UNIVERSITY
OF OSLO

Thomas Kleine Büning

Learning in the Presence of Cooperative, Adversarial and Strategic Agents

Thesis submitted for the degree of Philosophiae Doctor

Department of Informatics
Faculty of Mathematics and Natural Sciences

2024

© Thomas Kleine Büning, 2024

Series of dissertations submitted to the
Faculty of Mathematics and Natural Sciences, University of Oslo
No. 2796

ISSN 1501-7710

All rights reserved. No part of this publication may be
reproduced or transmitted, in any form or by any means, without permission.

Cover: UiO.
Print production: Graphic center, University of Oslo.

In memory of Klaus

Preface

This thesis is submitted in partial fulfillment of the requirements for the degree of *Philosophiae Doctor* at the University of Oslo. The research presented here was conducted at the University of Oslo, under the supervision of Christos Dimitrakakis and Ingrid Chieh Yu. This work was supported by the Norwegian Research Council through the grant No 302203 "Algorithms and Models for Socially Beneficial AI".

The thesis is a collection of five papers, which are preceded by three introductory chapters. The papers were authored in collaboration with several co-authors. I am the main contributor for four out of the five papers. All authors of the remaining paper contributed equally.

Acknowledgements

I want to thank my advisors, Christos and Ingrid, for their continuous support and for being accepting of all the research directions I wanted to explore during my PhD. Many thanks also to the other members of Christos' group, including Meirav and Marie for the fun trips and discussions in Oslo; Emilio, Hannes, Divy, and Milad for the interesting chats during group meetings, retreats, and at Chalmers; and Victor and Andreas for their warm welcome when visiting Neuchatel. I also want to thank my other co-authors Aadirupa, Debabrota, and Haifeng for their support and the things they taught me. Finally, I am also grateful to the many people at UiO for their help when setting up in Oslo.

Thomas Kleine Büning
Oslo, February 2024

Abstract

Machine learning has found application in various fields, such as healthcare, robotics, and personalized recommendations. In many of these applications, learning algorithms interact with other agents that can exhibit diverse behaviors. For instance, in scenarios of human-AI collaboration, the objective of the learning algorithm is to jointly complete a task in cooperation with a benevolent human. These situations frequently arise in, e.g., robotics and human-assisted autonomous driving, where effective collaboration requires the learning algorithm to understand and adapt to its human partner. In yet other scenarios such as online recommendation, learning algorithms interact with users who directly influence decision outcomes and whose preferences may evolve over time. Sometimes agents may also respond strategically to the learning algorithm to maximize their own benefit. For example, e-commerce retailers or web designers may game a learning algorithm, e.g., a recommendation system or search engine, to improve their rank and maximize their exposure to potential customers.

We see that there are many ways a learning algorithm can interact with agents. Here, we distinguish between three types of agents depending on their objectives. Generally speaking, *cooperative agents* are aligned with the goals of the learning algorithm as is the case in human-AI collaboration. Conversely, *adversarial agents* oppose the learning algorithm's objective and act maliciously so as to impede the learning algorithm as much as possible. Finally, we call agents *strategic* when they are neither fully aligned nor opposed to the learning algorithm's goals, but instead have an own objective which places them somewhere between cooperative and adversarial behavior. Learning in the presence of each of these types of agents brings its own challenges and peculiarities. In this thesis, we study these within the reinforcement learning framework, which involves sequential interactions with an unknown environment and cooperative, adversarial, or strategic agents directly or indirectly influencing the environment and the rewards the learning algorithm receives.

As an instance of learning in the presence of cooperative agents, we study the problem of collaborating with a potentially suboptimal human partner without access to the *joint* reward function. This connects to the problem of inferring a reward function from demonstrations, called inverse reinforcement learning, and we propose interactive learning setups which allow for actively querying information about the unknown reward function from a human partner. We theoretically and empirically demonstrate the benefits of inverse reinforcement learning in a collaborative environment where the learning algorithm gets to repeatedly interact with a human and probe their behavior.

In the adversarial setting, we first study the scenario where the environment undergoes adversarial changes over time, which could be due to malicious attacks

or evolving user preferences. Here, we focus on dynamic regret minimization in non-stationary dueling bandits, which requires the learning algorithm to detect and adapt to changes in an online fashion. We study and discuss several notions of non-stationary complexity in dueling bandits and propose a learning algorithm that achieves near-optimal dynamic regret w.r.t. the number of best arm switches, without prior knowledge of the number of switches. As another instance of learning in the presence of adversarial agents, we also study the situation where an adversary chooses a worst-case problem instance (or distribution over problem instances) in response to our learning algorithm. Here, we consider the Bayesian setting where the problem can be viewed as a minimax-Bayes game. We show that solutions to this minimax game between the learning algorithm and the adversary can yield more robust reinforcement learning policies.

Finally, we study online learning in the presence of agents that respond strategically to the learning algorithm so as to maximize their payoffs. We propose a strategic variant of the multi-armed bandit problem and construct an incentive-aware learning algorithm that incentivizes desirable agent strategies while minimizing regret. We thereby connect online learning and mechanism design, two popular and influential research areas, which, however, have been mostly studied separately so far. For the proposed strategic multi-armed bandit problem, we derive trade-offs between regret minimization and incentivizing all agents to act in a desirable fashion. Moreover, our work provides insights into the complexity of online mechanism design under uncertainty.

Sammendrag

Maskinlæring har funnet anvendelse på en rekke områder, for eksempel innen helsevesenet, robotteknologi og personaliserte anbefalinger. I mange av disse bruksområdene samhandler læringsalgoritmer med andre agenter som kan oppføre seg på ulike måter. I scenarier med menneske-AI-samarbeid er målet for læringsalgoritmen for eksempel å fullføre en oppgave i samarbeid med et velvillig menneske. Slike situasjoner oppstår ofte innen for eksempel robotteknologi og menneskeassistert autonom kjøring, der et effektivt samarbeid krever at læringsalgoritmen forstår og tilpasser seg den menneskelige partneren. I andre scenarier, for eksempel i forbindelse med nettbaserte anbefalinger, samhandler læringsalgoritmer med brukere som direkte påvirker beslutningsutfallet og hvis preferanser kan endre seg over tid. Noen ganger kan også agenter reagere strategisk på læringsalgoritmen for å maksimere sine egne fordeler. For eksempel kan nettbutikker eller webdesignere spille på en læringsalgoritme, f.eks. et anbefalingssystem eller en søkemotor, for å forbedre sin egen rangering og maksimere eksponeringen for potensielle kunder.

Vi ser at det er mange måter en læringsalgoritme kan samhandle med agenter på. Her skiller vi mellom tre typer agenter avhengig av hvilke mål de har. Generelt sett er *kooperative agenter* på linje med læringsalgoritmens mål, slik tilfellet er i menneske-AI-samarbeid. Motsatt motsetter *adversarial agenter* seg læringsalgoritmens mål og opptrer ondsinnet for å hindre læringsalgoritmen så mye som mulig. Til slutt kaller vi agenter *strategiske* når de verken er helt på linje med eller motstander av læringsalgoritmens mål, men i stedet har et eget mål som plasserer dem et sted mellom samarbeid og motstand. Læring i nærvær av hver av disse agenttypene medfører sine egne utfordringer og særegenheter. I denne avhandlingen studerer vi disse innenfor rammeverket for forsterkningslæring, som innebærer sekvensielle interaksjoner med et ukjent miljø og kooperative, kontradiktoriske eller strategiske agenter som direkte eller indirekte påvirker miljøet og belønningen læringsalgoritmen mottar.

List of Papers

Paper I

Interactive Inverse Reinforcement Learning for Cooperative Games. Thomas Kleine Buening, Anne-Marie George, Christos Dimitrakakis. *In 39th International Conference on Machine Learning (ICML) 2022.*

Paper II

Environment Design for Inverse Reinforcement Learning. Thomas Kleine Buening, Christos Dimitrakakis. *Presented at the Human in the Loop Learning Workshop at NeurIPS 2022.*

Paper III

ANACONDA: An Improved Dynamic Regret Algorithm for Adaptive Non-Stationary Dueling Bandits. Thomas Kleine Buening, Aadirupa Saha. *In 26th International Conference on Artificial Intelligence and Statistics (AISTATS) 2023.*

Paper IV

Minimax-Bayes Reinforcement Learning. Thomas Kleine Buening*, Christos Dimitrakakis*, Hannes Eriksson*, Divya Grover*, Emilio Jorge*. *In 26th International Conference on Artificial Intelligence and Statistics (AISTATS) 2023.*

Paper V

Bandits Meet Mechanism Design to Combat Clickbait in Online Recommendation. Thomas Kleine Buening, Aadirupa Saha, Christos Dimitrakakis, Haifeng Xu. *To appear in 12th International Conference on Learning Representations (ICLR) 2024.*

(* denotes equal contribution)

Other publications by the author that are not included in this thesis are:

On Meritocracy in Optimal Set Selection. Thomas Kleine Buening, Meirav Segal, Debabrota Basu, Anne-Marie George, Christos Dimitrakakis. *In Equity and Access in Algorithms, Mechanisms, and Optimization (EAAMO) 2022.*

Contents

Chapter 1

Introduction

The field of artificial intelligence, and specifically machine learning, has seen tremendous advances in recent years, driven by breakthroughs in deep learning, reinforcement learning, and the availability of large datasets. As a result, AI systems have been deployed in various applications, including natural language processing, autonomous vehicles, healthcare diagnostics, and online recommendation. In many cases, these systems operate in environments where other agents are present and directly or indirectly influence the environment and decision outcomes.

Despite this, the nature of other agents is frequently overlooked. For example, we may assume that a human user is oblivious and impartial in the sense that they do not wish to help, hinder, or influence the AI system in any specific way. Of course, this is often not the case. In collaborative scenarios, such as human-assisted autonomous driving, the human user actively tries to assist the AI system in driving safely. Conversely, malicious actors may manipulate the environment and training data to impede the AI system's learning process and cause failures. Adversarial notions are also useful for modeling arbitrary behavior or changes in the environment. For example, user behavior may drastically shift due to seasonality or other external factors, which then an online recommendation system must detect and adapt to. In other cases, agents may act strategically so as to maximize their own benefit, neither fully aligning with nor fully opposing the AI system's goals. For instance, vendors on e-commerce platforms may game the recommendation system to maximize their exposure and click-rates.

We see that there are different ways in which an AI system can interact with its environment and the agents that are a part of it. In this thesis, we analyze such situations in the context of reinforcement learning and study learning in the presence of *cooperative*, *adversarial*, and *strategic* agent behavior.

Reinforcement learning is a learning paradigm where one or several agents interact with an unknown environment to maximize the total reward they receive over time. For instance, the reinforcement learning framework can model human-AI collaboration tasks as two agents interacting with the environment and maximizing a *common* reward. When acting in the presence of adversarial agents, we typically assume that these adversaries try to impede the learning algorithm as much as possible by choosing worst-case problem instances or altering the environment over time. In strategic problems, we can consider the situation where the learner and the strategic agents all interact with the same environment, however, every party wants to maximize their own reward, which may be different for all agents. We study several different reinforcement learning frameworks in this thesis, ranging from multi-armed bandits to two-player Markov games, and hope to shed some light on the advantages and disadvantages of learning in the

presence of cooperative, adversarial, and strategic agents.

1.1 Thesis Outline

The thesis is structured as follows.

- The remainder of this **Chapter 1** states the main research questions. This is then followed by an overview and a short summary of the papers.
- **Chapter 2** contains a brief introduction to the reinforcement learning framework and provides some basic background to the problems studied in this thesis.
- **Chapter 3** discusses the main findings of this thesis and derives conclusions from them. The main contributions of the thesis are summarized and presented with respect to the main research questions. In addition, a few future directions for research are outlined.
- Finally, **Papers** contains all the papers which are a part of this thesis.

1.2 Research Questions

We wish to understand both the benefits and the challenges when learning in the presence of *cooperative*, *adversarial*, and *strategic* agents. We organize our main research questions accordingly. We first provide a one to two sentence high-level question which is then followed by a brief discussion.

Q1. (Cooperative): *Can we learn an unknown reward function more precisely and more efficiently by actively seeking information from a human partner through repeated interaction? And if so, how much is the benefit of such repeated interaction?*

As AI systems become more powerful, it is important to align such systems' goals with that of their human users and society as a whole, which was recently once again highlighted by the development and dissemination of powerful large language models [Kad+23; Zie+19]. This is generally known as the value alignment problem and is based on the premise that a human system designer cannot reliably hand-specify a goal to the AI system prior to deployment [Gab20; Rus21]. In fact, the challenge of specifying suitable reward functions, i.e., a numerical objective to maximize, is one of the main barriers to the wider application of reinforcement learning in real-world settings. In particular, when manually designing reward functions, unsafe agent behavior as well as phenomena such as reward hacking have been observed [CA16; Ska+22].

A popular approach to address this challenge within the reinforcement learning framework is Inverse Reinforcement Learning (IRL) which aims to infer the reward function from human demonstrations [NR+00; Rus98]. The basic IRL setup assumes that the learner observes the human demonstrate the

task in the environment first, on the basis of which the learner then attempts to estimate the reward function that the human is (implicitly) maximizing. However, IRL suffers from several limitations, one of the most severe being that it is generally impossible to fully recover the human's true reward function [CCS21; Kim+21]. As a result, even though some recent work has attempted to quantify the estimation error of IRL and derive theoretical guarantees [LKR22; Zen+22, e.g.], the inherent limitations of IRL in its classical setup cannot be overcome. In view of this, it is natural to ask what additional assumptions or different learning setups may improve the reward inference and could enable us to provably recover the human's true reward function.

Naturally, the human expert in IRL can be assumed to be cooperative and willing to help the AI system infer the reward function. We can thus consider the IRL problem as a cooperative game. Taking this perspective, prior work has viewed the human as a teacher with the goal of finding the best teaching strategy for the human demonstrator in order to provide better data for the learner [BN19; CL12; Had+16; Tsc+19]. However, such teaching strategies may be difficult to implement for a human and even if implemented can be insufficient to recover the true reward function. Alternatively, we can hypothesize that direct interaction of the learner and the human via joint completion of a collaborative task may help the inference of the reward function. For example, when the learner and human jointly solve a collaborative task, as in robotic assistance scenarios, the human partner may indirectly provide feedback that allows us to learn a robust and safe reward function. If the IRL agent is able to properly interpret such interactions with the human, it may benefit the reward inference due to the diversity of interactions and the learner's ability to query for specific scenarios. The inference thus becomes an active (or interactive) learning problem [BCN18; LKR22; LMM09]. More generally, in view of the limitations of IRL, it is interesting to study what data would be sufficient to learn a (near-)optimal representation of the true reward functions, and how we could actively seek such data from a human demonstrator.

Q2. (Adversarial): *What if the environment is chosen adversarially and changes over time? How does this impact our ability to learn efficiently?*

Typically, reinforcement learning is studied under the assumption that the environment is fixed a priori and does not change over time. However, this assumption is often violated in practice, as many real-world environments are dynamic and undergo changes.

To address such non-stationary environments, many adversarial versions of traditional reinforcement learning problems have been studied [Aue+95; Pin+17; RM20]. In these models, the environment is no longer a passive entity providing transitions and rewards; instead, it actively impedes the learner. This requires the learning algorithm to detect and adapt to changes in the environment to be robust, which is crucial for systems that operate over extended periods, where optimal actions may shift over time.

