Estimating α-Rank from A Few Entries with Low Rank Matrix Completion

Yali Du [*] [1] **Xue Yan** [*] [2] **Xu Chen** [3] **Jun Wang** [1] **Haifeng Zhang** [2]

Abstract

Multi-agent evaluation aims at the assessment of an agent's strategy on the basis of interaction with others. Typically, existing methods such as α-rank and its approximation still require to exhaustively compare all pairs of joint strategies for an accurate ranking, which in practice is computationally expensive. In this paper, we aim to reduce the number of pairwise comparisons in recovering a satisfying ranking for n strategies in two-player meta-games, by exploring the fact that agents with similar skills may achieve similar payoffs against others. Two situations are considered: the first one is when we can obtain the true payoffs; the other one is when we can only access noisy payoff. Based on these formulations, we leverage low-rank matrix completion and design two novel algorithms for noise-free and noisy evaluations respectively. For both of these settings, we theorize that $O(nr \log n)$ (n is the number of agents and r is the rank of the payoff matrix) payoff entries are required to achieve sufficiently well strategy evaluation performance. Empirical results on evaluating the strategies in three synthetic games and twelve real world games demonstrate that strategy evaluation from a few entries can lead to comparable performance to algorithms with full knowledge of the payoff matrix.

1. Introduction

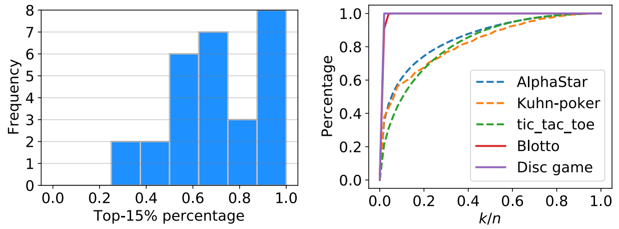
Evaluation of multi-agent strategies is of vital importance to drive the progress of learning strategies in combating various tasks. In the multi-agent reinforcement learning community, renowned evaluations include Elo, TrueSkill, mElo2k, α-rank, (Elo, 1978; Herbrich et al., 2006; Balduzzi et al., 2018; Omidshafiei et al., 2019). The Elo and TrueSkill both deal with ratings of agents in the pairwise competitions. Though it is simple to implement and widely used, it is only limited to transitive scenarios and is not suitable for games like rock-paper-scissors of which consistent winners do not exist. Baking in Hodge decomposition theory (Jiang et al., 2011), Balduzzi et al. introduces multi-dimensional elo (mElo2k) that decomposes a game into transitive and cyclic components to tackle intransitive evaluations and computes nash-averaging to evaluate different strategies. α-rank (Omidshafiei et al., 2019) is the most advanced algorithm which can both tackle intransitive evaluations and be tractably computed in general games with more than two players, such as Mahjong, Poker. It constructs Markov transition matrix based on payoffs of joint strategies and the invariant distribution of the constructed Markov chain yields the strategy profile rankings.

Despite these advances, existing evaluating algorithms, however, need to evaluate all joint strategy profiles to obtain the raw payoffs, before computing nash-averaging or α-rank. Exhaustively comparing any pair of agents is neither feasible for real-world matches such as football matches, nor computation-efficient for computerized agents evaluation such as in game AI. A single Go match for two players can take one hour to finish. It is even longer for some team-based matches, such as football and basketball. On one hand, each agent requires many interactions against agents (or tasks) to obtain a confident estimate of expected winning rate (or performance). On the other hand, exhaustive evaluation of any joint strategy profiles further increases the burden of computation. Rowland et al. (2019) approximates α-rank from incomplete data and examines how many games are needed for each agent pair, in order to achieve accurate evaluation of α-rank. Rashid et al. (2021) selects the agent pairs whose payoff has highest expected information gain to a belief over α-ranks. However, computationally this is still expensive as all the agent pairs have to be enumerated and compared.

In this paper, we consider the evaluation of meta-strategies in two-player games, and we extend the approximation of α-rank in (Rowland et al., 2019) by removing the necessity of exhaustively comparing all agent pairs. Our study is based on the observation that an agent's performance is not independent. Agents who have similar skills might perform

[*]Equal contribution [1] University College London, UK [2] Institute of Automation, Chinese Academy of Sciences [3]Beijing Key Laboratory of Big Data Management and Analysis Methods, GSAI, Renmin University of China. Correspondence to: Yali Du <yali.dux@gmail.com>, Haifeng Zhang <haifeng.zhang@ia.ac.cn>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).



(a) Histograms of 15% singular values percentage

(b) Percentage of top singular values

Figure 1. Illustrations of ranks and singular values for 28 real-world games from Czarnecki et al. (2020). (a) Histograms of top 15% of singular values' percentage. We take top-k ($k = \lfloor 15\% \times n \rfloor$) singular values then compute their percentage in the sum of all singular values. (b) Percentage of top-k singular values growing with k. x-axis is k normalized by n.

similarly. This is evidenced by the low rank nature of competition payoff matrix between agents, as shown in Figure 1, for multiple multi-agent competition performance data sets. This allows us to build our framework based on the theory of low-rank matrix completion to reduce the computation burden (Candes & Plan, 2010). Specifically, given the payoff of a set of randomly chosen pairs, we build an incomplete pairwise rating matrix $\boldsymbol{M}^{\Omega}$ that contains these observed entries. Then we apply a (low-rank) matrix completion algorithm to recover a completed rating matrix $\overline{\boldsymbol{M}}$, and then apply α-rank to obtain the final ranking of all strategies (see Figure 2). When the noise-free payoff is not accessible, we sample pairwise comparisons to obtain empirical evaluation. We theoretically justify that $O(nr\log n)$ (n is number of agents and r is the rank of the payoff matrix) pairs of comparisons are required to achieve sufficiently well evaluation performance in both noise-free and noisy settings.

We have tested our evaluation solutions in three synthetic games and twelve real-world games. Our empirical contributions are three-fold. Firstly, we demonstrate that in the noise-free case, a much lower number of pairwise evaluations (compared to n^2) can lead to accurate α-rank performance in terms of ranking error and convergence measure. Secondly, in the noisy setting, we observe that OptEval can achieve comparable performance compared to baselines that require the complete payoff matrix. Lastly, in real world games, we show that a lower rank approximation of payoff matrices can achieve satisfactory performance. The demo and code for this project are released under `https://github.com/yalidu/optEval.git`.

2. Related work

Evaluation has wide applications in machine learning in ranking of agents (Silver et al., 2017; Lai, 2015; Arneson et al., 2010; Gruslys et al., 2018) and in improving the searching for stronger strategies (Muller et al., 2020; Czarnecki et al., 2020). Below we review the famous evaluation algorithms.

