Estimating α-Rank by Maximizing Information Gain

Tabish Rashid * tabish.rashid@cs.ox.ac.uk
Department of Computer Science, University of Oxford

Cheng Zhang cheng.zhang@microsoft.com
Kamil Ciosek kamil.ciosek@microsoft.com
Microsoft Research, Cambridge, UK

Abstract

Game theory has been increasingly applied in settings where the game is not known outright, but has to be estimated by sampling. For example, meta-games that arise in multi-agent evaluation can only be accessed by running a succession of expensive experiments that may involve simultaneous deployment of several agents. In this paper, we focus on α-rank, a popular game-theoretic solution concept designed to perform well in such scenarios. We aim to estimate the α-rank of the game using as few samples as possible. Our algorithm maximizes information gain between an epistemic belief over the alpha-ranks and the observed payoff. This approach has two main benefits. First, it allows us to focus our sampling on the entries that matter the most for identifying the alpha rank. Second, the Bayesian formulation provides a facility to build in modeling assumptions by using a prior over game payoffs. We show the benefits of using information gain as compared to the confidence interval criterion of ResponseGraphUCB (Rowland et al., 2019), and provide theoretical results justifying our method.

Keywords: Game Theory, α-rank, Information Gain

1. Introduction

Traditionally, game theory is applied in situations where the game is fully known. More recently, it is being used in situations where the game is not fully known and can only be interacted with through sampling, known as Empirical Game Theory (Wellman, 2006). One area in which this is becoming increasingly common is the ranking of trained agents relative to one another. Specifically, in the field of Reinforcement Learning game-theoretic rankings are used not just as a metric for measuring algorithmic progress (Balduzzi et al., 2018), but also as an integral component of many population-based training methods (Muller et al., 2020; Lanctot et al., 2017; Vinyals et al., 2019). In particular, for ranking, two popular solution concepts have recently emerged: Nash averaging (Balduzzi et al., 2018; Nash, 1951) and α-rank (Omidshafiei et al., 2019).

We use the α-rank solution concept for two reasons. First, it admits a unique solution whose computation easily scales to N-player games. Second, unlike older schemes such as Elo (Elo, 1978), α-rank is designed with intransitive interactions in mind. Because measuring payoffs can be very expensive, it is important to do it by using as few samples as possible.

∗. Work done during an internship at Microsoft Research Cambridge.

For example, playing a match of chess (Silver et al., 2017) in a self-play algorithm (Lanctot et al., 2017; Muller et al., 2020) can take roughly 40 minutes[1]. *Our objective is thus to accurately estimate the α-rank using a small number of payoff queries.*

Rowland et al. (2019) proposed ResponseGraphUCB (RG-UCB) for this purpose, inspired by the pure exploration bandit literature. RG-UCB aims to correctly determine the ordering between entries relevant to the computation of α-rank by maintaining confidence intervals over their values, and concluding that the correct ordering has been found between two entries when their confidence intervals do not overlap. While they prove that this is sufficient to determine the true α-ranking with a high probability in the infinite α regime, their approach has two important limitations. First, since the frequentist criterion is indirect, relying on payoff ordering, the obtained payoffs aren't always used optimally. Second, it is not straightforward to include useful domain knowledge about the entries or structure of the payoff matrix.

To remedy these problems, we propose a Bayesian approach. Specifically, we utilize a Gaussian Process to maintain an epistemic belief over the entries of the payoff matrix, providing a powerful framework in which to to supply domain knowledge. This payoff distribution induces an epistemic belief over the α-ranking. We determine which payoff to sample by maximizing information gain between the α-rank belief and the obtained payoff. This allows us to focus our sampling on the entries that are expected to have the largest effect on our belief over possible α-ranks.

Contributions Theoretically, we justify the use of information gain by showing a regret bound for a version of our criterion in the infinite-α regime. Empirically, our contribution is twofold. First, we compare to RG-UCB on stylized games, showing that maximizing information gain provides competitive performance by focusing on sampling the more relevant payoffs. Second, we evaluate another objective based on minimizing the Wasserstein divergence, which offers competitive performance while being computationally much cheaper.

2. Background

A game with K players, each of whom can play S strategies is fully characterized by its payoff matrix $M \in \mathcal{R}^N$, where $N = S^K$ (Fudenberg and Tirole, 1991). The α-rank $r \in \mathcal{R}^S$ (Omidshafiei et al., 2019; Rowland et al., 2019) of a game is defined[2] as the unique stationary distribution of a Markov Chain C'. Please see Appendix A for a more detailed description.

3. Method

On a high level, our method works by maintaining an epistemic belief over alpha ranks and selecting payoffs that lead to the maximums reduction in the entropy of that belief. Figure 1 provides a pictorial overview. In the middle of the figure, we maintain an explicit distribution over the entries of the payoff matrix. This payoff distribution induces a belief over α-ranks, shown on the left. When deciding which payoff to sample, we examine hypothetical belief states after sampling, striving to end up with a belief with the lowest entropy. One such hypothetical, or 'hallucinated' belief is shown on the right. We now describe our method formally, first describing the probabilistic model and then the implementation.

1. Assuming a typical game-length of 40 and up to 1 minute per move.
2. We focus on the single population case $K = 1$. Our method can be extended to multiple populations in a straightforward way, but we don't do this for simplicity.

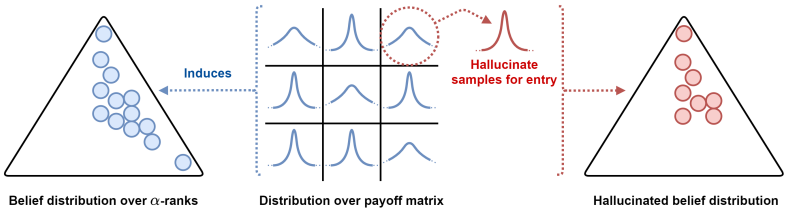

Figure 1: On the left, a belief over α-ranks is induced by a belief over the payoff matrix, shown in the middle. A hallucinated belief distribution is shown on the right. See Section 3 for detailed description.

Payoffs: Ground Truth and Belief We denote the unknown true payoff vector as $M_\star$. To quantify our uncertainty about what this true payoff is, we employ a Gaussian Process M, which also allows us to encode prior knowledge about payoff dependencies. The GP models noise in the payoffs as $\tilde{M} = M + \epsilon$, where $\epsilon \sim \mathcal{N}(0, I\sigma_A^2)$. When interacting with the game sequentially, the received payoffs are assumed to be generated as $m_t = M_\star^{a_t} + \epsilon'_t$. Here, ϵ'_t are i.i.d. random variables with support on the interval $[-\sigma_A, \sigma_A]$. While it may at first seem surprising that we use Gaussian observation noise in the GP model, while assuming a truncated observation noise for the actual observation, this does not in fact affect our theoretical guarantees. We provide more details in Section 4. We denote by H_t the history of interactions at time t. Because of randomness in the observations, H_t is a random variable. The sequence of random variables $H_1, H_2, \ldots$ forms a filtration. We use the symbol h_t to denote particular realization of history, so that $h_t = a_1, m_1, \ldots, a_{t-1}, m_{t-1}$.

Belief over α-ranks Our explicit distribution over the entries of the payoff matrix induces an implicit belief distribution over the α-ranks $r = f(M)$, where $P(r) = P(M \in f^{-1}(r))$ and f^{-1} denotes the pre-image of r under f. In other words, the probability assigned to an α-rank r is the probability assigned to its pre-image by our belief over the payoffs. Since r is represented implicitly, we cannot query its mass function directly. Instead, we access r via sampling. This is done by first drawing a payoff from $m \sim M$ and then computing the resulting α-rank $f(m)$.

Picking Payoffs To Query At time t, we query the payoff that provides us with the largest information gain about the α-rank. Formally,

$$
\begin{aligned}
a_t &= \arg\max_a \mathbb{I}(r\,;\,(\tilde{M}_t^a, a) \mid H_t = h_t) \\
&= \arg\max_a \mathbb{H}\left(r \mid H_t = h_t\right) - \mathbb{E}_{\tilde{m}_t \sim \tilde{M}_t^a}\left[\mathbb{H}\left(r \mid H_t = h_t, A_t = a, \tilde{M}_t^a = \tilde{m}_t\right)\right] \qquad (1) \\
&= \arg\min_a \mathbb{E}_{\tilde{m}_t \sim \tilde{M}_t^a}\left[\mathbb{H}\left(r \mid H_t = h_t, A_t = a, \tilde{M}_t^a = \tilde{m}_t\right)\right]. \qquad (2)
\end{aligned}
$$

In equation (1), $\mathbb{H}\left(r \mid H_t = h_t\right)$ is the entropy of our current belief distribution over α-ranks, which does not depend on a and can be dropped from the maximization, producing equation (2). The expectation in (2) has an intuitive interpretation as the expected negative entropy of our *hallucinated* belief, i.e. belief obtained by conditioning on a sample $\tilde{m}_t$ from the current model. In essence, we are pretending to receive a sample for entry a, and then computing what our resulting belief over α-ranks will be. By picking the entry as in (2), we are picking the entry whose sample will lead to the largest reduction in the entropy of our belief over α-ranks in expectation.