In adversarial problems, we often adopt a worst-case perspective, granting

the adversary unlimited power over the environment. However, this assumption can be overly pessimistic in practice and can hinder our understanding of the relationship between non-stationarity and learning efficiency. For this reason, it can be insightful to study the situation where the adversary has bounded influence [BGZ14; GM11]. The first question we then have to address is how to measure the adversary's influence on the environment, i.e., non-stationarity, and what constitutes a good (or even the right) measure of non-stationary complexity. In a next step, our goal becomes to design algorithms which adapt to the amount of non-stationarity they experience and thereby perform well under different degrees of non-stationary complexity. In particular, deriving performance guarantees with dependencies on the amount of non-stationarity, e.g., the number of times the environment changes, can help us to better understand the challenges of learning in adversarial environments.

The second challenge we address is robustness against misspecification. In real-world applications, there is often uncertainty in the dynamics of the environment and the model parameters. Bayesian methods provide a natural framework for modeling and updating uncertainty [Gha+15; Str00]. At the basis of the Bayesian RL framework is a subjective prior belief over the environments. However, it is not clear how such a prior can be selected from first principles if we have no domain knowledge, but still want to be robust. One idea is to assume that an adversary chooses a worst-case distribution over environments in response to the learning algorithm [Ber13]. This leads to a minimax-Bayes formulation of the reinforcement learning problem. We are then interested in the properties of this minimax-Bayes game and whether solving for minimax solutions can yield more robust policies.

Q3. (Strategic): *When learning in the presence of agents that are neither purely cooperative nor adversarial but instead act strategically so as to maximize their own benefit, how can we incentivize desirable agent behavior under uncertainty while simultaneously minimizing regret? What is the cost of mechanism design under uncertainty and what are the trade-offs between regret minimization and incentive design?*

In some cases the other agents do not pursue the same goals as the AI system, nor do they want to explicitly harm it. Instead, each agent may have their own objective (i.e., utility) which they try to maximize by strategizing in response to the AI system. For example, consider an e-commerce platform deploying an online recommendation system to suggest products to sequentially arriving customers. A strategic agent, such as an e-commerce retailer, may attempt to maximize their exposure and click-through rate by manipulating item descriptions or misreporting parameters to the platform.

By viewing such self-interested agents as purely adversarial, i.e., simply assuming worst-case behavior, we could try to achieve robustness. However, the result would be an extremely pessimistic system, which fails to utilize the control the AI system has over the interactions and the utility of the other agents. Could we make the agents behave in a desirable fashion by aligning agent incentives

Paper No.	Title	Research Question
Paper I	Interactive Inverse Reinforcement Learning for Cooperative Games [BGD22]	Q1. (Cooperative)
Paper II	Environment Design for Inverse Reinforcement Learning [BD22]	Q1. (Cooperative)
Paper III	An Improved Dynamic Regret Algorithm for Adaptive Non-Stationary Dueling Bandits [BS23]	Q2. (Adversarial)
Paper IV	Minimax-Bayes Reinforcement Learning [Bue+23b]	Q2. (Adversarial)
Paper V	Bandits Meet Mechanism Design to Combat Clickbait in Online Recommendation [Bue+23a]	Q3. (Strategic)

with our, e.g., the e-commerce platform's, goals?

In game theory this question is studied under the name of *mechanism design* [Mye89; Nis+07]. In mechanism design, the goal is to create incentives (through mechanisms) which—provided that agents act rationally—result in desirable game outcomes. The problem formulation typically involves a principal committing to a mechanism, e.g., an allocation rule, and several agents strategically responding to the committed mechanism. Here, a mechanism is said to be incentive-compatible if being truthful, i.e., sharing private information truthfully with the principal, is a dominant strategy for all agents. That is, being truthful is as least as good as any other strategy regardless of what other agents do. Such inverse game design is at the core of many real-world applications, including efficient market design, auction design, and network routing.

We quickly notice that the objectives of online regret minimization and mechanism design may clash when combining the two areas. While in the former our goal is to minimize regret by learning an optimal policy and playing optimal actions, the latter is primarily interested in incentivizing truthful agent behavior. In many cases, algorithmic actions that serve as incentives for the agents could be costly (i.e., suboptimal) for the algorithm. For example, to ensure truthful behavior across all agents, we may have to allocate customers to bad retailers as well. This could result in a trade-off between incentivizing all agents to be truthful and minimizing regret. Such dynamics could even be exacerbated when the environment and agent strategies are unknown to us in advance and must be learned through interaction. In this case, it is also not clear how to design mechanisms which learn over time and incentivize agents under environment- and strategy-uncertainty.

1.3 Summary of Papers

We here briefly summarize the papers and match them to the stated research questions. A more thorough discussion of the contributions of the papers can be found in Chapter 3.

Paper I studies the situation where the learner has to solve a task in collaboration with a human without access to the joint reward function. This is modeled by an episodic two-player Stackelberg game in which the learner commits to their policy first. We analyze how the learner should act in order to learn the joint reward function as quickly as possible and so that the joint policy is as close to optimal as possible.

Paper II formulates a framework of environment design for inverse reinforcement learning in which the learner can choose environments, i.e., transition dynamics, for the human expert to demonstrate the task in. We propose a minimax-regret objective to choose these environments and empirically show the benefits of learning from demonstrations in a diverse set of environments.

Paper III studies non-stationary dueling bandits and various notions of non-stationary complexity. We propose a schedule-based algorithm that achieves near-optimal regret w.r.t. the number of best arm switches adaptively, i.e., without prior knowledge of the non-stationary complexity.

Paper IV studies minimax-Bayes solutions in reinforcement learning. Here, the problem is viewed as a game between learning algorithm (i.e., policy) and nature which select a worst-case prior distribution. We analyze the properties of this game and show that minimax-Bayes policies can be more robust than those that assume a standard (e.g., uniform) prior.

Paper V proposes a strategic variant of the multi-armed bandit problem, called the strategic click-bandit. This model is motivated by applications in online recommendation where the choice of recommended items depends on both the click-through rates and the post-click rewards. Like in classical bandits, rewards follow a fixed unknown distribution. However, we assume that the click-through rate of each arm is chosen strategically by the arm in order to maximize the number of times it gets clicked. To solve this problem, we design an incentive-aware learning algorithm, which simultaneously incentivizes desirable arm strategies and minimizes regret.

References

[Aue+95] Auer, P. et al. “Gambling in a rigged casino: The adversarial multi-armed bandit problem”. In: *Proceedings of IEEE 36th annual foundations of computer science*. IEEE. 1995, pp. 322–331.

[BCN18] Brown, D. S., Cui, Y., and Niekum, S. “Risk-aware active inverse reinforcement learning”. In: *Conference on Robot Learning*. PMLR. 2018, pp. 362–372.

[BD22] Buening, T. K. and Dimitrakakis, C. “Environment Design for Inverse Reinforcement Learning”. In: *arXiv preprint arXiv:2210.14972* (2022).

[Ber13] Berger, J. O. *Statistical decision theory and Bayesian analysis*. Springer Science & Business Media, 2013.

[BGD22] Buening, T. K., George, A.-M., and Dimitrakakis, C. “Interactive Inverse Reinforcement Learning for Cooperative Games”. In: *International Conference on Machine Learning*. PMLR. 2022, pp. 2393–2413.

[BGZ14] Besbes, O., Gur, Y., and Zeevi, A. “Stochastic multi-armed-bandit problem with non-stationary rewards”. In: *Advances in neural information processing systems* vol. 27 (2014).

[BN19] Brown, D. S. and Niekum, S. “Machine teaching for inverse reinforcement learning: Algorithms and applications”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01. 2019, pp. 7749–7758.

[BS23] Buening, T. K. and Saha, A. “ANACONDA: An Improved Dynamic Regret Algorithm for Adaptive Non-Stationary Dueling Bandits”. In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2023, pp. 3854–3878.

[Bue+23a] Buening, T. K. et al. “Bandits Meet Mechanism Design to Combat Clickbait in Online Recommendation”. In: *arXiv preprint arXiv:2311.15647* (2023).

[Bue+23b] Buening, T. K. et al. “Minimax-Bayes Reinforcement Learning”. In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2023, pp. 7511–7527.

[CA16] Clark, J. and Amodei, D. “Faulty reward functions in the wild”. In: *Internet: https://blog. openai. com/faulty-reward-functions* (2016).

[CCS21] Cao, H., Cohen, S., and Szpruch, L. “Identifiability in inverse reinforcement learning”. In: *Advances in Neural Information Processing Systems* vol. 34 (2021), pp. 12362–12373.

[CL12] Cakmak, M. and Lopes, M. “Algorithmic and Human Teaching of Sequential Decision Tasks”. In: *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*. AAAI’12. Toronto, Ontario, Canada: AAAI Press, 2012, pp. 1536–1542.

[Gab20] Gabriel, I. “Artificial intelligence, values, and alignment”. In: *Minds and machines* vol. 30, no. 3 (2020), pp. 411–437.

[Gha+15] Ghavamzadeh, M. et al. “Bayesian reinforcement learning: A survey”. In: *Foundations and Trends® in Machine Learning* vol. 8, no. 5-6 (2015), pp. 359–483.

[GM11] Garivier, A. and Moulines, E. “On upper-confidence bound policies for switching bandit problems”. In: *International Conference on Algorithmic Learning Theory*. Springer. 2011, pp. 174–188.

[Had+16] Hadfield-Menell, D. et al. “Cooperative inverse reinforcement learning”. In: *Advances in neural information processing systems* vol. 29 (2016).

[Kad+23] Kaddour, J. et al. “Challenges and applications of large language models”. In: *arXiv preprint arXiv:2307.10169* (2023).

[Kim+21] Kim, K. et al. “Reward identification in inverse reinforcement learning”. In: *International Conference on Machine Learning*. PMLR. 2021, pp. 5496–5505.

[LKR22] Lindner, D., Krause, A., and Ramponi, G. “Active exploration for inverse reinforcement learning”. In: *Advances in Neural Information Processing Systems* vol. 35 (2022), pp. 5843–5853.

[LMM09] Lopes, M., Melo, F., and Montesano, L. “Active learning for reward estimation in inverse reinforcement learning”. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2009, pp. 31–46.

[Mye89] Myerson, R. B. *Mechanism design*. Springer, 1989.

[Nis+07] Nisan, N. et al. *Algorithmic Game Theory*. Cambridge University Press, 2007.

[NR+00] Ng, A. Y., Russell, S., et al. “Algorithms for inverse reinforcement learning.” In: *Icml*. Vol. 1. 2000, p. 2.

[Pin+17] Pinto, L. et al. “Robust adversarial reinforcement learning”. In: *International Conference on Machine Learning*. PMLR. 2017, pp. 2817–2826.

[RM20] Rosenberg, A. and Mansour, Y. “Stochastic shortest path with adversarially changing costs”. In: *arXiv preprint arXiv:2006.11561* (2020).

[Rus21] Russell, S. “Human-compatible artificial intelligence”. In: *Human-like machine intelligence* (2021), pp. 3–23.

[Rus98] Russell, S. “Learning agents for uncertain environments”. In: *Proceedings of the eleventh annual conference on Computational learning theory*. 1998, pp. 101–103.

[Ska+22] Skalse, J. et al. “Defining and characterizing reward gaming”. In: *Advances in Neural Information Processing Systems* vol. 35 (2022), pp. 9460–9471.

[Str00] Strens, M. “A Bayesian framework for reinforcement learning”. In: *ICML*. Vol. 2000. 2000, pp. 943–950.

[Tsc+19] Tschiatschek, S. et al. “Learner-aware teaching: Inverse reinforcement learning with preferences and constraints”. In: *Advances in neural information processing systems* vol. 32 (2019).

[Zen+22] Zeng, S. et al. “Maximum-likelihood inverse reinforcement learning with finite-time guarantees”. In: *Advances in Neural Information Processing Systems* vol. 35 (2022), pp. 10122–10135.

[Zie+19] Ziegler, D. M. et al. “Fine-tuning language models from human preferences”. In: *arXiv preprint arXiv:1909.08593* (2019).

Chapter 2

The Reinforcement Learning Framework

We now give a brief introduction to the reinforcement learning framework, ranging from multi-armed bandits to Markov games. In short, reinforcement learning concerns learning how to act in an unknown environment from interaction so as to maximize rewards. When formalizing the various models that fall under the umbrella of reinforcement learning, the differences mainly boil down to different interpretations of the *environment*, the *actions*, and the type of *reward signal*. However, the common ground is that in all of these settings a *learning agent*, also called the *learning algorithm* or simply the *learner*, sequentially takes actions, upon which some feedback is observed.

Throughout this chapter we try to be as concise as possible and restrict our attention to the classical problem setups only. For further details we refer to the several great textbooks in this field, including [DO18; Put90; SB+98] as well as [LS20; Sli+19] who specifically discuss the bandit problem.

2.1 Bandits

Multi-Armed Bandits (MABs) [Tho33] have been extensively studied in the past several decades, both due to their practical applications, such as clinical trials or recommendation systems, but also because the multi-armed bandit problem is perhaps the simplest instance of the exploration-exploitation dilemma. As a result, many fundamental algorithmic and technical tools for online regret minimization such as the optimism in the face of uncertainty principle [ACF02; LR85], were first developed for bandits and then later extended to richer reinforcement learning models like the Markov decision process [AJO08, e.g.].

There is an abundance of extensions and variants to the classical multi-armed bandit problem, which are far too many to cover here. In the following, we therefore only introduce the standard stochastic MAB problem as well as the case of stochastic preference-based feedback, also known as the dueling bandit.

2.1.1 Stochastic Multi-Armed Bandits

The stochastic MAB consists of $K \in \mathbb{N}$ so-called arms with each arm i being associated with a reward distribution P_i. In MABs, the learner and environment interact over the course of $T \in \mathbb{N}$ rounds. In each round $t \in [T]$, the learner selects an arm $i_t \in [K]$ and receives a numerical reward $r_{t,i_t} \in \mathbb{R}$ independently drawn from arm i_t's reward distribution P_{i_t}. It is assumed that the learner has

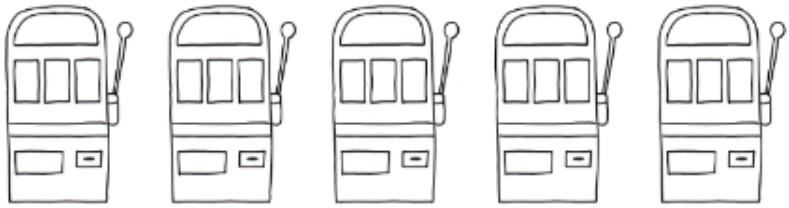


Figure 2.1: Multiple one-armed bandit slot machines next to one another. Suppose that each slot machine follows some unknown payoff distribution and we want to maximize our cumulative payoff over T rounds [Pik23].

no prior knowledge of the reward distributions and acts on the basis of past reward observations only (except for knowledge of K and possibly T as well).

A typical choice for the family of reward distributions is the Bernoulli distribution, which can be a natural choice for applications in online platforms which aim to maximize click-rates. More generally, the standard assumption is that of sub-Gaussian reward distributions, which includes all distributions with bounded support and which ensures that the rewards concentrate around their mean at a sufficiently fast rate. Without such restrictions on the reward distributions, more sophisticated concentration bounds and approaches are required (see, e.g., heavy-tailed bandits [AJK21; BCL13; Yu+18]).