Evaluation algorithms Elo and α-rank and two renowned evaluation algorithms. The Elo (Elo, 1978) is widely used in scenarios where two players are competing, which assigns a rating score to each player (Silver et al., 2017; Lai, 2015; Arneson et al., 2010; Gruslys et al., 2018). The score is updated based on the result of both players losing or winning. Ratings can be used not only to rank players, but also to quantify their abilities, further predicting the probability of winning a match, and so can be used for opponent matching. TrueSkill (Herbrich et al., 2006) generalized Elo by handling player skill uncertainties under Bayesian framework. While it can only be applied to transitive scenarios, Multidimensional Elo (mElo2k) improves Elo to handle intransitive strategies in two-player, zero-sum settings (Balduzzi et al., 2018; Tuyls et al., 2018), which proposed to use Combinatorial Hodge to decompose a game into the sum of the transitive component and the cyclic component. Then it adopts Nash Averaging for evaluation, but it is not generally applicable due to the intractability of computation and selection of the Nash equilibrium (Harsanyi et al., 1988; Daskalakis et al., 2009). α-rank (Omidshafiei et al., 2019; Yang et al., 2020) is inspired from evolutionary theories and is more general than Elo in ranking of n-player ($n \geq 2$) and handling intransitive abilities. α-rank constructs a Markov transition matrix based on the payoffs of the joint strategy profiles, called response graph (Lanctot et al., 2017), and solves the ranking by computing the unique invariant distributions.

Noisy payoffs Both Elo-based systems and α-rank assume the noise-free payoff table is available. However, it is seldom available in empirical games, and we have to let agents pit against each other for a sufficient number of times to obtain confident statistics about the meta-payoffs. Rowland et al. (2019) theorized that at least how many simulations are needed to get a confident evaluation for each strategy profile. α-IG (Rashid et al., 2021) improves α-rank by choosing pairs that have the largest information gain to ranking results.

However, they all do not consider the repeated strategies in the empirical games. Our work aims to reduce the number of evaluations from a different perspective: we focus on the number of pairwise evaluations required for accurate ranking of n strategies. We consider both noise-free and noisy payoffs. In the later case, we build our algorithm and theory based on ResponseGraphUCB (RG-UCB) (Rowland et al., 2019).

Ranking and low-rank In many game scenarios, there may exist repeated or similar agents in multi-agent systems.

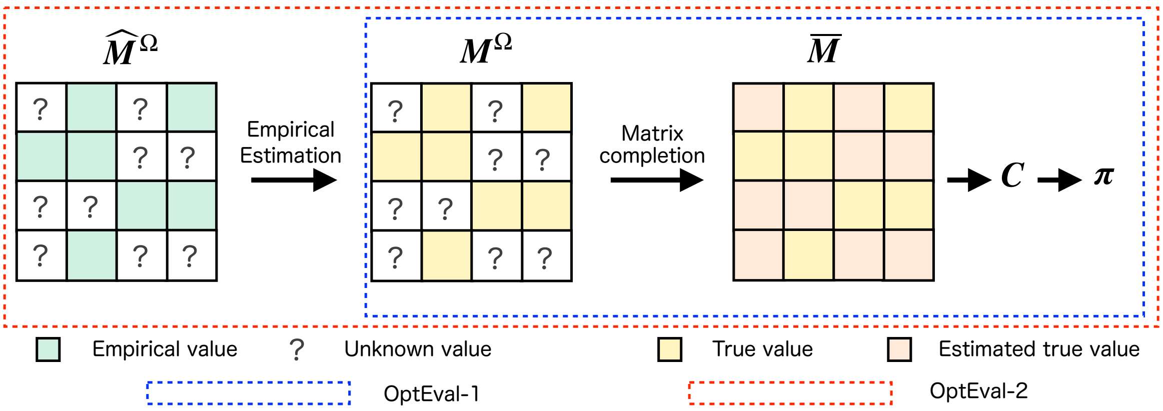


Figure 2. Illustrations of OptEval. The blue box shows the process of OptEval-1 which performs α-rank based on noise-free payoff table. The red box shows the process of OptEval-2 based on noisy payoff table, of which $\widehat{M}^{\Omega}$ represents an empirical estimation of true payoffs in M^{Ω} .

So low rank properties can be used for evaluation in these scenarios. For a matrix with low-rank attributes, a rich literature studies the exact recovery guarantees (Candes & Recht, 2012; Candès & Tao, 2010; Recht, 2011). Keshavan & Oh (2009) uses a small number of randomly selected entries with noise to reconstruct the matrix. Rajkumar & Agarwal (2016) adapts low-rank property in solving classic ranking problems. It is claimed that some special stochastic transitive models such as the Bradley-Terry-Luce model have low rank properties, thus Rajkumar & Agarwal (2016) used the low rank matrix completion algorithm to estimate the winning probability matrix. Our work leverages OptSpace algorithm (Keshavan & Oh, 2009) to perform matrix completion on the incomplete payoff matrices.

3. Methodolody

3.1. Preliminaries

Games and α-rank We consider two-player meta-games, each of whom can play S strategies, indicating that they are sharing the same population of strategies. Let $M(i,j)$, or M_{ij} for short, denote the payoff when the first player plays S_i and the second player plays S_j. At a higher meta-level, a strategy $s \in S$ corresponds to a machine learning agent and the matrix M captures expected payoffs when agents play against one another in some given tasks. From this perspective, the "agents" and "strategies" are synonyms in this paper.

We consider the α-rank $r \in \mathcal{R}^{|S|}$ (Omidshafiei et al., 2019). According to the setting of α, for the finite-α case, we can calculated the Markov transition matrix C as below:

$$\mathbf{C}_{i,j} = \begin{cases} \eta \frac{1-\exp(-\alpha(M_{ji}-M_{ij}))}{1-\exp(-\alpha p(M_{ji}-M_{ij}))} & \text{if } M_{ji} \neq M_{ij}, \\ \frac{\eta}{p} & \text{otherwise ,} \end{cases} \tag{1}$$

where the coefficient η is defined as $\eta = \frac{1}{|S|-1}$. $\alpha \geq 0$, $p \in \mathbb{N}$ are hyperparameters to be chosen. And $\forall i \in [n]$, $C_{ii} = 1 - \sum_{j \neq i} C_{ij}$ ensures that transition probabilities are valid. The invariant distribution of this Markov chain π yields the score of strategy profiles. We denote $\widehat{M}$ as the empirical payoff matrix, where $\widehat{M}_{ij}$ is the empirical payoff measured by agent i pitting against agent j for a plausible number of times. Full details of α-rank can be found in Appendix A.

3.2. Algorithms

We consider the evaluation under two scenarios: games with noise-free and noisy payoffs and propose two algorithms, named OptEval-1 and OptEval-2 respectively. The main idea is to leverage low rank matrix completion algorithms to recover the complete payoff matrix based on a few observations, and then apply α-rank to generate the rankings of all strategies. The main difference between the two algorithms is that, entries are evaluated exactly (yellow square in Figure 2) in OptEval-1 and are estimated empirically (green square in Figure 2) in OptEval-2. We present the details of the two algorithms below.