Algorithm 1 αIG algorithm. (NSB) and (Bin) variants differ in entropy estimator (line 7).

1: **for** $t = 1, 2, \dots T$ **do**
2: **for** $a = 1, 2, \dots N$ **do**
3: **for** $i = 1, 2, \dots N_e$ **do**
4: $\tilde{m}_t \sim \tilde{M}_t^a$ ▷ 'Hallucinate' a payoff.
5: Obtain posterior payoff distribution $P(\tilde{M}_t^a | H_t = h_t, A_t = a, \tilde{M}_t^a = \tilde{m}_t)$
6: $D = \{r_1, \dots, r_{N_b}\}$, where $r_i \sim f(\tilde{M}_t^a)$ i.i.d.
7: $\hat{h}_a^i =$ ESTIMATE-ENTROPY(D)
8: **end for**
9: $\hat{h}_a = \frac{1}{N_e} \sum_{i=1}^{N_e} \hat{h}_a^i$
10: Query payoff $a_t = \arg\min_a \hat{h}_a$. ▷ Implements equation (1).
11: **end for**
12: **end for**

Implementation The αIG method is summarized in algorithm 1. In line 4, we use our epistemic model to obtain a 'hallucinated' outcome resulting from selecting payoff a. In line 7, we use empirically estimate the entropy of the resulting distribution over ranks. In line 9, we average out entropy estimates obtained from N_e different possible hallucinated payoffs. In line 10, we use these estimates to perform query selection as in (1).

Our algorithm depends on an entropy estimator ESTIMATE-ENTROPY, used in line 7. We present results for 2 different entropy estimators: simple binning and NSB. The simple binning estimator estimates the entropy using a histogram. For comparison, we also used NSB (Nemenman et al., 2002), an entropy estimator designed to produce better estimates in the small-data regime.

In addition, we perform two optimizations when deploying the algorithm in practice. To save computational cost, in line 10, we observe the same payoff N_r times rather than once. Moreover, to obtain better differentiation between beliefs arising from sampling different payoffs, we heuristically perform conditioning in line 5 N_c times.

4. Theoretical Results

Notions of Regret We quantify the performance of our method by measuring regret. Our main analysis relies on Bayesian regret (Russo and van Roy, 2018), defined as

$$J_t^B = 1 - \mathbb{E}_{h_t} \left(P(r = r_\star | H_T = h_t) \right), \tag{3}$$

where we used $r_\star$ to denote the alpha rank with the highest probability under r at time t. In (3), the expectation is over realizations of the observation model. Since J_t^B, like all purely Bayesian notions, does not involve the ground truth payoff, we need to justify its practical relevance. We do this by benchmarking it against two notions of frequentist regret. First, $J_t^F = 1 - \mathbb{E}_{h_t} \left(P(r = r_{\text{GT}} | H_T = h_t) \right)$, where $r_{\text{GT}} = f(M_\star)$. Second, $J_t^M = 1 - \mathbb{E}_{h_t} \left(\delta \Big[f(M_\mu) = r_{\text{GT}} \Big] \right)$, where we denote the mean of the payoff belief with M_μ and the symbol $\delta \Big[\text{predicate} \Big]$ evaluates to 1 or 0 depending on whether the predicate is true or false. In section 5, we empirically conclude that the three notions of regret are closely coupled in practice, changing at a comparable rate.

Separability Assumption Similarly to the work of Rowland et al. (2019), we limit ourselves to payoffs that are distinguishable in order to make α-rank robust to small changes in the payoffs. We assume that there exists a constant $\Delta > 0$ such that for all payoff indices i, j

$$\left|M_\star^i - M_\star^j\right| \geq \Delta. \tag{4}$$

Regret Bounds As an intermediate step before discussing information gain on the alpha-ranks, we first analyze the behavior of a query selection rule which maximizes information gain over the payoffs.

$$\pi_{\text{IGM}}(a|H_t = h_t) = \arg\max_a \mathbb{I}(M\,;\,(M', a) \mid H_t = h_t). \tag{5}$$

The following result shows that using sampling strategy π_{IGM} for T timesteps leads to a decay in regret of at least $Te^{\mathcal{O}(-\Delta^2 T)}$.

Proposition 1 (Regret Bound For Information Gain on Payoffs) *If we select actions using strategy π_{IGM}, regret is bounded as*

$$J_T^B \leq J_T^F \leq 1 - \mathbb{E}_{h_t}\left(P(r = r_{GT}|H_T = h_T)\right) \leq 1 - Te^{g(T)} \;\; \textit{where} \;\; g(T) = \mathcal{O}(-\sqrt[3]{\Delta^2 T}). \tag{6}$$

The proof, and an explicit form of g are found in supplementary material. We now proceed to our second result, where we maximize information gain on the alpha ranks directly. Consider a querying strategy that is an extension of (1) to T-step look-ahead, defined as

$$\pi_{\text{IGR}} = \arg\max_{a_1,\ldots,a_T} \mathbb{I}(r\,;\,M_1'^a, \ldots, M_T'^a). \tag{7}$$

We quantify regret achieved by π_{IGR} in the proposition below.

Proposition 2 (Regret Bound For Information Gain on Belief over Alpha Ranks) *If we select actions using strategy π_{IGR}, regret is bounded as*

$$1 - P(r = r_\star|H_T = h_t) \leq \delta\Big[z(T) \leq h_b(|R|^{-1})\Big]\, h_b^{-1}(z(T)) + \delta\Big[z(T) > h_b(|R|^{-1})\Big],$$

where $z(T) = \delta\Big[Te^{g(T)} \geq \frac{1}{2}\Big]\, h_b(Te^{g(T)}) + \delta\Big[Te^{g(T)} \leq \frac{1}{2}\Big]\, N \log N$ *and $g(T)$ is defined as in Proposition 1.*

Proof is provided in supplementary material. The symbol $\delta\Big[\text{predicate}\Big]$ evaluates to 1 or 0 depending on whether the predicate is true or false. In practice, to avoid the combinatorial expense of selecting action sequences using π_{IGR}, we use the greedy query selection strategy in equation (1). While the regret result above does not carry over, this idealized setting at least provides some justification for information gain as a query selection criterion.

5. Experiments

In this section, we describe our results on 3 synthetic games, graphing the notions of regret described in Section 4. We benchmark two versions of our algorithms, αIG (Bins) and αIG (NSB), which differ in the employed entropy estimator. We also benchmark αWass, a variant of our algorithm optimising a Wasserstein based objective which is described in more detail

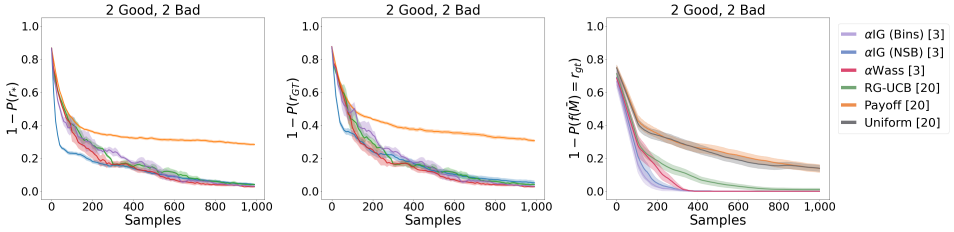

Figure 2: Results for 2 Good, 2 Bad. Graphs show the the mean and standard error of the mean over multiple runs (shown in brackets) of 10 repeats each.

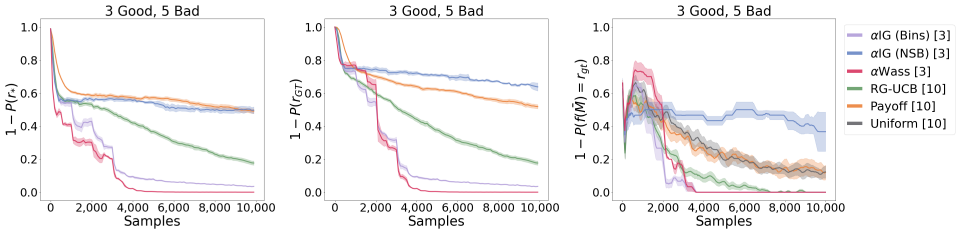

Figure 3: Results for 3 Good, 3 Bad. Graphs show the the mean and standard error of the mean over multiple runs (shown in brackets) of 10 repeats each.

in Appendix C. A detailed explanation of the experimental setup and details on the baselines and hyperparameters are included in Appendix F.

The first 2 synthetic games we examine feature X Good and Y Bad agents, in which the Good agents always beat the Bad agents. In these games it is highly beneficial to concentrate sampling on the payoffs between the Good agents. Figures 2 and 3 demonstrate the benefits of our methods, particularly on the larger scenario, since they are able to concentrate their sampling on the relevant payoffs. The final game we examine features a randomly generated payoff matrix with Gaussian observation noise, providing empirical confirmation of our theoretical results.

Additional experiments and details of the games are included in Appendix E.

6. Conclusions

We described αIG, an algorithm for estimating the α-rank of a game using a small number of payoff evaluations. αIG works by maximizing information gain. It achieves competitive sample efficiency and allows a way of building in prior knowledge about the payoffs.

