}; and game-theoretic, multiplayer, or player-vs.-task notions (e.g., game balance, level of competition, social equality or welfare of the optimal game solution, etc.)^{21,68,178}. Overall, it is complex (and, arguably, not very useful) to introduce a unifying definition of interestingness that covers all of the above perspectives.

Thus, we focus here on a specialized notion of interestingness from the perspective of AI training, linked to the work of Czarnecki et al.³³. As mentioned earlier, Czarnecki et al.³³ introduce the notion of Games of Skill, which are of interest, in the sense of AI training, due to two axes of interactions between agents: a transitive axis enabling progression in terms of relative strength or skill, and a radial axis representing diverse intransitive/cyclical interactions between strategies of similar strength levels. The overall outlook provided by the above paper is that in these games of interest there exist many average-strength strategies with intransitive relations among them, whereas the level of intransitivity decreases as transitive strength moves towards an extremum (either very low, or very high strength); the topology of strategies in a Game of Skill is, thus, noted to resemble a spinning top.

Through the spinning top paradigm, Czarnecki et al.³³ identifies several real-world games (e.g., Hex, Tic-Tac-Toe, Connect-Four, etc.) as Games of Skill. Notably, the lower left cluster of games in Figure 1 highlights precisely these games. Interestingly, while the Random Game of Skill, AlphaStar League, and Elo game (noise=0.5) are also noted as Games of Skill in Czarnecki et al.³³, they are found in a distinct cluster in our landscape. On closer inspection of these payoffs for this trio of games, there exists a strong correlation between the number of intransitive relations and the transitive strength of strategies, in contrast to other Games of Skill such as Hex. Our landscape also seems to highlight non-interesting games. Specifically, variations of Blotto, Rock-Paper-Scissors, and the Disc Game are noted to not be Games of Skill in Czarnecki et al.³³, and are also found to be in distinct clusters in Figure 1.

Overall, these results highlight how topological analysis of multiplayer games can be used to not only study individual games, but also identify clusters of related (potentially interesting) games. For completeness, we also conduct additional studies in Supplementary Note 2, which compare our taxonomization of 2×2 normal-form games (and clustering into potential classes of interest) to that of Bruns¹⁹.

The Problem Problem and Procedural Game Structure Generation

Having now established various graph-theoretic tools for characterizing games of interest, we revisit the so-called Problem Problem, which targets automatic generation of interesting environments. Here we focus on the question of how we can leverage the topology discovered by our method to procedurally generate collections of new games (which can be subsequently analyzed, used for training, characterized as interesting or not per previous discussions, and so on).

Full details of our game generation procedure are provided in the Methods Section. At a high level, we establish the feedback loop visualized in Figure 3, enabling automatic generation of games as driven by our graph-based analytical workflow. We generate the payoff structure of a game (i.e., as opposed to the raw underlying game rules, e.g., as done by Browne and Maire¹⁸). Thus, the generated payoffs can be considered either direct representations of normal form games, or empirical games indirectly representing underlying games with complex rules. At a high level, given a parameterization of a generated game that specifies an associated payoff tensor, we synthesize its response graph and associated measures of interest (as done when generating the earlier landscape of games). We use the multidimensional Elo parameterization for generating payoffs, due to its inherent ability to specify complex transitive and intransitive games⁵. We then specify an objective function of interest to optimize over these graph-based measures. As only the evaluations of such graph-based measures (rather than their gradients) are typically available, we use a gradient-free approach to iteratively generate games optimizing these measures (CMA-ES⁵⁵ is used in our experiments). The overall generation procedure used can be classified as a Search-Based Procedural Content Generation technique

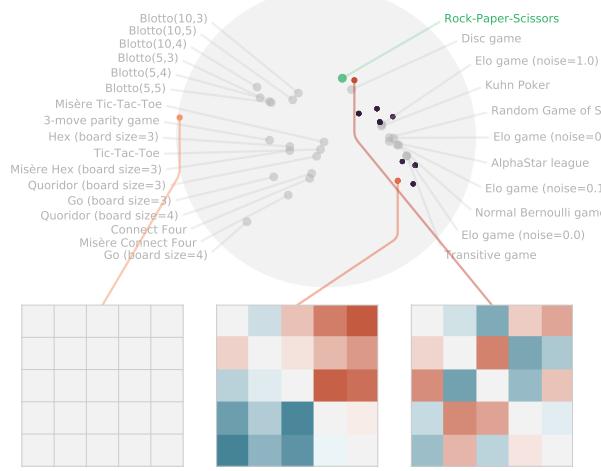
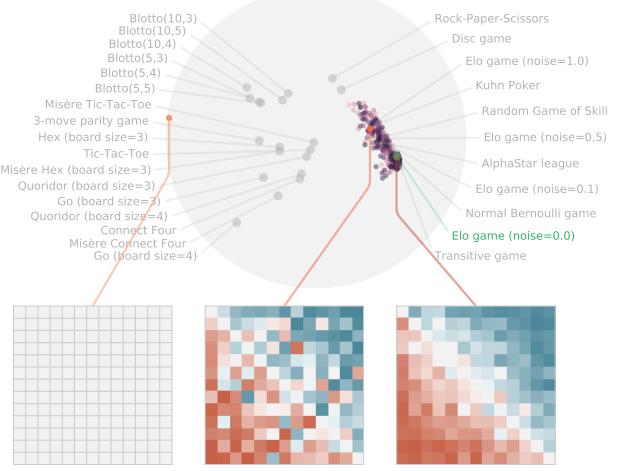
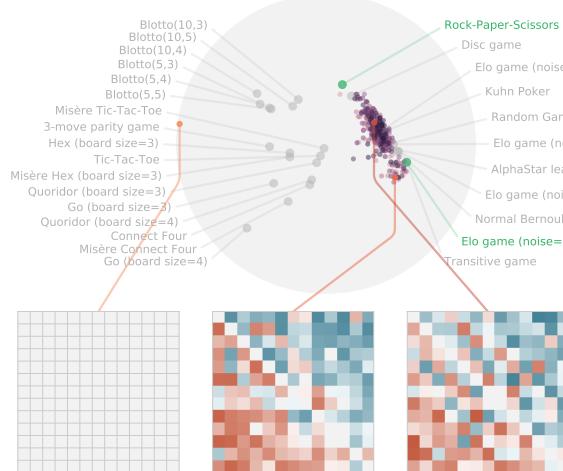
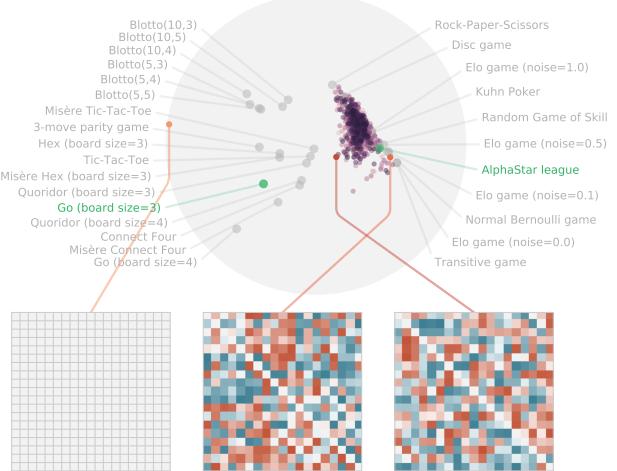
a**b****c****d**

Figure 10: Visualization of procedural game structure generation projected in the games landscape. Each figure visualizes the generation of a game of specified size, which targets a pre-defined game (or mixture of games) of a different size. The three payoffs in each respective figure, from left to right, correspond to the initial procedural game parameters, intermediate parameters, and final optimized parameters. **a** 5 × 5 generated game with the target game set to Rock–Paper–Scissors (3×3). **b** 13 × 13 generated game with the target game set to Elo (1000×1000). **c** 13 × 13 generated game with the target game set to the mixture of Rock–Paper–Scissors (3×3) and Elo game (1000×1000). **d** 19 × 19 generated game with the target game set to the mixture of AlphaStar League and Go (board size=3). Strategies are sorted by mean payoffs in **b** and **c** to more easily identify transitive structures expected from an Elo game.

(SBPCG)¹⁴⁸. Specifically, in accordance to the taxonomy defined by Togelius et al.¹⁴⁸, our work uses a direct encoding representation of the game (as the generated payoffs are represented as real-valued vectors of mElo parameters), with a theory-driven direct evaluation used for quantifying the game fitness/quality (as we rely on graph theory to derive the game features of interest, then directly minimize distances of principal components of generated and target games).

Naturally, we can maximize any individual game complexity measures, or a combination thereof, directly (e.g., entropy of the α -Rank distribution, number of 3-cycles, etc.). More interestingly, however, we can leverage our low-dimensional landscape of games to directly drive the generation of new games towards existing ones with properties of interest. Consider the instance of game generation shown in Figure 10a, which shows an overview of the above pipeline generating a 5×5 game minimizing Euclidean distance (within the low-dimensional complexity landscape) to the standard 3×3 Rock–Paper–Scissors game. Each point on this plot corresponds to a generated game instance. The payoffs visualized, from left to right, respectively correspond to the initial procedural game parameters (which specify a game with constant payoffs), intermediate parameters, and final optimized parameters; projections of the corresponding games within the games landscape are also indicated, with the targeted game of interest (Rock–Paper–Scissors here) highlighted in green. Notably, the final optimized game exactly captures the underlying rules that specify a general-size Rock–Paper–Scissors game, in that each strategy beats as many other strategies as it loses to. In Figure 10b, we consider a larger 13×13 generated game, which seeks to minimize distance to a 1000×1000 Elo game (which is transitive in structure, as in our earlier motivating example in Figure 2a). Once again, the generated game captures the transitive structure associated with Elo games.

Next, we consider generation of games that exhibit properties of mixtures of several target games. For example, consider what happens if 3×3 Rock–Paper–Scissors were to be combined with the 1000×1000 Elo game above; one might expect a mixture of transitive and cyclical properties in the payoffs, though the means of generating such mixed payoffs directly is not obvious due to the inherent differences in sizes of the targeted games. Using our workflow, which conducts this optimization in the low-dimensional graph-based landscape, we demonstrate a sequence of generated games targeting exactly this mixture in Figure 10c. Here, the game generation objective is to minimize Euclidean distance to the mixed principal components of the two target games (weighted equally). The payoffs of the final generated game exhibit exactly the properties intuited above, with predominantly positive (blue) upper-triangle of payoff entries establishing a transitive structure, and the more sporadic positive entries in the lower-triangle establishing cycles.

Naturally, this approach opens the door to an important avenue for further investigation, targeting generation of yet more interesting combinations of games of different sizes and rule-sets (e.g., as in Figure 10d, which generates games targeting a mixture of Go (board size=3) and the AlphaStar League), and subsequent training of AI agents using such a curriculum of generated games. Moreover, we can observe interesting trends when analyzing the Nash equilibria associated with the class of normal-form games considered here (as detailed in Supplementary Note 2). Overall, these examples illustrate a key benefit of the proposed graph-theoretic measures in that it captures the underlying structure of various classes of games. The characterization of games achieved by our approach directly enables the navigation of the associated games landscape to generate never-before-seen instances of games with fundamentally related structure.

Discussion

In 1965, mathematician Alexander Kronrod stated that “chess is the Drosophila of artificial intelligence”⁹², referring to the genus of flies used extensively for genetics research. This parallel drawn to biology invites the question of whether a family, order, or, more concretely, shared structures linking various games can be identified. Our work demonstrated a means of revealing this topological structure, extending beyond related works investigating this question for small classes of games (e.g., 2×2 games^{32,115,118}). We believe that such a topological landscape of games can help to identify and generate related games of interest for AI agents to tackle, as targeted by the Problem Problem, hopefully significantly extending the reach of AI system capabilities. As such, this paper presented a comprehensive study of games under the lens of graph theory and empirical game theory, operating on the response graph of any game of interest. The proposed approach applies to general-sum, many-player games, enabling richer understanding of the inherent relationships between strategies (or agents), contraction to a representative (and smaller) underlying game,

and identification of a game’s inherent topology. We highlighted insights offered by this approach when applied to a large suite of games, including canonical games, empirical games consisting of trained agent policies, and real-world games consisting of representative sampled policies, extending well beyond typical characterizations of games using raw payoff visualizations, cardinal measures such as strategy or game tree sizes, or strategy rankings. We demonstrated that complexity measures associated with the response graphs analyzed correlate well to the computational complexity of solving these games, and importantly enable the visualization of the landscape of games in relation to one another (as in Figure 1). The games landscape exposed here was then leveraged to procedurally generate games, providing a principled means of better understanding and facilitating the solution of the so-called Problem Problem.

While the classes of games generated in this paper were restricted to the normal-form (e.g., generalized variants of Rock–Paper–Scissors), they served as an important validation of the proposed approach. Specifically, this work provides a foundational layer for generating games that are of interest in a richer context of domains. In contrast to some of the prior works in taxonomization of games (e.g., that of Bruns¹⁹, Liebrand⁸⁷, Rapoport and Guyer¹¹⁵), a key strength of our approach is that it does not rely on human expertise, or manual isolation of patterns in payoffs or equilibria to compute a taxonomy over games. We demonstrate an example of this in Supplementary Note 2, by highlighting similarities and differences of our taxonomization with those of Bruns¹⁹ over a set of 144 2 × 2 games. Importantly, these comparisons highlight that our response graph-based approach does not preclude more classical equilibrium-centric analysis from being conducted following clustering, while also avoiding the need for a human-in-the-loop analysis of equilibria for classifying the games themselves.

While our graph-based game analysis approach (i.e., the spectral analysis and clustering technique) applies to general-sum, many-player games, the procedural game structure generation approach used in our experiments is limited to zero-sum games (due to our use of the mElo parameterization). However, any other general-sum payoff parameterization approaches (e.g., even direct generation of the payoff entries) can also be used to avoid the zero-sum constraint. The study of normal-form games continues to play a prominent role in the game theory and machine learning literature^{11,37,42,83,86,142,177} and as such the procedural generation of normal-form games can play an important role in the research community. An important line of future work will involve investigating means of generalizing this approach to generation of more complex classes of games. Specifically, one way to generate more complex underlying games would be to parameterize core mechanisms of such a games class (either explicitly, or via mechanism discovery, e.g., using a technique such as that described by Charity et al.²⁴). Subsequently, one could train AI agents on a population of such games, constructing a corresponding empirical game, and using the response graph-based techniques used here to analyze the space of such games (e.g., in a manner reminiscent of Wang et al.¹⁶⁷, though under our graph-theoretic lens). The connection to the SBPCG literature mentioned in The Problem Problem and Procedural Game Structure Generation section also offers an avenue of alternative investigations into game structure generation, as the variations of fitness measures and representations previously explored in that literature¹⁴⁸ may be considered in lieu of the approach used here.

Moreover, as the principal contribution of this paper was to establish a graph-theoretic approach for investigating the landscape of games, we focused our investigation on empirical analysis of a large suite of games. As such, for larger games, our analysis relied on sampling of a representative set of policies to characterize them. An important limitation here is that the empirical game-theoretic results are subject to the policies used to generate them. While our sensitivity analysis (presented in Supplementary Note 2) seems to indicate that the combination of our policy sampling scheme and analysis pipeline produce fairly robust results, this is an important factor to revisit in significantly larger games. Specifically, such a policy sampling scheme can be inherently expensive for extremely large games, making it important to further investigate alternative sampling schemes and associated sensitivities. Consideration of an expanded set of such policies (e.g., those that balance the odds for players by ensuring a near-equal win probability) and correlations between the empirical game complexity and the complexity of the underlying policy representations (e.g., deep versus shallow neural networks or, whenever possible, Boolean measures of strategic complexity^{46,126}) also seem interesting to investigate. Moreover, formal study of the propagation of empirical payoff variance to the topological analysis results is another avenue of future interest, potentially using techniques similar to Rowland et al.¹²¹. Another direction of research for future work is to analyze agent-vs.-task games (e.g., those considered in Balduzzi et al.⁵) from a graph-theoretic lens. Finally, our approach is general enough to be applicable to other areas in social and life sciences^{91,94,129,135,141}, characterized by complex ecologies

often involving a large number of strategies or traits. In particular, these processes may be modeled through the use of large-scale response graphs or invasion diagrams^{40,48,70,159,161}, whose overall complexity (and how it may vary with the inclusion or removal of species, strategies, and conflicts) is often hard to infer.

Overall, we believe that this work paves the way for related investigations of theoretical properties of graph-based games analysis, for further scientific progress on the Problem Problem and task theory, and further links to related works investigating the geometry and structure of games^{6,32,33,69,115,118}.

Methods

Games

Our work applies to K -player, general-sum games, wherein each player $k \in [K]$ has a finite set S^k of pure strategies. The space of pure strategy profiles is denoted $S = \prod_k S^k$, where a specific pure strategy profile instance is denoted $s = (s^1, \dots, s^K) \in S$. For a given profile $s \in S$, the payoffs vector is denoted $\mathbf{M}(s) = (\mathbf{M}^1(s), \dots, \mathbf{M}^K(s)) \in \mathbb{R}^K$, where $\mathbf{M}^k(s)$ is the payoff for each player $k \in [k]$. We denote by s^{-k} the profile of strategies used by all but the k -th player. A game is said to be zero-sum if $\sum_k \mathbf{M}^k(s) = 0$ for all $s \in S$. A game is said to be symmetric if all players have the same strategy set, and $\mathbf{M}^k(s^1, \dots, s^K) = \mathbf{M}^{\rho(k)}(s^{\rho(1)}, \dots, s^{\rho(K)})$ for all permutations ρ , strategy profiles s , and player indices $k \in [K]$.

Empirical games

For the real-world games considered (e.g., Go, Tic–Tac–Toe, etc.), we conduct our analysis using an empirical game-theoretic approach^{112,156,166,174,175}. Specifically, rather than consider the space of all pure strategies in the game (which can be enormous, even in the case of, e.g., Tic–Tac–Toe), we construct an empirical game over meta-strategies, which can be considered higher-level strategies over atomic actions. In empirical games, a meta-strategy s^k for each player k corresponds to a sampled policy (e.g., in the case of our real-world games examples), or an AI agent (e.g., in our study of AlphaGo, where each meta-strategy was a specific variant of AlphaGo). Empirical game payoffs are calculated according to the win/loss ratio of these meta-strategies against one another, over many trials of the underlying games. From a practical perspective, game-theoretic analysis applies to empirical games (over agents) in the same manner as standard games (over strategies); thus, we consider strategies and agents as synonymous in this work. Overall, empirical games provide a useful abstraction of the underlying game that enables the study of significantly larger and more complex interactions.

Finite population models and α -Rank

In game theory^{52,67,98,123,172}, one often seeks algorithms or models for evaluating and training strategies with respect to one another (i.e., models that produce a score or ranking over strategy profiles, or an equilibrium over them). As a specific example, the Double Oracle algorithm⁹³, which is used to quantify the computational complexity of solving games in some of our experiments, converges to Nash equilibria, albeit only in two-player zero-sum games. More recently, a line of research has introduced and applied the α -Rank algorithm^{97,106,121} for evaluation of strategies in general-sum, n -player many-strategy games. α -Rank leverages notions from stochastic evolutionary dynamics in finite populations^{70,103,104,144,151,152} in the limit of rare mutations^{48,161,162}, which are subsequently analyzed to produce these scalar ratings (one per strategy or agent). At a high level, α -Rank models the probability of a population transitioning from a given strategy to a new strategy, by considering the additional payoff the population would receive via such a deviation. These evolutionary relations are considered between all strategies in the game, and are summarized in its so-called response graph. α -Rank then uses the stationary distribution over this response graph to quantify the long-term propensity of playing each of the strategies, assigning a scalar score to each.

Overall, α -Rank yields a useful representation of the limiting behaviors of the players, providing a summary of the characteristics of the underlying game—albeit a 1-dimensional one (a scalar rating per strategy profile). In our work, we exploit the higher-dimensional structural properties of the α -Rank response graph, to make more informed characterizations of the underlying game, rather than compute scalar rankings.