Regret. The goal of the learner is to maximize the sum of rewards collected over the course of all T rounds, given by $\sum_{t=1}^{T} r_{t,i_t}$. This quantity is random and we are usually happy with just maximizing the expected return $\sum_{t=1}^{T} \mu_{i_t} = \mathbb{E}[\sum_{t=1}^{T} r_{t,i_t}]$, where $\mu_i := \mathbb{E}[r_{t,i}]$ denotes the mean of arm i's reward distribution P_i. To evaluate the learner, we then compare the learner's expected return against that of the optimal policy, defined as the policy which picks the arm with largest mean reward $\mu^* := \max_{i \in [K]} \mu_i$ every round. This quantity is called the *regret* of the learner, formally defined as

$$R_T = T \cdot \mu^* - \sum_{t=1}^{T} \mu_{i_t}.$$

Note that the learner's decisions i_t may be random, since it can depend on the randomness of observed rewards or a deliberate randomization of the learner's selection rule. Again, we are usually happy to minimize the *expected regret* $\mathbb{E}[R_T]$ and not care too much about the specific realization of R_T as we often derive high-probability bounds on R_T.

2.1.2 Dueling Bandits

The stochastic MAB framework has been generalized to different settings, among which a popular variant is known as the dueling bandit [Yu+18]. Dueling bandits

are a preference-based version of MABs, where at every round t, the learner cannot directly observe the random rewards of an arm, but can only indirectly compare two arms. This is frequently used to model human preferences [Ben+21; Sui+18], where you ask a person to compare two items, without asking them to give an absolute evaluation for each one. The randomness in that case can be due to the random sampling of individuals or individually stochastic responses.

For that reason, instead of reward distributions, it is simpler (and more general) to define the K-armed (stochastic) dueling bandit through a preference matrix $\mathbf{P} = [p_{i,j}]_{i,j\in[K]} \in [0,1]^K$ satisfying $p_{i,j} = 1 - p_{j,i}$. The probability $p_{i,j}$ is interpreted as the probability that arm i is winning in a duel against arm j. If $p_{i,j} > 0.5$, we then say that arm i is preferred over arm j and write $i \succ j$ to express this relation. The interaction proceeds again in rounds. Each round $t \in [T]$, the learner selects two arms $i_t, j_t \in [K]$ upon which the winner of the duel is observed, where i_t wins the duel against j_t with probability p_{i_t,j_t}.

In stochastic MABs, it is well-known that the optimality gap $\Delta_i = \mu^* - \mu_i$ can be used to characterize the learning complexity of a given MAB problem. The reason for this is that the gap Δ_i governs the number of samples required to distinguish arm i's mean reward μ_i from the maximal mean μ^*. In dueling bandits, we observe that the closer $p_{i,j}$ is to 0.5, the more difficult it becomes to distinguish arm i and j. Hence, a reasonable notion of *gap* in dueling bandits, which is sometimes also called the *preference strength*, is

$$\delta_{i,j} := p_{i,j} - 0.5.$$

Solution Concepts. It is not immediately clear how to measure the performance of a learning algorithm in dueling bandits. An intuitive choice for a *benchmark* is to compare the algorithm's actions against the arm that is preferred over any other arm, the so-called *Condorcet winner*, defined as $i^* \in [K]$ such that $p_{i^*,i} > 0.5$ for all $i \in [K] \setminus \{i^*\}$. With the Condorcet winner as a benchmark the learner's regret is then defined as

$$R_T = \sum_{t=1}^{T} \frac{\delta_{i^*,i_t} + \delta_{i^*,j_t}}{2}.$$

However, notice that the Condorcet winner may not always exist, namely, if for all $i \in [K]$ there exists $j \neq i$ with $j \succ i$. For this reason, several other solution concepts, where a "best arm" always exists, have been studied in the dueling bandit literature.

The *Copeland winner* is the arm that is preferred over the most other arms [Zog+15]. More precisely, let $c_i = \frac{1}{K-1}\#\{j \in [K] \colon p_{i,j} > 0.5\}$ denote the normalized Copeland score of arm i. The Copeland winner is then given by $i^* \in \operatorname{argmax}_{i\in[K]} c_i$ and regret defined as

$$R_T = \sum_{t=1}^{T} (2c_{i^*} - c_{i_t} - c_{j_t}).$$

The *Borda winner* follows a similar idea and is the arm that has the highest winning probability against a uniformly selected opponent arm [BSH14; Bus+13]. Formally, letting $b_i = \frac{1}{K-1}\sum_{j\neq i} p_{i,j}$, we define the Borda winner as $i^* \in \operatorname{argmax}_{i\in[K]} b_i$ and define regret as

$$R_T = \sum_{t=1}^{T}(2b_{i^*} - b_{i_t} - b_{j_t}).$$

Arguably the most elegant definition of a benchmark in dueling bandits is that of the *von Neumann winner* [Dud+15]. This notion of benchmark is particularly intriguing as it naturally brings out the close relationship between dueling bandits and normal-form zero-sum games. Given a preference matrix $\mathbf{P}$ we can define a zero-sum game matrix as $\mathbf{Q} = 2\mathbf{P} - 1$ (i.e., $q_{i,j} = 2\delta_{i,j}$). The interpretation of $\mathbf{Q}$ is as follows. Let the outcome of a duel (i, j) be $+1$ when arm i wins and -1 if j wins. Then, the entry $q_{i,j}$ denotes the *expected outcome* of the duel (i, j). We assume that a duel (i, j) is equivalent to the negation of the duel (j, i) as well as $q_{ii} = 0$, so that $\mathbf{P}$ is skew-symmetric, i.e. $\mathbf{P}^\top = -\mathbf{P}$. Here, von Neumann's minimax theorem ensures the existence of a maximin strategy $\mathbf{w} \in \Delta(K)$, that is, $\mathbf{w}^\top\mathbf{Q}\mathbf{u} \geq 0$ for all $\mathbf{u} \in \Delta(K)$. This maximin strategy $\mathbf{w}$ is called the *von Neumann winner* and is a direct generalization of the Condorcet winner in the sense that if there exists a Condorcet winner i^*, then the pure strategy $\mathbf{w} = \text{Dirac}(i^*)$ is maximin. The regret is then defined similarly to the Condorcet winner regret as

$$R_T = \max_{k\in[K]} \sum_{t=1}^{T} \frac{\delta_{k,i_t} + \delta_{k,j_t}}{2}.$$

Additional Assumptions on the Preference Model. In dueling bandits, there are certain preference matrices that are particularly hard to deal with such as preference matrices with cyclic preferences where $i \succ j$ and $j \succ k$, but $k \succ i$. Such preferences can be difficult to learn efficiently since the estimation of $p_{i,j}$ and $p_{j,k}$ does not necessarily yield any useful information about $p_{i,k}$. However, in practice it is often reasonable to assume some additional properties of the preference model. For instance, if a user prefers action movies over comedies and comedies over thrillers, then action movies should also be preferred over thrillers. For this reason, dueling bandits are often studied under two additional assumptions on the preference model: Strong Stochastic Transitivity (SST) and the Stochastic Triangle Inequality (STI).

(SST) If $i \succ j \succ k$, then $\delta_{i,k} \geq \max\{\delta_{i,j}, \delta_{j,k}\}$.

(STI) If $i \succ j \succ k$, then $\delta_{i,k} \leq \delta_{i,j} + \delta_{j,k}$.

SST and STI imply some useful properties of the preference matrix, which sometimes allow us to approach the dueling bandit problem similarly to MABs. Most notably, for $i \succ j$ and $i \succ k$, we obtain $\delta_{i,j} \leq 2\delta_{i,k} + \delta_{k,j}$, which can be useful to decompose the regret (see, e.g., Paper III).

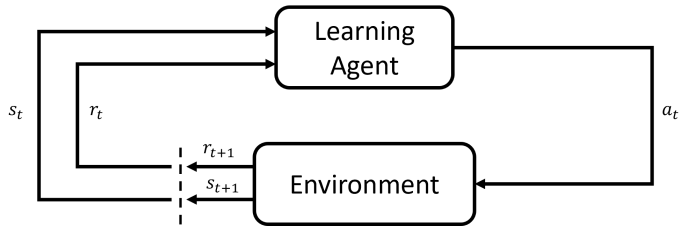


Figure 2.2: The interaction protocol in a Markov decision process.

2.2 Markov Decision Processes

We now introduce the Markov Decision Process (MDP), which extends the reinforcement learning problem to the case where actions have to be taken in different situations, i.e., states of the environment.

In an MDP, every round $t \in [T]$, the learner observes the current state of the environment $s_t \in S$ and chooses an action $a_t \in A$, where S denotes the state space and A the action space of the MDP. In the next time step, the environment transitions to state $s_{t+1} \in S$ and the learner receives a reward $r_{t+1} \in \mathbb{R}$. Here, the transition to the next state s_{t+1} is assumed to depend only on the prior state s_t and action a_t and is modeled by a transition function $\mathcal{P}(\cdot \mid s, a) := \mathbb{P}(\cdot \mid s_t = s, a_t = a)$, mapping state-action pairs to a distribution over next states. The reward signal r_{t+1} also depends on the state-action pair (s_t, a_t) and is assumed to be sampled from a reward distribution $\rho(\cdot \mid s, a) := \mathbb{P}(\cdot \mid s_t = s, a_t = a)$. However, often it is enough and more convenient to consider the expected reward function $R\colon S \times A \to \mathbb{R}$, defined by $R(s, a) = \mathbb{E}_\rho[r_{t+1} \mid s_t = s, a_t = a]$. In MDPs, we define the utility of the learner as

$$U = \sum_{t=1}^{T} \gamma^{t-1} r_t,$$

where $\gamma \in (0, 1]$ is a *discount factor*. We then also define the utility after round ℓ as $U_\ell = \sum_{t=1}^{T-\ell} \gamma^{t-1} r_{\ell+t}$.

Broadly speaking, there are two types of situations that require slightly different modeling assumptions. The first is that of episodic tasks, which consist of a series of episodes with each episode comprising a finite number of rounds. In this case, the utility is often undiscounted, i.e., $\gamma = 1$. The second situation is that of an indefinitely continuing tasks, i.e., infinite horizon $T = \infty$. In this case, the objective of maximizing U can become ill-posed and the discount factor serves the practical purpose of ensuring that the utility is finite for $\gamma < 1$. In the remainder, to save us some notation, we will consider infinite horizon discounted MDPs, i.e., $T = \infty$ and $\gamma < 1$.

Policies and Value Functions. A history-dependent policy is defined as a distribution over actions given past observations and the current state, denoted

$\pi(a_t \mid s_t, \dots, s_1, a_{t-1}, \dots, a_1)$. Often it is enough to restrict our attention to memoryless policies for which $\pi(a_t \mid s_t) = \pi(a_t \mid s_t, \dots, s_1, a_{t-1}, \dots, a_1)$.

Under a given policy π, we then define the *value* of state s as $V_\pi(s) = \mathbb{E}_\pi[U_t \mid s_t = s]$. Similarly, we can define the value of a state-action pair under policy π as $Q_\pi(s, a) = \mathbb{E}_\pi[U_t \mid s_t = s, a_t = a]$, which is often called the Q-value. The optimal policy π^* in a given MDP is then defined as the policy that maximizes the value function in every state, i.e., π^* satisfies $V_{\pi^*}(s) \geq V_\pi(s)$ for all $s \in S$ and policies π.

2.2.1 Markov Games

Markov games, which are sometimes also called multi-agent MDPs or stochastic games, extend the MDP formulation to the situation where multiple agents simultaneously act in the same environment. An n-player Markov game consists of n possibly different action spaces $A_1, \dots, A_n$ and reward distributions $\rho_1, \dots, \rho_n$. In each round $t \in [T]$, every agent $i \in [n]$ takes an action $a_{t,i} \in A_i$ upon which the next state is sampled from the transition function $\mathcal{P}(\cdot \mid s_t, a_{t,1}, \dots, a_{t,n})$, which depends on all agent actions. Each agent i then receives a reward $r_{t,i}$ from their reward distribution $\rho_i(\cdot \mid s_t, a_{t,1}, \dots, a_{t,n})$, and attempts to maximize their utility $U^i = \sum_{t=1}^{T} \gamma^{t-1} r_{t,i}$, where γ is some discount factor.

In Markov games the dynamics between agents when the agents' objectives are cooperative, competitive, or mixed-motive are particularly interesting. In the fully cooperative case, all agents have a joint reward function so that $r_{t,1} = \dots = r_{t,n}$. In a two-player Markov game, an interesting special case is that of fully competitive objectives, i.e., the game becomes zero-sum so that $r_{t,1} = -r_{t,2}$ always.

Another important aspect of learning and interacting in Markov games is the knowledge each agent possesses and is able to share with other agents. This can range from a fully centralized setting, where all agents make joint observations, to a fully decentralized setting with private observations and no communication between agents.

References

[ACF02] Auer, P., Cesa-Bianchi, N., and Fischer, P. "Finite-time analysis of the multiarmed bandit problem". In: *Machine learning* vol. 47 (2002), pp. 235–256.

[AJK21] Agrawal, S., Juneja, S. K., and Koolen, W. M. "Regret minimization in heavy-tailed bandits". In: *Conference on Learning Theory*. PMLR. 2021, pp. 26–62.

[AJO08] Auer, P., Jaksch, T., and Ortner, R. "Near-optimal regret bounds for reinforcement learning". In: *Advances in neural information processing systems* vol. 21 (2008).

[BCL13] Bubeck, S., Cesa-Bianchi, N., and Lugosi, G. "Bandits with heavy tail". In: *IEEE Transactions on Information Theory* vol. 59, no. 11 (2013), pp. 7711–7717.

[Ben+21] Bengs, V. et al. "Preference-based online learning with dueling bandits: A survey". In: *The Journal of Machine Learning Research* vol. 22, no. 1 (2021), pp. 278–385.

[BSH14] Busa-Fekete, R., Szörenyi, B., and Hüllermeier, E. "PAC rank elicitation through adaptive sampling of stochastic pairwise preferences". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 28. 1. 2014.

[Bus+13] Busa-Fekete, R. et al. "Top-k selection based on adaptive sampling of noisy preferences". In: *International Conference on Machine Learning*. PMLR. 2013, pp. 1094–1102.

[DO18] Dimitrakakis, C. and Ortner, R. "Decision making under uncertainty and reinforcement learning". In: *Book available at http://www. cse. chalmers. se* (2018).

[Dud+15] Dudik, M. et al. "Contextual dueling bandits". In: *Conference on Learning Theory*. PMLR. 2015, pp. 563–587.

[LR85] Lai, T. L. and Robbins, H. "Asymptotically efficient adaptive allocation rules". In: *Advances in applied mathematics* vol. 6, no. 1 (1985), pp. 4–22.

[LS20] Lattimore, T. and Szepesvári, C. *Bandit algorithms*. Cambridge University Press, 2020.

[Pik23] Pike-Burke, C. *Multi-Armed Bandits*. online. 2023.

[Put90] Puterman, M. L. "Markov decision processes". In: *Handbooks in operations research and management science* vol. 2 (1990), pp. 331–434.

[SB+98] Sutton, R. S., Barto, A. G., et al. *Introduction to reinforcement learning*. Vol. 135. MIT press Cambridge, 1998.

[Sli+19] Slivkins, A. et al. "Introduction to multi-armed bandits". In: *Foundations and Trends® in Machine Learning* vol. 12, no. 1-2 (2019), pp. 1–286.

[Sui+18] Sui, Y. et al. "Advancements in Dueling Bandits." In: *IJCAI*. 2018, pp. 5502–5510.

[Tho33] Thompson, W. R. "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples". In: *Biometrika* vol. 25, no. 3-4 (1933), pp. 285–294.

[Yu+18] Yu, X. et al. "Pure Exploration of Multi-Armed Bandits with Heavy-Tailed Payoffs." In: *UAI*. 2018, pp. 937–946.