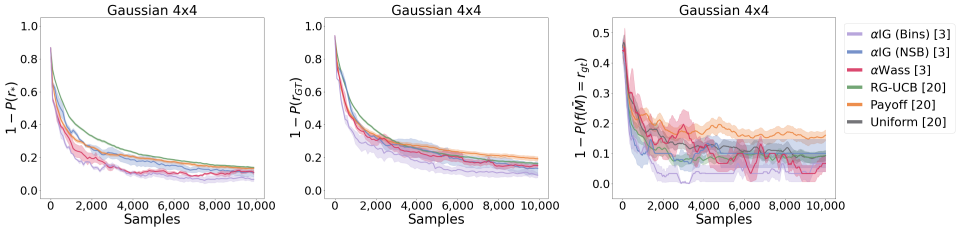

Figure 4: Results for the 4x4 Gaussian Game. Graphs show the the mean and standard error of the mean over multiple runs (shown in brackets) of 10 randomly sampled games.

References

David Balduzzi, Karl Tuyls, Julien Perolat, and Thore Graepel. Re-evaluating evaluation. In *Advances in Neural Information Processing Systems*, pages 3268–3279, 2018.

Nicolas Bonneel, Michiel Van De Panne, Sylvain Paris, and Wolfgang Heidrich. Displacement interpolation using lagrangian mass transport. In *Proceedings of the 2011 SIGGRAPH Asia Conference*, pages 1–12, 2011.

Shuo Chen and Thorsten Joachims. Modeling intransitivity in matchup and comparison data. In *Proceedings of the ninth acm international conference on web search and data mining*, pages 227–236, 2016.

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

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

Ronald Fagin, Ravi Kumar, Mohammad Mahdian, D Sivakumar, and Erik Vee. Comparing partial rankings. *SIAM Journal on Discrete Mathematics*, 20(3):628–648, 2006.

R'emi Flamary and Nicolas Courty. Pot python optimal transport library, 2017. URL `https://pythonot.github.io/`.

Drew Fudenberg and Jean Tirole. Game theory, 1991. *Cambridge, Massachusetts*, 393(12): 80, 1991.

Ralf Herbrich, Tom Minka, and Thore Graepel. Trueskill$^{\text{TM}}$: a bayesian skill rating system. In *Advances in neural information processing systems*, pages 569–576, 2007.

Patrick R Jordan, Yevgeniy Vorobeychik, and Michael P Wellman. Searching for approximate equilibria in empirical games. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 2*, pages 1063–1070, 2008.

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 *Advances in Neural Information Processing Systems*, pages 4190–4203, 2017.

Simone Marsili. ndd - bayesian entropy estimation from discrete data. URL `https://github.com/simomarsili/ndd`.

Tom Minka, Ryan Cleven, and Yordan Zaykov. Trueskill 2: An improved bayesian skill rating system. *Tech. Rep.*, 2018.

Paul Muller, Shayegan Omidshafiei, Mark Rowland, Karl Tuyls, Julien Perolat, Siqi Liu, Daniel Hennes, Luke Marris, Marc Lanctot, Edward Hughes, Zhe Wang, Guy Lever, Nicolas Heess, Thore Graepel, and Remi Munos. A generalized training approach for multiagent learning. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=Bkl5kxrKDr`.

John Nash. Non-cooperative games. *Annals of mathematics*, pages 286–295, 1951.

Ilya Nemenman, Fariel Shafee, and William Bialek. Entropy and inference, revisited. In *Advances in neural information processing systems*, pages 471–478, 2002.

Shayegan Omidshafiei, Christos Papadimitriou, Georgios Piliouras, Karl Tuyls, Mark Rowland, Jean-Baptiste Lespiau, Wojciech M Czarnecki, Marc Lanctot, Julien Perolat, and Remi Munos. α-rank: Multi-agent evaluation by evolution. *Scientific reports*, 9(1):1–29, 2019.

Mark Rowland, Shayegan Omidshafiei, Karl Tuyls, Julien Perolat, Michal Valko, Georgios Piliouras, and Remi Munos. Multiagent evaluation under incomplete information. In *Advances in Neural Information Processing Systems*, pages 12270–12282, 2019.

Daniel Russo and Benjamin van Roy. Learning to optimize via information-directed sampling. *Operations Research*, 66(1):230–252, 2018.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017.

Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

Karl Tuyls, Julien Perolat, Marc Lanctot, Edward Hughes, Richard Everett, Joel Z Leibo, Csaba Szepesvári, and Thore Graepel. Bounds and dynamics for empirical game theoretic analysis. *Autonomous Agents and Multi-Agent Systems*, 34(1):7, 2020.

Cédric Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575 (7782):350–354, 2019.

William E Walsh, David C Parkes, and Rajarshi Das. Choosing samples to compute heuristic-strategy nash equilibrium. In *International Workshop on Agent-Mediated Electronic Commerce*, pages 109–123. Springer, 2003.

Michael P Wellman. Methods for empirical game-theoretic analysis. In *AAAI*, pages 1552–1556, 2006.

Yaodong Yang, Rasul Tutunov, Phu Sakulwongtana, and Haitham Bou Ammar. α α-rank: Practically scaling α-rank through stochastic optimisation. *arXiv preprint arXiv:1909.11628*, 2019.

Appendix A. Background

Games and α-Ranks A game with K players, each of whom can play S strategies is fully characterized by its payoff matrix $M \in \mathcal{R}^N$, where $N = S^K$ (Fudenberg and Tirole, 1991). The α-rank $r \in \mathcal{R}^S$ (Omidshafiei et al., 2019; Rowland et al., 2019) of a game is defined[3] as the unique stationary distribution of a Markov Chain C', i.e.

$$r^\top C' = r^\top.$$

Here, C' is a small perturbation of the chain $C \in \mathcal{R}^{S \times S}$, motivated by evolutionary dynamics in the population of strategies and defined as

$$C_{\sigma,\tau} = \begin{cases} (S-1)^{-1}(1-\epsilon) & \text{if } M^{(\tau,\sigma)} > M^{(\sigma,\tau)}, \\ (S-1)^{-1}\epsilon & \text{if } M^{(\tau,\sigma)} < M^{(\sigma,\tau)}, \\ 0.5(S-1)^{-1} & \text{if } M^{(\tau,\sigma)} = M^{(\sigma,\tau)}. \end{cases}$$

for $\sigma \neq \tau$ and $C_{\sigma,\sigma} = 1 - \sum_{\tau \neq \sigma} C_{\sigma,\tau}$ for transitions from a strategy σ to itself. We abstract the above computation into the α-rank function $f : \mathcal{M} \to \mathcal{R}^S$. It represents the special case $\alpha \to \infty$ of a more general α-rank concept (Omidshafiei et al., 2019; Rowland et al., 2019).

Wasserstein Divergence Let p and q be probability distributions supported on $\mathcal{X}$, and $c : \mathcal{X} \times \mathcal{X} \to [0, \infty)$ be a distance. Define Π as the space of all joint probability distributions with marginals p and q. Wasserstein divergence (Villani, 2008) with cost function c, is defined as:

$$\mathcal{W}_c(p, q) := \min_{\pi \in \Pi} \int_{\mathcal{X} \times \mathcal{X}} c(x, y) d\pi(x, y).$$

In this paper, we will utilize the Wasserstein distance between our belief distributions over α-rank, and so we set $\mathcal{X} = \Delta^{S-1}$ and use $c(x, y) = \frac{1}{2}\|x - y\|_1$. We will drop the suffix and denote this simply as $\mathcal{W}$.

Appendix B. Related Work

There are many methods related to the ranking and evaluation of agents in games. ELO (Elo, 1978) and TrueSkill (Herbrich et al., 2007; Minka et al., 2018) both quantify the performance of an agent using a single number, which means they are unable to model *intransitive* interactions. Chen and Joachims (2016) extend TrueSkill to better model such interactions, while Balduzzi et al. (2018) do the same for ELO, improving its predictive power by introducing additional parameters. Balduzzi et al. (2018) also re-examines the use of Nash equilibrium, proposing to disambiguate across possible equillibria by picking the one with maximum entropy. However, it is well known that computing the Nash equilibrium is computationally difficult (Daskalakis et al., 2009) and only computationally tractable for restricted classes of games. In this paper, we focus on α-rank (Omidshafiei et al., 2019) since it has been designed with *intransitive* interactions in mind, is computationally tractable for N-player games and shows considerable promise as a component of self-play frameworks (Muller et al., 2020).

3. We focus on the single population case $K = 1$. Our method can be extended to multiple populations in a straightforward way, but we don't do this for simplicity.

Empirical Game Theory (Wellman, 2006) is concerned with situations in which a game can only be interacted with through sampling. The most related work to ours investigates sampling strategies and concentration inequalities for the Nash equilibrium as opposed to the α-rank. Walsh et al. (2003) introduce Heuristic Payoff Tables (HPTs) in order to choose the samples that provide the most information about the currently chosen Nash equilibrium, where information is quantified as the reduction in estimated error. This differs from our approach both in the use of α-rank as opposed to the Nash equilibrium as our solution concept, and in the criterion used to select the observed payoff. Tuyls et al. (2020) provide concentration bounds for estimated Nash equilibria. Jordan et al. (2008) find Nash equilibria from limited data by using information gain on distributions over strategies, a concept different from our information gain on distributions over ranks. We also utilizes α-rank as the solution concept, rather than Nash equilibria.