OptEval-1 Suppose we can get exact payoff, we propose an algorithm OptEval-1 to precisely evaluate n strategies only through evaluation of a small portion of pairs. We define a sampling operator $\Omega \in [n] \times [n]$ which randomly samples m pairs. For each pair $(i,j) \in \Omega$, we can get the true payoff M_{ij} and the true payoff on Ω is M^{Ω}. Thus we conduct rank $\hat{r}$ approximate matrix completion on M^{Ω}. The recovered payoff matrix is denoted as $\overline{M}$, then we perform α-rank on $\overline{M}$ and get the invariant distribution of n strategies. This framework is compatible with many low-rank matrix completion algorithms with exact recovery guarantee (Candes & Recht, 2012; Recht, 2011). Here we choose OptSpace (Keshavan & Oh, 2009), which takes as input a parameter rank $\hat{r}$ and randomly sampled M^{Ω}. OptSpace

Algorithm 1 OptEval-1: estimating α-rank with noise-free payoff.

```
Input: n strategies, a chosen rank $\hat{r}$, sampling operator $\Omega \in [n] \times [n]$.
Output: The invariant distribution $\overline{\boldsymbol{\pi}}$ of $n$ strategies.
  Randomly sample $m$ pairs from the entire sample space $[n] \times [n]$ by the sampling operator $\Omega$.
  Get pairwise comparison results $\boldsymbol{M}^{\Omega}$
  Calculate the reconstructing payoff matrix $\overline{\boldsymbol{M}}$ according to OptSpace with rank $\hat{r}$.
  Construct the Markov chain $\overline{\boldsymbol{C}}$ through Eq. (1)
  Solve the invariant distribution $\overline{\boldsymbol{\pi}}$ of $\overline{\boldsymbol{C}}$
  Return $\overline{\boldsymbol{\pi}}$
```

Algorithm 2 OptEval-2: estimating α-rank with noisy payoff.

```
Input: $n$ strategies, a chosen rank $\hat{r}$, a sampling operator $\Omega \in [n] \times [n]$
Output: The invariant distribution $\overline{\boldsymbol{\pi}}$ of $n$ strategies.
  Randomly sample $m$ pairs from the entire sample space $[n] \times [n]$ by $\Omega$.
  Call RG-UCB on $\Omega$ to get noisy pairwise comparison results $\widehat{\boldsymbol{M}}^{\Omega}$
  Perform OptSpace on $\widehat{\boldsymbol{M}}^{\Omega}$ with rank $\hat{r}$ and calculate the reconstructing payoff matrix $\overline{\widehat{\boldsymbol{M}}}$
  Construct the Markov chain $\overline{\widehat{\boldsymbol{C}}}$ through Eq. (1)
  Solve the invariant distribution $\bar{\bar{\boldsymbol{\pi}}}$ of $\overline{\widehat{\boldsymbol{C}}}$
  Return $\bar{\bar{\boldsymbol{\pi}}}$
```

recovers a matrix similar to the original matrix with a guaranteed sampling complexity of ($O(nr\log n)$). Algorithm 1 gives the details of OptEval-1. Details of OptSpace algorithm are in Appendix A.

OptEval-2 OptEval-2 solves α-rank based on noisy (empirical) payoffs of a selected set of agent pairs in Ω. To obtain empirical payoffs $\widehat{\boldsymbol{M}}_{ij}, \forall (i,j) \in \Omega$, agent i and j need to compete against each other for a sufficient number of times. We employ RG-UCB (Rowland et al., 2019) as the sampling algorithm on estimating $\widehat{\boldsymbol{M}}_{ij}$, which is composed by sampling scheme $\mathcal{S}$ and a stopping condition $\mathcal{C}(\delta)$. In our implementation, we adopt Uniform-exhaustive (UE) which randomly samples a pair from Ω at each time, and Hoeffding (UCB) as confidence-bound for stopping the evaluation of $\boldsymbol{M}_{ij}$. We run RG-UCB to evaluate pairs in Ω to obtain $\widehat{\boldsymbol{M}}^{\Omega}$, then with a chosen rank $\hat{r}$, we apply OptSpace algorithm to get the recovered matrix $\overline{\widehat{\boldsymbol{M}}}$. At last, we perform α-rank on $\overline{\widehat{\boldsymbol{M}}}$ and get the invariant distribution of n strategies. Algorithm 2 gives the details of OptEval-2. Details of RG-UCB algorithm are deferred in Appendix A.

Discussions Compared to the original RG-UCB, which evaluates the empirical payoffs of all pairs in $[n] \times [n]$, we only need to evaluate agent pairs in Ω, thus reducing the cost for evaluations by a large margin. Details of RG-UCB can be found in Appendix A.

Our algorithms compute α-rank based on $m (\ll n^2)$ entries from payoff table. Theories suggest m can be selected by a line search on C with $m = C \cdot nr\log n$, or more generally, with $m = C \cdot n\log n$, since the true rank r is usually unknowable. Formal results about sampling complexity on the payoff matrix are presented in Section 4.

4. Theoretical Analysis

The low-rank hypothesis of the payoff matrix implies that when repeated strategies exist, one can expect a lower number of competitions between agents. We now prove that, in the noise-free setting, one can obtain the accurate α-rank distribution with $O(nr\log n)$ payoff entries. Our results are based on the low-rank matrix completion theorem. The (μ_0, μ_1)-Incoherence is required for low-rank matrix completion theory which characterizes the spread of singular values in different coordinates. We leave these details to Appendix B. The following proposition provides a theoretical justification of Algorithm 1.

Proposition 1 (Noise-free payoff)**.** *Let $\boldsymbol{M} \in \mathbb{R}^{n\times n}$ denote the payoff matrix of n agents with rank r and it meets the (μ_0, μ_1)-Incoherence. Let $\mu = \max\{\mu_0, \mu_1\}$. Define $\kappa = (\Sigma_{\max}/\Sigma_{\min})$ as the ratio between maximal and minimal singular value. Let $\Omega \subseteq [n] \times [n]$ be a randomly selected set of pairs to be evaluated, then there exists a constant C such that if Ω satisfies*

$$|\Omega| \geq Cnr\kappa^2 \max\left\{\mu_0 \log n, \mu^2 r\kappa^4\right\},$$

then we can obtain the invariant distribution $\overline{\boldsymbol{\pi}}$, that is exactly the same as $\boldsymbol{\pi}$ obtained from complete comparisons of all strategies with high probability.

The proof is straightforward by applying low rank matrix completion theorem (see Theorem 3 Appendix B), since we can obtain an exact payoff matrix $\boldsymbol{M}$ based on $\boldsymbol{M}^{\Omega}$.