[Zog+15] Zoghi, M. et al. "Copeland dueling bandits". In: *Advances in neural information processing systems* vol. 28 (2015).

Chapter 3

Main Findings and Conclusions

In this chapter, we outline the main findings of this thesis. We organize these along our research questions from Section 1.2 and the included five papers. Finally, future research directions are discussed in Section 3.1.

Towards Research Question Q1. (Cooperative): *Can we learn rewards more precisely and efficiently by actively seeking information from a human partner through repeated interaction?*

In Paper I, we address this question first from a human-AI collaboration perspective and consider the problem of repeatedly assisting a human partner in a two-player Markov game *without knowledge or observations* of the *joint* reward function. We quickly notice that in order to infer the joint reward function from interactions with the human, we must be able to contextualize observed human behavior. To this end, when the interaction between learner and human is sequential, it is natural to assume that the human will adapt their behavior to the learner's policy. By equipping the human with a behavioral model, such as Boltzmann-rationality, and a belief over the learner's policy in the next episode, this can be made concrete and can allow us to properly interpret any observed behavior by viewing the human's actions as an (approximately) best response to the learner's policy.

We distill this model further in Paper I by studying the *Stackelberg* formulation of the Markov game in which at the beginning of every episode the learner publicly commits to a policy before the human does. With knowledge of the learner's policy, the human can be viewed as planning and acting in the *marginalized* single-agent MDP, which is obtained from the two-player Markov game by marginalizing over the learner's policy. Consequently, for the purpose of reward inference, the learner can actively seek human demonstrations in various scenarios by committing to a specific policy which in turn implies a marginalized single-agent MDP for the human to act in.

Based on this observation, Paper I shows that not only is it possible to learn about the joint reward function when simultaneously acting in a two-player Markov game, but by observing the human respond to various policies we can learn a better representation of the reward function with fewer observations compared to standard IRL settings. In fact, while other work [CCS21; Kim+21, e.g.] shows that in the standard IRL formulation recovering the true reward function is generally impossible, we show under mild assumptions that through repeated interaction we can provably learn a reward function that yields an optimal policy for the learner when optimized.

The perspective of actively seeking demonstrations in specific scenarios, i.e., MDPs, also leads us to questions about what scenarios carry the most useful

information about the reward function. In Paper II, we expand on this question and study how to design environments for the human expert to act in so as to learn their reward function more efficiently and precisely. Our experimental results suggest that a minimax regret-based objective yields useful demonstration environments which help us to infer nearly all performance-relevant aspects of the reward function. More generally, the results of Paper II illustrate the benefits of active IRL through environment design, and we observe that the reward functions learned through the interactive environment design process are more robust against variations in the transition dynamics and transfer better to new environments.

Towards Research Question Q2. (Adversarial): *What if the environment is chosen adversarially and changes over time? How does this impact our ability to learn efficiently?*

In Paper III, we study non-stationary dueling bandits, where the underlying preference matrix undergoes adversarial changes over time. Prior work [KBH22; SG22] studied the number of changes in the preference matrix as a measure of non-stationary complexity. While the number of such preference switches indeed relates to the hardness of the problem, it is without doubt a pessimistic measure of non-stationarity. For example, a change in the preference between two widely suboptimal arms or a minor change in the preference matrix under which the optimal arm remains optimal should not significantly impact our ability to achieve low regret.

To this end, we propose three new notions of non-stationary complexity for dueling bandits: (1) the number of Condorcet winner switches, (2) the total variation in the sequence of Condorcet winners, and (3) the number of "significant" Condorcet winner switches. The novelty of our proposed non-stationarity measures lies in capturing only the non-stationarity observed for the 'best arms' of the preference sequences. They remain unaffected by any changes in the suboptimal arms, which makes them less pessimistic. We then design a learning algorithm which achieves near-optimal dynamic regret w.r.t. the number of Condorcet winner switches *without* prior knowledge of the number of switches, i.e., adaptively. Under additional assumptions on the preference model, we then also derive sublinear dynamic regret bounds w.r.t. Condorcet winner variation and significant Condorcet winner switches.

Interestingly, our results suggested that the effect of non-stationarity in dueling bandits is more severe than in classical multi-armed bandits. While it is always possible to achieve sublinear dynamic regret w.r.t. *significant* switches in multi-armed bandits [SK22], to do so for dueling bandits we had to impose additional transitivity properties on the preference model, namely, strong stochastic transitivity and the stochastic triangle inequality (see Section 2.1.2). In fact, [SA23b] recently showed that it is generally impossible to achieve sublinear dynamic regret w.r.t. significant Condorcet winner switches if the preference matrices do not satisfy the aforementioned transitivity assumptions.

In Paper IV, we study worst-case prior distributions in Bayesian reinforcement

learning, which is modeled as a minimax game between the learning algorithm, i.e., policy, and an adversarial nature that selects a worst-case prior distribution over problem instances, i.e., MDPs. We show that while the minimax-Bayes game in terms of utility can be degenerate with vacuous solutions, the game where the learning algorithm minimizes, and the adversary maximizes, regret is well-defined and minimax theorems hold. We find that minimax-Bayes policies not only appear to be feasible, but also that such policies can be significantly more robust than those based on standard uninformative priors.

Towards Research Question Q3. (Strategic): *When learning in the presence of agents that are neither purely cooperative nor adversarial but instead act strategically so as to maximize their own benefit , how can we incentivize desirable agent behavior under uncertainty while simultaneously minimizing regret? What is the cost of mechanism design under uncertainty and what are the trade-offs between regret minimization and incentive design?*

In Paper V, we introduce the strategic click-bandit problem in which each arm is associated with a click-rate, chosen strategically by the arms, and an immutable post-click reward. The algorithm designer does not know the post-click rewards nor the arms' actions (i.e., strategically chosen click-rates) in advance, and must learn both values over time. To model the arms' strategic behavior, we assume that the arms respond in Nash equilibrium to the learning algorithm, that is, they choose strategies so as to maximize their total number of clicks given the algorithm and the environment.

We show that designing the right incentives for the arms by means of an incentive-aware selection policy is necessary to achieve low regret in the strategic click-bandit. In particular, incentive-agnostic algorithms, i.e., those that do not account for the arms' strategic behavior, imply undesirable equilibria among arms, which results in linear regret. To address this challenge, we then design an incentive-aware online learning algorithm, called UCB-S, that combines a UCB-type selection policy with an additional screening rule. The idea behind the screening rule is to sanction arms with elimination if they are deviating from the desired strategies. Due to arm strategies and reward distributions being unknown in advance, we use confidence bounds to ensure that any elimination is justified and credible.

Due to the learner's uncertainty about strategies and reward distributions, the mechanism design of UCB-S is *approximate* and leaves room for arms to exploit the learner's uncertainty. This leads to an interesting regret bound which makes the intuition precise that arms can exploit the learner's uncertainty about their strategies. More precisely, we observe that the cost of incentive design and the strategic behavior of the arms is of order $\sqrt{KT}$, which primarily stems from "optimal" arms deviating by roughly $\sqrt{K/T}$ from the desired equilibrium.

Hereto related, we find that the selection policy directly impacts the truthfulness of the arms, since more frequently selected arms are forced to choose strategies closer to the desired equilibrium because our estimate of their strategies is better. This results in a trade-off between incentivizing *all* arms to

be truthful by selecting the arms almost uniformly at random and minimizing regret by selecting only the best arms. More generally, our results suggest that under strategy-uncertainty, i.e., when strategies are not directly observable but must be learned over time, precisely incentivizing desirable equilibria is generally impossible, but, instead, the mechanism design has to be approximate. Moreover, the "approximation error" and, by extension, the cost of strategy-uncertainty usually depends on the observational model. For example, in the strategic click-bandit, the estimates of the arms' strategies concentrated at a similar rate as the estimates of the post-click rewards, so that the cost of strategy-uncertainty roughly matched the MAB learning complexity.

3.1 Future Directions

There are many future directions and open questions related to learning in the presence of either cooperative, adversarial, or strategic agents. Here, I will highlight two specific research directions that have received limited attention in the literature so far.

Modeling Human Choices for Human-AI Alignment. Understanding human choices is fundamental for building intelligent systems that can interact with users effectively, align with their preferences, and contribute to the development of ethical and user-centric AI applications. Recently, this challenge was once again prominently highlighted by the development and dissemination of large language models such as ChatGPT, which are fine-tuned using human evaluations [Ouy+22]. However, the human choice models used in both research and practice are often overly simplistic and can fail to capture actual human decision-making. For example, the literature on learning from human demonstrations, including this thesis' work on IRL, almost exclusively assumes humans to act rationally as modeled by a Boltzmann distribution [JMD20], or, even worse, assumes humans to act optimally w.r.t. some reward model [NR+00]. Moreover, several other assumptions about human decision-making remain mostly unchallenged in the human-AI alignment literature [Ji+23; LE22]. For example, current choice models assume that human decisions are unbiased, and do not take into account systematic or cognitive biases [Sha+19]. Moreover, it is typically assumed that human behavior is static in the sense that human choices do not depend on past interactions or accumulated knowledge.

Clearly, most of these assumptions are violated in practice. Despite this, so far, there have been barely any attempts at incorporating richer human choice models into human-AI alignment, let alone studies of the risks of overly restrictive modeling assumptions [Cas+23; Ji+23]. To this end, to ensure effective human-AI alignment and collaboration, it is important to reevaluate current choice models and assumptions, for which we may want to derive insights from other research areas, such as behavioral psychology, as well. There are also obvious technical challenges arising from deploying more realistic human models, and we can expect that designing tractable algorithms for richer human models will

be more difficult. In practice, this could lead to a trade-off between choice model complexity and tractability of the learning algorithms. Moreover, based on the premise that mathematical choice models cannot accurately model real-world human decision-making, it is also important to study the effect of human model misspecification on the performance, robustness, and safety of AI systems, e.g., those that are trained using human evaluations, or those that use reward functions learned through IRL [FSD21; HBD22; SA23a].

Online Learning and Mechanism Design. Online learning and algorithmic mechanism design, two in themselves widely popular areas of research, have been mostly studied as separate streams. However, incorporating mechanism design into the algorithm design, that is, taking into account the incentives created by an algorithm, holds the promise of more efficient and robust systems that discourage harmful behavior and encourage collaboration among multiple self-interested agents.

One interesting direction is to study problems at the intersection of reinforcement learning and mechanism design. So far, this has only been studied in a few specific cases such as auction design (e.g., pay-per-click auctions) [BKS15; BSS09] and certain strategic-variants of multi-armed bandits [Bra+19; FPX20] including our Paper V. However, a clear and general understanding of incentive-aware regret minimization and the trade-offs between incentive design and regret is still missing from the literature. There have also been some attempts at solving mixed-motive Markov games with the help of incentive design, e.g., by equipping each agent with the ability to share rewards with other agents [Wil+23; Yan+20]. While these works do not necessarily view this as a mechanism design problem, it could be treated as a problem of decentralized mechanism design, where each agent has the ability to influence the game outcome by designing incentives for the other agents.

There has also been a number of works studying strategic agent behavior in classification, where individuals strategically manipulate their attributes in response to a classifier so as to obtain a desired classification outcome [Don+18; Har+16]. While these works take the strategic responses of individuals into account, they do not design incentives, but instead treat the problem similar to a robust classification task. As a result, to make these problems tractable, it is necessary to assume that agents are limited by either a budget or suffer a cost for manipulating their attributes. However, when the classification task is online and we encounter the same agents repeatedly, we may be able to design incentives so as to prevent harmful manipulation and unwanted strategic behavior at its root. Thus, another interesting direction for future work could be to introduce active incentive design into online learning problems, apart from reinforcement learning, such as repeated classification or algorithmic recourse.

References

[BKS15] Babaioff, M., Kleinberg, R. D., and Slivkins, A. "Truthful mechanisms with implicit payment computation". In: *Journal of the ACM (JACM)* vol. 62, no. 2 (2015), pp. 1–37.

[Bra+19] Braverman, M. et al. "Multi-armed bandit problems with strategic arms". In: *Conference on Learning Theory*. PMLR. 2019, pp. 383–416.

[BSS09] Babaioff, M., Sharma, Y., and Slivkins, A. "Characterizing truthful multi-armed bandit mechanisms". In: *Proceedings of the 10th ACM conference on Electronic commerce*. 2009, pp. 79–88.

[Cas+23] Casper, S. et al. "Open problems and fundamental limitations of reinforcement learning from human feedback". In: *arXiv preprint arXiv:2307.15217* (2023).

[CCS21] Cao, H., Cohen, S., and Szpruch, L. "Identifiability in inverse reinforcement learning". In: *Advances in Neural Information Processing Systems* vol. 34 (2021), pp. 12362–12373.

[Don+18] Dong, J. et al. "Strategic classification from revealed preferences". In: *Proceedings of the 2018 ACM Conference on Economics and Computation*. 2018, pp. 55–70.

[FPX20] Feng, Z., Parkes, D., and Xu, H. "The intrinsic robustness of stochastic bandits to strategic manipulation". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 3092–3101.

[FSD21] Freedman, R., Shah, R., and Dragan, A. "Choice set misspecification in reward inference". In: *arXiv preprint arXiv:2101.07691* (2021).

[Har+16] Hardt, M. et al. "Strategic classification". In: *Proceedings of the 2016 ACM conference on innovations in theoretical computer science*. 2016, pp. 111–122.

[HBD22] Hong, J., Bhatia, K., and Dragan, A. "On the Sensitivity of Reward Inference to Misspecified Human Models". In: *arXiv preprint arXiv:2212.04717* (2022).

[Ji+23] Ji, J. et al. "Ai alignment: A comprehensive survey". In: *arXiv preprint arXiv:2310.19852* (2023).

[JMD20] Jeon, H. J., Milli, S., and Dragan, A. "Reward-rational (implicit) choice: A unifying formalism for reward learning". In: *Advances in Neural Information Processing Systems* vol. 33 (2020), pp. 4415–4426.

[KBH22] Kolpaczki, P., Bengs, V., and Hüllermeier, E. "Non-stationary dueling bandits". In: *arXiv preprint arXiv:2202.00935* (2022).

[Kim+21] Kim, K. et al. "Reward identification in inverse reinforcement learning". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 5496–5505.

[LE22] Lindner, D. and El-Assady, M. "Humans are not Boltzmann Distributions: Challenges and Opportunities for Modelling Human Feedback and Interaction in Reinforcement Learning". In: *arXiv preprint arXiv:2206.13316* (2022).

[NR+00] Ng, A. Y., Russell, S., et al. "Algorithms for inverse reinforcement learning." In: *Icml*. Vol. 1. 2000, p. 2.

[Ouy+22] Ouyang, L. et al. "Training language models to follow instructions with human feedback". In: *Advances in Neural Information Processing Systems* vol. 35 (2022), pp. 27730–27744.

[SA23a] Skalse, J. and Abate, A. "Misspecification in inverse reinforcement learning". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37. 12. 2023, pp. 15136–15143.

[SA23b] Suk, J. and Agarwal, A. "When Can We Track Significant Preference Shifts in Dueling Bandits?" In: *arXiv preprint arXiv:2302.06595* (2023).

[SG22] Saha, A. and Gupta, S. "Optimal and efficient dynamic regret algorithms for non-stationary dueling bandits". In: *International Conference on Machine Learning*. PMLR. 2022, pp. 19027–19049.

[Sha+19] Shah, R. et al. "On the feasibility of learning, rather than assuming, human biases for reward inference". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 5670–5679.

[SK22] Suk, J. and Kpotufe, S. "Tracking most significant arm switches in bandits". In: *Conference on Learning Theory*. PMLR. 2022, pp. 2160–2182.