Rowland et al. (2019) introduce ResponseGraphUCB (RG-UCB), which can be viewed as a frequentist analogue to our method. They prove regret bounds in the infinite-α regime and also provide a method for obtaining uncertainty estimates in the finite-α regime, which is, however, not used as part of an adaptive sampling strategy.

Muller et al. (2020) utilise α-rank as part of a PSRO (Lanctot et al., 2017) framework. They do not use an adaptive sampling strategy for deciding which entries to sample, but are a natural application for applying our algorithm (and RG-UCB). Yang et al. (2019) introduce an approximate gradient-based algorithm which does not require access to the entire payoff matrix at once in order to compute α-rank. Although their method does not require the entire payoff matrix at every iteration, it is not designed for operating in the same incomplete information setting that we explore in this paper since they assume every entry can be cheaply queried with no noise.

Srinivas et al. (2009) prove regret bounds for Bayesian optimization with GPs. We use their concentration result to derive our bounds as well as as inspiration for our information gain criterion.

Appendix C. Query Selection by Maximizing Wasserstein Divergence

While the query objective proposed in (2) is backed both by an appealing intuition and a theoretical argument (see Section 4), is can be expensive to evaluate due to the cost of accurate entropy estimation. To address this difficulty, we also investigate an alternative involving the Wasserstein distance. The objective we consider is

$$\arg\max_a \mathbb{E}_{\tilde{m}_t \sim \tilde{M}_t}[\mathcal{W}(P(r|H_t = h_t), P(r|H_t = h_t, A_t = a, \tilde{M}_t^a = \tilde{m}_t))]. \tag{8}$$

Since the computation of Wasserstein distance from empirical distributions can be achieved by solving a linear program (Bonneel et al., 2011), equation (8) naturally lends itself to being approximated via samples. In our implementation we use POT (Flamary and Courty, 2017) to approximate this distance.

The Wasserstein distance is built on the notion of cost, which allows a practitioner the opportunity to supply additional prior knowledge. In our case, since α-ranks are probability distributions, a natural way to measure accuracy is to use the total variation distance, which corresponds to setting the cost to $c(x, y) = \frac{1}{2}\|x - y\|_1$. On the other hand, in cases where we are interested in finding the relative ordering of agents under the α-rank, an alternative cost,

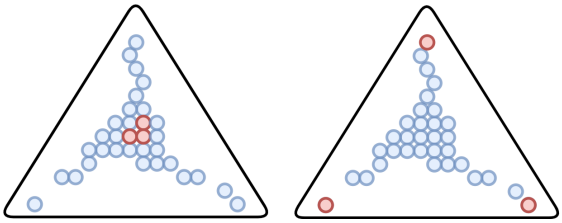

Figure 5: Diagram depicting the current belief (Blue) and 2 different hallucinated beliefs (Red). We are assuming a discrete distribution over α-ranks, where the belief is uniform across the relevant circles.

such as the Kendall Tau metric (Fagin et al., 2006) could be used. While we emphasize the ability of the Wasserstein divergence to work with any cost, we leave the empirical study of non-standard costs for future work.

It is important to note that the objective in (8) is qualitatively different to the information gain objective proposed in (2). Figure 5 provides a diagram illustrating a major difference between the two objectives. The entropy for both belief distributions shown in red is the same. In contrast, the Wasserstein distance in (8) between the current belief in blue and the hallucinated belief in red is much smaller for the distribution on the left compared to the distribution on the right.

Appendix D. Implementation Details

D.1 αIG (Bins).

For this binning entropy estimator we split $[0, 1]$ into 101 equal bins of width 0.005 (implemented by rounding to the nearest second decimal place). We then estimated the entropy using a histogram.

D.2 αIG (NSB).

The NSB estimator requires an upper bound on the total number of atoms, but since we do not know the true upper bound we utilize an estimate on the total number of possible α-ranks, which we describe below. We use the open-source implementation provided in (Marsili).

D.3 Upper bound on number of α-ranks

In the Infinite-α regime there are a finite number of possible α-ranks.

This is because only the ordering between relevant entries in the payoff matrix changes the transition matrix of the Markov Chain produced in the computation of α-rank (Rowland et al., 2019).

Let there be k populations each with S strategies. Then there are S^k strategies considered and so the transition matrix of the Markov Chain has S^k rows, one for each of the possible joint-strategies.

Each possible joint-strategy σ can transition to at most $k(S-1)$ other strategies $\tau \neq \sigma$. The probability of a self-transition is uniquely determined based on these probabilities.

This gives at most $2^{(k(S-1))}$ unique values for that row.

There are then $[2^{(k(S-1))}]^{S^k} = 2^{S^k(k(S-1))}$ unique transition matrices. Thus, the possible number of unique α-ranks is upper-bounded by $2^{S^k(k(S-1))}$. This bound is not tight, since there are many transition matrices with equal stationary distributions.

In our experiments with $K = 1$ this gives $2^{S(S-1)}$. Note that this produces a tighter bound than (10) used in our theory.

D.4 Conditioning of the belief distribution

In our experiments we found that setting $N_c = 1$ as suggested by theory is not always sufficient and use $N_c = 100$ for all experiments.

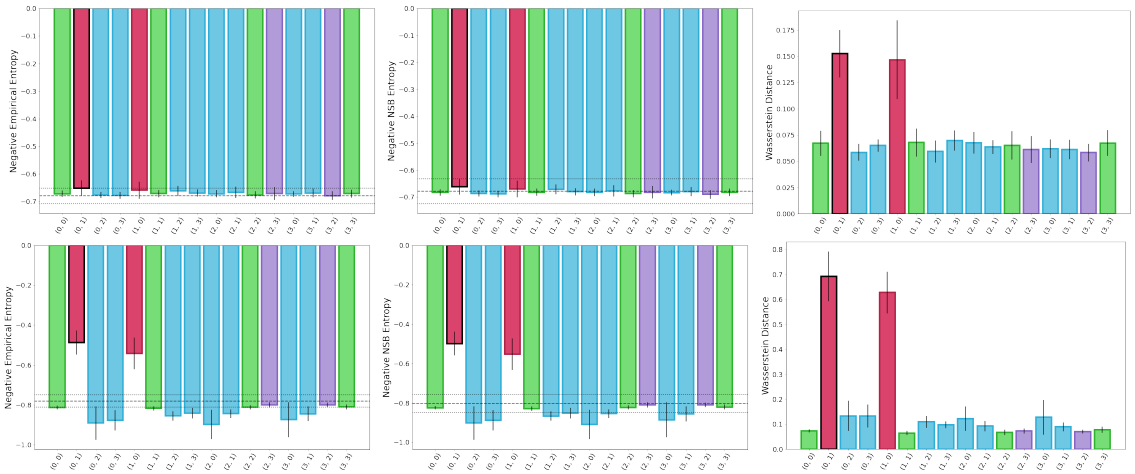

Figure 6: Comparing the values of the objectives for each entry after sampling 5 values for every entry. **Top** shows the results for $N_c = 1$. **Bottom** shows the results for $N_c = 100$. Mean and standard deviation are plotted across 10 seeds, maximum entry is highlighted in black. The mean and standard deviation of the (estimated) entropy of the current belief distribution is also plotted as a dashed horizontal line.

After drawing a sample $m'_t \sim M_t + \epsilon$, we then condition our belief distribution over α-rank on this sample N_c times and then approximate the Entropy of the resulting hallucinated belief distribution (or the Wasserstein distance between the current belief and the hallucinated belief). Theory suggests that setting $N_c = 1$ is sufficient, however empirically we found that this did not produce satisfactory results. Figure 6 shows that only conditioning once produces very little separation between the values for the different entries. Additionally, we can see that there is very little separation between the current belief's entropy and the hallucinated belief's entropy. In contrast, we can see that conditioning 100 times produces significantly more separation. Figure 7 shows the same trend, after additionally sampling 250 values for the red entries. The Wasserstein objective shows the same trend, that conditioning more

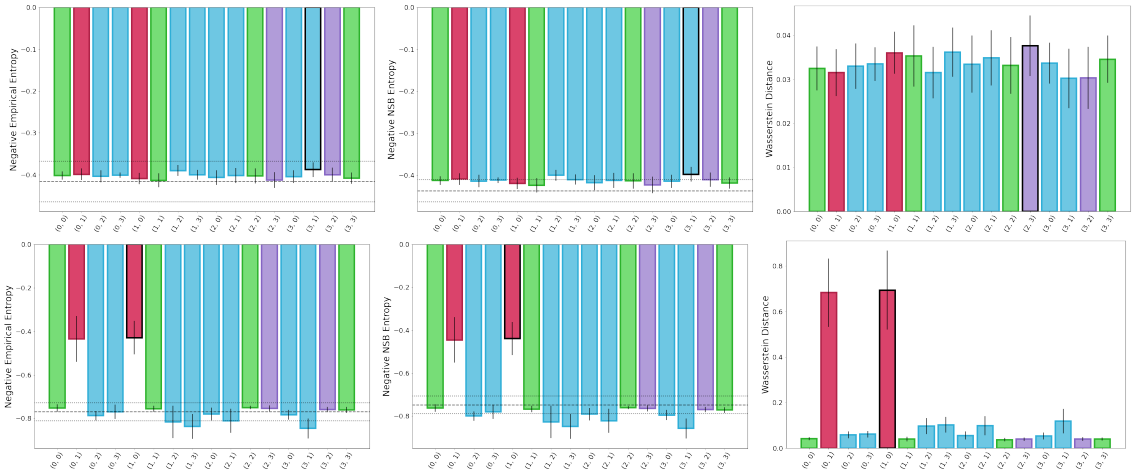

Figure 7: Comparing the values of the objectives for each entry after sampling 5 values for every entry, and then additionally sampling 250 values for the Red entries. **Top** shows the results for $N_c = 1$. **Bottom** shows the results for $N_c = 100$. Mean and standard deviation are plotted across 10 seeds, maximum entry is highlighted in black. The mean and standard deviation of the (estimated) entropy of the current belief distribution is also plotted as a dashed horizontal line.

than once produces significantly more separation. A Wasserstein Distance of 0 indicates that the two distributions are identical.