Response graphs

The α -Rank response graph¹⁰⁶ provides the mathematical model that underpins our analysis. It constitutes an analogue (yet, not equivalent) model of the invasion graphs used to describe the evolution dynamics in finite populations in the limit when mutations are rare (see, e.g., Fudenberg and Imhof⁴⁸, Hauert et al.⁵⁹, Van Segbroeck et al.¹⁵⁹, Vasconcelos et al.¹⁶¹). In this small-mutation approximation, directed edges stand for the fixation probability¹⁰⁴ of a single mutant in a monomorphic population of resident individuals (the vertices), such that all transitions are computed through processes involving only two strategies at a time. Here, we use a similar approach. Let us consider a pure strategy profile $s = (s^1, \dots, s^K)$. Consider a unilateral deviation (corresponding to a mutation) of player k from playing $s^k \in S^k$ to a new strategy $\sigma^k \in S^k$, thus resulting in a new profile $\sigma = (\sigma^k, s^{-k})$. The response graph associated with the game considers all such deviations, defining transition probabilities between all pairs of strategy profiles involving a unilateral deviation. Specifically, let $\mathbf{E}_{s,\sigma}$ denote the transition probability from s to σ (where the latter involves a unilateral deviation), defined as

$$\mathbf{E}_{s,\sigma} = \begin{cases} \eta \frac{1 - \exp(-\alpha(\mathbf{M}^k(\sigma) - \mathbf{M}^k(s)))}{1 - \exp(-\alpha m(\mathbf{M}^k(\sigma) - \mathbf{M}^k(s)))} & \text{if } \mathbf{M}^k(\sigma) \neq \mathbf{M}^k(s) \\ \frac{\eta}{m} & \text{otherwise,} \end{cases} \quad (1)$$

where η is a normalizing factor denoting the reciprocal of the total number of unilateral deviations from a given strategy profile, i.e., $\eta = (\sum_{l=1}^K (|S^l| - 1))^{-1}$. Furthermore, $\alpha \geq 0$ and $m \in \mathbb{N}$ are parameters of the underlying evolutionary model considered and denote, respectively, to the so-called selection pressure and population size.

To further simplify the model and avoid sweeps over these parameters, we consider here the limit of infinite- α introduced by Omidshafiei et al.¹⁰⁶, which specifies transitions from lower-payoff profiles to higher-payoff ones with probability $\eta(1 - \varepsilon)$, the reverse transition with probability $\eta\varepsilon$, and transition between strategies of equal payoff with probability $\eta/2$, where $0 < \varepsilon \ll 1$ is a small perturbation factor. We use $\varepsilon = 1e-10$ in our experiments, and found low sensitivity of results to this choice given a sufficiently small value. For further theoretical exposition of α -Rank under this infinite- α regime, see Rowland et al.¹²¹. Given the pairwise strategy transitions defined as such, the self-transition probability of s is subsequently defined as,

$$\mathbf{E}_{s,s} = 1 - \sum_{\substack{k \in [K] \\ \sigma | \sigma^k \in S^k \setminus \{s^k\}}} \mathbf{E}_{s,\sigma}. \quad (2)$$

As mentioned earlier, if two strategy profiles s and σ do not correspond to a unilateral deviation (i.e., differ in more than one player's strategy), no transition occurs between them under this model (i.e., $\mathbf{E}_{s,\sigma} = 0$).

The transition structure above is informed by particular models in evolutionary dynamics as explained in detail in Omidshafiei et al.¹⁰⁶. The introduction of the perturbation term ε effectively ensures the ergodicity of the associated Markov chain with row-stochastic transition matrix \mathbf{E} . This transition structure then enables definition of the α -Rank response graph of a game.

Definition 1 (Response graph) *The response graph of a game is a weighted directed graph (digraph) $G = (S, \mathbf{E})$ where each node corresponds to a pure strategy profile $s \in S$, and each weighted edge $\mathbf{E}_{s,\sigma}$ quantifies the probability of transitioning from profile s to σ .*

For example, the response graph associated with a transitive game is visualized in Figure 3b, where each node corresponds to a strategy s , and directed edges indicate transition probabilities between nodes. Omidshafiei et al.¹⁰⁶ define α -Rank $\boldsymbol{\pi} \in \Delta^{|S|-1}$ as a probability distribution over the strategy profiles S , by ordering the masses of the stationary distribution of \mathbf{E} (i.e., solution of the eigenvalue problem $\boldsymbol{\pi}^T \mathbf{E} = \boldsymbol{\pi}^T$). Effectively, the α -Rank distribution quantifies the average amount of time spent by the players in each profile $s \in S$ under the associated discrete-time evolutionary population model⁴⁸. Our proposed methodology uses the α -Rank response graphs in a more refined manner, quantifying the structural properties defining the underlying game, as detailed in the workflow outlined in Figure 3 and, in more detail, below.

Spectral, clustered, and contracted response graphs

This section details the workflow used to for spectral analysis of games' response graphs (i.e., the steps visualized in Figure 3c to Figure 3e). Response graphs are processed in two stages: i) symmetrization (i.e., transformation of the directed response graphs to an associated undirected graph), and ii) subsequent spectral analysis. This two-phase approach is a standard technique for analysis of directed graphs, which has proved effective in a large body of prior works (see Malliaros and Vazirgiannis⁹⁰, Van Lierde¹⁵⁸ for comprehensive surveys). Additionally, spectral analysis of the response graph is closely-associated with the eigenvalue analysis required when solving for the α -Rank distribution, establishing a shared formalism of our techniques with those of prior works.

Let \mathbf{A} denote the adjacency matrix of the response graph G , where $\mathbf{A} = \mathbf{E}$ as G is a directed weighted graph. We seek a transformation such that response graph strategies with similar relationships to neighboring strategies tend to have higher adjacency with one another. Bibliometric symmetrization¹²⁷ provides a useful means to do so in application to directed graphs, whereby the symmetrized adjacency matrix is defined $\tilde{\mathbf{A}} = \mathbf{A}\mathbf{A}^T + \mathbf{A}^T\mathbf{A}$. Intuitively, in the first term, $\mathbf{A}\mathbf{A}^T$, the (s, σ) -th entry captures the weighted number of other strategies that both s and σ would deviate to in the response graph G ; the same entry in the second term, $\mathbf{A}^T\mathbf{A}$, captures the weighted number of other strategies that would deviate to both s and σ . Hence, this symmetrization captures the relationship of each pair of response graph nodes (s, σ) with respect to all other nodes, ensuring high values of weighted adjacency when these strategies have similar relational roles with respect to all other strategies in the game. More intuitively, this ensures that in games such as Redundant Rock–Paper–Scissors (see Figure 5, first row), sets of redundant strategies are considered to be highly adjacent to each other.

Following bibliometric symmetrization of the response graph, clustering proceeds as follows. Specifically, for any partitioning of the strategy profiles S into sets $S_1 \subset S$ and $\bar{S}_1 = S \setminus S_1$, define $w(S_1, \bar{S}_1) = \sum_{s \in S_1, \sigma \in \bar{S}_1} \mathbf{E}_{s,\sigma}$. Let the sets of disjoint strategy profiles $\{S_k\}_{k \in [K]}$ partition S (i.e., $\bigcup_{k \in [K]} S_k = S$). Define the K -cut of graph G under partitions $\{S_k\}_{k \in [K]}$ as

$$cut(\{S_k\}) = \sum_k w(S_k, \bar{S}_k), \quad (3)$$

which, roughly speaking, measures the connectedness of points in each cluster; i.e., a low cut indicates that points across distinct clusters are not well-connected. A standard technique for cluster analysis of graphs is to choose the set of K partitions, $\{S_k\}_{k \in [K]}$, which minimizes (3). In certain situations, balanced clusters (i.e., clusters with similar numbers of nodes) may be desirable; here, a more suitable metric is the so-called normalized K -cut, or $Ncut$, of graph G under partitions $\{S_k\}_{k \in [K]}$,

$$Ncut(\{S_k\}) = \sum_k \frac{w(S_k, \bar{S}_k)}{w(S_k, S)}. \quad (4)$$

Unfortunately, the minimization problem associated with (4) is NP-hard even when $K = 2$ (see Shi and Malik¹³⁴). A typical approach is to consider a spectral relaxation of this minimization problem, which corresponds to a generalized eigenvalue problem (i.e., efficiently solved via standard linear algebra); interested readers are referred to Shi and Malik¹³⁴, Van Lierde¹⁵⁸ for further exposition. Define the Laplacian matrix $\mathbf{L} = \mathbf{D} - \tilde{\mathbf{A}}$ (respectively, $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2}\tilde{\mathbf{A}}\mathbf{D}^{-1/2}$), where degree matrix \mathbf{D} has diagonal entries $D_{i,i} = \sum_j \tilde{\mathbf{A}}_{i,j}$, and zeroes elsewhere. Then the eigenvectors associated with the lowest nonzero eigenvalues of \mathbf{L} provide the desired spectral projection of the datapoints (i.e., spectral response graph), with the desired number of projection dimensions corresponding to the number of eigenvectors kept. We found that using the unnormalized graph Laplacian $\mathbf{L} = \mathbf{D} - \tilde{\mathbf{A}}$ yielded intuitive projections in our experiments, which we visualize 2-dimensionally in our results (e.g., see Figure 3c).

The relaxed clustering problem detailed above is subsequently solved by application of a standard clustering algorithm to the spectral-projected graph nodes. Specifically, we use agglomerative average-linkage clustering in our experiments (see Rokach and Maimon¹²⁰, Chapter 15 for details). For determining the appropriate number of clusters, we use the approach introduced by Pham et al.¹¹¹, which we found to yield more intuitive clusterings than the gap statistic¹⁴⁶ for the games considered.

Following computation of clustered response graphs (e.g., Figure 3d), we contract clustered nodes (summing edge probabilities accordingly), as in Figure 3e. Note that for clarity, our visualizations only show

edges corresponding to transitions from lower-payoff to higher-payoff strategies in the standard, spectral, and clustered response graphs, as these bear the majority of transition mass between nodes; reverse edges (from higher- to lower-payoff nodes) and self-transitions are not visually indicated, despite being used in the underlying spectral clustering. The exception is for contracted response graphs, where we do visualize weighted edges from higher- to lower-payoff nodes; this is due to the node contraction process potentially yielding edges with non-negligible weight in both directions.

Low-dimensional landscape & game generation

We next detail the approach used to compute the games landscape and to procedurally generate games, as respectively visualized in Figures 1 and 10.

To compute the low-dimensional games landscape, we use principal component analysis (PCA) of key features associated with the games' response graphs; we use a collection of features we found to correlate well with the underlying computational complexity of solving these games (as detailed in the Results section).

Specifically, using the response graph G of each game, we compute in- and out-degrees for all nodes, the entropy of the α -Rank distribution π , and the total number of 3-cycles. We normalize each of these measures as follows: dividing node-wise in- and out-degrees via the maximum possible degrees for a response graph of the same size; dividing the α -Rank distribution entropy via the entropy of the uniform distribution of the same size; finally, dividing the number of 3-cycles via the same measure for a fully connected directed graph.

We subsequently construct a feature vector consisting of the normalized α -Rank distribution entropy, the normalized number of 3-cycles, and statistics related to normalized in- and out-degrees. Specifically, we consider the mean, median, standard deviation, skew, and kurtosis of the in- and out-degrees across all response graph nodes, similar to the NetSimile⁹ approach, which characterized undirected graphs. This yields a feature vector of fixed size for all games. We subsequently conduct a PCA analysis of the resulting feature vectors, visualizing the landscape in Figure 1 via projection of the feature vectors onto the top two principal components, yielding a low-dimensional embedding \mathbf{v}_g for each game g .

Procedural game structure generation

The games generated in Figures 2 and 10 use the multidimensional Elo (mElo) parametric structure⁵, an extension of the classical Elo⁴⁴ rating system used in Chess and other games. In mElo games, each strategy i is characterized by two sets of parameters: i) a scalar rating $r_i \in \mathbb{R}$ capturing the strategy's transitive strength, and ii) a $2k$ -dimensional vector \mathbf{c}_i capturing the strategy's intransitive relations to other strategies. The payoff a strategy i receives when played against a strategy j in a mElo game is defined by $M(i, j) = \sigma(r_i - r_j + \mathbf{c}_i^T \boldsymbol{\Omega} \mathbf{c}_j)$ where $\sigma(z) = (1 + \exp(-z))^{-1}$, $\boldsymbol{\Omega} = \sum_{i=1}^k (\mathbf{e}_{2i-1} \mathbf{e}_{2i}^T - \mathbf{e}_{2i} \mathbf{e}_{2i-1}^T)$, and \mathbf{e}_i is the unit vector with coordinate i equal to 1. This parametric structure is particularly useful as it enables definition of a wide array of games, ranging from those with fully transitive strategic interactions (e.g., those with a single dominant strategy, as visualized in Figure 2a), to intransitive interactions (e.g., those with cyclical relations, as visualized in Figure 2d), to a mix thereof.

The procedural game structure generation visualized in Figure 10 is conducted as follows. First, we compute the low-dimensional game embeddings for the collection of games of interest, as detailed above. Next, for an initially randomly-generated mElo game of the specified size and rank k , we concatenate the associated mElo parameters r_i and \mathbf{c}_i for all strategies, yielding a vector of length $|S|(1 + 2k)$ fully parameterizing the mElo game, and constituting the decision variables of the optimization problem used to generate new games. We used a rank 5 mElo parameterization for all game generation experiments. For any such setting of mElo parameters, we compute the associated mElo payoff matrix M , then the associated response graph and features, and finally project these features onto the principal components previously computed for the collection of games of interest, yielding the projected mElo components \mathbf{v}_{mElo} .

Subsequently, given a game g of interest that we would like to structurally mimic via our generated mElo game, we use a gradient-free optimizer, CMA-ES⁵⁵, to minimize $\|\mathbf{v}_{mElo} - \mathbf{v}_g\|_2^2$ by appropriately setting the mElo parameters. For targeting mixtures of games (e.g., as in Figures 10c and 10d), we simply use a weighted mixture of their principal components \mathbf{v}_g (with equal weights used in our experiments). We found the open-source implementation of CMA-ES⁵⁶ to converge to suitable parameters within 20 iterations for

all experiments, with the exception of the larger game generation results visualized in Figure 10d, which required 40 iterations.

Statistics

To generate the distributions of Double Oracle iterations needed to solve the motivating examples (Figure 2), we used 20 generated games per class (transitive, cyclical, random), showing four examples of each in Figures 2a, 2d and 2g. For each of these 20 games, we used 10 random initializations of the Double Oracle algorithm, reporting the full distribution of iterations. To generate the complexity results in Figure 9, we likewise used 10 random initializations of Double Oracle per game, with standard deviations shown in the scatter plots (which may require zooming in). For the Spearman correlation coefficients shown in each of Figures 9a to 9c, the reported p-value is two-sided and rounded to two decimals.

Data Availability

We use OpenSpiel⁸¹ (https://github.com/deepmind/open_spiel) as the backend providing many of the games and associated payoff datasets studied here (see Supplementary Methods 1 for details). Payoff datasets for empirical games in the literature are referenced in the main text.

Code Availability

We use OpenSpiel⁸¹ (https://github.com/deepmind/open_spiel) for the implementation of α -Rank and Double Oracle.

References

- [1] Krzysztof R Apt and Sunil Simon. A classification of weakly acyclic games. *Theo. Decis.*, 78(4):501–524, 2015.
- [2] Ayala Arad and Ariel Rubinstein. The 11-20 money request game: A level-k reasoning study. *Am. Econ. Rev.*, 102(7):3561–73, 2012.
- [3] Bowen Baker, Ingmar Kanitscheider, Todor Markov, Yi Wu, Glenn Powell, Bob McGrew, and Igor Mordatch. Emergent tool use from multi-agent autocurricula. In *Proc. Int. Conf. Learn. Represent.*, 2020. URL <https://openreview.net/forum?id=SkxpxJBKwS>.
- [4] David Balduzzi, Sébastien Racanière, James Martens, Jakob Foerster, Karl Tuyls, and Thore Graepel. The mechanics of n -player differentiable games. In *Proc. Int. Conf. Mach. Learn.*, 2018.
- [5] David Balduzzi, Karl Tuyls, Julien Perolat, and Thore Graepel. Re-evaluating evaluation. In *Proc. Neural Inf. Process. Syst.*, 2018.
- [6] David Balduzzi, Marta Garnelo, Yoram Bachrach, Wojciech M Czarnecki, Julien Perolat, Max Jaderberg, and Thore Graepel. Open-ended learning in symmetric zero-sum games. In *Proc. Int. Conf. Mach. Learn.*, 2019.
- [7] Marc G. Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The Arcade Learning Environment: An evaluation platform for general agents. *J. Artif. Intel. Res.*, 47:253–279, 2013.
- [8] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proc. Int. Conf. Mach. Learn.*, 2009.
- [9] Michele Berlingero, Danai Koutra, Tina Eliassi-Rad, and Christos Faloutsos. NetSimile: A scalable approach to size-independent network similarity. *Preprint at arXiv:1209.2684*, 2012.
- [10] Jordi E Bieger and Kristinn R Thórisson. Task analysis for teaching cumulative learners. In *Proc. Int. Conf. on Artif Gen. Intel.*, 2018.
- [11] Daan Bloembergen, Karl Tuyls, Daniel Hennes, and Michael Kaisers. Evolutionary dynamics of multi-agent learning: A survey. *J. Artif. Intel. Res.*, 53:659–697, 2015.
- [12] Stefano Boccaletti, Vito Latora, Yamir Moreno, Martin Chavez, and D-U Hwang. Complex networks: Structure and dynamics. *Phys. rep.*, 424(4-5):175–308, 2006.
- [13] Danail D Bonchev and Dennis Rouvray. *Complexity in chemistry, biology, and ecology*. Springer Science & Business Media, 2007.
- [14] Emile Borel. La théorie du jeu et les équations intégrales à noyau symétrique. *Comp. Rend. de l'Acad. Sci.*, 173(1304-1308):58, 1921.
- [15] Branislav Bosansky, Albert Xin Jiang, Milind Tambe, and Christopher Kiekintveld. Combining compact representation and incremental generation in large games with sequential strategies. In *Proc. AAAI Conf. on Artif. Intel.*, 2015.
- [16] Michael Bowling, Neil Burch, Michael Johanson, and Oskari Tammelin. Heads-up limit hold’em poker is solved. *Science*, 347(6218):145–149, 2015.
- [17] David Braben and Ian Bell. Elite. *Firebird, Acornsoft and Imagineer*, 1984.
- [18] Cameron Browne and Frederic Maire. Evolutionary game design. *IEEE T. Comp. Intel. AI*, 2(1):1–16, 2010.
- [19] Bryan Randolph Bruns. Names for games: Locating 2×2 games. *Games*, 6(4):495–520, 2015.