[Wil+23] Willis, R. et al. "Resolving social dilemmas with minimal reward transfer". In: *arXiv preprint arXiv:2310.12928* (2023).

[Yan+20] Yang, J. et al. "Learning to incentivize other learning agents". In: *Advances in Neural Information Processing Systems* vol. 33 (2020), pp. 15208–15219.

Papers

Remarks

There have been no modifications made to contents of the published versions of the papers, except for minor attempts at unifying the notation in Paper I and II.

Paper I

Interactive Inverse Reinforcement Learning for Cooperative Games

Thomas Kleine Buening, Anne-Marie George, Christos Dimitrakakis

Published in *39th International Conference on Machine Learning (ICML)*, 2022.

Abstract

We study the problem of designing autonomous agents that can learn to cooperate effectively with a potentially suboptimal partner while having no access to the joint reward function. This problem is modeled as a cooperative episodic two-agent Markov decision process. We assume control over only the first of the two agents in a Stackelberg formulation of the game, where the second agent is acting so as to maximise expected utility given the first agent's policy. How should the first agent act in order to learn the joint reward function as quickly as possible and so that the joint policy is as close to optimal as possible? We analyse how knowledge about the reward function can be gained in this interactive two-agent scenario. We show that when the learning agent's policies have a significant effect on the transition function, the reward function can be learned efficiently.

I.1 Introduction

Recent applications of autonomous systems in our daily lives show that autonomous agents are no longer deployed in isolation only, but in situations where they are in close interaction with humans. To facilitate successful and safe cooperation between autonomous systems and humans, we need to design agents that can learn about human preferences as well as adapt to suboptimal human behaviour. We focus on the situation where the autonomous agent and the human simultaneously act in the same environment. As a result, observed human behaviour, which could be used to infer preferences, depends on the learning agent's actions. This leads to the problem of learning preferences and intentions from interactions. Learning in these interactive scenarios brings its own challenges, but also significant benefits as we will see in the following.

In this paper, we consider the problem of learning to cooperate with a potentially suboptimal partner while having no access to the joint reward function.

This problem is modeled as a cooperative episodic Markov Decision Process (MDP) between two agents $\mathcal{A}_1$ and $\mathcal{A}_2$. While agent $\mathcal{A}_2$ (the human) knows the joint reward function, we take the perspective of agent $\mathcal{A}_1$ (the learner) that has to cooperate with $\mathcal{A}_2$ without knowing or observing the rewards. As an example, consider a maze in which the human tries to reach a target while the learning agent can unlock doors to help the human move, but without knowing the precise target location. We focus on the Stackelberg formulation of the game, in which at the beginning of each episode the learner commits to a policy before the human does. This allows us to view the learning agent as a *designer of environments* that the human operates in. For instance, when the learning agent's actions correspond to unlocking doors in a grid world, then, in the Stackelberg game, we can interpret the learner's policy as choosing a maze layout, which is communicated to the human at the beginning of the episode and in which she operates.

Inverse Reinforcement Learning (IRL) [Rus98] can be used to infer the reward function of an agent from observations of that agent's behaviour, which is assumed to be (near-)optimal. In our case, the learner also obtains observations of the human's behaviour through interactions, which could then be used to infer the joint reward function. However, the human's actions, e.g. the path taken in a maze, depend on the learner's policy, e.g. the maze layout, so that in contrast to the standard IRL formulation the learner now *actively influences* the demonstrations of the human expert. This leads to an interesting Interactive IRL setting, where the learner can actively seek information about the joint reward function by playing specific policies. In this paper, we analyse how to infer the unknown (joint) reward function from interactions with the expert and how the learner should choose its policy so that the two agents collaborate efficiently over both the short and long term. We lay an emphasis on the role of the learner as the designer of environments and investigate what environments allow the learning agent to infer the reward function quickly while achieving high levels of cooperation.

Outline and Contribution. We discuss related work in Section I.2 and formally introduce the setting in Section I.3. Section I.4 considers the case where $\mathcal{A}_2$ plays *optimally*. We show how to learn about the reward function from interactions with $\mathcal{A}_2$ and prove the existence of ideal reward learning environments. We then construct an algorithm that is no-regret under mild assumptions. Section I.5 considers the case where $\mathcal{A}_2$ responds *suboptimally*. In Section I.5.1, we adapt conventional Bayesian IRL methods for estimating the reward function to our setting. We then analyse optimal commitment strategies for cooperating with suboptimal followers in Section I.5.2. Section I.6 describes the experiments, which we perform on random MDPs and specially constructed maze problems. Our experiments support our theoretical results and show that the interactive nature of our setting allows the learning agent to obtain a much better estimate of the reward function (compared to the standard IRL setting). We thus achieve better cooperation by intelligently probing the human's responses. Future work

is discussed in Section I.7. Finally, omitted proofs, experimental details and algorithms are collected in the appendix.

I.2 Related Work

Since our setting requires (a) inferring the joint reward function, as in IRL, and (b) collaborating with a potentially suboptimal agent, in this section we present related work in those two domains.

Inverse Reinforcement Learning. IRL [Rus98] aims to find a reward function that explains observed behaviour of an agent. We face the same problem, with the main difference being that *two* agents act in the environment simultaneously, one of which (the human) knows the reward function and the other (the learner) does not. Our algorithm for the case when $\mathcal{A}_2$ is optimal is based on a characterisation of reward functions consistent with an optimal policy, similarly to [NR00]. We extend their characterisation to our interactive setting and prove the existence of ideal (reward) learning environments. [RA07] adopt a Bayesian perspective to the IRL problem as it provides a principled way to reason under uncertainty. The Bayesian formulation of the IRL problem can naturally account for suboptimal demonstrations as well as partial information and we will show how to translate the Bayesian approach to our interactive IRL setting.

[Had+16] introduce the problem of cooperative IRL in which a robot must cooperate with a human but does not initially know the reward function. Their work focuses on apprenticeship learning, where the robot and the human *take turns* demonstrating and performing a task. In particular, they examine the problem of calculating optimal human demonstrations for the robot to observe. Instead, we consider the situation when the agents *interact* by simultaneously acting in the same environment.

Our setting also notably differs from apprenticeship learning [AN04] and imitation learning [RBZ06] more generally in that our goal is *not* to mimic the behaviour of $\mathcal{A}_2$, as effective cooperation between $\mathcal{A}_1$ and $\mathcal{A}_2$ may require both agents to perform entirely different tasks. [NS13] consider a cross-training approach in which a human expert and a robot repeatedly switch roles. In the first of two phases, the expert operates in an environment, which is influenced by the robot. The learner then observes the expert and updates its estimates of the reward function. In the second phase, the robot then demonstrates the learned policy while the expert influences the transitions. Crucially, in this approach the human steers the learning of the robot similar to teaching approaches for IRL [BN19; Par+19]. In contrast, we consider the situation where the learner actively seeks information from the human over whom we have no control.

[Nat+10] consider a multi-agent extension of IRL in which the learner observes multiple experts maximising a joint reward function. Similarly, [LAB19] address the problem of multi-agent IRL in certain general-sum games. In contrast to their work, we consider the case where the learner is *not* a passive observer,

but interacts with the other agent and thereby influences what observations it collects.

[ZP08] and [ZPC09] consider the problem of *environment design*: how to modify an environment so as to influence an agent's decisions. They analyse how to construct *reward incentives* to induce a particular policy when the reward function of the acting agent is unknown. In our setting, we can also view the learner as a designer of environments that the human operates in, however, with the difference that the learner influences transitions, but not the underlying reward function. Moreover, our goal is generally not to steer the human towards certain behaviour, but rather to learn from and cooperate with a human expert.

Cooperating with suboptimal partners. In the context of human-AI collaboration, there have been recent efforts addressing the problem of cooperating with a potentially suboptimal partner *when the reward function is known.* In particular, [Dim+17] and [Rad+19] consider a setting where the human responds suboptimally to the learning agent's policy. The former focuses on a single-stage Stackelberg game, while the latter on an online learning variant of the problem. However, in both cases the learning agent knows the human's reward function.

Our work also has some links to the problem of optimal commitment in Stackelberg games [CS06; Let+12]. While prior work assumes optimal responses and a potentially competitive game, we focus on finding optimal commitment strategies when playing with a *suboptimal* follower in a strictly *cooperative* setting.

I.3 Setting

We model this problem as a cooperative two-agent MDP $(S, A_1, A_2, \mathcal{P}, R, \gamma)$ between agents $\mathcal{A}_1$ and $\mathcal{A}_2$, where S denotes a finite state space, A_i a finite action space of agent $\mathcal{A}_i$ with $i \in \{1, 2\}$, $\mathcal{P} : S \times A_1 \times A_2 \to \Delta(S)$ the transition function, $R : S \to \mathbb{R}$ the *joint* reward function and $\gamma \in [0, 1)$ the discount factor. We will take the perspective of agent $\mathcal{A}_1$ that, without knowing or observing the joint reward function, aims to cooperate with its partner $\mathcal{A}_2$. We assume that the interaction between the two agents and the environment takes place in a sequence of episodes, where at the beginning of each episode, $\mathcal{A}_1$ commits to a policy π^1 first. Agent $\mathcal{A}_2$ then responds with a policy π^2 and the joint policy is executed until the end of the episode.[1] We assume that agents $\mathcal{A}_1$ and $\mathcal{A}_2$ know the transition function.

Interaction. The repeated interaction of both agents can be specified as the following *Stackelberg game*. In episode t:

1) $\mathcal{A}_1$ commits to policy π^1_t,

[1] Even in MDPs without termination condition, discounting corresponds to episodes that end with probability $1 - \gamma$ each time step.

2) $\mathcal{A}_2$ observes π_t^1 and responds with policy π_t^2,

3a) $\mathcal{A}_1$ observes the *fully specified policy* π_t^2, *or*

3b) $\mathcal{A}_1$ observes a *trajectory* $\tau_t = (s_0, a_0, b_0, \ldots, s_H, a_H, b_H)$ of length $H+1$.

Alternative 3a) describes the *full information* setting in which the complete policy π_t^2 is available to the learner at the end of each episode. This could, for instance, be the case when interaction takes place for a sufficiently long time in each episode, or the same policy is committed by $\mathcal{A}_1$ several times so that $\mathcal{A}_1$ can effectively observe $\mathcal{A}_2$'s response. Alternative 3b) corresponds to the *partial information* setting, where $\mathcal{A}_1$ interacts with $\mathcal{A}_2$ in a series of $H+1$ time steps and observes the generated trajectory only.

I.3.1 Preliminaries

By a slight abuse of notation, we sometimes refer to functions $f : S \to \mathbb{R}$ as vectors $f \in \mathbb{R}^{|S|}$. For instance, when convenient, we treat reward functions $R : S \to \mathbb{R}$ as vectors $R \in \mathbb{R}^{|S|}$. Let $\mathcal{V}_{\pi^1,\pi^2}$ denote the value function under the joint policy (π^1, π^2). The value function satisfies the Bellman equation, which we can concisely express in matrix-form as

$$\mathcal{V}_{\pi^1,\pi^2} = (I - \gamma \mathcal{P}_{\pi^1,\pi^2})^{-1} R,$$

where $\mathcal{V}_{\pi^1,\pi^2}$ and R are column vectors and $\mathcal{P}_{\pi^1,\pi^2}$ is the transition matrix obtained from $\mathcal{P}$ by marginalising over policy (π^1, π^2). Let $Q_{\pi^1,\pi^2}(s, a, b)$ denote the value of taking joint action (a, b) in state s under policy (π^1, π^2). When $\mathcal{A}_1$ commits to a policy π^1 first, agent $\mathcal{A}_2$ gets to plan under the marginalised transitions $\mathcal{P}_{\pi^1} : S \times A_2 \to \Delta(S)$ given by $\mathcal{P}_{\pi^1}(s'|s, b) = \mathbb{E}_{a\sim\pi^1}[\mathcal{P}(s'|s, a, b)]$. The Q-values for $\mathcal{A}_2$ under $\mathcal{P}_{\pi^1}$ equal $Q_{\pi^1,\pi^2}(s, b) = \mathbb{E}_{a\sim\pi^1}[Q_{\pi^1,\pi^2}(s, a, b)]$ and we denote the optimal Q-value with respect to $\mathcal{P}_{\pi^1}$ by $Q^*_{\pi^1}(s, b) = \max_{\pi^2} Q_{\pi^1,\pi^2}(s, b)$.

Behavioural Models for $\mathcal{A}_2$. A typical assumption about the behaviour of a partner (or opponent) in game theory [Nis+07] and IRL [NR00] is that of optimal behaviour, sometimes referred to as fully rational behaviour. In our case, this means that in episode t, agent $\mathcal{A}_2$ plays an optimal response $\pi_t^2(\pi_t^1)$ to the policy π_t^1 committed by agent $\mathcal{A}_1$. Note that we will simply write π_t^2 when the dependence on π_t^1 is clear from the context.

We are also interested in the case when $\mathcal{A}_2$ is suboptimal. A common decision-model for suboptimal human behaviour in IRL [JMD20], economics [Luc59], and cognitive science [BST09] are Boltzmann-rational policies for which the probability of choosing an action is exponentially dependent on its expected value:

$$\pi^2(b \mid s, \pi^1) \propto \exp\big(\beta Q^*_{\pi^1}(s, b)\big).$$

Here, $\beta \geq 0$ is called the inverse temperature of the distribution and indicates how rationally $\mathcal{A}_2$ is behaving. In particular, for $\beta = 0$, $\mathcal{A}_2$ acts uniformly at

random, and for $\beta \to \infty$, $\mathcal{A}_2$ acts perfectly rational, i.e. optimally in response to $\mathcal{A}_1$'s committed policy.

Objective and Regret. Agent $\mathcal{A}_1$ aims to maximise the expected sum of discounted rewards by learning about the joint reward function and cooperating with $\mathcal{A}_2$. In general, due to the possibly suboptimal nature of $\mathcal{A}_2$, we have that $\max_{\pi^1} \mathcal{V}_{\pi^1,\pi^2(\pi^1)} \preceq \max_{\pi^1,\pi^2} \mathcal{V}_{\pi^1,\pi^2}$, i.e. the value of the game under $\mathcal{A}_2$'s behavioural model is bounded by the value of the joint optimal policy. For an initial state distribution D, we define the value of the optimal commitment strategy as

$$V^* = \max_{\pi^1} \mathbb{E}_{s_0 \sim D}\left[\mathcal{V}_{\pi^1,\pi^2(\pi^1)}(s_0)\right],$$

where $\pi^2(\pi^1)$ denotes the response of $\mathcal{A}_2$ to policy π^1. Note that the optimal value V^* may only be well-defined with respect to a specific initial state distribution as a dominating commitment strategy may fail to exist when $\mathcal{A}_2$ responds suboptimally (see Section I.5.2). We define the (per-episode) regret of playing policy π^1 as the difference $\mathscr{L}(\pi^1) = V^* - \mathbb{E}_{s_0 \sim D}[\mathcal{V}_{\pi^1,\pi^2(\pi^1)}(s_0)]$. Similarly, we define the (online) regret of playing policies $\pi^1_1, \ldots, \pi^1_T$ as the sum $\mathscr{L}(\pi^1_1, \ldots, \pi^1_T) = \sum_{t=1}^{T} \mathscr{L}(\pi^1_t)$.

I.3.2 Interactive IRL

In the classical IRL problem, the learner is able to observe an expert performing a task. The observations are then interpreted as demonstrations of approximately optimal behaviour in a *fixed* single-agent MDP with unknown reward function. Our setting is substantially different, as two agents must collaborate in the same two-agent MDP, with the first agent not knowing the common reward function. As a result, the second agent's demonstrations depend on the first agent's policy and so become *context-dependent*. In addition, learning must take place in an *online* fashion, as the first agent must adapt its policy to extract information and to better collaborate.