Appendix E. Further Results

Good-Bad Games To investigate our algorithm, we study two environments whose payoffs are shown in Figure 8. We start with the relatively simple environment with 4 agents. Figure

Figure 8: Payoff matrices for 2 Good, 2 Bad (Left) and 3 Good, 5 Bad (Right). Best viewed in color.

8 (Left) shows the expected payoffs, which we can interpret as the win-rate. Samples are drawn from a Bernoulli distribution with the appropriate mean. We refer to the environment as '2 Good, 2 Bad' since agents 1 and 2 are much stronger than the other 2 agents, winning 100% of the games against them. Since, the ordering between agents 3 and 4 has no effect

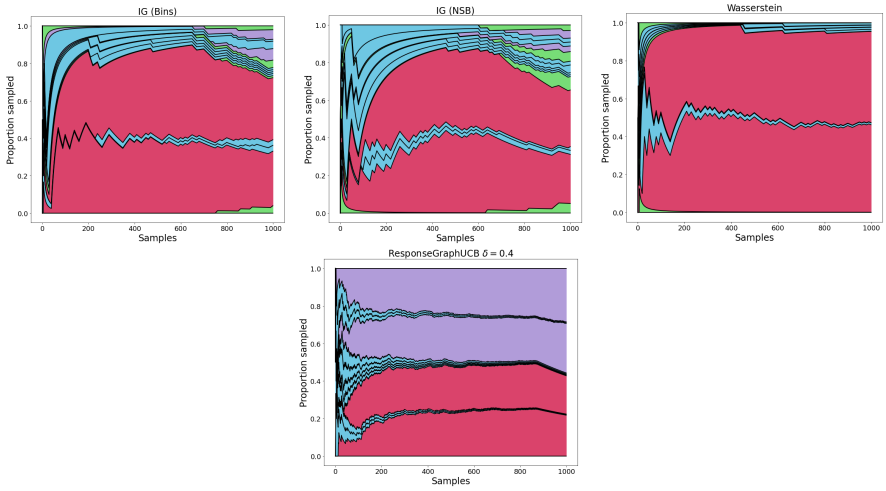

Figure 9: Proportion of entries sampled on 2 Good, 2 Bad for different methods and objectives.

on the α-rank, gathering samples to determine this ordering (highlighted in Purple) does not affect the belief distribution over α-ranks. Furthermore, since we treat this as a 1-population game, the entries highlight in Green where each agent plays against themselves do not affect the α-rank. Entries that are are necessary to determine the ordering between agents 1 and 2 are the most relevant for the α-rank and highlighted in Red. Since agent 2 is slightly better than agent 1, the true α-rank is $(0, 1, 0, 0)$. However, it can be difficult to determine the correct ordering between agents 1 and 2 without drawing many samples from these entries. The game thus provides a model for the common scenario of agents with clustered performance ratings.

Focusing on Relevant Payoffs Figure 9 presents the behavior of our method and RG-UCB on this task. As expected, RG-UCB splits its sampling between the Red entries and the Purple entries, whereas our method concentrates its sampling much more significantly on the relevant entries, determining the ordering between agents 1 and 2. This is because, in contrast to our method, RG-UCB aims to correctly determine the ordering between **all** entries used in the calculating of α-rank, irrespective of whether they matter for the final outcome.

Wasserstein Payoff Selection Does Well Comparing the Wasserstein Criterion with Information Gain payoff section, we can see that it enjoys better concentration of the sampling on the Red entries, and improved performance towards the end of training. Appendix **??** provides a more detailed analysis of this.

Bayesian and Frequentist Regret Go Down Figure 2 shows the resulting performance of the methods on this task, measured by the regret. Due to the relative simplicity of the game, there is limited benefit to our method over RG-UCB, but there is a clear benefit over more naive methods that systematically or uniformly sample the entries. We can see that the Bayesian regret J_t^B and Frequentist regrets J_t^F and J_t^M are highly correlated, providing

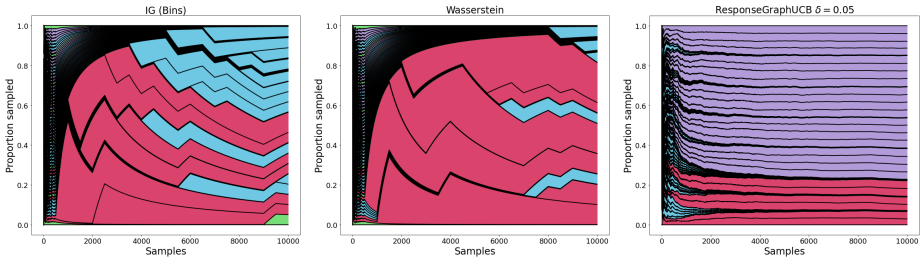

Figure 10: Proportion of entries sampled on 3 Good, 5 Bad.

empirical justification for minimizing J_t^B and validating that our method is concentrating on the ground truth.

Comparing Entropy Estimators We also investigate a larger scale version of 2 Good, 2 Bad with 3 good and 5 bad agents. Figure 3 shows the results, demonstrating a clear benefit for our method using the Binning estimator for the Information Gain or the Wasserstein objective. The performance of the NSB entropy estimator is not surprising, given the significantly larger nature of this task compared to '2 Good, 2 Bad'. A necessary part of the NSB estimator is an upper-bound on the total number of atoms in the distributions, for which we only have a crude approximation that grows exponentially with the size of the payoff matrix. Figure 10 shows the proportion of entries sampled for αIG (Bins), the Wasserstein objective and RG-UCB. Once again, RG-UCB spends a significant part of its sampling budget determining the ordering between agents that do not have an effect on the α-rank of the game (in this task agents 3 to 8). In contrast, our methods concentrate their sampling on the Red entries that determine the payoffs between the top 3 agents, and hence the true α-rank. In general, our algorithm does not depend as much on accurate estimates on entropy but on identifying the distribution with a lowest entropy, for which the NSB estimator isn't tuned.

Figure 11 shows the values used by the different objectives during training. The top row shows the values after sampling 5 values for each entry, showing a clear seperation between the Red entries and the rest. The bottom row shows the values after additionally sampling 250 values for the Red entries. We can then see a large difference between the Wasserstein and Entropy-based objectives. As desired the Wasserstein-based objective shows a large separation between the Red entries and the others, additionally assigning the smallest values to the irrelevant Green, and Purple entries.

Figures 13, 14 and 15 show similar results for 3 Good, 5 Bad.

Gaussian Games Finally, Figure 4 shows the results on 4x4 games with Gaussian noise, demonstrating improved performance across all 3 regret metrics for the αIG (Bins). This is empirical confirmation of our theoretical results, and shows that our method achieves better performance compared to RG-UCB on general games.

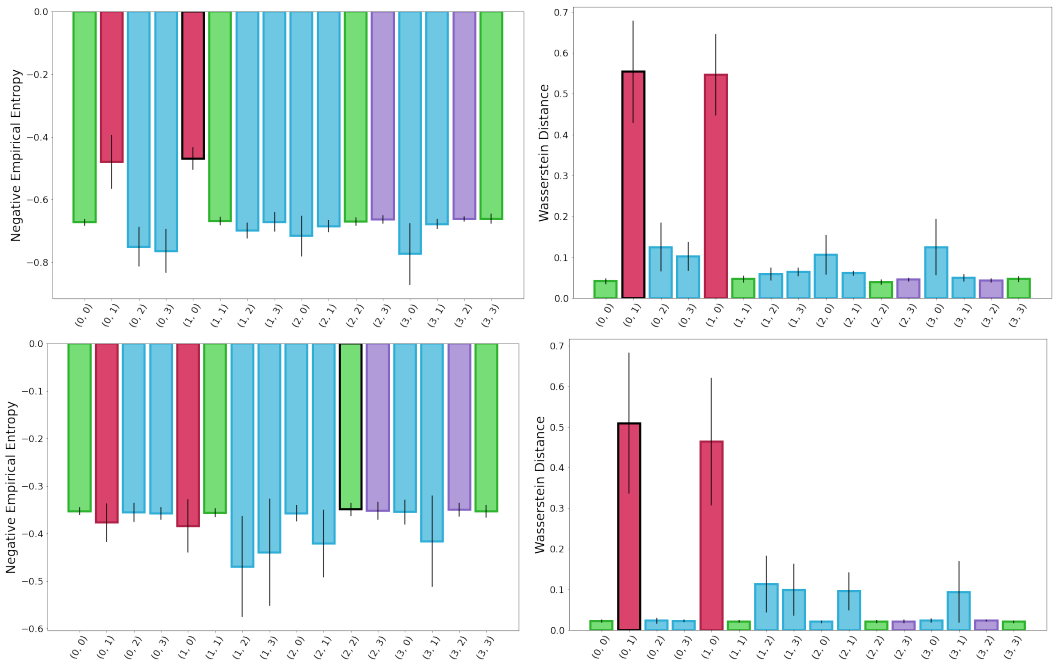

Figure 11: Value of the objectives for each entry after sampling 5 values for every entry (top) and additionally sampling 250 values for the red entries (below). Mean and standard deviation are plotted across 10 seeds, maximum entry is highlighted in black.