- [20] Neil Burch, Michael Johanson, and Michael Bowling. Solving imperfect information games using decomposition. In *Proc. AAAI Conf. on Artif. Intel.*, 2014.
- [21] Andrew Byde. Applying evolutionary game theory to auction mechanism design. In *Proc. EEE Int. Conf. on E-Comm.* IEEE, 2003.
- [22] Elizabeth Camilleri, Georgios N Yannakakis, and Alexiee Dingli. Platformer level design for player believability. In *Proc. IEEE Conf. Compu. Intel.*, 2016.
- [23] Murray Campbell, A Joseph Hoane Jr., and Feng-hsiung Hsu. Deep Blue. *Artif. Intel.*, 134(1-2):57–83, 2002.
- [24] Megan Charity, Michael Cerny Green, Ahmed Khalifa, and Julian Togelius. Mech-elites: Illuminating the mechanic space of gvgai. *Preprint at arXiv:2002.04733*, 2020.
- [25] Xi Chen, Xiaotie Deng, and Shang-Hua Teng. Settling the complexity of computing two-player Nash equilibria. *J. ACM*, 56(3):1–57, 2009.
- [26] Jeff Clune. AI-GAs: AI-generating algorithms, an alternate paradigm for producing general artificial intelligence. *Preprint at arXiv:1905.10985*, 2019.
- [27] Karl Cobbe, Oleg Klimov, Chris Hesse, Taehoon Kim, and John Schulman. Quantifying generalization in reinforcement learning. In *Proc. Int. Conf. Mach. Learn.*, 2019.
- [28] Michael Cook and Simon Colton. Multi-faceted evolution of simple arcade games. In *Proc. IEEE Conf. Comp. Intel. Games*, 2011.
- [29] Michael Cook, Simon Colton, and Jeremy Gow. The ANGELINA videogame design system—part I. *IEEE T. Comp. Intel. AI*, 9(2):192–203, 2016.
- [30] Luciano da F Costa, Francisco A Rodrigues, Gonzalo Travieso, and Paulino Ribeiro Villas Boas. Characterization of complex networks: A survey of measurements. *Adv. Phys.*, 56(1):167–242, 2007.
- [31] Rémi Coulom. Efficient selectivity and backup operators in Monte-Carlo tree search. In *Proc. Int. Conf. Comp. Games*, 2006.
- [32] Jacob W Crandall, Mayada Oudah, Fatimah Ishowo-Oloko, Sherief Abdallah, Jean-François Bonnefon, Manuel Cebrian, Azim Shariff, Michael A Goodrich, and Iyad Rahwan. Cooperating with machines. *Nat. Commun.*, 9(1):1–12, 2018.
- [33] Wojciech Marian Czarnecki, Gauthier Gidel, Brendan Tracey, Karl Tuyls, Shayegan Omidshafiei, David Balduzzi, and Max Jaderberg. Real world games look like spinning tops. *Proc. Neural Inf. Process. Syst.*, 2020.
- [34] Constantinos Daskalakis. On the complexity of approximating a Nash equilibrium. *ACM T. Algorithms*, 9(3):1–35, 2013.
- [35] Constantinos Daskalakis, Paul W Goldberg, and Christos H Papadimitriou. The complexity of computing a Nash equilibrium. *SIAM J. on Comput.*, 39(1):195–259, 2009.
- [36] Robyn M Dawes. Social dilemmas. *Annu. Rev. Psychol.*, 31(1):169–193, 1980.
- [37] Christian Schröder de Witt, Jakob N. Foerster, Gregory Farquhar, Philip H. S. Torr, Wendelin Boehmer, and Shimon Whiteson. Multi-agent common knowledge reinforcement learning. In *Proc. Neural Inf. Process. Syst.*, 2019.
- [38] Matthias Dehmer. *Structural analysis of complex networks*. Springer Science & Business Media, 2010.
- [39] Sebastian Deterding. The lens of intrinsic skill atoms: A method for gameful design. *Hum.-Comput. Interact.*, 30(3-4):294–335, 2015.

- [40] Kate Donahue, Oliver P Hauser, Martin A Nowak, and Christian Hilbe. Evolving cooperation in multichannel games. *Nat. Commun.*, 11(1):1–9, 2020.
- [41] Debora Donato, Luigi Laura, Stefano Leonardi, and Stefano Millozzi. Large scale properties of the webgraph. *Eur. Phys. J. B*, 38(2):239–243, 2004.
- [42] Ishan Durugkar, Elad Liebman, and Peter Stone. Balancing individual preferences and shared objectives in multiagent reinforcement learning. In *Int. Joint Conf. Artif. Intel.*, 2020.
- [43] Jeffrey L Elman. Learning and development in neural networks: The importance of starting small. *Cognition*, 48(1):71–99, 1993.
- [44] Arpad Elo. *The Rating of Chess players, Past and Present*. Ishi Press International, 1978.
- [45] Richard Everett, Adam Cobb, Andrew Markham, and Stephen Roberts. Optimising worlds to evaluate and influence reinforcement learning agents. In *Proc. Auton. Agent Multi-Ag.*, 2019.
- [46] Jacob Feldman. Minimization of boolean complexity in human concept learning. *Nature*, 407(6804):630–633, 2000.
- [47] Carlos Florensa, David Held, Markus Wulfmeier, Michael Zhang, and Pieter Abbeel. Reverse curriculum generation for reinforcement learning. In *Conf. on Robot Learn. (CoRL)*, 2017.
- [48] Drew Fudenberg and Lorens A Imhof. Imitation processes with small mutations. *J. Econ. Theory*, 131(1):251–262, 2006.
- [49] Hello Games. No man’s sky. *Hello Games*, 2016.
- [50] Michael Genesereth, Nathaniel Love, and Barney Pell. General game playing: Overview of the AAAI competition. *AI Mag.*, 26(2):62–62, 2005.
- [51] Bertrand Georgeot, Olivier Giraud, and Dima L Shepelyansky. Spectral properties of the Google matrix of the world wide web and other directed networks. *Phys. Rev. E*, 81(5):056109, 2010.
- [52] Herbert Gintis. *Game Theory Evolving*. Princeton University Press, 2nd edition, 2009.
- [53] Alex Graves, Marc G Bellemare, Jacob Menick, Remi Munos, and Koray Kavukcuoglu. Automated curriculum learning for neural networks. In *Proc. Int. Conf. Mach. Learn.*, 2017.
- [54] Guy Hacohen and Daphna Weinshall. On the power of curriculum learning in training deep networks. In *Proc. Int. Conf. Mach. Learn.*, 2019.
- [55] Nikolaus Hansen. The CMA evolution strategy: A comparing review. In *Towards a new evolutionary computation*, pages 75–102. Springer, 2006.
- [56] Nikolaus Hansen, Youhei Akimoto, and Petr Baudis. CMA-ES/pycma on Github. Zenodo, DOI:10.5281/zenodo.2559634, February 2019.
- [57] Wang Hao and Sun Chuen-Tsai. Game reward systems: Gaming experiences and social meanings. In *Proc. DiGRA Int. Conf.*, 2011.
- [58] Garrett Hardin. The tragedy of the commons. *Science*, 162(3859):1243–1248, 1968.
- [59] Christoph Hauert, Arne Traulsen, Hannelore Brandt, Martin A Nowak, and Karl Sigmund. Via freedom to coercion: The emergence of costly punishment. *Science*, 316(5833):1905–1907, 2007.
- [60] Ricardo Hausmann, César A Hidalgo, Sebastián Bustos, Michele Coscia, Alexander Simoes, and Muhammed A Yildirim. *The atlas of economic complexity: Mapping paths to prosperity*. MIT Press, 2014.
- [61] Johannes Heinrich, Marc Lanctot, and David Silver. Fictitious self-play in extensive-form games. In *Proc. Int. Conf. Mach. Learn.*, 2015.

- [62] Daniel Hennes, Dustin Morrill, Shayegan Omidshafiei, Rémi Munos, Julien Perolat, Marc Lanctot, Audrunas Gruslys, Jean-Baptiste Lespiau, Paavo Parmas, Edgar Duéñez-Guzmán, and Karl Tuyls. Neural replicator dynamics: Multiagent learning via hedging policy gradients. In *Proc. Auton. Agent Multi-Ag.*, 2020.
- [63] José Hernández-Orallo. *The Measure of All Minds: Evaluating Natural and Artificial Intelligence*. Cambridge University Press, 2017.
- [64] José Hernández-Orallo. Evaluation in artificial intelligence: From task-oriented to ability-oriented measurement. *Artif. Intel. Rev.*, 48(3):397–447, 2017.
- [65] José Hernández-Orallo, Peter A. Flach, and César Ferri. A unified view of performance metrics: translating threshold choice into expected classification loss. *J. Mach. Learn. Res.*, 13:2813–2869, 2012.
- [66] José Hernández-Orallo, Marco Baroni, Jordi Bieger, Nader Chmait, David L Dowe, Katja Hofmann, Fernando Martínez-Plumed, Claes Strannegård, and Kristinn R Thórisson. A new AI evaluation cosmos: Ready to play the game? *AI Mag.*, 38(3):66–69, 2017.
- [67] Josef Hofbauer and Karl Sigmund. Evolutionary games and population dynamics. *Cambridge University Press*, 1998.
- [68] Vincent Hom and Joe Marks. Automatic design of balanced board games. In *Proc. AAAI Conf. on Artif. Intel. and Inter. Dig. Ent.*, 2007.
- [69] Álvaro Francisco Huertas-Rosero. A cartography for 2x2 symmetric games. In *III Colomb. Cong. and I Int. Andean Conf. Op. Res.*, 2004.
- [70] Lorens A Imhof, Drew Fudenberg, and Martin A Nowak. Evolutionary cycles of cooperation and defection. *Proc. Natl Acad Sci USA*, 102(31):10797–10800, 2005.
- [71] Manish Jain, Dmytro Korzhyk, Ondřej Vaněk, Vincent Conitzer, Michal Pěchouček, and Milind Tambe. A double oracle algorithm for zero-sum security games on graphs. In *Proc. Auton. Agent Multi-Ag.*, 2011.
- [72] Arthur Juliani, Ahmed Khalifa, Vincent-Pierre Berges, Jonathan Harper, Ervin Teng, Hunter Henry, Adam Crespi, Julian Togelius, and Danny Lange. Obstacle tower: A generalization challenge in vision, control, and planning. In Sarit Kraus, editor, *Int. Joint Conf. Artif. Intel.*, pages 2684–2691. ijcai.org, 2019. doi: 10.24963/ijcai.2019/373. URL <https://doi.org/10.24963/ijcai.2019/373>.
- [73] Niels Justesen, Ruben Rodriguez Torrado, Philip Bontrager, Ahmed Khalifa, Julian Togelius, and Sebastian Risi. Illuminating generalization in deep reinforcement learning through procedural level generation. In *Proc. NeurIPS Deep Rein. Learn. Workshop*, 2018.
- [74] Ahmed Khalifa, Michael Cerny Green, Gabriella Barros, and Julian Togelius. Intentional computational level design. In *Proc. Genetic and Evo. Comp. Conf.*, 2019.
- [75] Jongkwang Kim and Thomas Wilhelm. What is a complex graph? *Physica A*, 387(11):2637–2652, 2008.
- [76] Raph Koster. *Theory of fun for game design*. O'Reilly Media, Inc., 2013.
- [77] Danai Koutra, Joshua T Vogelstein, and Christos Faloutsos. Deltacon: A principled massive-graph similarity function. In *IEEE Data Min.*, 2013.
- [78] Jakub Kowalski and Marek Szykuła. Evolving chess-like games using relative algorithm performance profiles. In *Proc. Europ. Conf. Appl. of Evo. Comp.*, 2016.
- [79] Kai A Krueger and Peter Dayan. Flexible shaping: How learning in small steps helps. *Cognition*, 110(3):380–394, 2009.

- [80] Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Pérolat, David Silver, and Thore Graepel. A unified game-theoretic approach to multiagent reinforcement learning. In *Proc. Neural Inf. Process. Syst.*, 2017.
- [81] Marc Lanctot, Edward Lockhart, Jean-Baptiste Lespiau, Vinicius Zambaldi, Satyaki Upadhyay, Julien Pérolat, Sriram Srinivasan, Finbarr Timbers, Karl Tuyls, Shayegan Omidshafiei, Daniel Hennes, Dustin Morrill, Paul Muller, Timo Ewalds, Ryan Faulkner, János Kramár, Bart De Vylder, Brennan Saeta, James Bradbury, David Ding, Sebastian Borgeaud, Matthew Lai, Julian Schrittwieser, Thomas Anthony, Edward Hughes, Ivo Danihelka, and Jonah Ryan-Davis. OpenSpiel: A framework for reinforcement learning in games. *Preprint at arXiv:1908.09453*, 2019.
- [82] Nicole Lazzaro. Why we play: affect and the fun of games. *Hum.-Comput. Interact.*, 155:679–700, 2009.
- [83] Joel Z Leibo, Vinicius Zambaldi, Marc Lanctot, Janusz Marecki, and Thore Graepel. Multi-agent reinforcement learning in sequential social dilemmas. In *Proc. Auton. Agent Multi-Ag.*, 2017.
- [84] Joel Z Leibo, Edward Hughes, Marc Lanctot, and Thore Graepel. Autocurricula and the emergence of innovation from social interaction: A manifesto for multi-agent intelligence research. *Preprint at arXiv:1903.00742*, 2019.
- [85] Annick Lesne. Complex networks: From graph theory to biology. *Lett. Math. Phys.*, 78(3):235–262, 2006.
- [86] Zun Li and Michael P. Wellman. Structure learning for approximate solution of many-player games. In *Proc. AAAI Conf. on Artif. Intel.*, 2020.
- [87] Wim BG Liebrand. A classification of social dilemma games. *Sim. & Games*, 14(2):123–138, 1983.
- [88] Siqi Liu, Guy Lever, Nicholas Heess, Josh Merel, Saran Tunyasuvunakool, and Thore Graepel. Emergent coordination through competition. In *Proc. Int. Conf. Learn. Represent.*, 2019.
- [89] Marlos C. Machado, Marc G. Bellemare, Erik Talvitie, Joel Veness, Matthew J. Hausknecht, and Michael Bowling. Revisiting the Arcade Learning Environment: Evaluation protocols and open problems for general agents. *J. Artif. Intel. Res.*, 61:523–562, 2018.
- [90] Fragkiskos D Malliaros and Michalis Vazirgiannis. Clustering and community detection in directed networks: A survey. *Phys. Rep.*, 533(4):95–142, 2013.
- [91] Robert M May, Simon A Levin, and George Sugihara. Ecology for bankers. *Nature*, 451(7181):893–894, 2008.
- [92] John McCarthy. AI as sport. *Science*, 276(5318):1518–1519, 1997.
- [93] H Brendan McMahan, Geoffrey J Gordon, and Avrim Blum. Planning in the presence of cost functions controlled by an adversary. In *Proc. Int. Conf. Mach. Learn.*, 2003.
- [94] John H Miller and Scott E Page. *Complex adaptive systems: An introduction to computational models of social life*. Princeton university press, 2009.
- [95] Vahab S Mirrokni and Adrian Vetta. Convergence issues in competitive games. In *Approximation, randomization, and combinatorial optimization: Algorithms and techniques*. Springer, 2004.
- [96] Bojan Mohar. The Laplacian spectrum of graphs. *Graph Theo., Comb., and Appl.*, 2(871-898):12, 1991.
- [97] Paul Muller, Shayegan Omidshafiei, Mark Rowland, Karl Tuyls, Julien Perolat, Siqi Liu, Daniel Hennes, Luke Marrs, Marc Lanctot, Edward Hughes, Zhe Wang, Guy Lever, Nicolas Heess, Thore Graepel, and Remi Munos. A generalized training approach for multiagent learning. In *Proc. Int. Conf. Learn. Represent.*, 2020.

- [98] John F Nash. Equilibrium points in n -person games. *Proc. Natl. Acad. Sci. USA*, 36(1):48–49, 1950.
- [99] Mark J Nelson and Michael Mateas. Towards automated game design. In *Proc. Cong. Ital. Assoc. Artif. Intel.*, 2007.
- [100] Mark J Nelson, Julian Togelius, Cameron Browne, and Michael Cook. Rules and mechanics. In *Proc. Proced. Content Gen. Games*, pages 99–121. Springer, 2016.
- [101] Allen Newell and Herbert A Simon. Computer science as empirical inquiry: Symbols and search. *Commun. ACM*, 1976.
- [102] Thorbjørn S Nielsen, Gabriella AB Barros, Julian Togelius, and Mark J Nelson. Towards generating arcade game rules with VGDL. In *Proc. IEEE Conf. Compu. Intel.*, 2015.
- [103] Martin A Nowak and Karl Sigmund. Evolutionary dynamics of biological games. *Science*, 303(5659):793–799, 2004.
- [104] Martin A Nowak, Akira Sasaki, Christine Taylor, and Drew Fudenberg. Emergence of cooperation and evolutionary stability in finite populations. *Nature*, 428(6983):646–650, 2004.
- [105] Mancur Olson. *The Logic of Collective Action*. Harvard University Press, 1974.
- [106] Shayegan Omidshafiei, Christos Papadimitriou, Georgios Piliouras, Karl Tuyls, Mark Rowland, Jean-Baptiste Lespiau, Wojciech M Czarnecki, Marc Lanctot, Julien Perolat, and Rémi Munos. α -Rank: Multi-agent evaluation by evolution. *Sci. Rep.*, 9, 2019.
- [107] OpenAI, Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, Rafal Józefowicz, , Scott Gray, Catherine Olsson, Jakub Pachocki, Michael Petrov, Henrique Pondé de Oliveira Pinto, Jonathan Raiman, Tim Salimans, Jeremy Schlatter, Jonas Schneider, Szymon Sidor, Ilya Sutskever, Jie Tang, Filip Wolski, and Susan Zhang. Dota 2 with large scale deep reinforcement learning. *Preprint at arXiv:1912.06680*, 2019.
- [108] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.
- [109] Georgios A Pavlopoulos, Maria Secrier, Charalampos N Moschopoulos, Theodoros G Soldatos, Sophia Kossida, Jan Aerts, Reinhard Schneider, and Pantelis G Bagos. Using graph theory to analyze biological networks. *BioData Min.*, 4(1):10, 2011.
- [110] Diego Perez-Liebana, Jialin Liu, Ahmed Khalifa, Raluca D Gaina, Julian Togelius, and Simon M Lucas. General video game AI: A multitrack framework for evaluating agents, games, and content generation algorithms. *IEEE Trans. Games*, 11(3):195–214, 2019.
- [111] Duc Truong Pham, Stefan S Dimov, and Chi D Nguyen. Selection of K in K -means clustering. *P. I. Mech. Eng. C-J Mec.*, 219(1):103–119, 2005.
- [112] Steve Phelps, Simon Parsons, and Peter McBurney. An evolutionary game-theoretic comparison of two double-auction market designs. In *AAMAS Workshop on Agent-Med. Elec. Comm.*, 2004.
- [113] John Platt. Social traps. *Am. Psychol.*, 28(8):641, 1973.
- [114] Marc Prensky. Fun, play and games: What makes games engaging. *Dig. Game-based Learn.*, 5(1):5–31, 2001.
- [115] Anatol Rapoport and Melvin Guyer. A taxonomy of 2×2 games. *Gen. Sys.*, 11:203–214, 1966.
- [116] Kevin Regan and Craig Boutilier. Regret-based reward elicitation for Markov decision processes. In *Proc. Uncertainty in Artif. Intel.*, 2009.