$\mathcal{A}_1$ as an MDP Designer. When the learner, $\mathcal{A}_1$, commits to a policy π^1 at the beginning of an episode, then — with knowledge of π^1 — the expert, $\mathcal{A}_2$, can be seen as planning in a single-agent MDP with transition function $\mathcal{P}_{\pi^1}$. Consequently, from the perspective of the learner, choosing a policy π^1 is equivalent to designing single-agent MDPs for the human expert to act in. While the state space, $\mathcal{A}_2$'s action space, the (unknown) reward function as well as the discount factor remain the same across these simplified MDPs, $\mathcal{A}_2$ may face different environment dynamics $\mathcal{P}_{\pi^1}$ depending on $\mathcal{A}_1$'s policy. This is in contrast to the standard IRL setting in which demonstrations always take place in the same fixed MDP. An abstract example where the learner creates different environments for the expert to operate in is illustrated in Figure I.1(a).

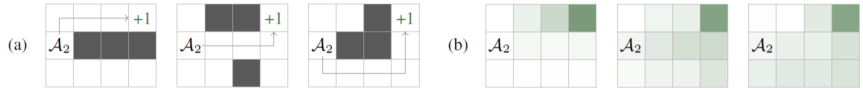


Figure I.1: (a) $\mathcal{A}_1$ designs a maze for $\mathcal{A}_2$ to navigate in and collect a reward in the top right corner. $\mathcal{A}_2$ behaves differently, i.e. chooses a different path, depending on the maze created by $\mathcal{A}_1$. (b) The mean reward function computed using Bayesian IRL [RA07] when observing $\mathcal{A}_2$ navigate in each of the three mazes. Dark colours denote higher estimated rewards.

Context-Dependent Responses. The learner can now interpret the expert's response to a policy π^1 as a demonstration in the single-agent MDP $(S, A_2, \mathcal{P}_{\pi^1}, R^*, \gamma)$, where R^* is the true reward function that is unknown and unobserved by $\mathcal{A}_1$. Since $\mathcal{A}_2$ faces possibly different environment dynamics across episodes, we can also expect $\mathcal{A}_2$'s behaviour to vary between episodes. In Figure I.1(a), for instance, the expert adapts their policy to the specific maze layout created by the learner. As a result, $\mathcal{A}_2$'s responses (and thus demonstrations) become context-dependent in the sense that they always depend on $\mathcal{A}_1$'s policy, i.e. the environment that is implicitly generated by $\mathcal{A}_1$.

In particular, we see that even though the underlying reward function remains the same, the results of IRL methods vary depending on the environment in which demonstrations were provided. Figure I.1(b) also illustrates that reward learning may overfit to specific environment dynamics, which has also been observed by, e.g., [Toy+20]. While there may exist certain environment dynamics that are better suited for learning rewards, in this paper we focus on designing a sequence of environments, based on past data, to learn the reward function efficiently.

Online Learning. As the game progresses, the learner interacts with the expert in a series of episodes, thereby collecting a stream of observations. Then, in order to extract more information as well as to improve cooperation in the next episode, the learner may want to leverage the observations up to episode t to learn about the joint reward function and to inform its decisions in episode $t+1$. Naturally, since the learner *actively influences* the demonstrations by the expert, we ask ourselves whether demonstrations under some environment dynamics $\mathcal{P}_{\pi^1}$ are more informative than others. In particular, how much more information (if any) can be gained from demonstrations in unseen environments? In the following, we will address these questions both theoretically and empirically.

I.4 Cooperating with Optimal Agents

Here we consider the case when $\mathcal{A}_2$ responds optimally to the commitment of $\mathcal{A}_1$. In Section I.4.1, we characterise the set of *feasible* reward functions, i.e. those that are consistent with observed responses, and prove the existence of ideal (reward) learning environments. We then describe an algorithm that is no-regret under

an assumption on the identifiability of suboptimal behaviour in Section I.4.2. The omitted proofs from this section can be found in Appendix A.1.

I.4.1 Learning from Optimal Responses

For our theoretical analysis, we focus on the full information setting in which $\mathcal{A}_1$ observes the fully specified policy played by the expert at the end of each episode. In a first step, we define a *feasible* reward function under (π^1, π^2) as a reward function for which $\mathcal{A}_2$'s response to the commitment of $\mathcal{A}_1$ is optimal.

Definition I.4.1. We say that a reward function R is *feasible* when observing policy π^2 in response to π^1 if π^2 is optimal in the single-agent MDP $(S, A_2, \mathcal{P}_{\pi^1}, R, \gamma)$.

We now adapt the standard result by [NR00] to obtain a characterisation of the set of feasible reward functions under policies π^1 and π^2. Here, we let $\succeq$ denote element-wise inequality.

Theorem I.4.2 ([NR00])**.** *Let there be an MDP without reward function $(S, A_1, A_2, \mathcal{P}, \gamma)$. A reward function R is feasible under policies π^1 and π^2 if and only if*

$$\big(\mathcal{P}_{\pi^1,\pi^2} - \mathcal{P}_{\pi^1,b}\big)\big(I - \gamma\mathcal{P}_{\pi^1,\pi^2}\big)^{-1} R \succeq 0 \quad \forall b \in A_2,$$

where $\mathcal{P}_{\pi^1,b}$ is the one-step transition matrix under policy π^1 and action $b \in A_2$.

Since $\mathcal{A}_1$ and $\mathcal{A}_2$ repeatedly interact in a series of episodes, a reward function is feasible after t episodes if and only if it is feasible under all policies $\pi^1_1, \dots, \pi^1_t$ and corresponding responses $\pi^2_1, \dots, \pi^2_t$. As an immediate consequence of Theorem I.4.2, we then obtain the following characterisation of reward functions that are feasible under multiple observations.

Corollary I.4.3. *Let there be an MDP without reward function $(S, A_1, A_2, \mathcal{P}, \gamma)$. A reward function R is feasible when observing policies $(\pi^1_1, \pi^2_1), \dots, (\pi^1_t, \pi^2_t)$ if and only if*

$$\begin{aligned}
\big(\mathcal{P}_{\pi^1_1,\pi^2_1} - \mathcal{P}_{\pi^1_1,b}\big)\big(I - \gamma\mathcal{P}_{\pi^1_1,\pi^2_1}\big)^{-1} R \succeq 0 \quad \forall b \in A_2,\\
\dots\\
\big(\mathcal{P}_{\pi^1_t,\pi^2_t} - \mathcal{P}_{\pi^1_t,b}\big)\big(I - \gamma\mathcal{P}_{\pi^1_t,\pi^2_t}\big)^{-1} R \succeq 0 \quad \forall b \in A_2.
\end{aligned}$$

We denote the set of reward functions that satisfy these constraints by $\mathcal{R}_t = \mathcal{R}((\pi^1_1, \pi^2_1), \dots, (\pi^1_t, \pi^2_t))$.

The IRL problem is an inherently ill-posed problem as degenerate solutions such as constant reward functions explain any observed behaviour. In fact, we see that any reward function $R \in \mathbb{R}^{|S|}$ is indistinguishable from its positive affine transformations $\mathrm{Aff}(R) = \{\lambda_1 R + \lambda_2 \mathbf{1} \colon \lambda_1 \geq 0, \lambda_2 \in \mathbb{R}\}$.

Lemma I.4.4. *If $\mathcal{A}_2$ responds optimally to the commitment of $\mathcal{A}_1$, any reward function R is indistinguishable from its positive affine transformations, i.e. R is feasible iff every $\bar{R} \in \mathrm{Aff}(R)$ is feasible.*

In particular, Lemma I.4.4 states that all positive affine transformations of the true reward function R^* are always feasible.[2] However, since any reward function in $\text{Aff}(R^*)$ induces the same optimal (joint) policy, finding it is sufficient for optimally solving the IRL problem.

Perhaps surprisingly, we find that if $\mathcal{A}_1$'s policies can induce any transition matrix for $\mathcal{A}_2$, then there exists a policy π^1 such that its optimal response $\pi^2(\pi^1)$ can only be explained by positive affine transformations of the true reward function.

Theorem I.4.5. *(A) If $\mathcal{A}_2$ responds optimally and (B) if for all $\mathcal{T} : S \times A_2 \to \Delta(S)$ there exists π^1 such that $\mathcal{P}_{\pi^1} \equiv \mathcal{T}$, then there exists a policy π^1 with optimal response π^2 such that the feasible set of reward functions under (π^1, π^2) is given by $\text{Aff}(R^*)$, i.e. $\mathcal{R}((\pi^1, \pi^2)) = \text{Aff}(R^*)$.*

To emphasise the interpretation and relevance of Theorem I.4.5 in the standard single-agent IRL setting, we can also rephrase Theorem I.4.5 as follows:

Remark I.4.6. For any state space S, action space A, reward function R^* and discount factor $\gamma \in [0, 1)$, there exists a transition matrix $\mathcal{T} : S \times A \to \Delta(S)$ such that the optimal policy π in $(S, A, \mathcal{T}, R^*, \gamma)$ uniquely characterises R^* up to positive affine transformations.

This leads to the following corollary, which shows that it is possible to check in a single episode whether any given reward function is an affine transformation of R^*.

Corollary I.4.7. *Under Assumptions (A) and (B) of Theorem I.4.5, the learner can verify in any episode whether a reward function R is a positive affine transformation of the unknown and unobserved reward function R^*.*

We have shown that for any reward function R^* there exists an environment $\mathcal{T} : S \times A_2 \to \Delta(S)$ such that the optimal policy with respect to $\mathcal{T}$ and R^* characterises R^* up to positive affine transformations (Theorem I.4.5). This implied that the learner, without knowledge of R^*, can verify whether a reward function is element in $\text{Aff}(R^*)$ by playing a specific policy (Corollary I.4.7). However, the assumption that $\mathcal{A}_1$ can create any environment dynamics is very strong and we notice that, while retrieving the set $\text{Aff}(R^*)$ is clearly desirable, it is generally not necessary in order to cooperate optimally as other reward functions may also induce optimal behaviour. Thus, milder assumptions may be sufficient to learn about the reward function so that $\mathcal{A}_1$ is an optimal partner to $\mathcal{A}_2$. In the following, we propose an algorithm that learns about the reward function by adaptively designing environments and that is no-regret under mild assumptions.

[2] We generally denote the true underlying reward function by R^*. Note that R^* is unknown to and unobserved by $\mathcal{A}_1$.

Algorithm 1 Interactive IRL via Linear Programming

input: $(S, A_1, A_2, \mathcal{P}, \gamma)$, initial policy π_1^1
for $t = 1, 2, \ldots$ **do**
  commit to policy π_t^1
  observe response π_t^2
  get constraints $\mathcal{C}_t = \mathcal{C}((\pi_1^1, \pi_1^2), \ldots, (\pi_t^1, \pi_t^2))$
  sample objective vector c uniformly at random
  find solution $R_t \in \mathcal{R}_t$ of LP (I.1) for $\mathcal{C}_t$ and c
  compute $\pi_{t+1}^1 \in \Pi_1^{\mathtt{opt}}(R_t)$

I.4.2 An Algorithm for Interactive IRL

We now present an online algorithm for learning from and cooperating with an optimally responding agent $\mathcal{A}_2$ when agent $\mathcal{A}_1$ gets to observe the fully specified policy of $\mathcal{A}_2$ at the end of each episode. Note that we can always restrict the space of reward functions to the $|S|$-dimensional unit simplex $\Delta(S)$ as any positive affine transformation of $R \in \Delta(S)$ is equivalent to R in the sense that they are feasible under the same observations and induce the same optimal (joint) policies (Lemma I.4.4). Now, as the constraints characterising the feasible set $\mathcal{R}_t = \mathcal{R}((\pi_1^1, \pi_1^2), \ldots, (\pi_t^1, \pi_t^2))$ are linear in the reward function (Corollary I.4.3), we can use a Linear Program (LP) to find a reward function in $\mathcal{R}_t \cap \Delta(S)$. Let $\mathcal{C}((\pi_1^1, \pi_1^2), \ldots, (\pi_t^1, \pi_t^2))$ denote the set of constraints induced by $(\pi_1^1, \pi_1^2), \ldots, (\pi_t^1, \pi_t^2)$. In episode $t+1$, we then sample an $|S|$-dimensional objective function c uniformly at random and solve the following LP:

$$\max_{R \in \Delta^{|S|}} c^\top R \text{ subject to } \mathcal{C}((\pi_1^1, \pi_1^2), \ldots, (\pi_t^1, \pi_t^2)). \tag{I.1}$$

In the unlikely event that the LP computes the constant reward function in $\Delta(S)$, we resample the objective c and solve the LP again. Given a prospective reward function R, we then want to compute an optimal commitment strategy in $(S, A_1, A_2, \mathcal{P}, R, \gamma)$. We see that if $\mathcal{A}_2$ responds optimally, it suffices to find an optimal joint policy as it yields an optimal commitment strategy for $\mathcal{A}_1$.

Lemma I.4.8. *Let $(\bar{\pi}^1, \bar{\pi}^2)$ be an optimal joint policy. If agent $\mathcal{A}_2$ responds optimally to the commitment of $\mathcal{A}_1$, then $\mathcal{V}_{\bar{\pi}^1, \pi^2(\bar{\pi}^1)} = \mathcal{V}_{\bar{\pi}^1, \bar{\pi}^2}$. In particular, this entails that $\max_{\pi^1} \mathcal{V}_{\pi^1, \pi^2(\pi^1)} = \max_{\pi^1, \pi^2} \mathcal{V}_{\pi^1, \pi^2}$.*

Note that an optimal joint policy and thus an optimal commitment strategy for $\mathcal{A}_1$ can be computed in time polynomial in the number of states and actions. In episode $t+1$, the algorithm then commits to a policy $\pi_{t+1}^1 \in \Pi_1^{\mathtt{opt}}(R)$, where R is the solution of the LP (I.1) and $\Pi_1^{\mathtt{opt}}(R)$ is the set of optimal commitment strategies under R. A description of this approach is given by Algorithm 1. In fact, we can show that Algorithm 1 is *no-regret* under the assumption that reward functions that induce suboptimal joint policies are identifiable in the sense that these also induce suboptimal responses.

Proposition I.4.9. *Suppose that for any non-constant reward function $R \in \Delta(S)$ it holds that if an optimal joint policy (π^1, π^2) under R is suboptimal under R^*, then in return there exists an optimal response $\pi^2(\pi^1)$ under R^* that is suboptimal under R. Moreover, assume that $\mathcal{A}_2$ responds optimally and breaks ties between equally good policies uniformly at random. Then, the average regret suffered by Algorithm 1 converges to zero almost surely.*

Proof Sketch. The proof relies on a finite cover of the space of reward functions. We can show that in every step of the algorithm either an optimal policy was played (generating no regret) or with positive probability the reward functions in at least one of the sets of the cover become infeasible - thus ultimately reducing the set of feasible reward functions to only those that yield optimal policies. ■

I.5 Cooperating with Suboptimal Agents

We now consider the case when $\mathcal{A}_2$ responds suboptimally according to some behavioural model such as Boltzmann-rational policies. Section I.5.1 extends the Bayesian IRL formulation to our setting and Section I.5.2 analyses the problem of computing optimal commitment strategies when $\mathcal{A}_2$ is playing suboptimally. The omitted proofs from this section can be found in Appendix A.1.