Though we observe that many real world games have a low-rank payoff matrix, it is still possible that some games have a high or even full-rank payoff matrix. But as long as it is not too distant from a low-rank matrix, Algorithm 1 can still work. Based on matrix completion theory from noisy entries (Theorem 4 in Appendix B), we give our results in Theorem 1.

Theorem 1 (Approximate low-rank matrix)**.** *Let $\widehat{\boldsymbol{M}} = \boldsymbol{M} + \boldsymbol{Z}$ be the exact payoff matrix bounded in the interval $[-M_{\max}, M_{\max}]$. Define $L(\alpha, M_{\max}) = 2\alpha\exp(2\alpha M_{\max})$, $g(\alpha, \eta, p, M_{\max}) = \eta\frac{\exp(2\alpha M_{\max})-1}{\exp(2\alpha p M_{\max})-1}$.*

Assume that $\boldsymbol{M} \in \mathbb{R}^{n\times n}$ is a (μ_0, μ_1)-incoherent matrix of rank r, and $\boldsymbol{Z}$ is noise that satisfies $\|\boldsymbol{Z}\|_{\max} < \tau$. Define $\mu = \max\{\mu_0, \mu_1\}$, $\kappa = \Sigma_{\max}/\Sigma_{\min}$ as the ratio between maximal and minimal singular value of $\boldsymbol{M}$. Define $\Omega \subseteq [n]\times[n]$ as the sampling operator in which m payoffs are randomly selected for observation from all n^2 entries. Therefore, the observed payoff matrix by the sampling operator Ω is $\widehat{\boldsymbol{M}}^{\Omega} = \boldsymbol{M}^{\Omega} + \boldsymbol{Z}^{\Omega}$. By performing matrix completion algorithm OptSpace on $\widehat{\boldsymbol{M}}^{\Omega}$ to obtain $\widehat{\boldsymbol{M}}$, there exist constants C such that if the number of sampled entries satisfies

$$|\Omega| \geq C\kappa^2 n \max\left\{\mu_0 r \log n, \mu^2 r^2 \kappa^4\right\},$$

then $\max_{i\in[n]} |\bar{\hat{\boldsymbol{\pi}}}(i) - \hat{\boldsymbol{\pi}}(i)| \leq \epsilon$ is satisfied with probability at least $1 - \frac{1}{n^3}$, with $\epsilon \in (0, 18 \times 2^{-n} \sum_{i=1}^{n-1} \binom{n}{i} i^n)$, $\tau = \frac{\epsilon g(\alpha,\eta,p,M_{\max})}{18L(\alpha,M_{\max}) \sum\limits_{i=1}^{n-1} \binom{n}{i} i^n (2C'\kappa^2\sqrt{r}+1)n}$.

Remark. *This result suggests that if a payoff matrix is not low-rank, we can still apply low-rank matrix completion algorithms to obtain satisfactory results, as long as the maximal entry of $\boldsymbol{Z}$ is bounded.*

In practice, it is difficult for us to get the exact payoff value of the selected pair. Therefore, when reconstructing the payoff matrix, we want to not only select as few pairs as possible, but also do as little competitive simulation as possible for the selected pairs. Let $\widehat{\boldsymbol{M}}_{ij}$ denote the empirical payoff, and the empirical risk $\boldsymbol{Z}_{ij} = |\boldsymbol{M}_{ij} - \widehat{\boldsymbol{M}}_{ij}|$ can be minimized by simulating more interactions between i and j. The following theorem provides a theoretical justification of Algorithm 2.

Theorem 2 (Noisy payoff)**.** *Suppose the payoff matrix $\boldsymbol{M} \in \mathbb{R}^{n\times n}$ be a (μ_0, μ_1)-incoherent matrix of rank r, and payoffs are bounded in the interval $[-M_{\max}, M_{\max}]$. Define $\mu = \max\{\mu_0, \mu_1\}$. Define $L(\alpha, M_{\max}) = 2\alpha \exp(2\alpha M_{\max})$ and $g(\alpha, \eta, p, M_{\max}) = \eta\frac{\exp(2\alpha M_{\max})-1}{\exp(2\alpha p M_{\max})-1}$. Let $\epsilon \in (0, 18 \times 2^{-n} \sum_{i=1}^{n-1} \binom{n}{i} i^n)$. Let $\Omega \subseteq [n]\times[n]$ be the sampling operator by which m pairs are randomly sampled for evaluation. For each pair $(i,j) \in \Omega$, let $\widehat{\mathbf{M}}_{ij}$ be an empirical payoff constructed by taking K i.i.d. interactions of strategy i and j. $\bar{\hat{\boldsymbol{\pi}}}$ is the invariant distribution obtained by computing the α-rank on $\widehat{\boldsymbol{M}}$, that is obtained by running OptSpace on $\widehat{\boldsymbol{M}}^{\Omega}$. There exist constants C and C' such that if the number of randomly selected pairs satisfies*

$$|\Omega| \geq C\kappa^2 n \max\left\{\mu_0 r \log n, \mu^2 r^2 \kappa^4\right\}$$

and K satisfies

$$K \geq \frac{2592 M_{\max}^2 \log 2mn^3 L(\alpha, M_{\max})^2 (\sum_{i=1}^{n-1} \binom{n}{i} i^n)^2 {C'}^2 \kappa^4 r n^2}{\epsilon^2 g(\alpha, \eta, p, M_{\max})^2}$$

then $\max_{i\in[n]} |\bar{\hat{\boldsymbol{\pi}}}(i) - \boldsymbol{\pi}(i)| \leq \epsilon$ is satisfied with probability at least $1 - \frac{2}{n^3}$.

Remark. *This result suggests that by sampling at most $O(nr\log n)$ pairs and simulating K interactions for each pair, we can approximate each entry of the $\boldsymbol{\pi}$ in α-rank with a discrepancy at most ϵ. This result provides guidance in performing evaluations. Note that $g(\alpha, \eta, p, M_{\max})$ is inversely proportional to α, indicating that larger α will require higher number of interactions to obtain an accurate estimate of $\boldsymbol{\pi}$.*

5. Experiments

We consider the following three batteries of experiments with increasing complexity to evaluate the performance of OptEval in the scenarios of noise-free, noisy payoff and real world meta-games. Ablation studies of parameters δ, α, and additional results on real world games with more metrics can be found in Appendix C.

Gaussian games (Rashid et al., 2021). We randomly generate a two-player general-sum Gaussian game with payoff matrix $\boldsymbol{M}$, in which the payoff value $\boldsymbol{M}_{ij}$ means the expected reward of i compete with j. And we suppose simulation results of pairs (i, j) are samples from $\sigma(\boldsymbol{M}_{ij}, 1)$. Due to the extraordinary computation complexity of α-IG, we only generate a small size game called Gaussian(15): 15×15 with rank $r = 2$.