- [117] Sebastian Risi and Julian Togelius. Increasing generality in machine learning through procedural content generation. *Nature Mach. Intel.*, pages 1–9, 2020.
- [118] David Robinson and David Goforth. *The topology of the 2x2 games: A new periodic table*, volume 3. Psychology Press, 2005.
- [119] Julia Robinson. An iterative method of solving a game. *Ann. Math.*, pages 296–301, 1951.
- [120] Lior Rokach and Oded Maimon. Clustering methods. In *Data Min. Knowl. Disc. Hand.*, pages 321–352. Springer, 2005.
- [121] Mark Rowland, Shayegan Omidshafiei, Karl Tuyls, Julien Perolat, Michal Valko, Georgios Piliouras, and Remi Munos. Multiagent evaluation under incomplete information. In *Proc. Neural Inf. Process. Syst.*, 2019.
- [122] Arthur L Samuel. Programming computers to play games. In *Adv. Comput.*, volume 1, pages 165–192. Elsevier, 1960.
- [123] William H. Sandholm. *Population Games and Evolutionary Dynamics*. Economic Learning and Social Evolution. MIT Press, 2010.
- [124] A Sanfeliu and King-Sun Fu. A distance measure between attributed relational graphs for pattern recognition. *IEEE T. Sys., Man, Cyber.*, SMC-13(3), 1983.
- [125] Terence D Sanger. Neural network learning control of robot manipulators using gradually increasing task difficulty. *IEEE T. Robotic Autom.*, 10(3):323–333, 1994.
- [126] Fernando P Santos, Francisco C Santos, and Jorge M Pacheco. Social norm complexity and past reputations in the evolution of cooperation. *Nature*, 555(7695):242–245, 2018.
- [127] Venu Satuluri and Srinivasan Parthasarathy. Symmetrizations for clustering directed graphs. In *Proc. Int. Conf. on Ext. Database Tech.* ACM, 2011.
- [128] Jonathan Schaeffer. A gamut of games. *AI Mag.*, 22(3):29–29, 2001.
- [129] Marten Scheffer. *Critical transitions in nature and society*, volume 16. Princeton University Press, 2009.
- [130] John Scott. Social network analysis. *Sociol.*, 22(1):109–127, 1988.
- [131] Mohammad Shaker, Mhd Hasan Sarhan, Ola Al Naameh, Noor Shaker, and Julian Togelius. Automatic generation and analysis of physics-based puzzle games. In *Proc. IEEE Conf. Compu. Intel.*, 2013.
- [132] Noor Shaker, Georgios Yannakakis, and Julian Togelius. Towards automatic personalized content generation for platform games. In *Proc. Artif. Intel. and Int. Dig. Ent. Conf.*, 2010.
- [133] Noor Shaker, Julian Togelius, and Mark J Nelson. *Procedural content generation in games*. Springer, 2016.
- [134] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE T. Pattern Anal.*, 22(8):888–905, 2000.
- [135] Karl Sigmund. *The calculus of selfishness*, volume 6. Princeton University Press, 2010.
- [136] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

- [137] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, 362(6419):1140–1144, 2018.
- [138] Satinder P Singh, Michael J Kearns, and Yishay Mansour. Nash convergence of gradient dynamics in general-sum games. In *Proc. Uncertainty in Artif. Intel.*, 2000.
- [139] Adam M Smith and Michael Mateas. Answer set programming for procedural content generation: A design space approach. *IEEE T. Comp. Intel. AI*, 3(3):187–200, 2011.
- [140] Gillian Smith and Jim Whitehead. Analyzing the expressive range of a level generator. In *Proc. Workshop on Proced. Content Gen. Games*, 2010.
- [141] John Maynard Smith and John Maynard Smith. *Evolution and the Theory of Games*. Cambridge university press, 1982.
- [142] Thomas Spooner and Rahul Savani. Robust market making via adversarial reinforcement learning. In *Int. Joint Conf. Artif. Intel.*, 2020.
- [143] Andrea Tacchella, Matthieu Cristelli, Guido Caldarelli, Andrea Gabrielli, and Luciano Pietronero. Economic complexity: Conceptual grounding of a new metrics for global competitiveness. *J. of Econ. Dyn. Control*, 37(8):1683–1691, 2013.
- [144] Christine Taylor, Drew Fudenberg, Akira Sasaki, and Martin A Nowak. Evolutionary game dynamics in finite populations. *B. Math Biol.*, 66(6):1621–1644, 2004.
- [145] Kristinn R Thórisson, Jordi Bieger, Thröstur Thorarensen, Jóna S Siguroardóttir, and Bas R Steunebrink. Why artificial intelligence needs a task theory. In *Proc. Int. Conf. on Artif Gen. Intel.*, 2016.
- [146] Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic. *J. Royal Stats. Soc. B*, 63(2):411–423, 2001.
- [147] Julian Togelius and Jurgen Schmidhuber. An experiment in automatic game design. In *Proc. IEEE Conf. Compu. Intel.*, 2008.
- [148] Julian Togelius, Georgios N Yannakakis, Kenneth O Stanley, and Cameron Browne. Search-based procedural content generation: A taxonomy and survey. *IEEE T. Comp. Intel. AI*, 3(3):172–186, 2011.
- [149] Julian Togelius, Mark Nelson, and Antonios Liapis. Characteristics of generatable games. In *Work. Proced. Content Gen. Games*, 2014.
- [150] Michael Toy and Glenn Wichman. Rogue. Cross-platform, 1980.
- [151] Arne Traulsen, Martin A Nowak, and Jorge M Pacheco. Stochastic dynamics of invasion and fixation. *Phys. Rev. E*, 74:011909, 2006.
- [152] Arne Traulsen, Jorge M. Pacheco, and Lorens A. Imhof. Stochasticity and evolutionary stability. *Phys. Rev. E*, 74:021905, 2006.
- [153] Anton Tsitsulin, Davide Mottin, Panagiotis Karras, Alexander Bronstein, and Emmanuel Müller. NetLSD: Hearing the shape of a graph. In *Proc. Int. Conf. on Know. Disc. & Data Min. (SIGKDD)*, 2018.
- [154] Alan M Turing. Digital computers applied to games. In *Faster than thought: A symposium on digital computing machines*. Pitman Publishing, 1953.
- [155] Karl Tuyls, Julien Perolat, Marc Lanctot, Joel Z Leibo, and Thore Graepel. A generalised method for empirical game theoretic analysis. In *Proc. Auton. Agent Multi-Ag.*, 2018.

- [156] Karl Tuyls, Julien Pérolat, Marc Lanctot, Edward Hughes, Richard Everett, Joel Z. Leibo, Csaba Szepesvári, and Thore Graepel. Bounds and dynamics for empirical game theoretic analysis. *Proc. Auton. Agent Multi-Ag.*, 34(1):7, 2020.
- [157] Valeriu Ungureanu and Ana Botnari. Nash equilibria sets in mixed extended 2×3 games. *Comp. Sci. J. Mold.*, 13(2):38, 2005.
- [158] Hadrien Van Lierde. Spectral clustering algorithms for directed graphs. Master’s thesis, Université Catholique de Louvain., 2015.
- [159] Sven Van Segbroeck, Jorge M Pacheco, Tom Lenaerts, and Francisco C Santos. Emergence of fairness in repeated group interactions. *Phys. Rev. Lett.*, 108(15):158104, 2012.
- [160] Maarten Van Steen. Graph theory and complex networks: An introduction. 2010.
- [161] Vítor V Vasconcelos, Fernando P Santos, Francisco C Santos, and Jorge M Pacheco. Stochastic dynamics through hierarchically embedded Markov chains. *Phys. Rev. Lett.*, 118(5):058301, 2017.
- [162] Carl Veller and Laura K Hayward. Finite-population evolution with rare mutations in asymmetric games. *J. Econ. Theory*, 162:93–113, 2016.
- [163] Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, Laurent Sifre, Trevor Cai, John P. Agapiou, Max Jaderberg, Alexander S. Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, David Budden, Yury Sulsky, James Molloy, Tom L. Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wünsch, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy Lillicrap, Koray Kavukcuoglu, Demis Hassabis, Chris Apps, and David Silver. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.
- [164] Michael S Vitevitch. What can graph theory tell us about word learning and lexical retrieval? *J. Speech. Hear. Res.*, 2008.
- [165] Lev Vygotsky. Interaction between learning and development. *Read. Dev. Child.*, 23(3):34–41, 1978.
- [166] William E Walsh, Rajarshi Das, Gerald Tesauro, and Jeffrey O Kephart. Analyzing complex strategic interactions in multi-agent games. In *AAAI Workshop on Game Theo. Dec. Theo. Agents*, 2002.
- [167] Rui Wang, Joel Lehman, Jeff Clune, and Kenneth O Stanley. Paired open-ended trailblazer (POET): Endlessly generating increasingly complex and diverse learning environments and their solutions. *Preprint at arXiv:1901.01753*, 2019.
- [168] Rui Wang, Joel Lehman, Aditya Rawal, Jiale Zhi, Yulun Li, Jeff Clune, and Kenneth O Stanley. Enhanced POET: Open-ended reinforcement learning through unbounded invention of learning challenges and their solutions. *Preprint at arXiv:2003.08536*, 2020.
- [169] Xiaofeng Wang and Tuomas Sandholm. Reinforcement learning to play an optimal Nash equilibrium in team Markov games. In *Proc. Neural Inf. Process. Syst.*, 2003.
- [170] Stanley Wasserman and Katherine Faust. *Social network analysis: Methods and applications*. Cambridge University Press, 1994.
- [171] Kevin Waugh, Dustin Morrill, James Andrew Bagnell, and Michael Bowling. Solving games with functional regret estimation. In *Proc. AAAI Conf. on Artif. Intel.*, 2015.
- [172] Jorgen W Weibull. *Evolutionary Game Theory*. MIT Press, 1995.
- [173] Daphna Weinshall, Gad Cohen, and Dan Amir. Curriculum learning by transfer learning: Theory and experiments with deep networks. In *Proc. Int. Conf. Mach. Learn.*, 2018.

- [174] Michael P Wellman. Methods for empirical game-theoretic analysis. In *Proc. AAAI Conf. on Artif. Intel.*, 2006.
- [175] Michael P Wellman, Tae Hyung Kim, and Quang Duong. Analyzing incentives for protocol compliance in complex domains: A case study of introduction-based routing. In *Workshop on the Economics of Information Security*, 2013.
- [176] James R Wright and Kevin Leyton-Brown. A formal separation between strategic and nonstrategic behavior. *Preprint at arXiv:1812.11571*, 2018.
- [177] Mason Wright, Yongzhao Wang, and Michael P. Wellman. Iterated deep reinforcement learning in games: History-aware training for improved stability. In *Conf. Econ. Compu.*, 2019.
- [178] Georgios N Yannakakis and John Hallam. Evolving opponents for interesting interactive computer games. In *Proc. Int. Conf. on Sim. Adapt. Beh.*, 2004.
- [179] Georgios N Yannakakis and Julian Togelius. *Artificial intelligence and games*. Springer, 2018.

Acknowledgements

The authors gratefully thank Marc Lanctot and Thore Graepel for insightful feedback on the paper. FCS acknowledges the support of FCT-Portugal through grants PTDC/MAT/STA/3358/2014 and UIDB/50021/2020. The authors thank the three anonymous reviewers and editor for their constructive and insightful feedback, which helped to significantly improve the clarity of the proposed method, experimental results, and discussions thereof.

Author contributions

SO, KT, and FCS conceptualized the graph-based framework used in the paper. SO implemented and generated the experimental results (spectral analysis of response graphs, clusterings, landscape of games, procedural games generation). WMC devised and implemented the policy sampling scheme for real world games, and generated the payoff tables for the games listed in Figure 1. SO, KT, WMC, FCS, MR, JC, DH, PM, JP, BDV, AG, and RM contributed actively to discussions, analysis, writing, and review of the paper.

Competing interests

The authors declare no competing financial or non-financial interests.

Additional information

Materials & correspondence. Material requests and correspondence should be requested to Shayegan Omidshafiei (somidshafiei@google.com).

Supplementary Information: Navigating the Landscape of Multiplayer Games

Supplementary Methods 1

This section provides additional discussions and exposition of methods.

Policy sampling scheme

The policy sampling scheme we use for large, real-world games follows the policy space coverage procedure first specified by Czarnecki et al.³³, and is detailed as follows. First, we use a tree search algorithm, Alpha-Beta¹⁰¹ with varying tree depth limits d , and seeds s . This involves running the Alpha-Beta algorithm to depth d , and using random actions with seed s thereafter (i.e., if the game does not terminate). This yields policies of varying transitive strengths (controlled by depth d), with a range of related policies per depth (controlled by seed s). This also covers the case of a purely random policy when d is set to 0 (with seed s controlling the randomness). Second, we repeat the same procedure with negated game payoffs, thus also covering the space of policies that seek to lose the original game. Third, to further expand the policy space, we also define an augmented Alpha-Beta search, which assumes that branches of the game tree with depth beyond d have a value of 0. We likewise run this variant with negated payoffs as well. Finally, to cover the policy space for more difficult games, we further augment this sampling strategy by using MCTS³¹ on each game, with k simulations and varying seeds s .

For each variant of Alpha-Beta detailed above, we use depth parameters $d \in \{1, \dots, 9\}$. For MCTS, we use $k \in \{10, 100, 1000\}$ simulations. For all algorithms, we also sweep over seeds $s \in \{1, \dots, 50\}$. While this sampling procedure is a heuristic, it produces a range of policies with varying degrees of transitive and intransitive relations, and thus provide a useful approximation of the underlying game.

Description of games analyzed

We provide an overview of the games analyzed (noting that we omit descriptions of 11-20, AlphaGo, MuJoCo soccer, Blotto, and AlphaStar League, as they are detailed in the main text). As our methods operate on α -Rank response graphs, they apply to many-player general-sum games. For the specific instances of two-player zero-sum games analyzed here, we symmetrize payoffs and standardize them such that $M \in [-1, 1]$.

Redundant Rock–Paper–Scissors. We modify the standard Rock–Paper–Scissors payoffs, \mathbf{M}_{RPS} , by duplicating the first strategy (Rock), yielding the payoffs for the redundant variant, \mathbf{M}_{RRPS} ,

$$\mathbf{M}_{RPS} = \begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{bmatrix} \quad \mathbf{M}_{RRPS} = \begin{bmatrix} 0 & 0 & -1 & 1 \\ 0 & 0 & -1 & 1 \\ 1 & 1 & 0 & -1 \\ -1 & -1 & 1 & 0 \end{bmatrix}. \quad (5)$$

Disc game. The Disc game is a cyclical game, defined as a differentiable generalization of the standard game of Rock–Paper–Scissors⁶. We construct the Disc game payoffs as in Czarnecki et al.³³, by first uniformly sampling 1000 points, $\{S_i\}_{i \in [1000]}$, in the unit circle, subsequently defining payoffs,

$$M(i, j) = S_i^T \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} S_j. \quad (6)$$

Elo games and noisy variants. The variety of Elo games essentially correspond to the multidimensional Elo model detailed in the main paper, with the intransitive components removed, and payoffs rescaled. Noisy variants of these Elo games are generated via specifying a noise parameter σ^2 , adding zero-mean normally distributed noise of the specified variance to the non-noisy Elo payoff table, thus yielding noisy payoffs \mathbf{M}_{σ^2} . Finally, we symmetrize these payoffs, yielding final payoffs $\mathbf{M} = \mathbf{M}_{\sigma^2} - \mathbf{M}_{\sigma^2}^T$.

Games in motivating examples. The variants of transitively-structured games in the motivating examples are Elo games, as specified above. The cyclical games are $N \times N$ mElo games of rank $k = 1$, with all transitive r set to 0, and intransitive components specified as follows: $c_{0:N-2,0} = [0, 1, \dots, N-2]$, $c_{0:N-2,1} = [N-2, N-1, \dots, 0]$, and $c_{N-1,:} = [-1, 1]$. The random-structured games are mElo games with rank $k = 3$, and transitive and intransitive parameters i.i.d. sampled from $\mathcal{N}(0, 1)$.

Transitive game. The Transitive Game payoffs are simply set to +1 for the upper-triangle, -1 for the lower-triangle, and 0 across the diagonal.

Random Game of Skill. The Random Game of Skill is a 1000×1000 game, generated as defined by Czarnecki et al.³³. Specifically, payoffs are $\mathbf{M}(i, j) = 0.5(W_{ij} - W_{ji}) + S_i - S_j$, where W_{ij} , W_{ji} , S_i , and S_j are sampled from $\mathcal{N}(0, 1)$. In the Normal Bernoulli Game, parameters S_i and S_j are sampled likewise, while W_{ij} and W_{ji} are sampled from $\mathcal{U}(0, 1)$ and payoffs are specified as $\mathbf{M}(i, j) = W_{ij} - W_{ji} + S_i - S_j$.

3-move parity game. The parity game is generated per the definition in Czarnecki et al.³³.

Real-world games. We use the OpenSpiel⁸¹ implementations of the following games: Tic–Toe, Hex (board size=3), Quoridor (board size=3), Quoridor (board size=4), Go (board size=3), Go (board size=4), Connect Four, Kuhn Poker. For Go, we use a Komi (first-move advantage) of 6.5 points. With the exception of Kuhn Poker (which has only 12 information states, and fully enumerable policy space), we use the policy sampling scheme detailed above for this collection of games.

Supplementary Note 1: Related Work

This paper largely focuses on topological analysis and taxonomization of multiplayer games, via study of the interactions possible within them. As such, this work sits at the intersection of several different research disciplines, including game theory, machine learning, multiagent systems and network science.

A motivating application for such an approach is to classify or cluster games that present interesting challenges for artificial agents. The central question revolving around the interestingness of environments for artificial learning agents has a long history in machine learning, task theory¹⁴⁵, procedural content generation¹⁴⁸, and curriculum learning⁸.

An overview of the role of games in AI, including evaluation and generation of the games themselves, is provided in Yannakakis and Togelius¹⁷⁹. Thórisson et al.¹⁴⁵ emphasizes the need for a task theory in AI, a unifying and formal framework that enables comparison, abstraction, characterization, and decomposition of agent tasks (and the associated environments within which tasks are defined). The decomposability of tasks enabled by such a theory can prove useful for not only understanding the task-space, but also enabling more efficient agent training via curriculum-based teaching, as discussed in Bieger and Thórisson¹⁰. An overview of recent AI benchmarks and platforms is provided in Hernández-Orallo et al.⁶⁶, which also includes pointers to workshops and recent works investigating the potential taxonomization of such a wide suite of tasks and environments. Overall, our contribution can be broadly classified as falling under the task theory regime, as a step towards characterizing the particular set of tasks studied here (i.e., multiplayer games that can be characterized via empirical payoff tables) via graph- and game-theoretic techniques.

Procedural generation of game content has been used in commercially-available games for decades, spanning from the dungeon-crawler and space trading games Rogue and Elite, respectively released in 1980 and 1984^{17,150}, to more recent expansive 3D open-world games such as No Man’s Sky⁴⁹. Significant research has also been conducted in this field, thorough overviews provided by Risi and Togelius¹¹⁷, Shaker et al.¹³³, Togelius et al.¹⁴⁸. Other notable examples of procedural content generation and automated game design include Cook and Colton²⁸, Cook et al.²⁹, Nelson and Mateas⁹⁹, Smith and Mateas¹³⁹. The complexity of the environments generated by these and related techniques has been substantially increasing in recent years. Recently, Juliani et al.⁷² introduced a procedurally generated 3D tower environment for training AI agents, which increases complexity of associated skills required by the agent to solve the task. A technique for iterative procedural task generation and AI training was introduced recently by Wang et al.¹⁶⁷, and further generalized in Wang et al.¹⁶⁸.

The literature on General Game Playing⁵⁰, in particular, has close connections to the topic studied here. Specifically, works such as Browne and Maire¹⁸ investigate generation of complete rule-sets for extensive-form games, culminating in design of a commercially-available game (Yavalath). Related works on rule-based game generation¹⁴⁹ include examples generating pacman-like games¹⁴⁷ and chess-like games⁷⁸, with a useful survey provided in Nelson et al.¹⁰⁰. A comparative evaluation of games and agents generated is provided in Perez-Liebana et al.¹¹⁰.

Generation of games with attributes desirable by humans has also seen some attention, including querying of human preferences (specifically, fun, challenge, and frustration) for personalized content generation in Shaker et al.¹³², creation of more believable environments²², and creation of balanced board games⁶⁸. Works have also taken an agent-vs.-task perspective for evaluating both agent performance and task/game quality^{5,102}. A number of works have also considered specific notions of the interestingness of a game (or an evaluation of the fitness of a game from varying perspectives)^{3,18,21,39,50,57,68,82,102,114,117,148,165,167,178}; we refer readers to the Landscape of Games section of the main text for discussion of these works.

In curriculum learning, initial work concerned the use of curricula of tasks in training neural networks via supervised learning^{8,43,79,125}. This continues to be an active field of research to this day, with many new methods for curriculum learning being developed to match the increasingly intricate architectures used within deep learning^{53,54,173}, and the increasingly wide range of applications for deep neural networks, such as deep reinforcement learning^{27,47,73,137,163}. Recent work has brought open-ended learning via multiagent interaction and procedural environment generation to the forefront of curriculum learning^{6,33,45,73,84,167}.