I.5.1 Learning from Suboptimal Responses

When demonstrations are possibly suboptimal, it is natural to take a Bayesian perspective [RA07] as it provides a principled way to reason under uncertainty. Moreover, the Bayesian approach naturally extends to the partial information setting, where only trajectories generated by both agents' policies are available for learning. We assume that $\mathcal{A}_2$ responds with Boltzmann-rational policies with *unknown* inverse temperature β[3] and adapt the Bayesian IRL formulation to our setting. Suppose that in the first t episodes $\mathcal{A}_1$ observes $(\pi_1^1, \tau_1), \ldots, (\pi_t^1, \tau_t)$, where τ_i is the trajectory generated by $\mathcal{A}_1$'s policy π_i^1 and $\mathcal{A}_2$'s response $\pi_i^2(\pi_i^1)$ for $i \in [t]$.[4] Bayesian IRL aims to estimate the posterior

$$\begin{aligned}&\mathbb{P}(R, \beta \mid (\pi_1^1, \tau_1), \ldots, (\pi_t^1, \tau_t)) \\ &\qquad = \frac{\mathbb{P}((\pi_1^1, \tau_1), \ldots, (\pi_t^1, \tau_t) \mid R, \beta)\mathbb{P}(R)\mathbb{P}(\beta)}{\mathbb{P}((\pi_1^1, \tau_1), \ldots, (\pi_t^1, \tau_t))},\end{aligned}$$

given priors $\mathbb{P}(R)$ and $\mathbb{P}(\beta)$ over reward functions and inverse temperatures, respectively. We notice that the observations $(\pi_1^1, \tau_1), \ldots, (\pi_t^1, \tau_t)$ are conditionally independent under measure $\mathbb{P}(\cdot \mid R, \beta)$. As a result, we can express

[3] Note that any other parameterised behavioural model could also be modeled by this Bayesian formulation.

[4] For notational conciseness, we assume here that the length of a trajectory is fixed across all episodes.

their likelihood as $\mathbb{P}((\pi_1^1, \tau_1), \ldots, (\pi_t^1, \tau_t) \mid R, \beta) = \prod_{i=1}^{t} \mathbb{P}((\pi_i^1, \tau_i) \mid R, \beta)$. The likelihood for each observation (π_i^1, τ_i) can then be computed as

$$\begin{aligned}\mathbb{P}((\pi_i^1, \tau_i) \mid R, \beta) &= \prod_{h=0}^{H} \pi^2(b_{i,h} \mid s_{i,h}, \pi_i^1, R, \beta) \\ &\propto \exp\big(\beta \sum_{h=0}^{H} Q^*_{\pi_i^1}(s_{i,h}, b_{i,h}, R)\big).\end{aligned}$$

The Bayesian method we employ generates samples from the posterior via Markov Chain Monte Carlo (MCMC), similarly to [RA07; RD11]. At a high level, we employ a Metropolis-Hastings algorithm on the reward simplex, with a uniform prior on the reward function and an exponential prior on the inverse temperature (see Algorithm 4 in Appendix A.3).

I.5.2 Planning with Suboptimal Agents

Prior work on computing optimal commitment strategies in stochastic games typically assumes that the follower is responding optimally [Let+12; VS12]. In this section, we analyse optimal commitment strategies for the *cooperative* Stackelberg game from Section I.3 when agent $\mathcal{A}_2$, i.e. the follower, responds *suboptimally* according to some behavioural model, e.g. Boltzmann-rational policies or ε-greedy policies. For this, the concept of dominating policies play a crucial role.

Definition I.5.1. A policy π^1 is *dominating* if $\mathcal{V}_{\pi^1, \pi^2(\pi^1)}(s) \geq \mathcal{V}_{\bar{\pi}^1, \pi^2(\bar{\pi}^1)}(s)$ for all policies $\bar{\pi}^1$ and states $s \in S$.

The existence of dominating policies is closely linked to our capacity to compute an optimal commitment strategy efficiently as it is a key requirement for dynamic programming. We show that if $\mathcal{A}_2$ plays proportionally with respect to the expected value of taking an action, there may not exist dominating policy for $\mathcal{A}_1$ to commit to.

Theorem I.5.2. *If $\pi^2(b \mid s) \propto f(Q^*_{\pi^1}(s, b))$ for any strictly increasing function $f : [0, \infty) \to [0, \infty)$, then a dominating commitment strategy for agent $\mathcal{A}_1$ may not exist.*

In particular, this means that if $\mathcal{A}_2$ plays Boltzmann-rational policies, a dominating commitment strategy may fail to exist. Note that Theorem I.5.2 generally only holds for *strictly* increasing functions f, as, for instance, there always exists a dominating commitment strategy when $\mathcal{A}_2$ plays uniformly at random. However, even for behavioural models as simple as ε-greedy, we see that a dominating commitment strategy does not necessarily exist.

Lemma I.5.3. *If $\mathcal{A}_2$ plays ε-greedy, a dominating commitment strategy for $\mathcal{A}_1$ may not exist.*

Despite these difficulties, we provide algorithms to approximate optimal commitment strategies for the case of Boltzmann-rational responses (Algorithm 2) and ε-greedy responses (Algorithm 3), which can be found in Appendix A.2. The proposed methods correspond to approximate value iteration algorithms that keep track of two value functions, each modelling one agent. We include an empirical evaluation of the proposed algorithms in Appendix A.2.3, which demonstrates that accounting for the suboptimal nature of $\mathcal{A}_2$ reliably improves performance.

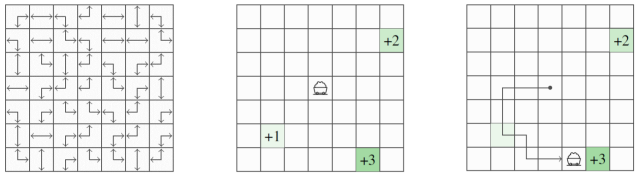
I.6 Experiments

In our experiments, we investigate how much the learner benefits from repeatedly interacting with the expert. To address this question and emphasise the potential benefit of demonstrations in different environments, we include the situation where $\mathcal{A}_1$ only observes the response of $\mathcal{A}_2$ to the initial policy π_1^1 played by $\mathcal{A}_1$. This resembles the standard IRL setting where we observe the expert only in a single fixed environment $(S, A_2, \mathcal{P}_{\pi_1^1}, R, \gamma)$.

Here, the initial policy π_1^1 is chosen uniformly at random. We model the standard IRL setting by repeatedly generating responses of $\mathcal{A}_2$ with respect to π_1^1, i.e. in the implied environment $\mathcal{P}_{\pi_1^1}$. Using these observations, we then estimate the reward function using standard IRL, compute the optimal policy with respect to the estimated rewards, and evaluate the regret of this policy. In contrast, in the Interactive IRL setting, the learner gets to choose a different policy in subsequent episodes. In this case, we report the online regret of the actually played policies, i.e. the actual regret of the learner. More details are provided in Appendix A.3.

I.6.1 Environments

Maze-Maker. In this environment, agents $\mathcal{A}_1$ and $\mathcal{A}_2$ jointly control a cart in a 7×7 grid world. In this grid world, the doors leading from one cell to the neighbouring ones are locked. However, $\mathcal{A}_1$ can unlock exactly two doors at any



(a) (b) (c)

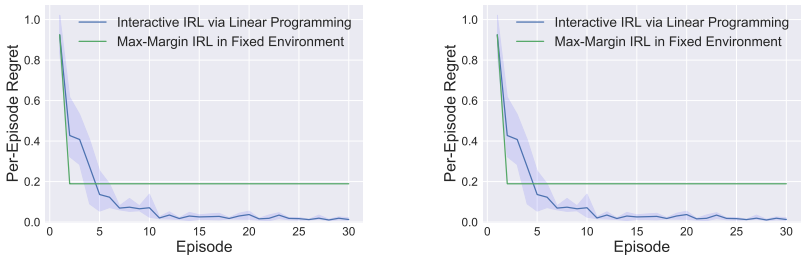
Figure I.2: Maze-Maker Environment. (a) The initial game setup with starting position in the center and three rewards scattered across the grid world. (b) When $\mathcal{A}_1$ commits to a policy it implicitly creates a maze for $\mathcal{A}_2$ to navigate the cart in. (c) An exemplary path taken by $\mathcal{A}_2$ in the maze implied by $\mathcal{A}_1$'s policy.

time step before they fall shut again. Agent $\mathcal{A}_2$ can attempt to move the cart through a door to a neighbouring cell. However, when the door is locked, the cart stays where it was. The agents are tasked with collecting three rewards of different value (+1, +2, +3), which disappear once collected. While the expert, $\mathcal{A}_2$, knows where the rewards are placed, the helper, $\mathcal{A}_1$, does not know their location. We model this environment as a two-agent MDP with 392 states (49×8) and discount factor $\gamma = 0.9$, where $\mathcal{A}_1$ has six actions (unlocking two out of four doors) and $\mathcal{A}_2$ four actions (moving the cart North, East, South, West). An illustration of the environment is given in Figure I.2.

Random MDPs. We also randomly generated MDPs with 200 states and four actions for each agent. We randomly draw the transition dynamics from a Dirichlet distribution, with restrictions on the influence of each agent on the transitions, and the rewards from an i.i.d. Beta distribution. The discount factor is set to $\gamma = 0.9$.

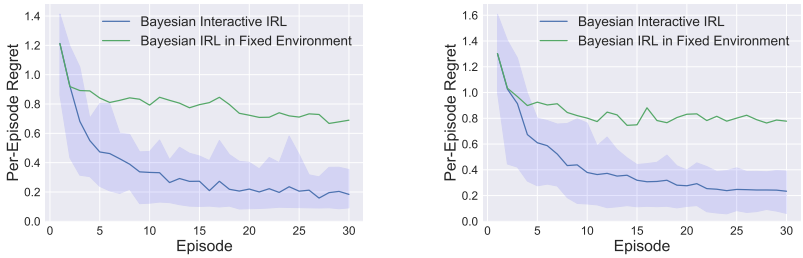
I.6.2 Results

Optimal Responses and Full Information. In Figure I.3a and I.3b, we observe that the per-episode regret suffered by Algorithm 1 in both environments decreases notably with the number of episodes played. In particular, we see that after only a few episodes the per-episode regret of Algorithm 1 is significantly lower than for maximum-margin IRL [NR00] when $\mathcal{A}_1$ only observes the response to the initial policy π_1^1. This roughly corresponds to the standard IRL setting in which demonstrations are obtained in a single environment only. We thus find that the learner significantly benefits from observing $\mathcal{A}_2$'s behaviour in new and different environments, i.e. with respect to different policies of $\mathcal{A}_1$. In particular, it appears to be necessary to observe the expert's response to several different policies in order to infer an approximately optimal reward function. The results are averaged over 5 runs.



(a) Maze-Maker (b) Random MDPs

Figure I.3: Optimal Responses and Full Information. Blue lines show the per-episode regret $\mathscr{L}(\pi_t^1)$ of Algorithm 1. Green lines correspond to the regret of maximum-margin IRL [NR00] performed with observation (π_1^1, π_1^2) only.



(a) Maze-Maker (b) Random MDPs

Figure I.4: Suboptimal Responses and Partial Information. Blue lines show the per-episode regret of Bayesian Interactive IRL (Algorithm 4 in Appendix A.3). Green lines refer to Bayesian IRL performed for trajectories repeatedly generated by π_1^1 and π_1^2.

Suboptimal Responses and Partial Information. For the case of suboptimal responses and partial information, we let $\mathcal{A}_2$ respond with Boltzmann-rational policies with inverse temperature $\beta = 10$ in both environments. We assume that the inverse temperature, i.e. the optimality of $\mathcal{A}_2$, is unknown to the learner and simulate the partial information setting by generating trajectories according to policies π_t^1 and π_t^2 in episode t. We let an episode end with probability $1 - \gamma$ each time step so that the lengths of observed trajectories are random.

Figure I.4a and I.4b show that Bayesian Interactive IRL (Algorithm 4) reliably improves its estimate of the true reward function with the number of episodes played and that the learner again substantially benefits from observing $\mathcal{A}_2$ act in different environments. While obtaining an increasing amount of trajectories in the *same* environment improves the estimate of the reward function as well, we see that trajectories generated in new environments, i.e. with respect to different policies of $\mathcal{A}_1$, yield much more information and thus allow for a better estimate of the unknown reward function. The results are averaged over 10 runs.

I.7 Discussion and Future Work

We considered an interactive cooperation problem when the objective is unknown to one of the agents. This can be seen as a two-agent version of the IRL problem, where one agent is actively trying to infer the preferences of the other in order to cooperate. While the classical IRL problem is generally ill-posed, the interactive version that we study here can indeed be solved if the learning agent has sufficient power to affect the transitions. This is supported by both our experimental and theoretical results. In particular, the experiments clearly show that we can more accurately estimate the reward function (and hence collaborate more effectively) if we intelligently probe the other agent's responses.

An open theoretical question is whether upper and lower problem-dependent bounds on the episodic regret could be obtained in this setting. We presume that such bounds would involve a characterisation of $\mathcal{A}_1$'s power to affect the

transitions. A natural extension of our setting would be the case where $\mathcal{A}_1$ does not reveal its policy to $\mathcal{A}_2$, but instead the latter simply observes the former's actions. In future work, it will also be interesting to construct Interactive IRL algorithms that scale to large state spaces (or continuous domains) and test these in real-world applications.

Our observation that reward learning benefits from demonstrations under different environment dynamics also opens up a new and interesting perspective on IRL more generally. While current IRL methods still struggle to learn satisfactory reward functions in certain domains (even with abundant data), it could be promising to try to infer the reward function from demonstrations in slight variations of the target environment (when possible). Moreover, our results suggest that receiving samples under new environment dynamics is generally more valuable than collecting additional samples from the same environment. Thus, such an approach could be useful in domains where resources are limited and samples expensive.

References

[AN04] Abbeel, P. and Ng, A. Y. "Apprenticeship learning via inverse reinforcement learning". In: *Proceedings of the twenty-first International Conference on Machine learning*. Banff, Alberta, Canada, 2004, p. 1.

[BN19] Brown, D. S. and Niekum, S. "Machine teaching for inverse reinforcement learning: Algorithms and applications". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01. 2019, pp. 7749–7758.

[BST09] Baker, C. L., Saxe, R., and Tenenbaum, J. B. "Action understanding as inverse planning". In: *Cognition* vol. 113, no. 3 (2009), pp. 329–349.

[CS06] Conitzer, V. and Sandholm, T. "Computing the optimal strategy to commit to". In: *Proceedings of the 7th ACM conference on Electronic commerce*. 2006, pp. 82–90.

[Dim+17] Dimitrakakis, C. et al. "Multi-View Decision Processes: The Helper-AI Problem". In: *Advances in Neural Information Processing Systems*. 2017, pp. 5449–5458.

[Gho+19] Ghosh, A. et al. "Towards deployment of robust AI agents for human-machine partnerships". In: *arXiv preprint arXiv:1910.02330* (2019).

[Had+16] Hadfield-Menell, D. et al. "Cooperative inverse reinforcement learning". In: *Advances in neural information processing systems* vol. 29 (2016).

[JMD20] Jeon, H. J., Milli, S., and Dragan, A. "Reward-rational (implicit) choice: A unifying formalism for reward learning". In: *Advances in Neural Information Processing Systems* vol. 33 (2020), pp. 4415–4426.

[LAB19] Lin, X., Adams, S. C., and Beling, P. A. "Multi-agent inverse reinforcement learning for certain general-sum stochastic games". In: *Journal of Artificial Intelligence Research* vol. 66 (2019), pp. 473–502.