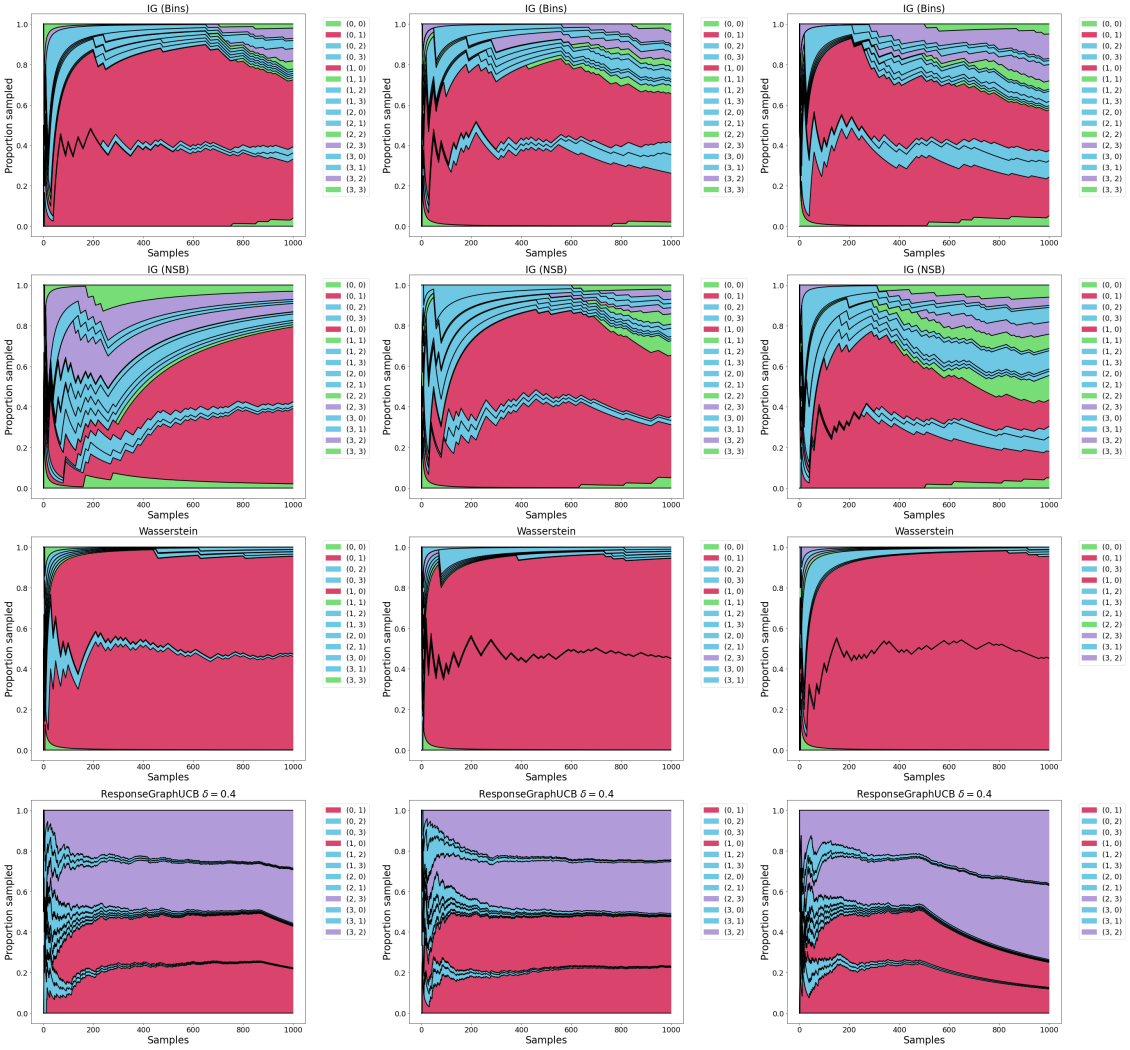

Figure 12: Proportion of entries sampled on 2 Good, 2 Bad for more seeds.

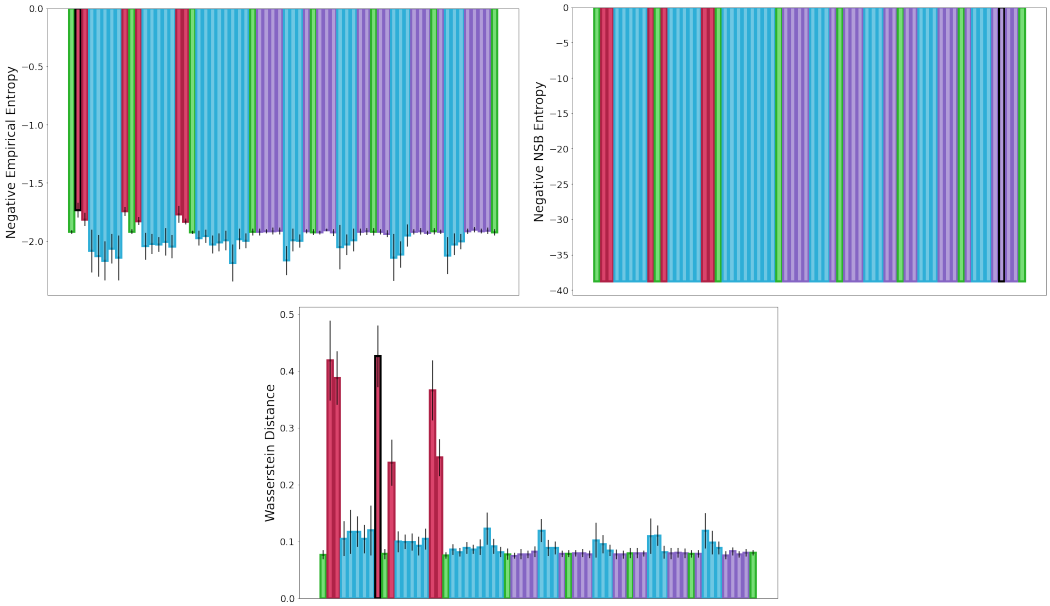

Figure 13: Values of the objectives for each entry on '3 Good, 5 Bad' after sampling 5 values for every entry. Mean and standard deviation across 10 seeds is shown, maximum highlighted in black.

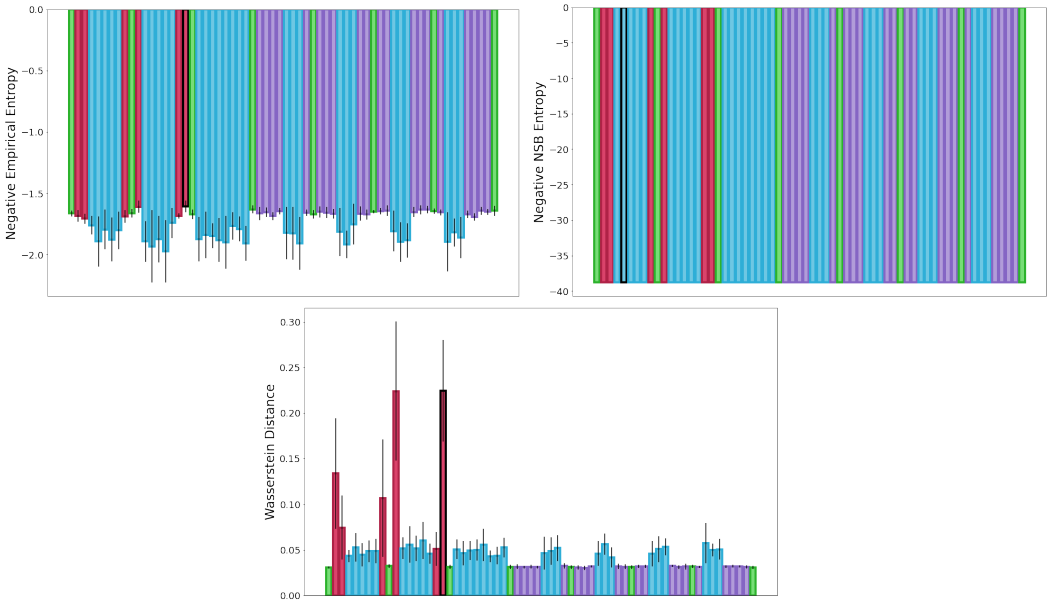

Figure 14: Values of the objectives for each entry on '3 Good, 5 Bad' after sampling 5 values for every entry, and additionally sampling 1000 values for the red entries. Mean and standard deviation across 10 seeds is shown, maximum highlighted in black.