This work contributes to this line of inquiry by framing the intrinsic interestingness of games as a driving factor in open-ended multiagent learning. While equilibrium computation in many classes of games is PPAD-complete (thus generally considered intractable^{25,34,35}), and while the computational complexity results in this work focus on the two-player zero-sum case, it is worth mentioning that our methods also directly apply to many-player general-sum settings (as exemplified by the analysis of the general-sum 11-20 game in the main text, and of many-player variants of Kuhn Poker in Figure 20).

Several prior works have analyzed game-theoretic response graphs from the perspective of agent evaluation^{1,95,97,106,121,157,169}, and more generally there is a wide literature concerning analysis of directed graphs across a range of domains and disciplines^{30,60,75,108}. Additionally, there is an established line of work in game theory that seeks a discrete classification of games based on payoff tables. The space of 2x2 normal-form games have been classified into groups with restrictions, such as ordering the payouts^{19,87,115}, and into a periodic table through topological and graph-theoretic techniques¹¹⁸. We provide a comparison against the approach of Bruns¹⁹ below. More recent work has introduced a framework for binary classification of agent behaviors (e.g., as strategic and non-strategic) in simultaneous-move normal form games¹⁷⁶.

Social dilemma games (such as Prisoners Dilemma, Chicken, or Hoarding), categorized by Dawes³⁶, Liebrand⁸⁷, are a subset of games encapsulating the trade-offs found in social dilemmas. Social dilemmas such as the tragedy of the commons show the potential conflict between individual and group self-interest^{58,105,113}. Early work in social dilemmas showed how the dilemmas can arise from reinforced behavior¹¹³. More recently, reinforcement learning (RL) has been applied to sequential social dilemmas⁸³ where strategic decisions to cooperate or defect were shown to coincide with the learning of policies. Agent behavior was also shown to depend on factors such as the relative difficulties of learning policies and in the case of cooperation, possibly non-occurring coordination sub-problems. RL has also been used to train agents that cooperate with humans across a variety of two-player repeated games³².

In addition to analysis of individual graphs, there has been much work on the analysis and clustering of collections of graphs in a variety of contexts^{9,77,124,153}. The graph-based analysis conducted herein draws on these works (e.g., using several local and global graph features and distributions to characterize our games, as in Berlingero et al.⁹). However, our approach focuses specifically on games, exploiting several measures related to the complexity of games, enabling navigation of their underlying landscape and subsequent generation of new games of interest.

A wide-spread strategy comparison method in machine learning is the Elo rating⁴⁴ which ranks agents against one another in a purely transitive manner. Balduzzi et al.⁵ showed the importance of intransitive strategies when comparing agents, leading to the so called multidimensional Elo model, modeling more complex relations between agents. Finally, in a line of work related to training of agents, Policy-Space Response Oracles (PSRO)⁸⁰ and Rectified PSRO⁶ lead to learning a collection of possibly intransitive strategies, but not to directly analyze them, as done in this paper.

Table 1: 2×2 games and row player payoffs from Bruns¹⁹. Important note: corresponding column player payoffs are defined as a transpose along the anti-diagonal in Bruns¹⁹ (i.e., not according to the typical convention of using standard transposes for symmetric games). This latter point has no impact on quantitative results, but is important for readers qualitatively analyzing the payoff structures and corresponding results below.

Chicken (Ch)	Battle (Ba)	Hero (Hr)	Compromise (Cm)	Deadlock (Dl)	Prisoner's dilemma (Pd)	Stag hunt (Sh)	Assurance (As)	Coordination (Co)	Peace (Pc)	Harmony (Ha)	Concord (Nc)
$\begin{bmatrix} 2 & 3 \\ 1 & 4 \end{bmatrix}$	$\begin{bmatrix} 3 & 2 \\ 1 & 4 \end{bmatrix}$	$\begin{bmatrix} 3 & 1 \\ 2 & 4 \end{bmatrix}$	$\begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix}$	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$	$\begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$	$\begin{bmatrix} 1 & 4 \\ 2 & 3 \end{bmatrix}$	$\begin{bmatrix} 1 & 4 \\ 3 & 2 \end{bmatrix}$	$\begin{bmatrix} 2 & 4 \\ 3 & 1 \end{bmatrix}$	$\begin{bmatrix} 3 & 4 \\ 2 & 1 \end{bmatrix}$	$\begin{bmatrix} 3 & 4 \\ 1 & 2 \end{bmatrix}$	$\begin{bmatrix} 2 & 4 \\ 1 & 3 \end{bmatrix}$

Supplementary Note 2: Additional results

This section provides additional results, including response graphs and experiments conducted for generating the landscapes in the main text.

Taxonomization of 2×2 games

Here we conduct a set of experiments comparing our approach against that of Bruns¹⁹ (focusing on the ordinal games results presented in Figure 1 of their paper).

Let us first provide a summary of the approach of Bruns¹⁹, prior to comparing our method against it. The taxonomy of 2×2 games proposed by Bruns¹⁹ relies on two elements: 1) a topological ordering of said games, according to the patterns of payoffs received by each player; 2) classification of payoff families, based on the respective payoffs received by each player at the Nash equilibria. Based on the earlier work of Robinson and Goforth¹¹⁸, Bruns¹⁹ identifies 12 core payoff patterns in ordinal 2×2 games (which we summarize for the row player in Table 1). Column player payoffs are defined in Bruns¹⁹ by transposing the row player payoffs along the anti-diagonal (i.e., not the conventional transpose). The combinations of row and column player payoffs yields a total of 144 2×2 games of interest. These games are subsequently categorized according to the relative payoffs received by the players in the Nash equilibria (e.g., win-win, cyclic, unfair outcomes, etc.), yielding a total of 7 classes (reproduced here in Figure 11a). Note that the 4 quadrants indicated by the thick blue lines in Figure 11a indicate so-called layers of 36 games each, where games in each layer have the same highest-payoff relationship between the two players. For example, games in the lower-left layer are all win-win (thus have the highest payoff for both players in the same payoff cell), whereas games in the top-right layer have the highest payoffs in diagonally-opposite payoff cells for the players.

To generate a clustering of these games using our approach, we run the graph-based analysis detailed in the paper (i.e., computing response graphs, collecting graph measures, and running PCA on the collection of 144 games). We subsequently cluster these games using their top-2 principal components, using 7 clusters (as in Bruns¹⁹). The resulting clusters are visualized in Figure 11b. Note that as our clusters are automatically discovered, and not hand-specified using Nash equilibrium payoffs, only their structure (and not color) should be compared against Bruns' clusters. While some similarities exist between these and Bruns' clusters (e.g., a visually-apparent division into the same 4 layers is present in our clustering, and certain games clusters such as ($Ch \times Co$, $Ch \times As$, $Ba \times Co$, $Ba \times Sh$, $Hr \times As$, $Hr \times Sh$) are shared between the two methods), notable differences are present.

To better understand these differences, consider again the means by which the Bruns clusters are derived: manual identification of patterns over the players' payoffs in the Nash equilibria of each game. Our approach, by contrast, relies on aggregation of statistics over the response graph of the games. As 2×2 games consist of 4 strategy profiles in total, the response graphs correspondingly have 4 nodes, implying that few variations are possible in their structure (in contrast to the much larger games targeted in our main results). Despite this, we can better tune our approach for these extremely small games. Specifically, one of the statistics used for computing our clustering is the α -Rank distribution entropy associated with the response graph. Typically, α -Rank is used with a high value of selection-intensity parameter, α , to increase the distribution mass over strong strategies. In these small 2×2 games, decreasing the value of α -Rank's selection-intensity parameter, α , causes the α -Rank distribution to increase in entropy, and

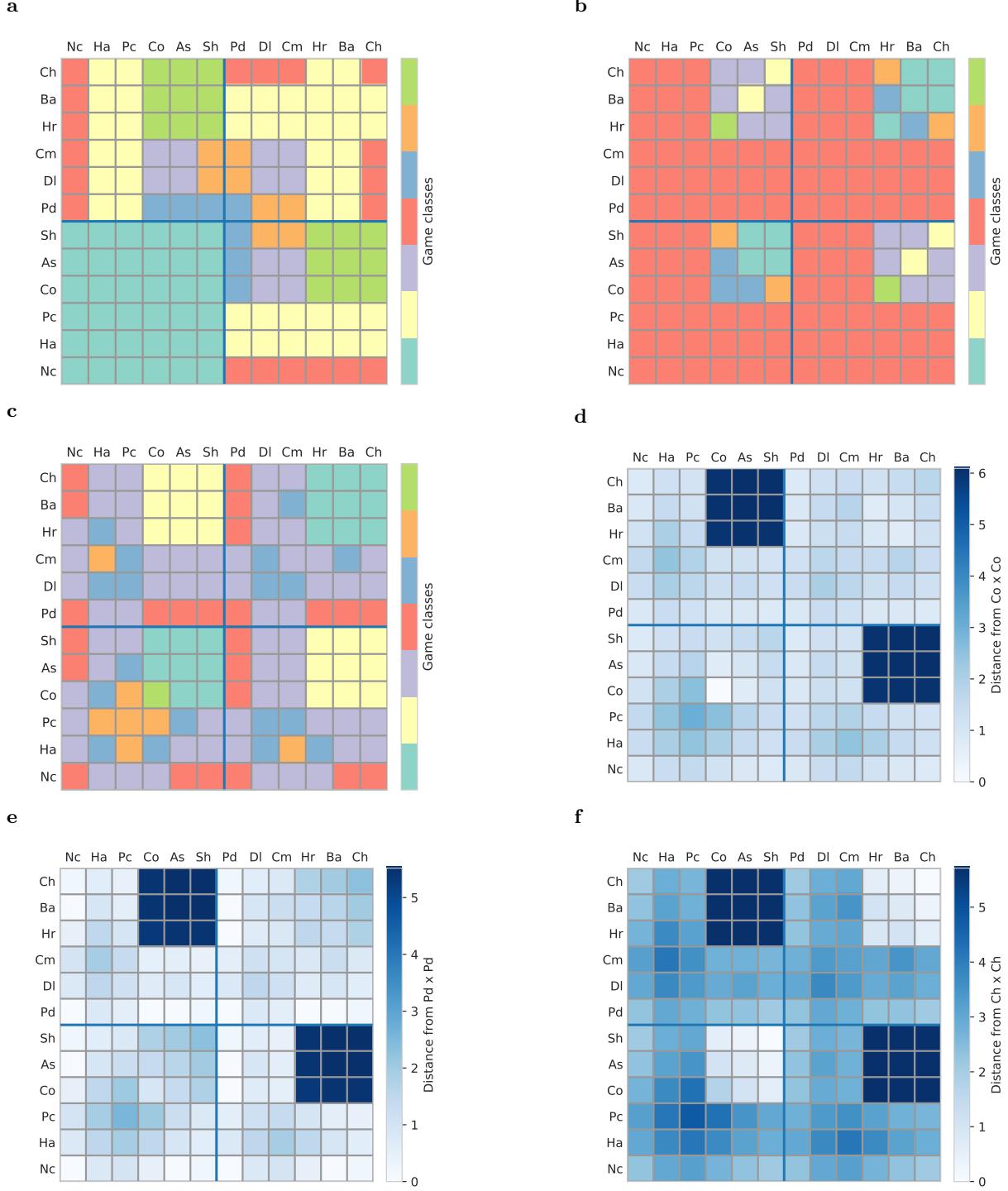


Figure 11: Comparison of our approach to that of Bruns¹⁹ for 2×2 games. **a** Clusters (Bruns). **b** Clusters (ours, high α -Rank α , where $\alpha = 0.2$). **c** Clusters (ours, lower α -Rank α , where $\alpha = 0.01$). **d** Game distances (ours, to $Co \times Co$). **e** Game distances (ours, to $Pd \times Pd$). **f** Game distances (ours, to $Ch \times Ch$).

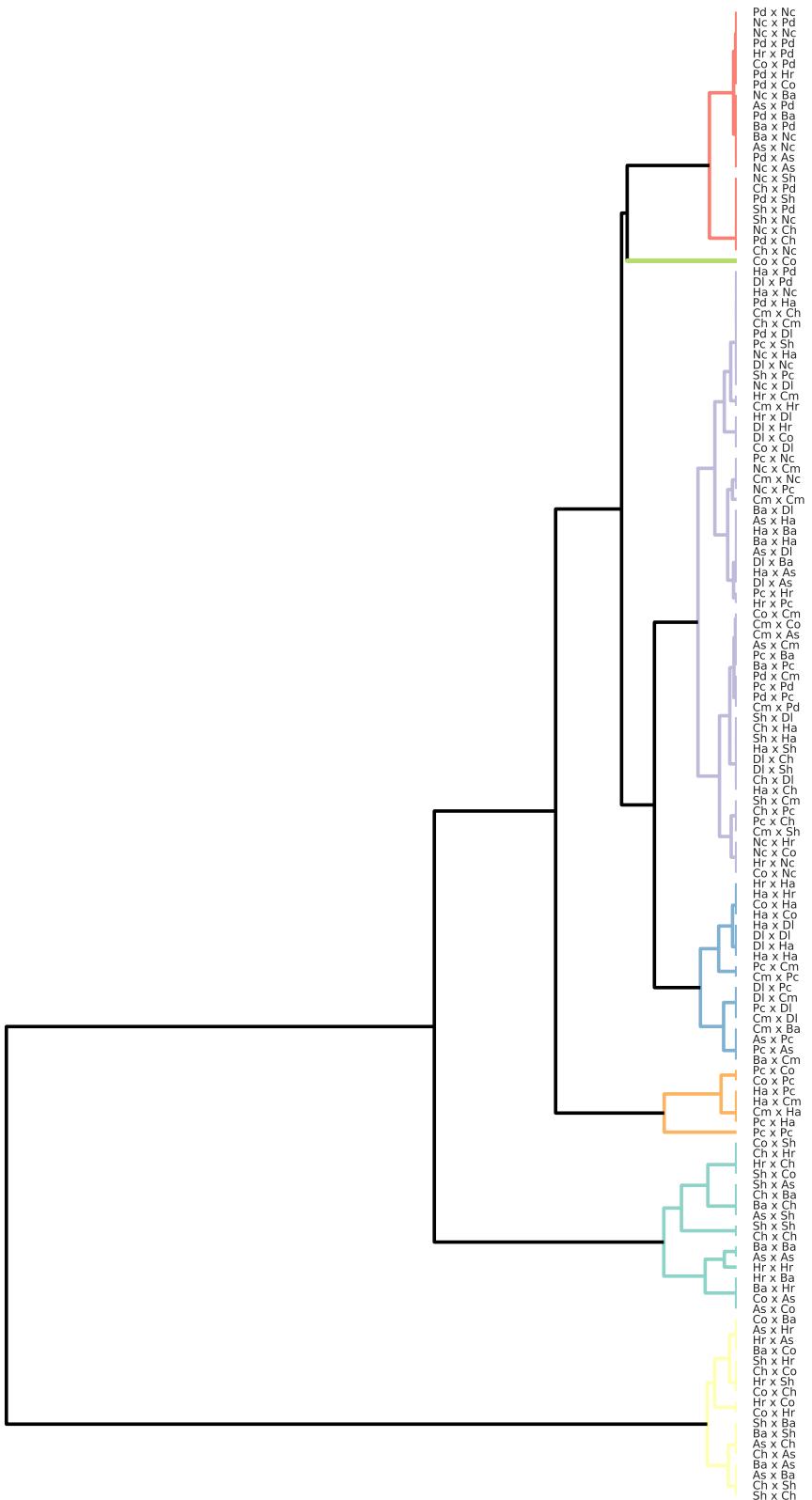


Figure 12: Dendrogram of 2×2 games, computed using our approach. All 144 games are visualized here, with classes colored in accordance to Figure 11c.

subsequently better capture the distribution of payoffs received over the profiles (akin to the Nash-based payoff comparisons used by Bruns to derive their clusters). The results of this modification, which brings our approach closer to that of Bruns, is visualized in Figure 11c. Notable similarities between our clusters and Bruns' are evident in this updated approach. Namely, variations of prisoner's dilemma (Pd) are clustered in our approach and theirs: $(Pd \times \{Co, As, Sh, Pd\})$ (and their transposed analogs). The cyclic game cluster of $\{Ch, Ba, Hr\} \times \{Co, As, Sh\}$ (and their transposed analogs) are identically clustered in the approaches.

Note also that due to the Nash-payoff based classification scheme of Bruns, variations in the lower-left quadrant (where all Nash equilibria are win-win) are not evident in their approach (Table 1). By contrast, our approach highlights differences between these games. For example, whereas Bruns classifies the symmetric Coordination Game ($Co \times Co$) in the same class as Stag Hunt ($Sh \times Sh$), ours places it in its own class altogether (better matching intuition, due to the anti-coordination outcomes of Stag Hunt not being present in the coordination game).

With these comparisons in place, we further note several benefits of our approach compared to that of Bruns¹⁹ and related works on classification of 2×2 games. First, our approach does not rely on a hand-crafted taxonomy or classification of games; our games classes are identified automatically, and without need for human expertise in determination of patterns in payoffs. Second, as the taxonomy introduced by Bruns' relies on enumeration of payoff orderings, it faces significant scalability issues if to be extended to larger games (involving more strategies and/or players). By contrast, our approach applies directly to all normal form games, and we are not aware of similar automated classification schemes for larger, many-player games. Moreover, whereas Bruns approach provides a hard classification based on hard-coded structural patterns in payoffs, ours provides a soft classification based on a similarity metric (specifically, based on the games' principal components). Thus, our approach can be used to compute distances between games, in contrast to Bruns' and related approaches. We visualize several examples of 2×2 such game distances in Figures 11d to 11f. A subsequent benefit of being able to compute distances between games is that it enables computation of a dendrogram over a collection of games. We use our approach to visualize this tree for Bruns' 2×2 games in Figure 12. Finally, the downstream usefulness of our approach is that such a similarity metric can be optimized to generate new games (as illustrated in the main text), in contrast to the hard classification scheme of Bruns.

Sensitivity analysis

As mentioned in the main text, the empirical game-theoretic results rely on sampling of a representative set of policies to characterize the underlying games. An important limitation here is that the empirical game-theoretic results are subject to the policies used to generate them. Here we conduct a set of experiments to test the sensitivity of the results to these sampled policies.

For each game (with the exception of Rock–Paper–Scissors, which is a canonical game and only 3×3 in size), we randomly subsample 50% of the policies. All other policies are discarded, thus yielding an empirical payoff matrix 4 times smaller in overall size. We subsequently run the full analysis pipeline over 10 trials of subsampling per game (see Figure 13 for results). Figure 13a indicates the quartiles of the first two principal components per game, summarized over all trials. Despite 50% of the policies being randomly discarded, the overall statistics are relatively robust across the different trials. There is some sensitivity of the principal components to the empirical policies sampled in the process, which on closer inspection seems to occur for highly cyclical games such as Blotto variations and the Disc game. We additionally recompute the Landscape of Games figure, independently for several of the subsampling trials, with 4 examples shown in Figures 13b to 13e. Note that while there are some quantitative differences (in terms of relative position of these games in the projected space), the overall trends and clusters are quite robust to the subsampling (compared to Figure 1 of the main paper, which shows the landscape generated using all policies).

Overall, this analysis seems to indicate that the combination of our policy sampling scheme and analysis pipeline produce robust results, though as mentioned in the Discussions section of the main text, this will be a useful aspect of the method to revisit when experimenting on significantly larger games.