Bernoulli games (Rowland et al., 2019). We randomly generate a two-player zero-sum Bernoulli game, in which the payoff value $\boldsymbol{M}_{ij}$ means i beating j with probability $\boldsymbol{M}_{ij}$. In addition, $\boldsymbol{M}_{ij} = 1 - \boldsymbol{M}_{ji}$ for it is a zero-sum game. We generate a game called Bern(100): 100×100 with rank $r = 11$. We can convert the winning probability matrix into an anti-symmetric matrix with rank 10 which is helpful for matrix completion.

Soccer meta-game (Liu et al., 2018). The payoffs of meta-game are taken from (Liu et al., 2018), where 10 agents learn to play soccer in Mujoco environments (Todorov et al., 2012). This also corresponds to a two-player zero-sum game with empirical payoffs that is evaluated by letting agents pitting against each other. We reproduce the 10×10 payoff matrix to a 200×200 matrix with each agent repeated 20 times. We intend to show the performance of OptEval in the repeated agent settings.

Real world games (Czarnecki et al., 2020) We select 12 real-world games from the OpenSpiel Library (Lanctot et al., 2019), which are also evaluated by Czarnecki et al.. The payoff matrices are baked in empirical game-theoretic analysis that construct abstract counterparts and simulate interactions to obtain the payoffs. Most of the meta-game payoffs have a rank that is much smaller than the size of the strategy set while some games, such as AlphaStar, tic_tac_toe, Kuhn-poker have full rank payoff matrices. We aim to show the effectiveness of OptEval in the complex meta-game evaluations. Table 1 summarizes the statistics of consisdered real world games.

For real-world games, the meta-payoffs are normalized such that $\boldsymbol{M}_{ij} \in [-1, 1]$ by Czarnecki et al. (2020). For the consistent presentation of results, meta-payoffs $\boldsymbol{M}$ for the Bernoulli and Soccer games are normalized to [-1, 1], converting from the winning rate matrix by $\boldsymbol{M} = 2\boldsymbol{P} - 1$. The Gaussian game is not normalized for representing the general asymmetric payoffs.

Table 1. Statistics of twelve real world games from (Czarnecki et al., 2020). k denote the number of dominant singular values that such that $\sum_i^k \Sigma_i / \sum_i^n \Sigma_i \geq 80\%$

Game	# policies	rank	k
3-move parity game 2	160	14	9
Blotto	1001	50	16
hex(board_size=3)	766	764	232
Disc game	1000	2	2
Normal Bernoulli game	1000	1000	499
Elo game	1000	38	2
Random game of skill	1000	1000	515
Transitive game	1000	2	2
Triangular game	1000	1000	137
AlphaStar	888	888	238
tic_tac_toe	880	880	285
Kuhn-poker	64	64	24

5.1. Baselines

We consider the following methods for comparisons.

- RG-UCB (Rowland et al., 2019) adopts a sampling scheme responsible for selecting the simulation pairs, and a stopping condition $\mathcal{C}(\delta)$ controling the number of observations for each pair. δ is the confidence level on the estimation of payoffs and is set to 0.01. Details are in Appendix A.

- α-IG (Rashid et al., 2021) is an active sampling strategy for estimating the α-rank through as few samples as possible. It selects pairs whose payoff leads to the largest reduction on the entropy over a belief of α-rank. Due to the high cost at each step of interaction (i.e., 8000 times computation of α-rank in a 4×4 Gaussian game), we only report its performance on Gaussian(15) with noisy payoffs.

- OptEval (ours): comes up with two algorithms. OptEval-1 estimates α-rank based on a set of noise-free payoffs. OptEval-2 computes α-rank based on noisy empirical payoffs.

We evaluate all methods on the finite-α regime with $\alpha = 0.001$. Four metrics are considered to evaluate both the correctness of the recovered matrix and the task performance. Details are given below:

- $\boldsymbol{M}$ **error** indicates the correctness of the recovered payoff matrix $\widehat{\boldsymbol{M}}$, which is defined as $\frac{1}{n^2}\|\boldsymbol{M} - \widehat{\boldsymbol{M}}\|_F^2$.

- $\boldsymbol{\pi}$ **error** indicates the max value in $|\boldsymbol{\pi} - \hat{\boldsymbol{\pi}}|$.

- α**-rank ranking error** is computed by Kendall's tau-b correlation coefficient (Signorino & Ritter, 1999): $K(\boldsymbol{\pi}, \hat{\boldsymbol{\pi}}) = \frac{1}{F}\sum_{(i,j)\in[|S|], i\neq j} \bar{K}_{i,j}(\boldsymbol{\pi}, \hat{\boldsymbol{\pi}})$, where:

$$\bar{K}_{i,j}(\boldsymbol{\pi}, \hat{\boldsymbol{\pi}}) = \operatorname{sign}(\boldsymbol{\pi}_i - \boldsymbol{\pi}_j) * \operatorname{sign}(\hat{\boldsymbol{\pi}}_i - \hat{\boldsymbol{\pi}}_j),$$

$$F = \sqrt{\sum_{i\neq j} \operatorname{sign}^2(\boldsymbol{\pi}_i - \boldsymbol{\pi}_j) * \sum_{i\neq j} \operatorname{sign}^2(\hat{\boldsymbol{\pi}}_i - \hat{\boldsymbol{\pi}}_j)},$$

  and $K(\boldsymbol{\pi}, \hat{\boldsymbol{\pi}}) \in [-1, 1]$. The larger value indicates the better consistency between $\boldsymbol{\pi}$ and $\hat{\boldsymbol{\pi}}$.

- α**-Conv** (Muller et al., 2020) measures the convergence of $\hat{\boldsymbol{\pi}}$. Define PBR-Score as $q(i; \boldsymbol{\pi}; S) = \sum_j \boldsymbol{\pi}_j \mathbb{1}[\boldsymbol{M}_{ij} > \boldsymbol{M}_{ji}]$, then α-Conv $= |\max_i q(i, \boldsymbol{\pi}, S) - \max_j q(j, \hat{\boldsymbol{\pi}}, S)|$. α-Conv being closer to 0 indicates that the estimated $\hat{\boldsymbol{\pi}}$ is converging to groundtruth $\boldsymbol{\pi}$.