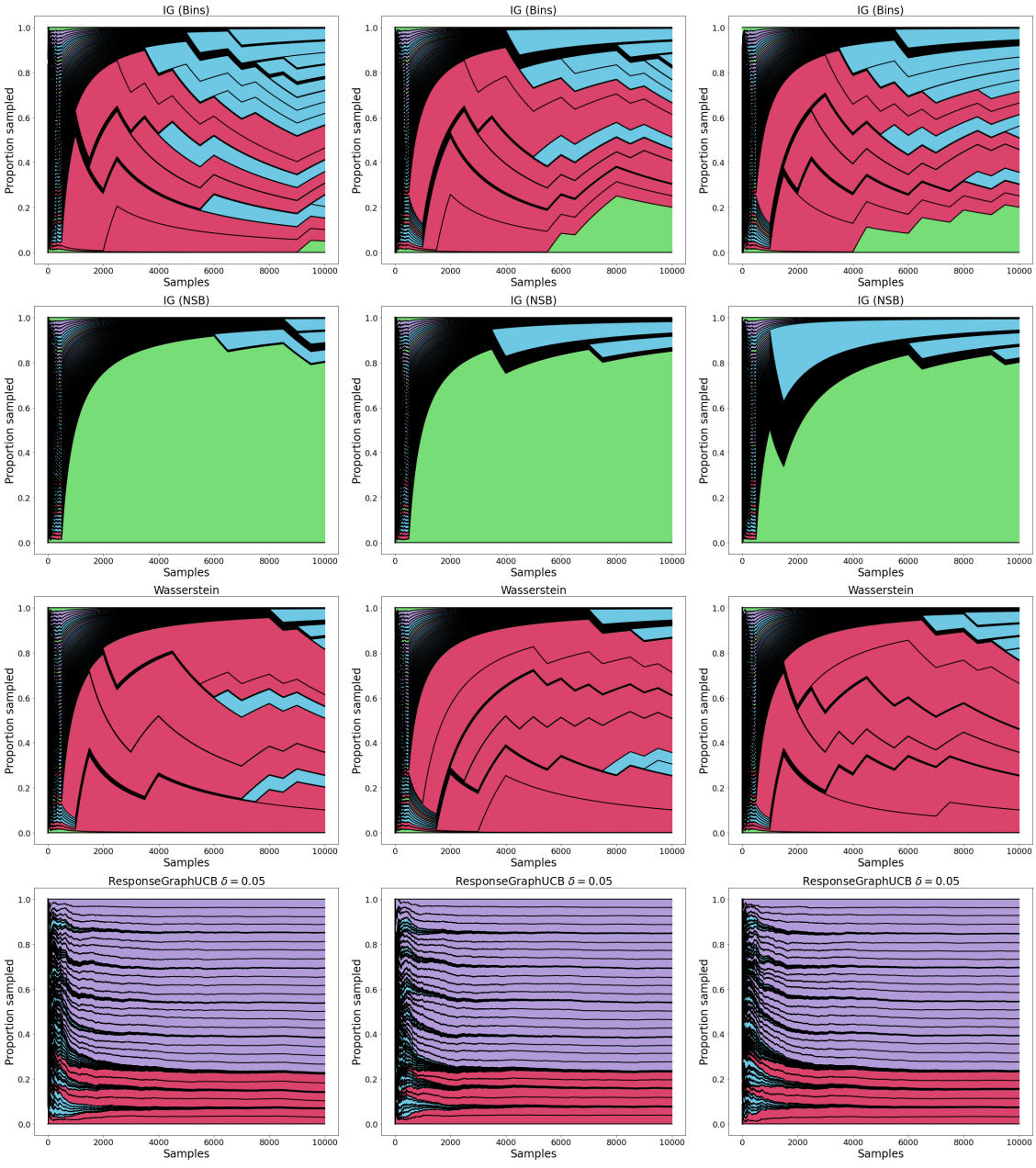

Figure 15: Proportion of entries sampled on 3 Good, 5 Bad for more seeds.

Appendix F. Experimental Setup

F.1 α-Rank

In the computation of α-rank we set $\epsilon = 10^{-6}$ in all of our experiments.

F.2 Baselines

ResponseGraphUCB, uses a Hoeffding Bound to construct the confidence interval: $(\mu - \sqrt{\frac{\log(2/\delta)(b-a)}{2N}}, \mu + \sqrt{\frac{\log(2/\delta)(b-a)}{2N}})$. Where δ is the confidence hyperparameter of the algorithm, b is the maximum value an entry can take, a is the minimum value, and N is the number of times a value has been seen for an entry. For all experiments we swept over $\delta \in \{0.4, 0.3, 0.2, 0.1, 0.05, 0.01, 0.001\}$, and the final value is selected by considering the area under the curve for $1 - P(f(\bar{M}) = r_{GT})$.

Uniform. The entry to sample if picked uniformly from all possible entries.

Payoff. The entry which maximises the information gain between its sample and the payoff distribution is chosen. For an isotropic Gaussian this is equivalent to picking the entry which the lowest count, which results in systematic sampling of each entry. For a non isotropic Gaussian the same procedure as (Srinivas et al., 2009) is used.

F.3 Graphs

$1 - P(f(\bar{M}) = r_{GT})$. At each timestep we compute the α-rank of the mean payoff matrix. Equality is determined if $|f(\bar{M}) = r_{GT})|_1 < 0.01$. The choice of 0.01 is largely arbitrary, we did not find the results to be sensitive to this.

$1 - P(r_{GT})$. 100 times during training (evenly spaced), we sample 2000 samples from the current belief distribution over α-ranks. $P(r_{GT})$ is determined from these 2000 samples (which are aggregated by rounding each value to the nearest $3d.p.$) by counting the number of sampled α-ranks r such that $|r - r_{GT}| < 0.01$.

$1 - P(r_*)$, is determined similarly to $1 - P(r_{GT})$, except we use the 2000 samples to calculate the mode.

For ResponseGraphUCB, we construct a distribution over the payoff entries as being uniform over the confidence intervals.

F.4 Environments

F.5 2 Good, 2 Bad

Observations are sampled from $Ber(x)$, where x is the value in the payoff matrix.

αIG(Bins), αIG(NSB), αWass. Prior used is $\mathcal{N}(\mu_0, \sigma_0^2)$, with aleatoric noise σ_A^2. $\mu_0 = 0.5$. Swept over $\sigma_0^2 \in \{0.5, 1\}, \sigma_A^2 \in \{0.25, 0.5, 1\}$. 20 samples are used to approximate the expectation, $N_e = 20$. 1000 samples are drawn from the belief distribution(s) to approximate the quantities inside the expectation, $N_b = 1000$. We set $N_r = 10$.

For all 3 methods we set $\sigma_0^2 = 1$. For αIG (Bins) and αIG (NSB) we set $\sigma_A^2 = 0.5$, and for αWass we set $\sigma_A^2 = 0.25$.

ResponseGraphUCB. We set $\delta = 0.4$. Maximum value is 1, minimum value is 0.

F.6 3 Good, 5 Bad

Observations are sampled from $Ber(x)$, where x is the value in the payoff matrix.

αIG(Bins), αIG(NSB), αWass. Prior used is $\mathcal{N}(\mu_0, \sigma_0^2)$, with aleatoric noise σ_A^2. $\mu_0 = 0.5$. Swept over $\sigma_0^2 \in \{0.5, 1\}, \sigma_A^2 \in \{0.25, 0.5, 1\}$. $N_e = 10$. $N_b = 500$. $N_r = 500$.

For all 3 methods we set $\sigma_0^2 = 1$. For αIG (Bins) and αIG (NSB) we set $\sigma_A^2 = 0.5$, and for αWass we set $\sigma_A^2 = 0.25$.

ResponseGraphUCB. We set $\delta = 0.05$.

F.7 4x4 Gaussian

To match the games considered in our theoretical analysis, Observations are sampled from $\mathcal{N}(x, 1)$ and then clipped to be within 1 of x, where x is the value of the entry in the payoff matrix. The values of x are uniformly drawn from $[0, 1)$.

αIG(Bins), αIG(NSB), αWass. Prior used is $\mathcal{N}(\mu_0, \sigma_0^2)$, with aleatoric noise σ_A^2. $\mu_0 = 0.5$. Swept over $\sigma_0^2 \in \{0.5, 1\}, \sigma_A^2 \in \{0.5, 1\}$. $N_e = 10$. $N_b = 500$. $N_r = 100$. For αIG(Bins) we set $\sigma_0^2 = 1$ and $sigma_A^2 = 1$. For αIG(NSB) we set $\sigma_0^2 = 0.5$ and $sigma_A^2 = 0.5$. For αWass we set $\sigma_0^2 = 1$ and $sigma_A^2 = 0.5$.

ResponseGraphUCB. We set $\delta = 0.3$. Maximum value is 2, minimum value is -1.