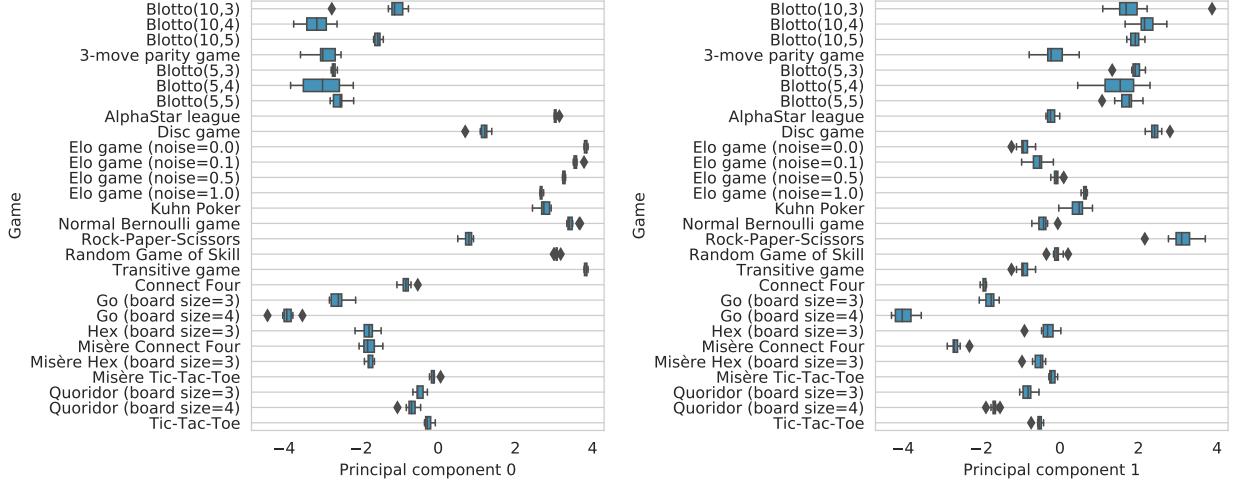
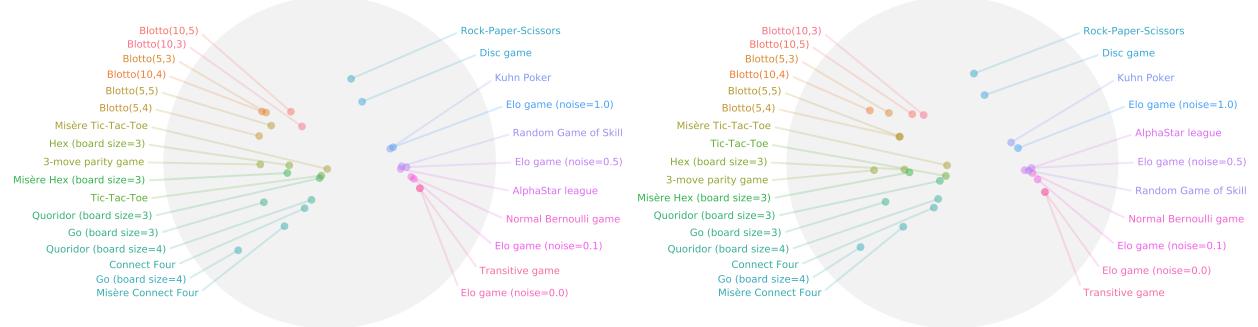
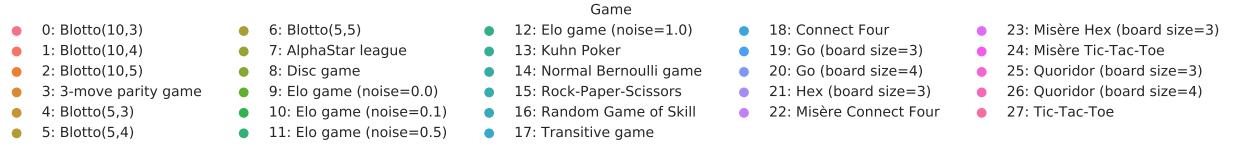
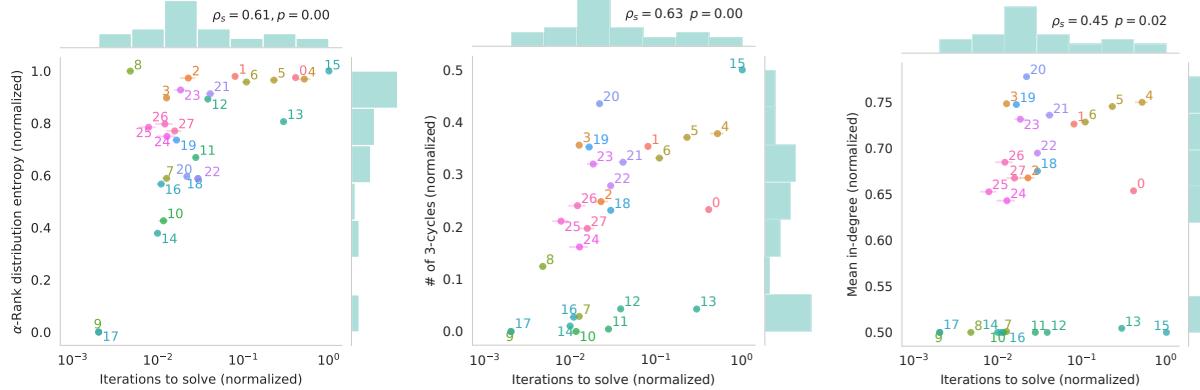
a**b****d**

Figure 13: Sensitivity to choice of empirical strategies, via subsampling of policy space. For each empirical game, we randomly subsample 50% of the policies (discarding the rest). We subsequently run the full analysis pipeline. 10 trials of subsampling are conducted per game (with the exception of Rock–Paper–Scissors, where we do not subsample strategies/policies due to it being a small canonical game). **a** indicates the quartiles of the first two principal components per game, over all trials. Specifically, the distribution of games' first two principal components under 10 trials of policy subsampling are illustrated, showing sensitivity to policy space changes. Boxplot elements are defined as follows: the box visualizes quartiles; whisker bars show data variability beyond the quartiles; the median is indicated by the center line; diamonds indicate outliers. **b–e** shows the landscape of games for 4 example trials. Note that game colors are kept the same as the original landscape of games visual (Figure 16a) for easier comparison.



a



b

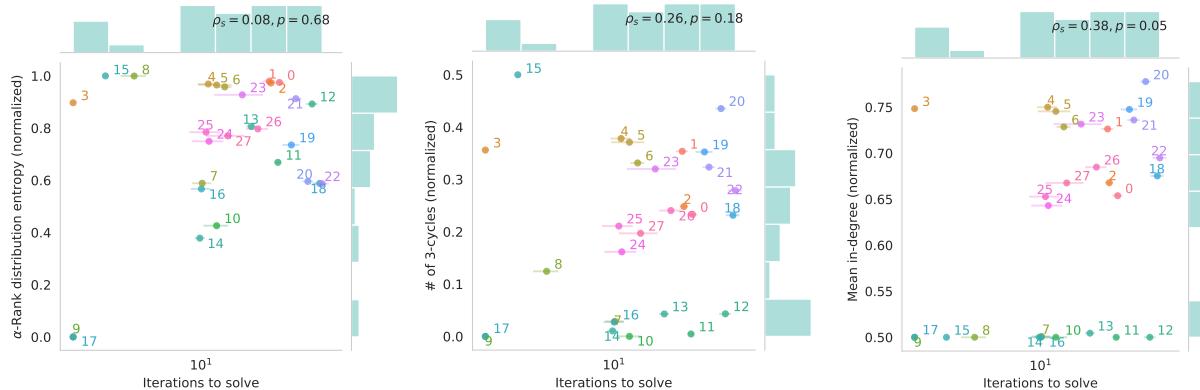
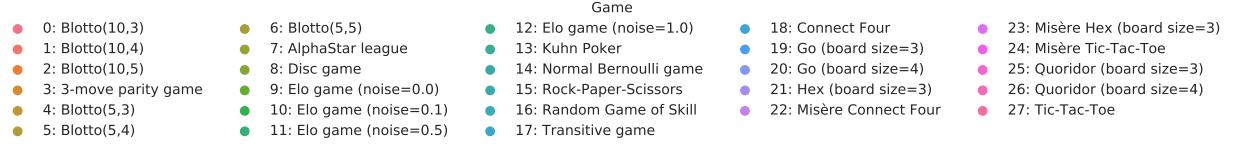


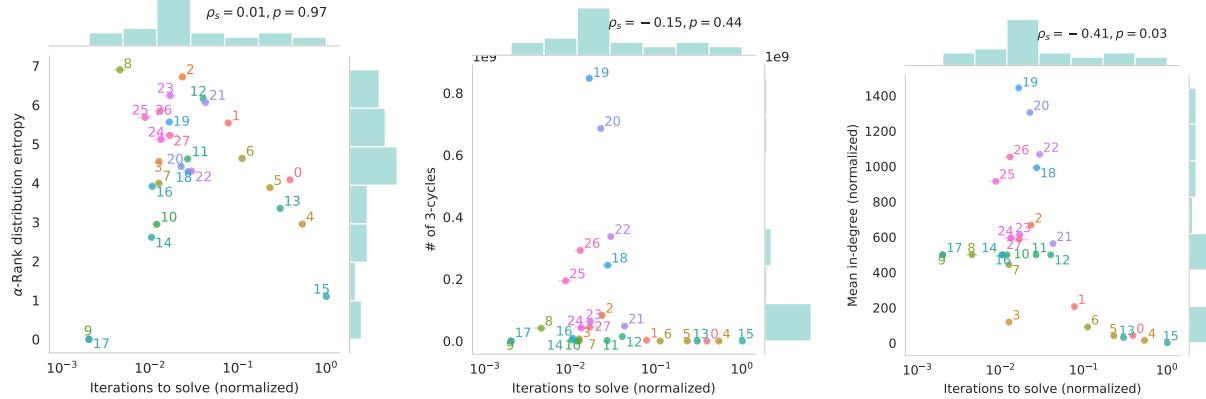
Figure 14: Response graph complexity vs. computational complexity of solving various games. Each column plots a respective measure of graph complexity against the number of iterations needed to solve the associated game via the double oracle algorithm. In both rows, the graph complexity measures are normalized. **a** shows results where the number of double oracle iterations is also normalized. **b** shows results where the number of double oracle iterations is not normalized.

Impact of normalization on complexity results

We conduct additional ablative analysis of the transformations of graph complexity measures considered in the main text. As noted in the main text, these results are not intended to propose that a specific definition of complexity (e.g., with respect to Nash) is explicitly useful for defining a topology / classification over games, but merely investigate correlations between these measures and those related to the raw response graphs. In Figures 14 and 15, we plot all complexity results, over all combinations of normalizing/not normalizing the graph measures (y-axes) and/or the number of iterations to solve (x-axes). Note that the case corresponding to the main paper is captured by Figure 14a, where both axes are normalized. The primary motivation for conducting this normalization (across both axes) is that the measures considered here would otherwise vary with increasing game size, which would make it easy to artificially inflate a game's complexity via arbitrary increase of the strategy space size via filler strategies.



a



b

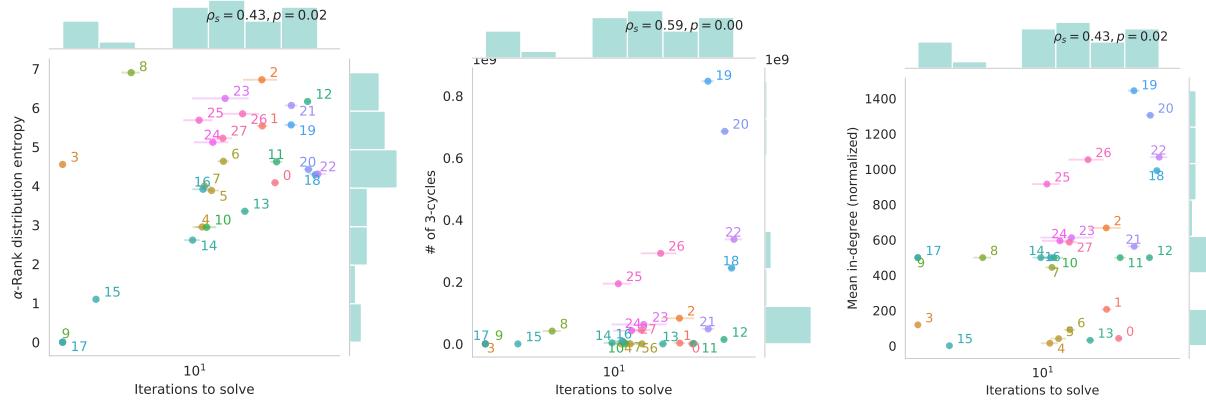
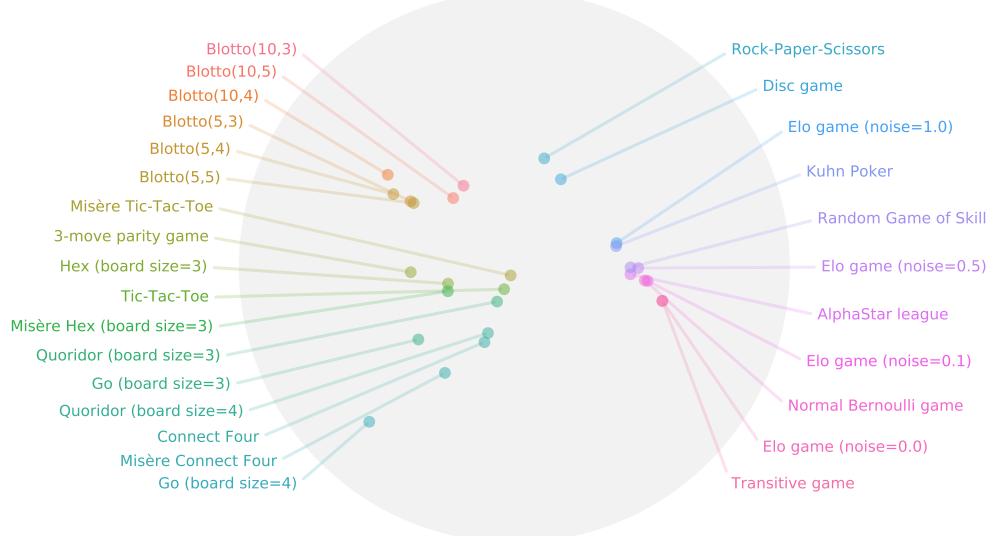


Figure 15: Response graph complexity vs. computational complexity of solving various games. Each column plots a respective measure of graph complexity against the number of iterations needed to solve the associated game via the double oracle algorithm. In both rows, the graph complexity measures are not normalized. **a** shows results where the number of double oracle iterations is normalized. **b** shows results where the number of double oracle iterations is not normalized.

Under this view, even the standard (i.e., non-repeated) version of Rock–Paper–Scissors has high computational complexity, in the sense that the Nash equilibrium requires these learning agents to discover and play the three pure strategies with equal probability. In other words, these results imply only that RPS is complex within the class of 3×3 games, as agents must fully explore the strategy space to equilibrate. Moreover, all larger variants of RPS be equally as complex under this view, in the sense that their Nash equilibria all have full support; thus, they would all deterministically require the same number of normalized iterations to solve (regardless of random initialization of the double oracle algorithm). Without this normalization, this particular notion of the complexity of a game could be inflated by artificially increasing the strategy space size, without affecting the underlying topology or type of strategic interactions needed to solve it.

Note that different (and potentially less useful) conclusions may indeed be drawn by considering ablations

a



b

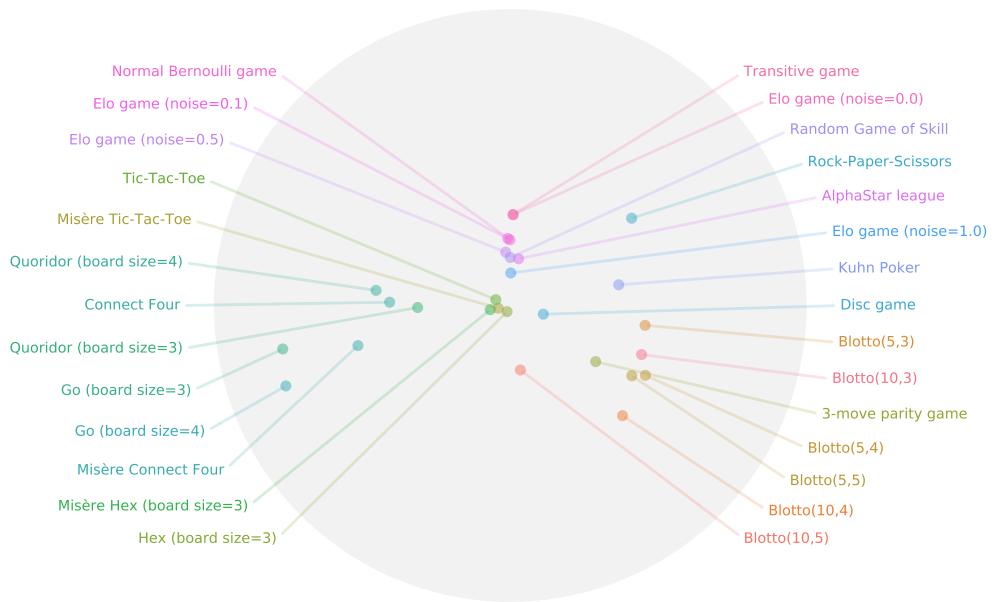


Figure 16: Landscape of games generated via normalized and unnormalized response graph measures. **a** Normalized graph measures. **b** Unnormalized graph measures.

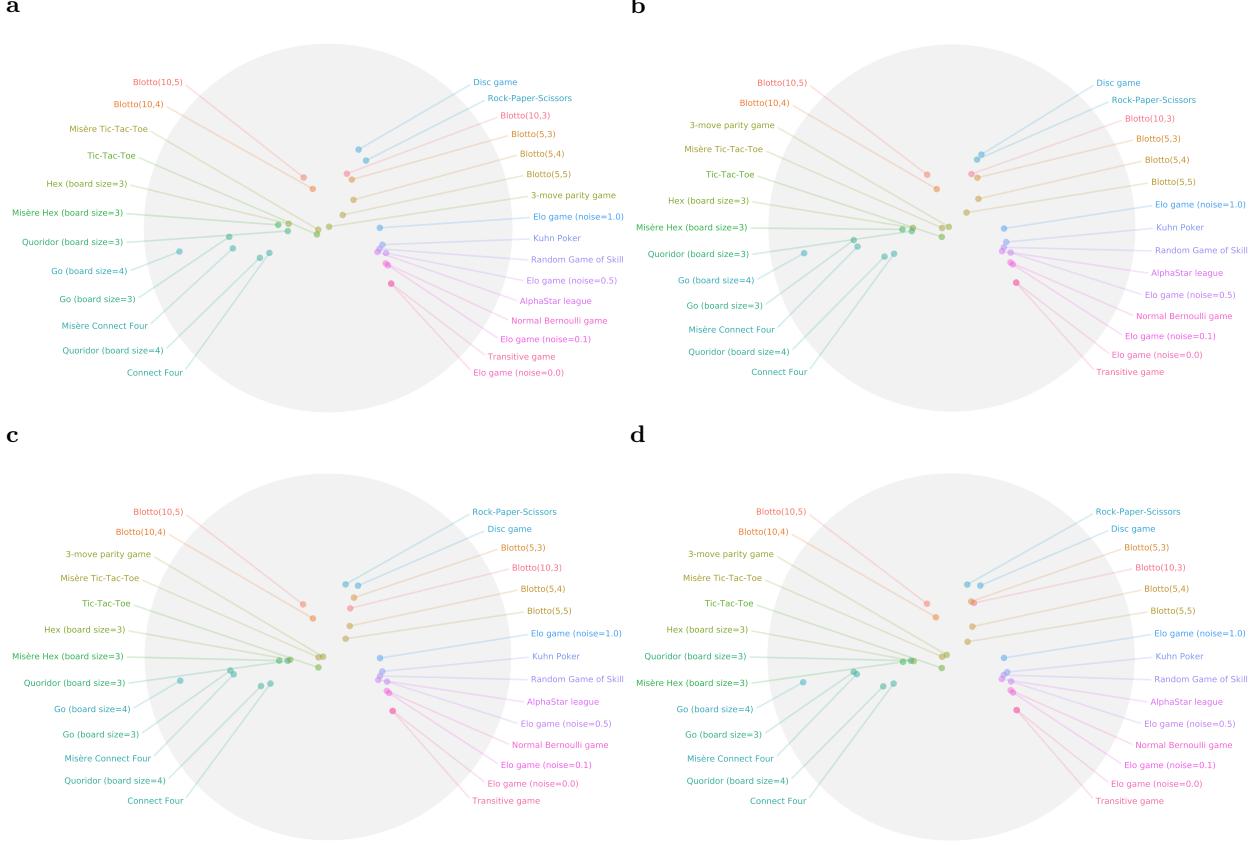


Figure 17: Sensitivity to random mixtures of policies. In each figure, 100 additional policies are included per game, and the landscape subsequently regenerated. Note that game colors are kept the same as the original landscape of games visual (Figure 16a) for easier comparison. **a-d** illustrate four such trials.

over the type of normalization done. For example, disabling this normalization yields Figure 14b, which implies that RPS has similar complexity to a fully Transitive Game (number 17), despite the latter’s Nash equilibrium being deterministically found after a single double oracle iteration. Moreover, in Figures 14 and 15, we find that normalizing both types of measures consistently maximizes their Spearman correlation (with low p-value, see top-right of each subfigure).