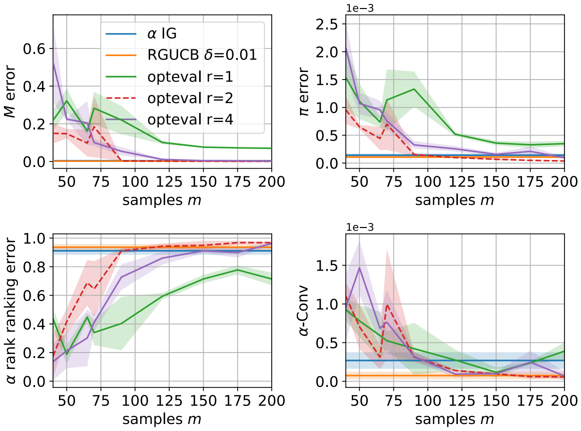
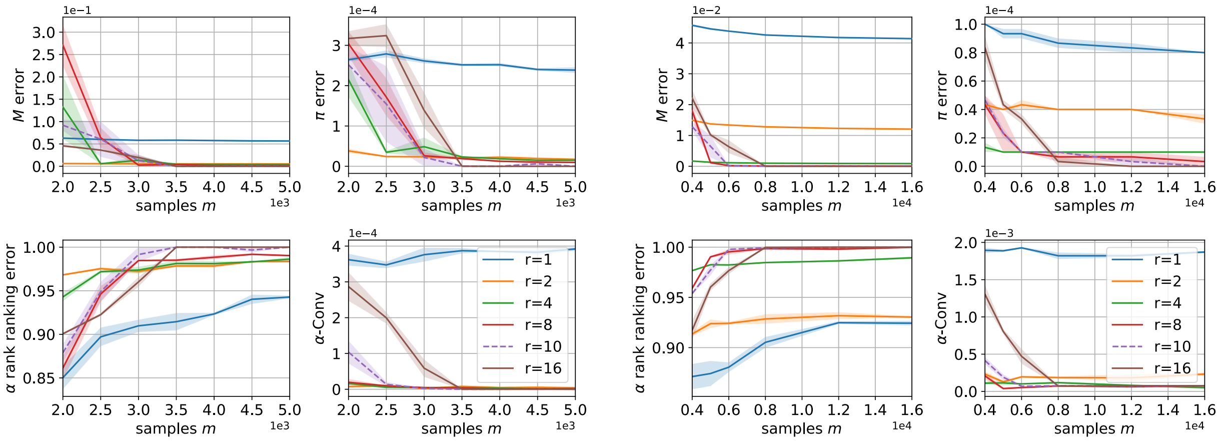


Figure 3. Noisy evaluations on Gaussian(15) with $r = 2, \alpha = 0.001, \delta = 0.01$.

5.2. Results

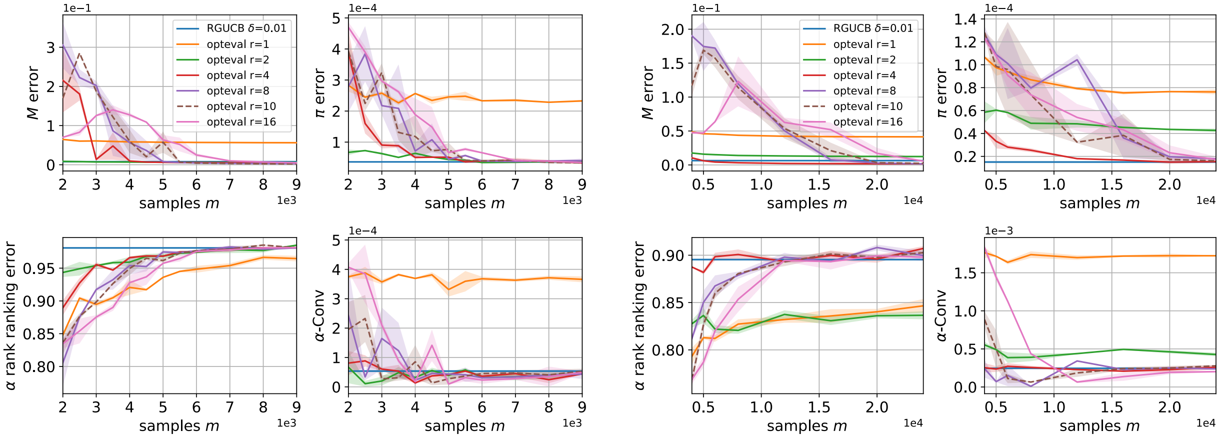
Results on Gaussian(15). Figure 3 examine the results in scenarios with noisy payoffs. Note that RG-UCB and α-IG



(a) Noise-free evaluations on Bernoulli games

(b) Noise-free evaluations on Soccer meta-game

Figure 4. Results in the Noise-free setting on Bernoulli and Soccer meta-game. Dashed lines indicating the performance of OptEval under true rank for reference. (a) Bern(100) game with $n = 100, r = 10, \alpha = 0.001$. (b) Soccer meta-game with $n = 200, r = 10, \alpha = 0.001$.



(a) Noisy evaluations on Bernoulli games

(b) Noisy evaluations on Soccer meta-game

Figure 5. Results in the Noisy setting on Bernoulli and Soccer meta-game. Dashed lines indicating the performance of OptEval under true rank for reference. (a) Bern(100) with $n = 100, r = 10, \alpha = 0.001, \delta = 0.01$; (b) Soccer meta-game with $n = 200, r = 10, \alpha = 0.001, \delta = 0.01$.

require the sampling over all pairs on the metagame, thus are horizontal lines here. On the contrary, OptEval-2 with $r = 2$ achieves the similar $\boldsymbol{\pi}$ error to RG-UCB and α-IG by around 100 sampled pairs, thus reducing the number of interactions by 60%. And OptEval-2 with $r = 2$ has lower ranking error and convergence error than both RG-UGB and α-IG. OptEval-2 with $r = 1$ can not fit the underlying payoff matrix of rank $r = 2$. OptEval-2 with $r = 4$ achieves satisfactory results around 150 sampled pairs reducing by 33%. It is noted that the convergence of OptEval-2 under $r = 4$ is slower than that under $r = 2$. One reason is that the model complexity for low rank matrix completion is higher, thus requiring larger samples for training.

Results on Bern(100) and Soccer Meta-game. Figure 4 shows the results on Bern(100) and Soccer meta-game the noise-free setting. In Bern(100), OptEval-1 learns to produce ranking and $\boldsymbol{\pi}$ with as few as about 3200 sampled pairs. In Soccer meta-game, OptEval-1 converges with around 6000 pairwise evaluations. Although both games have a groundtruth rank $r = 10$, Bern(100) can get small enough $\boldsymbol{\pi}$ and α-Conv error with a rank at 2. And in Soccer Meta-game, we can get a reliable α-rank when $r \geq 4$ and $m > 9000$. In Figure 4 (b) $r = 4$, even if the recovered matrix has difference from the original matrix, we can get an accurate α-rank ranking with ranking and α-Conv error close to 0.