Appendix G. Proofs

Permutation Property We begin by explicitly stating a property of the infinite-alpha version of α-rank.The function f computing the α-rank satisfies the permutation property, defined as

$$\pi(M_1) = \pi(M_2) \implies f(M_1) = f(M_2). \tag{9}$$

Here, $\pi(M)$ denotes the ordering of the elements of the vector M using the standard $\geq$ operation on real numbers. This is the same property exploited by frequentist analysis by Rowland et al. (2019). Property (9) implies that R is a finite set and

$$|R| \leq N!. \tag{10}$$

Information Gain and Entropy We recall a formula for the information gain in terms of the entropy.

$$\mathbb{I}(r\,;\,(M',a) \mid H_t = h_t) = \mathbb{H}\left(r \mid H_t = h_t\right) - \mathbb{H}\left(r \mid H_t = h_t, A_t = a, M'_t\right). \tag{11}$$

We now show a regret bound for a policy that maximizes information gain on the payoffs.

Proposition 1 [Regret Bound For Information Gain on Payoffs] If we select actions using strategy π_{IGM}, regret is bounded as

$$J_T \leq 1 - \mathbb{E}_{h_t}\left(P(r = r_{\mathrm{GT}} | H_T = h_T)\right) \leq 1 - Te^{g(T)} \text{ where } g(T) = \mathcal{O}(-\sqrt[3]{\Delta^2 T}). \tag{12}$$

Proof Fix a history h_T. By assumption of separability, we have

$$P(r = r_{\mathrm{GT}} | H_T = h_t) \geq P\left(|M_t - M^\star|_\infty \leq \frac{\Delta}{2}\right). \tag{13}$$

We now use concentration results for Gaussian Processes. Specifically, we invoke Corollary 5, stated in the appendix, together with an explicit formula for $g(T)$. ■

We move on to show a bound for a policy that maxinizes information gain on the alpha ranks.

Proposition 2[Regret Bound For Information Gain on Belief over Alpha Ranks] If we select actions using strategy π_{IGR}, regret is bounded as

$$1 - P(r = r_\star | H_T^{\mathrm{IGR}} = h_t) \leq \delta\Big[z(T) \leq h_b(|R|^{-1})\Big]\, h_b^{-1}(z(T)) + \delta\Big[z(T) > h_b(|R|^{-1})\Big], \tag{14}$$

where

$$z(T) = \delta\left[Te^{g(T)} \geq \frac{1}{2}\right] h_b(Te^{g(T)}) + \delta\left[Te^{g(T)} \leq \frac{1}{2}\right] N \log N \tag{15}$$

and $g(T)$ is as in Proposition 1.

Proof We start by bounding the entropy of the alpha-rank distribution. Denote by $h_b(p) = -(p\log p + (1-p)\log(1-p))$ the entropy of a Bernoulli random variable with parameter p. Also, introduce the abbreviation $p^\star = P(r = r_\star | H_T = h_t))$.

We have

$$\begin{aligned}
&\mathbb{H}\left(r \mid H_T^{\mathrm{IGR}}\right) \\
&\quad \overset{(a)}{\leq} \mathbb{H}\left(r \mid H_T^{\mathrm{IGM}}\right) \\
&\quad \overset{(b)}{\leq} \mathbb{E}_{h_T \sim H_T^{\mathrm{IGM}}}\left(h_b(p^\star) + (1-p^\star)\log(|R|)\right) \\
&\quad \overset{(c)}{\leq} \mathbb{E}_{h_T \sim H_T^{\mathrm{IGM}}}\left(h_b(p^\star) + (1-p^\star)N\log N\right).
\end{aligned}$$

Here, (a) follows from the definition of π_{IGR} and equation (11), (b) follows by Lemma 6 and (c) holds because $|R| \leq N!$ by equation (10). Combining the above with the bound $p^\star \geq Te^{g(T)}$ from Proposition 1, we have

$$\mathbb{H}\left(r \mid H_T^{\mathrm{IGR}}\right) \leq \underbrace{\delta\left[Te^{g(T)} \geq \frac{1}{2}\right] h_b(Te^{g(T)}) + \delta\left[Te^{g(T)} \leq \frac{1}{2}\right] N\log N}_{z(T)}. \tag{16}$$

We now proceed to bound the probability of $r_\star$ in terms of the entropy of the alpha-ranks. We have

$$h_b(P(r = r_\star | H_T^{\mathrm{IGR}} = h_t)) \leq \mathbb{H}\left(r \mid H_T^{\mathrm{IGR}}\right).$$

This, together with (16) implies

$$\begin{aligned}
&1 - P(r = r_\star | H_T^{\mathrm{IGR}} = h_t) \leq \\
&\quad \delta\Big[z(T) \leq h_b(|R|^{-1})\Big]\, h_b^{-1}(z(T)) + \delta\Big[z(T) > h_b(|R|^{-1})\Big].
\end{aligned}$$

Here, we denoted by h_b^{-1} the inverse of the restriction of h_b to the interval $(0, \frac{1}{2})$.

■

We use the following result by Srinivas et al. (2009, their Theorem 6), which we specialize in our notation. We use the term Gaussian Process despite the fact that the index set is finite, since the model includes observation noise.

Lemma 3 (Srinivas et al., Concentration for a Gaussian Process) *Consider a Gaussian Process M, with N indices. Assume M uses a zero-mean prior with constant variance σ_0^2 and observation noise σ_A. The posterior process M_t is obtained by conditioning on t observations. The observations are obtained as $m_t = M^\star[a_t] + \epsilon_t$, where ϵ_t are i.i.d random variables with support bounded by $[-\sigma_0, \sigma_0]$. Denote the RKHS norm of $M^\star$ under the GP prior with $\|M^\star\|_k$. Denote by $\gamma_t^\star$ the maximum information gain about M obtainable in t timesteps. Then, for any $\Delta > 0$, and for any timestep t, we have*

$$P\left[|M - M^\star|_\infty \leq \frac{\Delta}{2}\right] \geq 1 - te^{-\sqrt[3]{\frac{\left(\frac{\Delta}{2\sigma_t^{max}}\right)^2 - 2\|M^\star\|_k}{300\gamma_t^\star}}}. \tag{17}$$

The above lemma requires knowledge of the RKHS norm and the maximum obtainable information gain.

Lemma 4 (Worst-Case Constants) *For any kernel, we have*

$$\|M^\star\|_k \leq \frac{1}{\sigma_0^{-2}}\|M^\star\|_2^2 \;\; \textit{and} \;\; \gamma_t^\star \leq \frac{1}{2}\log\det(I + \sigma_A^{-2}K).$$

Moreover, for a strategy that maximizes information gain on payoffs, we have

$$\sigma_t^{max} \leq \frac{\sigma_A\sigma_0}{\sqrt{\sigma_A^2 + (\frac{T}{N}-1)\sigma_0^2}}.$$

Proof The inequalities for posterior variance and the RHKS norm are obtained by using the independent kernel, which represents the worst-case. The inequality for information gain follows by writing

$$\gamma_t^\star = \frac{1}{2}\log\frac{\det(I + \sigma_A^{-2}K)}{\det(I + \sigma_A^{-2}\Sigma)} \leq \frac{1}{2}\log\det(I + \sigma_A^{-2}K). \tag{18}$$

The inequality follows since the denominator is greater than one. Here, we denoted the prior covariance with K and the posterior covariance with Σ. ■

Corollary 5 *For a strategy that maximizes the payoff information gain and for any time-step T, we have:*

$$P\left[|M_t - M^\star|_\infty \leq \frac{\Delta}{2}\right] \geq 1 - Te^{g(T)}, \;\; \textit{where} \;\; g(T) = \mathcal{O}(-\sqrt[3]{\Delta^2 T})$$

Specifically,

$$g(T) = \frac{\left(\frac{\Delta}{2}\right)^2 \frac{\sigma_A^2 + (\frac{T}{N}-1)\sigma_0^2}{\sigma_A^2\sigma_0^2} - 2\frac{1}{\sigma_0^{-2}}\|M^\star\|_2^2}{300\frac{1}{2}\log\det(I + \sigma_A^{-2}K)}.$$

Lemma 6 (Upper Bound on Entropy) *For any discrete random variable x with n outcomes, we have, for each outcome i*

$$\mathbb{H}(x) \leq h_b(p_i) + (1 - p_i)\log(n-1).$$

Proof Without loss of generality, assume $i = 1$.

$$\begin{aligned}
\mathbb{H}(x) &= -p_1 \log p_1 - \sum_{j>1} p_j \log(p_j) \\
&= -p_1 \log p_1 - (n-1) \sum_{j>1} \frac{1}{n-1} p_1 \log(p_j) \\
&\overset{(a)}{\leq} -p_1 \log p_1 - (n-1) \left(\sum_{j>1} \frac{p_j}{n-1} \right) \log \left(\sum_{j>1} \frac{p_j}{n-1} \right) \\
&= -p_1 \log p_1 - (1-p_1) \log \left(\frac{1-p_1}{n-1} \right) \\
&= -p_1 \log p_1 - (1-p_1) \log \left(\frac{1-p_1}{n-1} \right) \\
&= h_b(p_j) + (1-p_j) \log(n-1)
\end{aligned}$$

There, (a) follows from Jensen's inequality applied to the function $x \log x$. ■