We also consider now the impact that this normalization has on the landscape of games figure (see Figure 16). Here, note that there are only 2 ablations involving only normalization over the graph measures (as double oracle/iterations to solve are not used in the spectral analysis). The normalized landscape (Figure 16a), used in the main paper, reveals coherent clusters of related games. By contrast, the variety of underlying payoff table sizes and associated graph measures implies that the unnormalized landscape (Figure 16b) is notably less structured. For instance, while games such as Rock–Paper–Scissors and the Disc Game are conceptually very similar, their distance is relatively larger in the unnormalized landscape due to their widely different payoff sizes (3×3 vs. 1000×1000 , respectively).

Effects of mixed policies

It is interesting to consider the effect that additional mixing of policies would have on the results, and conduct a suite of experiments focused on this here. Specifically, for all games in the landscape, we expand each payoff table by adding mixtures of policies. For each game, we uniformly sample a mixture over a random selection of half of the base policies; we repeat this 100 times, expanding the payoff table accordingly. For Rock–Paper–Scissors (due to the small size of the strategy space), we sample mixed policies uniformly over the full support of all 3 base strategies. Figure 17 shows the results for 4 independent trials of policy mixing.

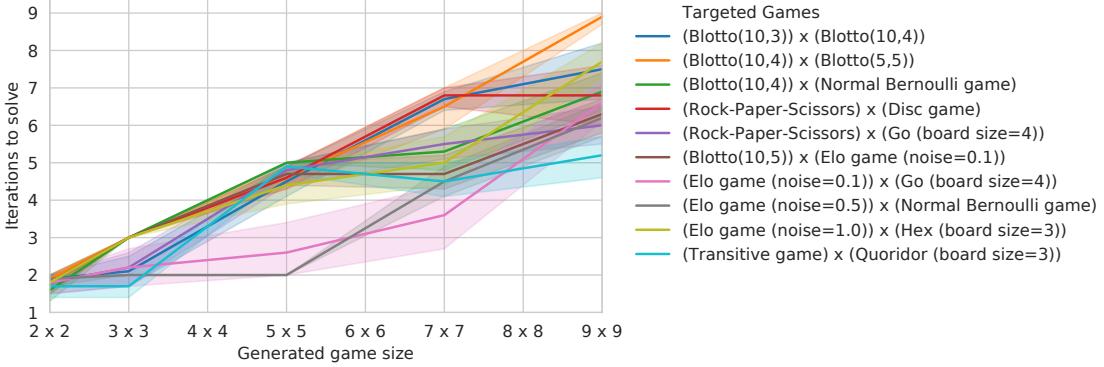


Figure 18: Iterations to solve procedurally-generated game structures of varying sizes. Error ranges indicate 95% confidence intervals over 10 trials.

Some observations can be made here, in comparing the trials to one another (and to the original landscape, visualized in Figure 16a). At a high-level, most of the prominent clusters found in the landscape presented in the main paper are also present here, with some specific details as follows: 1) Rock–Paper–Scissors and the Disc game (closely clustered in both, despite the size of Rock–Paper–Scissors increasing from 3×3 in the original landscape to 103×103 in the mixed policy landscapes) 2) Elo game($\text{noise}=0.0$) and Transitive game; 3) Elo game($\text{noise}=0.1$) and the Normal Bernoulli game; 4) Random Game of Skill, Elo game ($\text{noise}=0.5$), and the AlphaStar League; 5) Real-world games (e.g., Connect Four and Quoridor (board size=4)); 6) The cluster of Blotto games have somewhat shifted towards that of Rock–Paper–Scissors and the Disc Game. Notably, all of these games are highly cyclical, and Blotto requires players to play uniformly across all permutations of the token-selection strategies (due to the game rules itself being permutation-invariant), a characteristic shared by Rock–Paper–Scissors.

A closer look at generated games

While raw hybrids of games alone are unlikely to play a key role in establishing useful curricula, we note that discovery of core features or mechanisms in games has, indeed, been a key driver of generation of new games (see, e.g., Charity et al.²⁴, Khalifa et al.⁷⁴, Shaker et al.¹³¹). As such, we can observe interesting trends even in the specific class of normal-form games that we focus on here.

To more closely analyze trends in the games generated in the main paper, we conduct experiments that procedurally generate and subsequently train policies on these games via the double oracle algorithm. Specifically, we create a large class of generated games as follows. We first randomly sample 10 pairs of target games from the base games shown in the landscape of games (Figure 16a). We subsequently procedurally generate games of sizes 2×2 , 3×3 , 5×5 , 7×7 , and 9×9 targeting each sampled pair. This yields a total of 50 generated games of varying characteristics. We subsequently evaluate the complexity of solving these games via double oracle (of course, these can also be used for analysis of the dynamics of multiagent learning algorithms, as done in recent works such as Bloembergen et al.¹¹, Hennes et al.⁶². Figure 18 summarizes the results of this experiment, with each generated instance evaluated using 10 randomly-seeded trials of the double oracle algorithm.

We notice several trends in these generated games. First, the generated games targeting the three pairs $(\text{Blotto}(10,3) \times \text{Blotto}(10,4))$, $(\text{Blotto}(10,4) \times \text{Blotto}(5,5))$, and $(\text{Rock–Paper–Scissors} \times \text{Disc game})$ consistently require the largest number of iterations to solve, across all game sizes. This result is explained by the fact that each of these games targets a mixture over pairs of highly-cyclical underlying games; thus, each generated game likely has a large Nash support (despite the noise in the generation process), making them difficult to solve. Next, we observe a lower bound roughly established by generated games involving the target pairs $(\text{Elo game } (\text{noise}=0.1) \times \text{Go (board size}=4))$ and $(\text{Elo game } (\text{noise}=0.5) \times \text{Normal Bernoulli Game})$. Here, the noisy Elo game plays an important role, as its (roughly) transitive structure tends to reduce the number of strong strategies, thereby making it easier to solve. Interestingly, for the largest generated

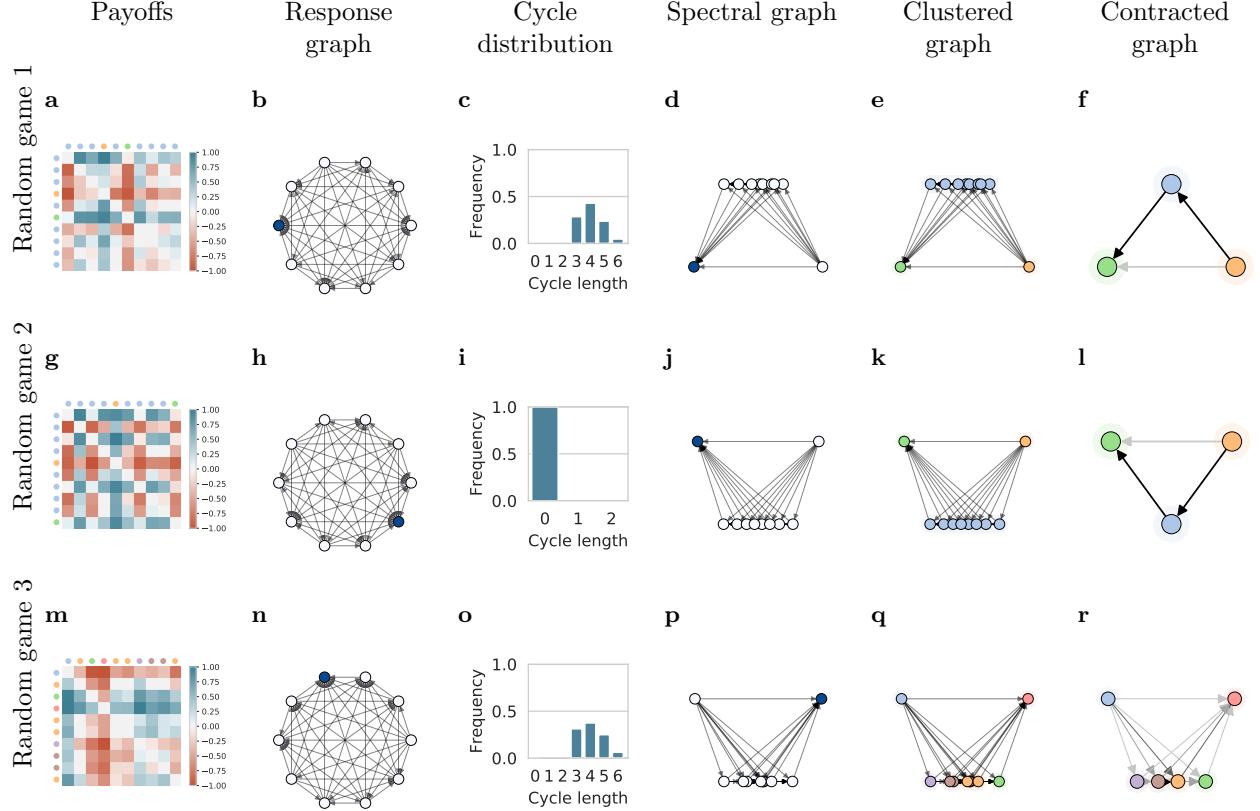


Figure 19: Results for randomly-generated games. Each column visualizes a different characteristic of the game or response graph, as discussed in the main text.

game size (9×9), the pair (Transitive game \times Quoridor (board size=3)) requires the fewest iterations to solve. Effectively, the lack of noise in the Transitive game dominates the structure of these larger generated games, in contrast to the noisy Elo games. Finally, the remaining generated games form a cluster with an intermediate number of iterations needed to solve them. This latter class of games primarily targets a mixture of cyclical and transitive games (e.g., (Blotto(10,5) \times Elo game (noise=0.1))), linking to the notion of interestingness from an AI perspective, as discussed in the main text.

Overall, generation of even the simple classes of games considered here exemplifies how one might use such an analysis to highlight relationships between generated games and existing, well-studied ones, to better understand and expand the space of multiplayer games.

Motivating examples: results for randomly-generated games

Figure 19 illustrates a sample of response graphs for generated games of random structure, as discussed in the Motivating Examples section of the main text.

Additional response graph analysis

For completeness, Figures 20 to 25 present the response graph-based analysis for the additional games considered in the main text. Figure 20 is of particular note here, as it exemplifies the application of the methodology to asymmetric, many-player games. Specifically, the empirical games constructed here correspond to agents trained via extensive-form fictitious play (XFP)⁶¹ in 2-, 3-, and 4-player variants of Kuhn Poker.

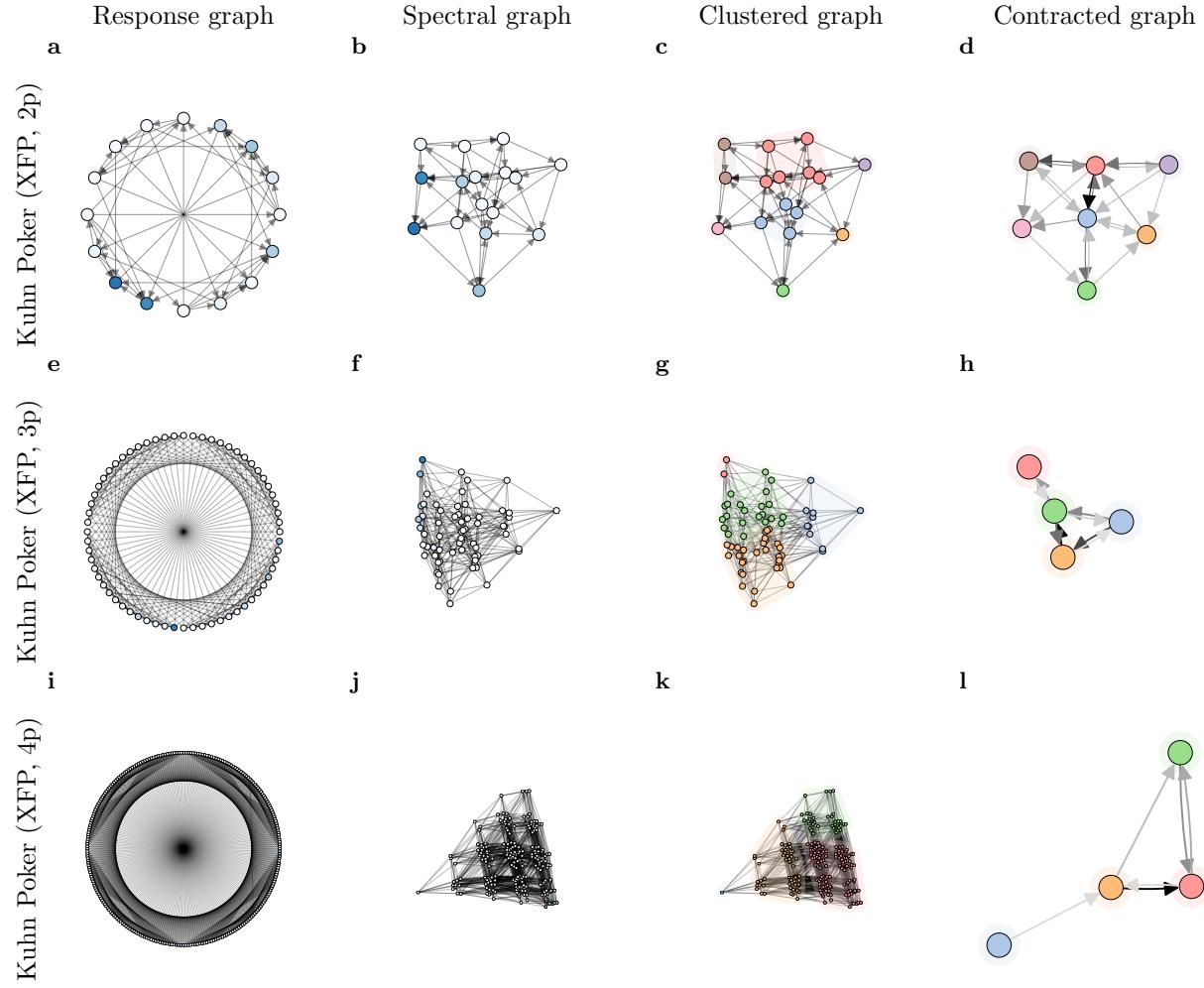


Figure 20: Additional response graph analysis results I. Specifically, these results pertain to Kuhn Poker agents trained via extensive-form fictitious play (XFP)⁶¹, with empirical games constructed as detailed in Omidshafiei et al.¹⁰⁶.

Additional complexity results

Figure 26 provides an overview of additional response graph-based measures, in comparison to the normalized number of iterations required to solve each of the games considered in the main text.

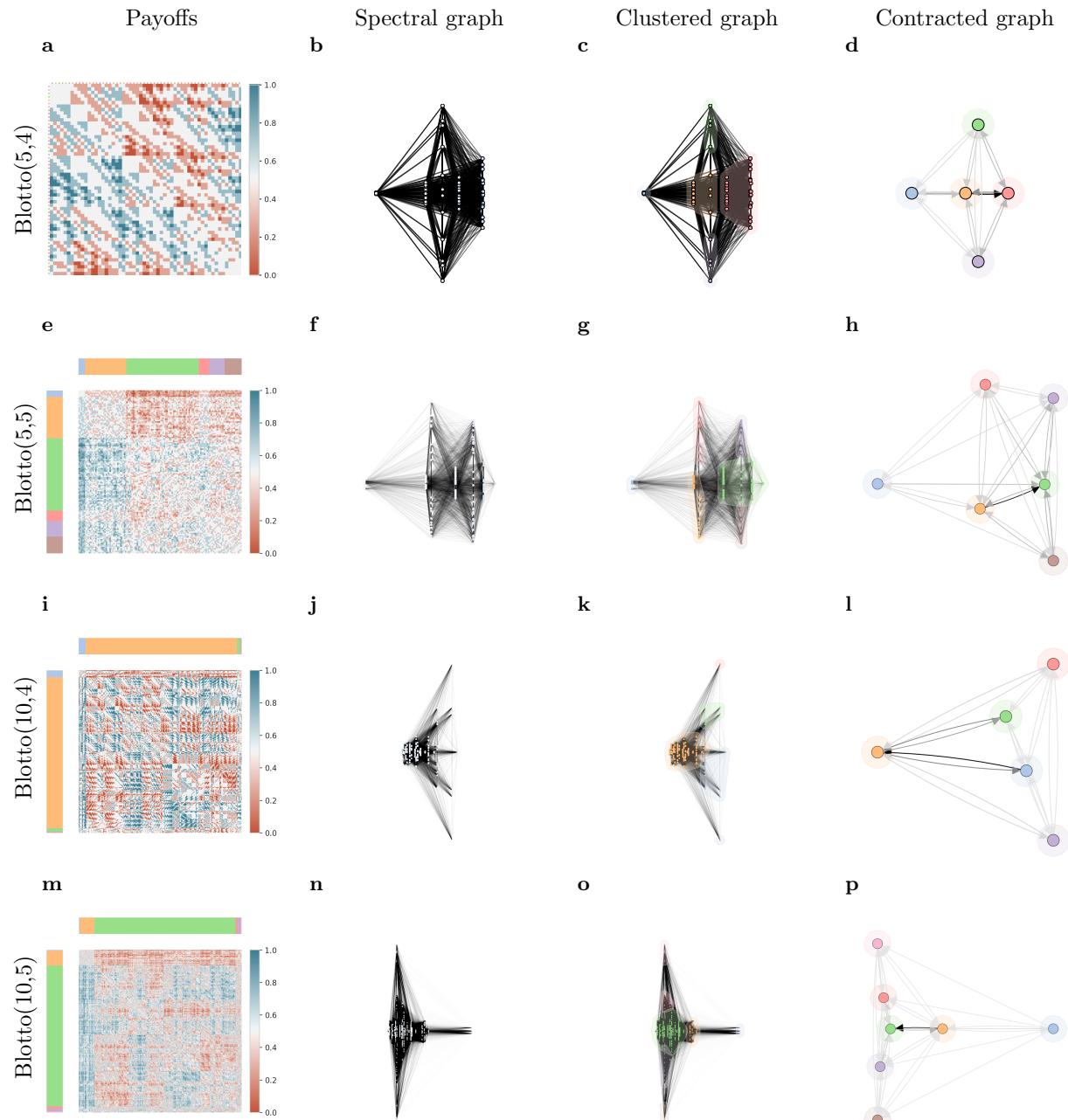


Figure 21: Additional response graph analysis results II. Each column visualizes a different characteristic of the game or response graph, as discussed in the main text.