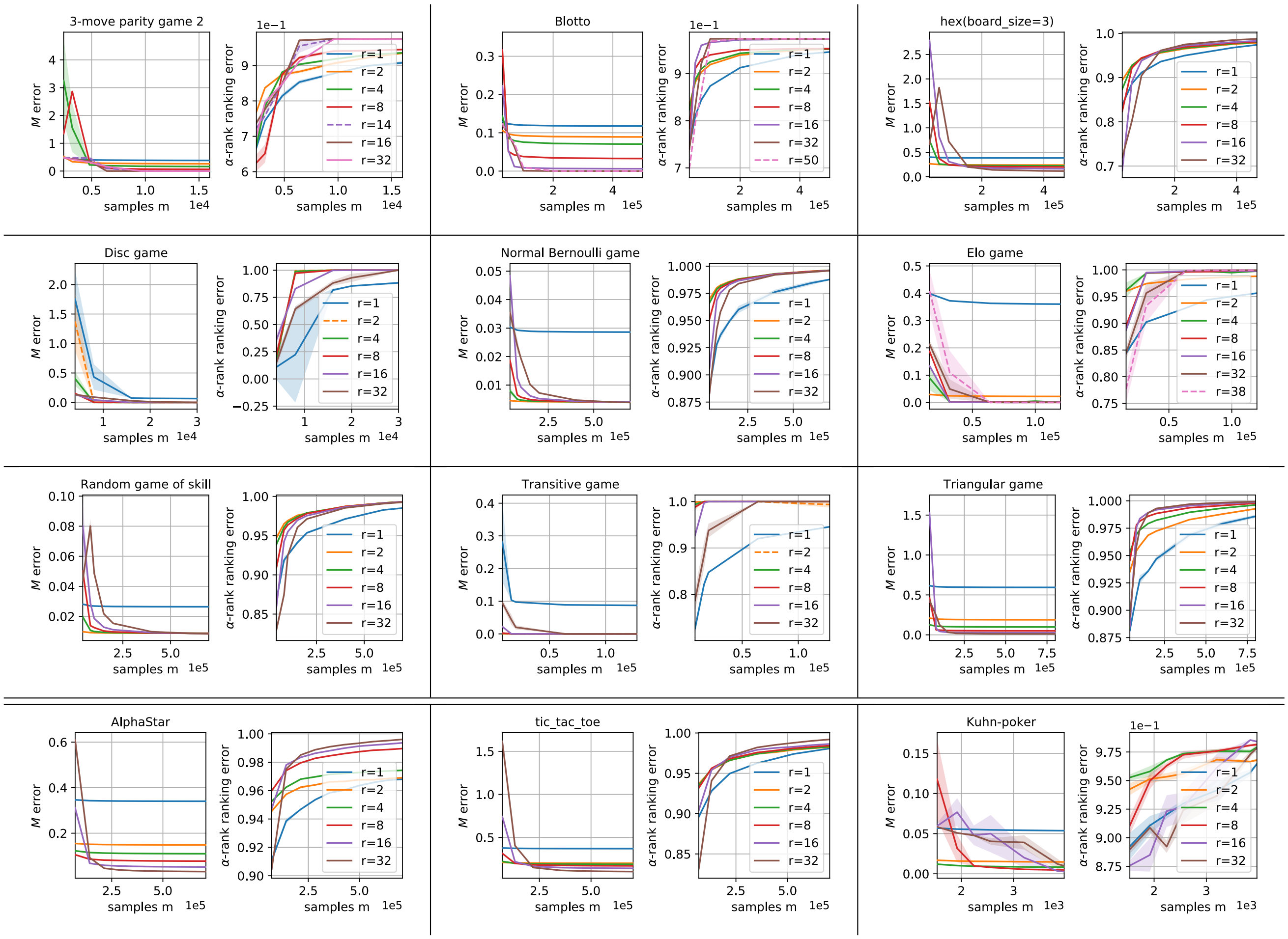


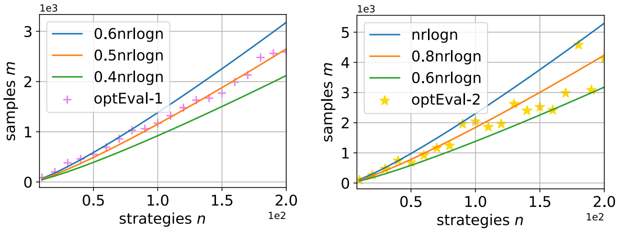
Table 2. Results on 12 real world games with noise free evaluations. (Left of plot) Recovery error on the payoff matrices. (Right of the plot) α-rank ranking error. The error on $\boldsymbol{M}$ indicates the performance of predicting the payoffs of unpitted pairs. The first three rows show the ranking results of low-rank payoff geometry, thus have lower error in predicting the payoffs of unobserved pairs. The last row shows the results of games that are not low-rank, and the ranking error is relatively larger than those games that are low-rank.

Figure 5 examine the results in scenarios with noisy payoffs. On Bern(100), the four error metrics show a similar trend with the change of number of samples and chosen rank. The matrix recovery error and ranking errors saturate at $r = 2$, indicating that a lower rank matrix can perform a satisfactory approximation to a higher rank payoff matrix. Besides, the sampled pairs are reduced to 6×10^3, leading to a 40% reduction compared to RG-UCB. On Soccer meta-game, the performance of OptEval-2 saturates at rank $r = 4$. Besides, the number of sampled pairs is reduced to $1e4$, leading to a 75% reduction compared to RG-UCB. Not only can we reduce the sample complexity of pairs m, we can also achieve a more accurate α-rank ranking compared to RG-UCB. Compared to RG-UCB, OptEval-2 achieves better performance in terms of the α-Conv measure on both Bern(100) (with $r \geq 2$) and Soccer meta-game (with ($r \geq 4$)).

Results on real world meta games Table 2 shows the results on real world meta games. For preserving the completeness, we show games that are both low-rank and high-rank, i.e. AlphaStar, Tick_tac_toe, which has a full rank meta payoffs. Though AlphaStar has a high rank at 888, we can approximate it with rank $r = 30$. One interesting observation is that, through cross reference to Table 1 in (Czarnecki et al., 2020), the games that show clear Game of Skill geometry have higher rank, such as AlphaStar, Quoridor, Go, etc. By contrast, the games that do not follow Game of Skill, have lower rank, Disc Game, Elo Game, Blotto etc. This indicates that strategies that are non-transitive contribute the most to the rank of the payoff geometry.

5.3. Empirical sampling complexity

To demonstrate that the empirical sampling complexity on agents pairs meets the theoretical results, we create one example as below. We create twenty Gaussian games with $n = 10, 20, ..., 200$ and $r = 5$ for all matrices. In both noise-free and noisy setting, we run OptEval with a chosen rank $r = 5$ and plot the number of samples and compare it to $m = c \cdot nr \log n, c = 0.4, 0.5, 0.6, 0.8, 1.0$. Figure 6 visualizes the number of payoff entries needed for each game. OptEval achieves a sampling complexity at around $0.5nr \log n$ in the noise-free setting, and $0.8nr \log n$ in the noisy setting, which are consistent with the theories.