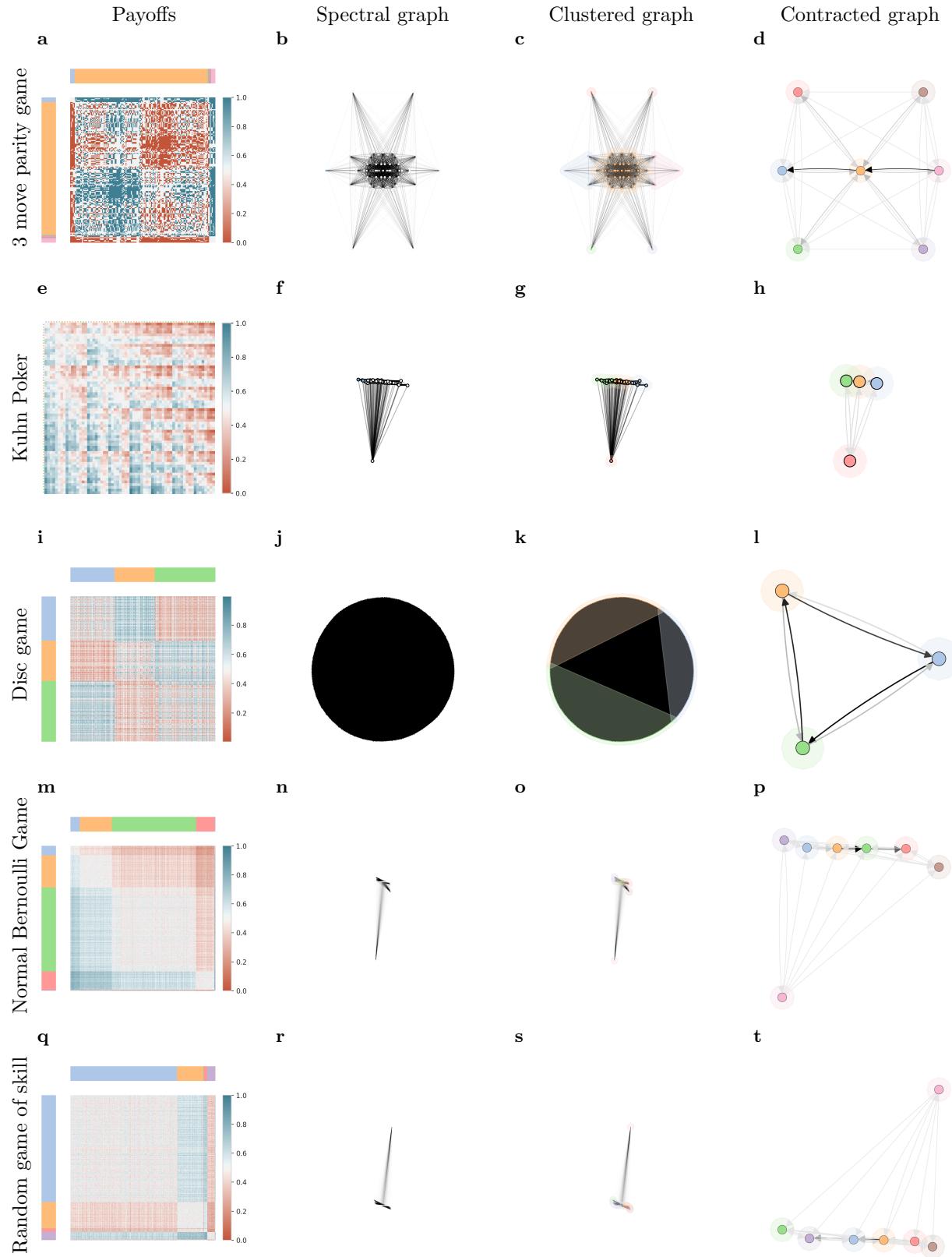


Figure 22: Additional response graph analysis results III. Each column visualizes a different characteristic of the game or response graph, as discussed in the main text.

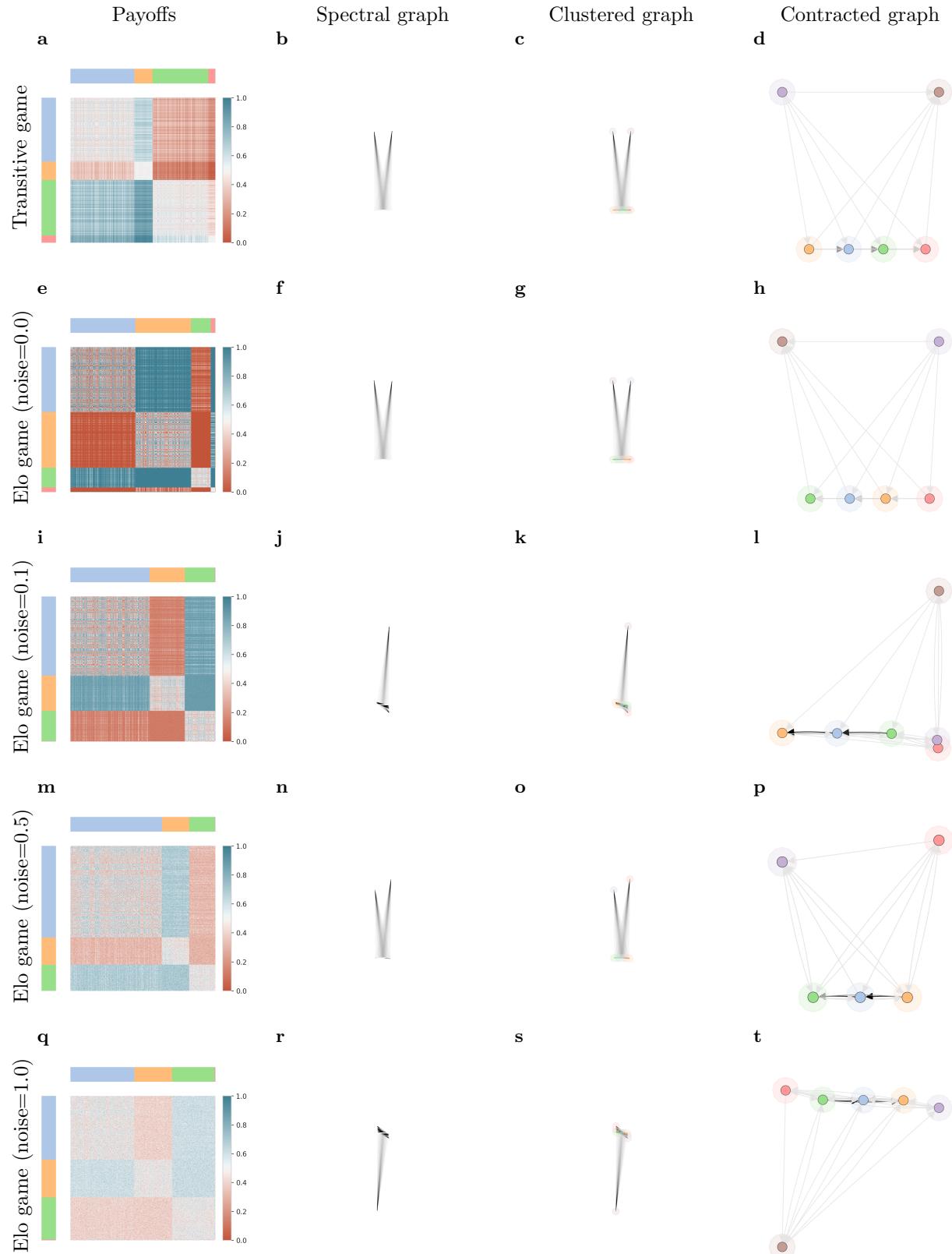


Figure 23: Additional response graph analysis results IV. Each column visualizes a different characteristic of the game or response graph, as discussed in the main text.

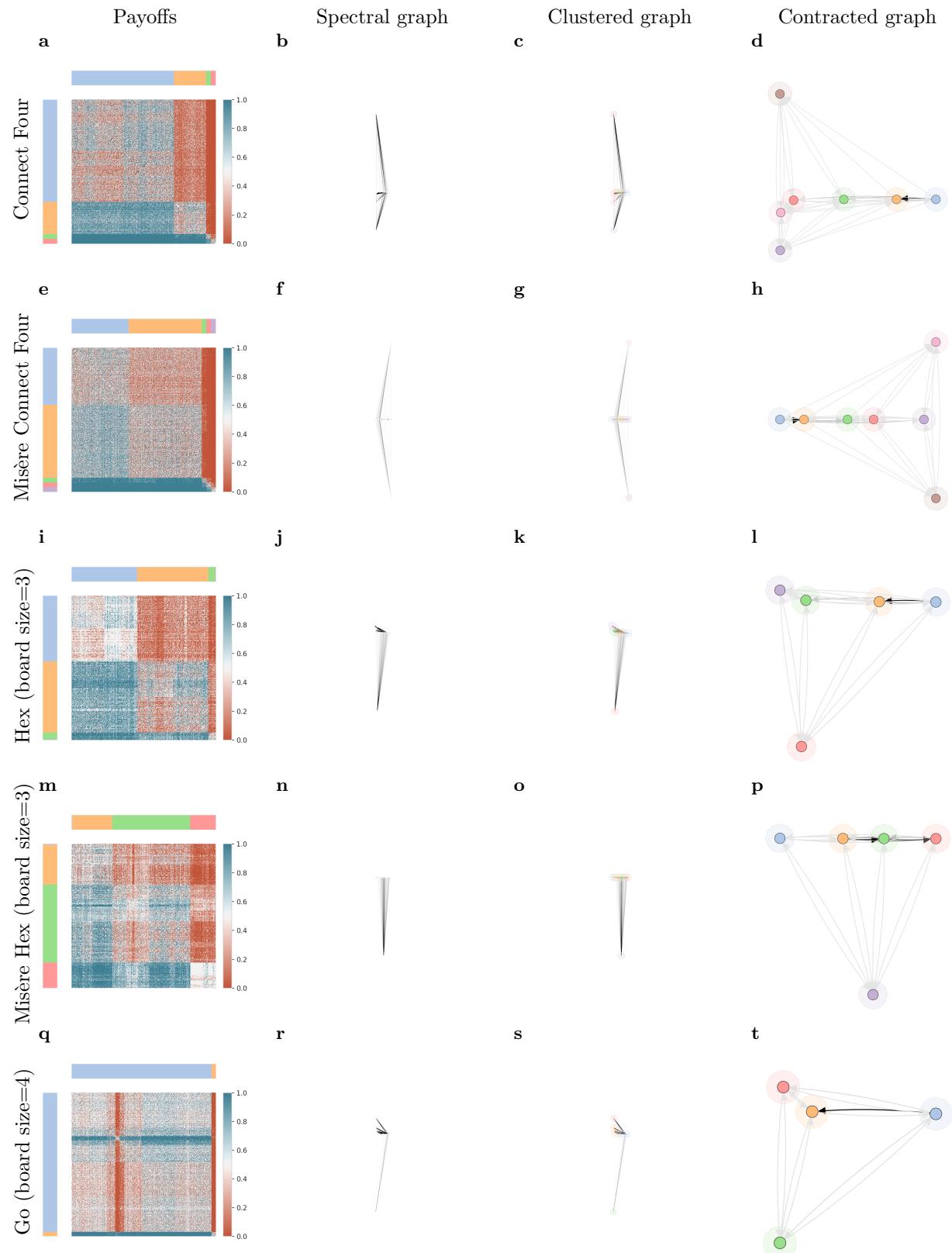


Figure 24: Additional response graph analysis results V. Each column visualizes a different characteristic of the game or response graph, as discussed in the main text.

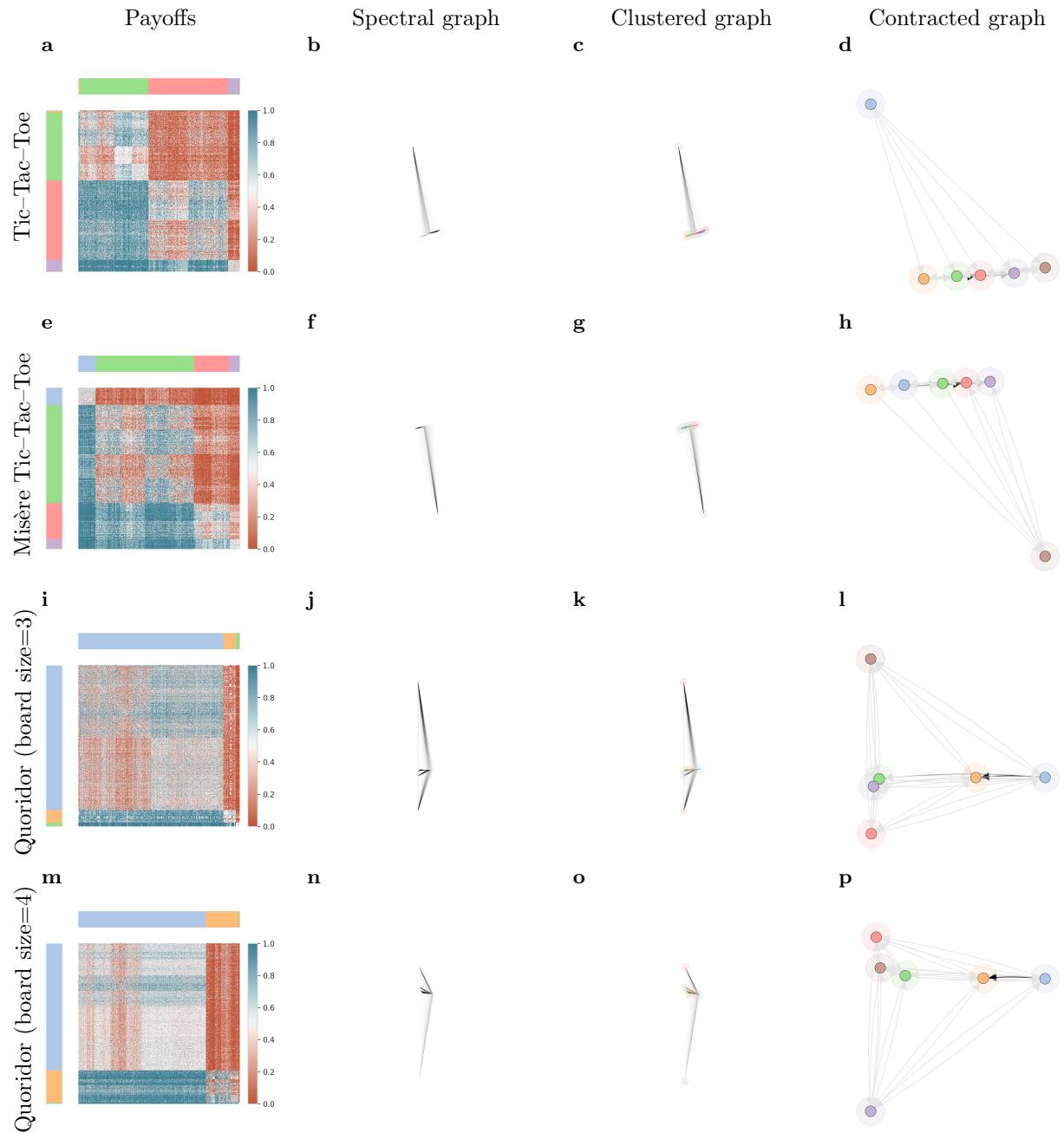


Figure 25: Additional response graph analysis results VI. Each column visualizes a different characteristic of the game or response graph, as discussed in the main text.

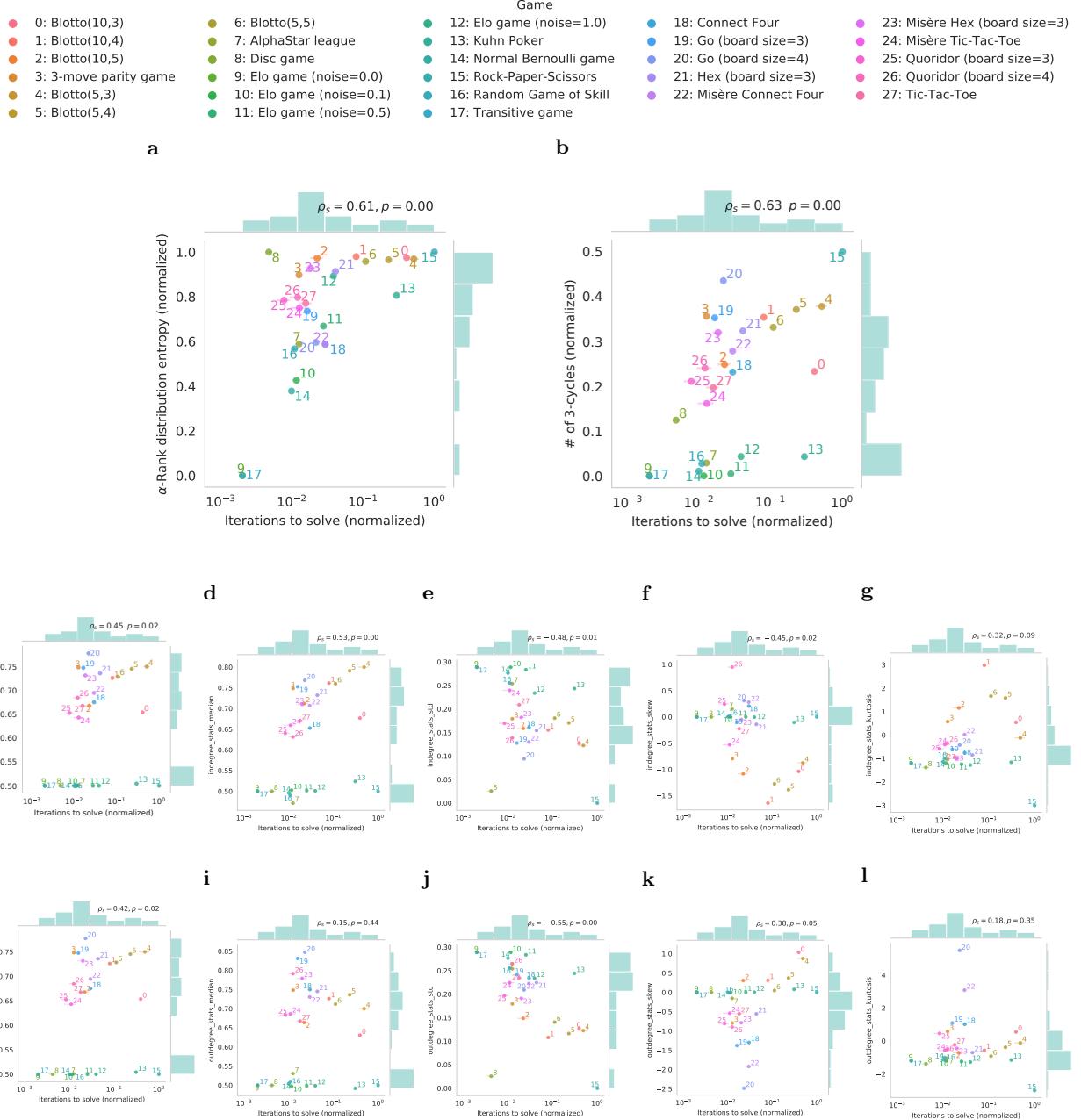


Figure 26: Response graph complexity vs. computational complexity of solving associated games. Each figure plots a respective measure of graph complexity against the normalized number of iterations needed to solve the associated game via the double oracle algorithm (with normalization done with respect to the total number of strategies in each underlying game). Note that mean node-wise in- and out-degrees (in **c** and **h**, respectively) match across the games here due to the degree sum formula; other distributional statistics, however, do not necessarily match across in- and out-degrees, as evident above.