(a) Noise-free evaluations on n-strategy games

(b) Noisy evaluations on n-strategy games

Figure 6. Comparisons of empirical sampling complexity and theoretical results on twenty games with $n = 10, 20, ..., 200, r = 5$. (a) the empirical sampling complexity of OptEval-1 when $\epsilon \leq 10^{-4}/n$ at a chosen rank $r = 5$. (b) the empirical sampling complexity when OptEval-2 outperforms RG-UCB.

6. Discussions

This work investigates the question of how many pairwise comparisons we need to produce a satisfied ranking for n agents based on α-rank. We bake our theory and algorithms based on the facts that strategies with similar skills may have similar payoff against the competitions with others, and repeated strategies may exist in multi-agent systems. We provide the theories and algorithms for scenarios with both noise-free and noisy evaluation of two-player meta-games. Experiment results show that with a much fewer number of comparisons, our method OptEval can reach comparable performance to ground truth results in noise-free case, and to RG-UCB which uses full payoff table in noisy case.

We are especially the first to perform experiments on large scale competitive games with more than 1000 strategies. For the future works, one line of research is to consider more complicated games such as Poker, Mahjong that require triple-wise or quadruplet-wise comparisons. Another direction is to consider active sampling of pairwise evaluations to further reduce the cost of evaluation.

Acknowledgements

We would like to thank Yaodong Yang for his constructive comments and Tianyi Zhou for helpful feedback on an early draft of this work. Haifeng Zhang thanks the support of the Strategic Priority Research Program of Chinese Academy of Sciences, Grant No. XDA27000000.

References

Arneson, B., Hayward, R. B., and Henderson, P. Monte carlo tree search in hex. *IEEE Transactions on Computational Intelligence and AI in Games*, 2(4):251–258, 2010.

Balduzzi, D., Tuyls, K., Perolat, J., and Graepel, T. Re-evaluating evaluation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 3268–3279, 2018.

Candes, E. and Recht, B. Exact matrix completion via convex optimization. *Communications of the ACM*, 55 (6):111–119, 2012.

Candes, E. J. and Plan, Y. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, 2010.

Candès, E. J. and Tao, T. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.

Czarnecki, W. M., Gidel, G., Tracey, B., Tuyls, K., Omidshafiei, S., Balduzzi, D., and Jaderberg, M. Real world games look like spinning tops. *Advances in Neural Information Processing Systems (NeurIPS)*, 33, 2020.

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

Elo, A. E. *The rating of chessplayers, past and present*. Arco Pub., 1978.

Gruslys, A., Dabney, W., Azar, M. G., Piot, B., Bellemare, M., and Munos, R. The reactor: A fast and sample-efficient actor-critic agent for reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2018.

Harsanyi, J. C., Selten, R., et al. A general theory of equilibrium selection in games. *MIT Press Books*, 1, 1988.

Herbrich, R., Minka, T., and Graepel, T. Trueskill™: a bayesian skill rating system. In *Proceedings of the 19th international Conference on Neural Information Processing systems (NeurIPS)*, pp. 569–576, 2006.

Jiang, X., Lim, L.-H., Yao, Y., and Ye, Y. Statistical ranking and combinatorial hodge theory. *Mathematical Programming*, 127(1):203–244, 2011.

Keshavan, R., Montanari, A., and Oh, S. Matrix completion from noisy entries. In Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C., and Culotta, A. (eds.), *Advances in*

Neural Information Processing Systems (NeurIPS), volume 22, pp. 952–960. Curran Associates, Inc., 2009.

Keshavan, R. H. and Oh, S. A gradient descent algorithm on the grassman manifold for matrix completion. *arXiv preprint arXiv:0910.5260*, 2009.

Keshavan, R. H., Montanari, A., and Oh, S. Matrix completion from a few entries. *IEEE transactions on Information Theory*, 56(6):2980–2998, 2010.

Lai, M. Giraffe: Using deep reinforcement learning to play chess. *arXiv preprint arXiv:1509.01549*, 2015.

Lanctot, M., Zambaldi, V., Gruslys, A., Lazaridou, A., Tuyls, K., Pérolat, J., Silver, D., and Graepel, T. A unified game-theoretic approach to multiagent reinforcement learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*, pp. 4193–4206, 2017.

Lanctot, M., Lockhart, E., Lespiau, J.-B., Zambaldi, V., Upadhyay, S., Pérolat, J., Srinivasan, S., Timbers, F., Tuyls, K., Omidshafiei, S., et al. Openspiel: A framework for reinforcement learning in games. *arXiv preprint arXiv:1908.09453*, 2019.

Liu, S., Lever, G., Merel, J., Tunyasuvunakool, S., Heess, N., and Graepel, T. Emergent coordination through competition. In *International Conference on Learning Representations (ICLR)*, 2018.

Muller, P., Omidshafiei, S., Rowland, M., Tuyls, K., Perolat, J., Liu, S., Hennes, D., Marris, L., Lanctot, M., Hughes, E., et al. A generalized training approach for multiagent learning. In *International Conference on Learning Representations (ICLR)*, pp. 1–35, 2020.

Omidshafiei, S., Papadimitriou, C., Piliouras, G., Tuyls, K., Rowland, M., Lespiau, J.-B., Czarnecki, W. M., Lanctot, M., Perolat, J., and Munos, R. α-rank: Multi-agent evaluation by evolution. *Scientific reports*, 9(1):1–29, 2019.

Rajkumar, A. and Agarwal, S. When can we rank well from comparisons of $o(n \log(n))$ non-actively chosen pairs? In *Conference on Learning Theory (COLT)*, pp. 1376–1401, 2016.

Rashid, T., Zhang, C., and Ciosek, K. Estimating α-rank by maximizing information gain. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 35, pp. 5673–5681, 2021.

Recht, B. A simpler approach to matrix completion. *Journal of Machine Learning Research*, 12(Dec):3413–3430, 2011.

Rowland, M., Omidshafiei, S., Tuyls, K., Perolat, J., Valko, M., Piliouras, G., and Munos, R. Multiagent evaluation under incomplete information. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 12291–12303, 2019.

Shalev-Shwartz, S. and Ben-David, S. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Signorino, C. S. and Ritter, J. M. Tau-b or not tau-b: Measuring the similarity of foreign policy positions. *International Studies Quarterly*, 43(1):115–144, 1999.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., and Bolton, A. Mastering the game of go without human knowledge. *Nature*, 550(7676):354, 2017.

Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012.

Tuyls, K., Perolat, J., Lanctot, M., Leibo, J. Z., and Graepel, T. A generalised method for empirical game theoretic analysis. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pp. 77–85, 2018.

Yang, Y., Tutunov, R., Sakulwongtana, P., and Ammar, H. B. α^{α}-rank: Practically scaling α-rank through stochastic optimisation. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pp. 1575–1583, 2020.