[Let+12] Letchford, J. et al. "Computing optimal strategies to commit to in stochastic games". In: *Twenty-Sixth AAAI Conference on Artificial Intelligence*. 2012, pp. 1380–1386.

[Luc59] Luce, R. D. *Individual choice behavior: A theoretical analysis*. Courier Corporation, 1959.

[Nat+10] Natarajan, S. et al. "Multi-agent inverse reinforcement learning". In: *2010 Ninth International Conference on Machine Learning and Applications*. 2010, pp. 395–400.

[Nis+07] Nisan, N. et al. *Algorithmic Game Theory*. Cambridge University Press, 2007.

[NR00] Ng, A. Y. and Russell, S. J. "Algorithms for Inverse Reinforcement Learning". In: *Proceedings of the Seventeenth International Conference on Machine Learning*. 2000, p. 2.

[NS13] Nikolaidis, S. and Shah, J. "Human-Robot Cross-Training: Computational Formulation, Modeling and Evaluation of a Human Team Training Strategy". In: *Proceedings of the 8th ACM/IEEE International Conference on Human-Robot Interaction*. HRI '13. Tokyo, Japan, 2013, pp. 33–40.

[Par+19] Parameswaran, K. et al. "Interactive teaching algorithms for inverse reinforcement learning". In: *28th International Joint Conference on Artificial Intelligence, 2019.* CONF. 2019.

[Put14] Puterman, M. L. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[RA07] Ramachandran, D. and Amir, E. "Bayesian Inverse Reinforcement Learning". In: *Proceedings of the 20th International Joint Conference on Artifical Intelligence*. 2007, pp. 2586–2591.

[Rad+19] Radanovic, G. et al. "Learning to collaborate in Markov decision processes". In: *International Conference on Machine Learning*. 2019, pp. 5261–5270.

[RBZ06] Ratliff, N. D., Bagnell, J. A., and Zinkevich, M. A. "Maximum margin planning". In: *Proceedings of the 23rd International Conference on Machine learning*. 2006, pp. 729–736.

[RD11] Rothkopf, C. A. and Dimitrakakis, C. “Preference elicitation and inverse reinforcement learning”. In: *Joint European conference on machine learning and knowledge discovery in databases*. 2011, pp. 34–48.

[Rus98] Russell, S. “Learning agents for uncertain environments”. In: *Proceedings of the eleventh annual conference on Computational learning theory*. 1998, pp. 101–103.

[Toy+20] Toyer, S. et al. “The MAGICAL Benchmark for Robust Imitation”. In: *Advances in Neural Information Processing Systems*. 2020.

[VS12] Vorobeychik, Y. and Singh, S. “Computing stackelberg equilibria in discounted stochastic games”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 26. 1. 2012, pp. 1478–1484.

[ZP08] Zhang, H. and Parkes, D. “Enabling environment design via active indirect elicitation”. In: *4th Multidisciplinary Workshop on Advances in Preference Handling*. 2008.

[ZPC09] Zhang, H., Parkes, D. C., and Chen, Y. “Policy Teaching through Reward Function Learning”. In: EC ’09. Stanford, California, USA: Association for Computing Machinery, 2009, pp. 295–304.

A.1 Proofs

A.1.1 Proof of Theorem I.4.2

Proof of Theorem I.4.2. Substituting transition matrix $\mathcal{P}$ by $\mathcal{P}_{\pi^1}$ in the proof by [NR00] readily implies Theorem I.4.2. Note that if $\pi^2(s) = \bar{b}$ for all $s \in S$, the inequality vacuously holds for $b = \bar{b}$. Thus, in general we obtain $|A_2| - 1$ many of the above vector inequalities. ■

A.1.2 Proof of Lemma I.4.4

Proof of Lemma I.4.4. We write $\mathcal{V}_{\pi^1,\pi^2}(R)$ for the value function under joint policy (π^1, π^2) and reward function R. The Bellman equation tells us that the value function under (π^1, π^2) and reward function $\lambda_1 R + \lambda_2 \mathbf{1} \in \mathrm{Aff}(R)$ is given by

$$\mathcal{V}_{\pi^1,\pi^2}(\lambda_1 R + \lambda_2 \mathbf{1}) = (I - \gamma \mathcal{P}_{\pi^1,\pi^2})^{-1}(\lambda_1 R + \lambda_2 \mathbf{1}).$$

Now, since $\mathcal{P}_{\pi^1,\pi^2}$ is a stochastic matrix, it is easy to check that $(I - \gamma \mathcal{P}_{\pi^1,\pi^2})^{-1}\mathbf{1} = (1-\gamma)^{-1}\mathbf{1}$. It then follows that

$$\mathcal{V}_{\pi^1,\pi^2}(\lambda_1 R + \lambda_2 \mathbf{1}) = \lambda_1 \mathcal{V}_{\pi^1,\pi^2}(R) + K,$$

where $K = \lambda_2 (1-\gamma)^{-1}\mathbf{1}$. Hence, we find that any policy π^2 that maximises $\mathcal{V}_{\pi^1,\pi^2}(R)$ also maximises $\mathcal{V}_{\pi^1,\pi^2}(\lambda_1 R + \lambda_2 \mathbf{1})$ for $\lambda_1 \geq 0$ and $\lambda_2 \in \mathbb{R}$, and vice versa. This means that R is feasible if and only if every $\bar{R} \in \mathrm{Aff}(R)$ is feasible. ■

A.1.3 Proof of Theorem I.4.5

For the proof of Theorem I.4.5, we will need the following technical lemma.

Lemma A.1.1. *Any (two-dimensional) plane $\mathcal{R} \subseteq \mathbb{R}^N$ can be uniquely characterized by the intersection of $N-1$ many half-spaces $H_i = \{x \in \mathbb{R}^N : \varphi_i^\top x \geq 0\}$, where $\varphi_1, \ldots, \varphi_{N-1} \in \mathbb{R}^N$ are vectors orthogonal to $\mathcal{R}$.*

Proof of Lemma A.1.1. W.lo.g. let $\mathcal{R}$ be some plane in $\mathbb{R}^N$ through the origin. Let the vectors $\mathcal{V}_1$ and $\mathcal{V}_2$ denote an orthogonal basis of $\mathcal{R}$, i.e. $\mathcal{R} = \{\lambda_1 \mathcal{V}_1 + \lambda_2 \mathcal{V}_2 : \lambda_1, \lambda_2 \in \mathbb{R}\}$ and $\mathcal{V}_1^\top \mathcal{V}_2 = 0$. We can then find vectors $\varphi_1, \ldots, \varphi_{N-2}$ such that $\{\varphi_1, \ldots, \varphi_{N-2}, \mathcal{V}_1, \mathcal{V}_2\}$ forms an orthogonal basis of $\mathbb{R}^N$. In particular, we then have $\varphi_i^\top x = 0$ for all $x \in \mathcal{R}$ and $i \in [N-2]$. Moreover, we define the vector

$$\varphi_{N-1} = -(\varphi_1 + \cdots + \varphi_{N-2})$$

and note that φ_{N-1} is orthogonal to $\mathcal{R}$ as well. Let the half-spaces induced by vectors $\varphi_1, \ldots, \varphi_{N-1}$ be given by $H_i = \{x \in \mathbb{R}^N : \varphi_i^\top x \geq 0\}$ for $i \in [N-1]$. We now show that $H_1 \cap \cdots \cap H_{N-1} = \mathcal{R}$.

We begin by verifying that $H_1 \cap \cdots \cap H_{N-1} \subseteq \mathcal{R}$. Suppose this is not true and there exists a vector $w \notin \mathcal{R}$ such that $\varphi_i^\top w \geq 0$ for all $i \in [N-1]$, i.e. $w \in H_1 \cap \cdots \cap H_{N-1}$. Then, we must have $\varphi_j^\top w > 0$ for some $j \in [N-2]$

as the orthogonal complement of $\operatorname{span}(\varphi_1, \dots, \varphi_{N-2})$ is given by $\mathcal{R}$ and we assumed $w \notin \mathcal{R}$. By definition of φ_{N-1}, we have $\varphi_1 + \dots + \varphi_{N-1} = 0$ and thus, $(\varphi_1 + \dots + \varphi_{N-1})^\top w = 0$. However, it also holds that

$$\varphi_1^\top w + \dots + \varphi_{N-1}^\top w > 0,$$

since $\varphi_i^\top w \geq 0$ for $i \in [N-1]$ and $\varphi_j^\top w > 0$ for some $j \in [N-2]$. Thus, such w cannot exist and we have shown that $H_1 \cap \dots \cap H_{N-1} \subseteq \mathcal{R}$. Finally, the relation $\mathcal{R} \subseteq H_1 \cap \dots \cap H_{N-1}$ also holds as $\varphi_1, \dots, \varphi_{N-1}$ are chosen orthogonal to $\mathcal{R}$ and thus, $\varphi_i^\top x = 0$ for all $i \in [N-1]$ and $x \in \mathcal{R}$.

Note that we can analogously prove that any line $\mathcal{C} = \{\lambda v : \lambda \in \mathbb{R}\}$ in $\mathbb{R}^N$ can be uniquely characterised by N half-spaces. In this case, we can find an orthogonal basis $\{\varphi_1, \dots, \varphi_{N-1}, v\}$ and define $\varphi_N = -(\varphi_1 + \dots + \varphi_{N-1})$. The remainder of the proof then follows the same line of argument as before. ■

Proof of Theorem I.4.5. Let $N = |S|$. We will now show that under the assumptions of Theorem I.4.5, there exists a policy π^1 with optimal response π^2 so that only positive affine transformations of R^* are feasible under observation (π^1, π^2), i.e. $\mathcal{R}((\pi^1, \pi^2) = \operatorname{Aff}(R^*)$.

First we observe that we can w.l.o.g. assume only two actions for $\mathcal{A}_2$, i.e. $|A_2| = 2$. To see this suppose that $|A_2| > 2$ and consider an action space $A_2' \subset A_2$ with $|A_2'| \geq 2$ and transition kernel $\mathcal{P}'_{\pi^1} : S \times A_2' \to \Delta(S)$ defined as $\mathcal{P}'_{\pi^1}(\cdot \mid s, b) = \mathcal{P}_{\pi^1}(\cdot \mid s, b)$ for $b \in A_2'$. If $\pi^2(s) \in A_2'$ for all $s \in S$, then the feasible set under action space A_2 is subset of the feasible set under action space A_2'. Thus, we can assume w.l.o.g. that $A_2 = \{b_1, b_2\}$. From hereon out, we assume that the true reward function R^* is non-constant. The special case of a constant true reward function is addressed at the end.

We first construct an orthogonal basis $\{\varphi_1, \dots, \varphi_N\}$ such that the corresponding half-spaces characterise $\operatorname{Aff}(R^*$ and then show that there exists π_1 such that

$$(\mathcal{P}_{\pi^1, b_1} - \mathcal{P}_{\pi^1, b_2})(I - \gamma \mathcal{P}_{\pi^1, b_1})^{-1} = (\varphi_1, \dots, \varphi_N)^\top.$$

For non-constant R^* we have that $\mathcal{R} \triangleq \operatorname{span}(R^*, \mathbf{1})$ describes a plane in $\mathbb{R}^N$ and $\operatorname{Aff}(R^*) \subset \mathcal{R}$. By Lemma A.1.1, there exist vectors $\varphi_1, \dots, \varphi_{N-1} \in \mathbb{R}^N$ such that $\varphi_i^\top x = 0$ for all $x \in \mathcal{R}$ and $H_1 \cap \dots \cap H_{N-1} = \mathcal{R}$ with $H_i = \{x \in \mathbb{R}^N \colon \varphi_i^\top x \geq 0\}$. In particular, it holds that $\varphi_i^\top \mathbf{1} = 0$, i.e. $\|\varphi_i\|_1 = 0$ for all $i \in [N-1]$.

Now, let us consider the orthogonal projection of R^* given by $R^* = \alpha \mathbf{1} + w$ for $\alpha \in \mathbb{R}$ and $w \in \mathbb{R}^N$ with $w^\top \mathbf{1} = 0$. It follows that $w^\top R^* = w^\top(\alpha \mathbf{1} + w) = w^\top w > 0$, since R^* is non-constant and thus, $w \neq \mathbf{0}$. Let us define $\varphi_N = \eta w$ for some scalar $\eta > 0$. Then, we have $\varphi_N^\top x \geq 0$ for all $x \in \{\lambda_1 R^* + \lambda_2 \mathbf{1} \colon \lambda \geq 0, \lambda_2 \in \mathbb{R}\}$, since $w^\top R^* > 0$ and $w^\top \mathbf{1} = 0$. Similarly, we have $\varphi_N^\top \hat{x} < 0$ for all $\hat{x} \in \{\lambda_1 R^* + \lambda_2 \mathbf{1} \colon \lambda_1 < 0, \lambda_2 \in \mathbb{R}\}$. It then follows that

$$H_1 \cap \dots \cap H_N = \mathcal{R} \cap H_N = \operatorname{Aff}(R^*),$$

where $H_N = \{x \in \mathbb{R}^N \colon \varphi_N^\top x \geq 0\}$. Note that every φ_i with $i \in [N]$ satisfies $\|\varphi_i\|_1 = 0$ and that the half-spaces H_i are invariant under positive linear

transformation of φ_i. We can therefore assume that $\varphi_1, \dots, \varphi_N$ take values in $[\frac{1}{N} - 1, \frac{1}{N}]$. We denote with $\Phi = (\varphi_1, \dots, \varphi_N)^\top$ the matrix with rows $\varphi_1, \dots, \varphi_N$.

Recall that $A_2 = \{b_1, b_2\}$. We will now show that there exists a policy π_1 such that

$$(\mathcal{P}_{\pi^1,b_1} - \mathcal{P}_{\pi^1,b_2})(I - \gamma \mathcal{P}_{\pi^1,b_1})^{-1} = \Phi.$$

By assumption, there exists a π^1 such that $\mathcal{P}_{\pi^1,b_1} \equiv B_1$ and $\mathcal{P}_{\pi^1,b_2} \equiv B_2$ for any two stochastic matrices B_1 and B_2. We set $\mathcal{P}_{\pi^1,b_1}(s' \mid s) = \frac{1}{N}$ for all $s, s' \in S$, which yields

$$\Phi(I - \gamma \mathcal{P}_{\pi^1,b_1}) = \Phi - \gamma \Phi \mathcal{P}_{\pi^1,b_1} = \Phi, \tag{2}$$

since $\|\varphi_i\|_1 = 0$ for all $i \in [N]$ and $\mathcal{P}_{\pi^1,b_1}$ is a constant matrix. Now, set $\mathcal{P}_{\pi^1,b_2} \equiv \mathcal{P}_{\pi^1,b_1} - \Phi$ and note that since $\|\varphi_i\|_1 = 0$ for all $i \in [N]$, the matrix $\mathcal{P}_{\pi^1,b_2}$ is indeed stochastic. It then follows that

$$(\mathcal{P}_{\pi^1,b_1} - \mathcal{P}_{\pi^1,b_2})(I - \gamma \mathcal{P}_{\pi^1,b_1})^{-1} = \Phi(I - \gamma \mathcal{P}_{\pi^1,b_1})^{-1} = \Phi,$$

by equation (2). Note that this means that indeed action b_1 is the optimal response to policy π^1 as $\Phi R^* \succeq 0$ by construction of Φ.[5] Therefore, from Theorem I.4.2 it follows that any feasible reward function R must satisfy

$$(\mathcal{P}_{\pi^1,b_1} - \mathcal{P}_{\pi^1,b_2})(I - \gamma \mathcal{P}_{\pi^1,b_1})^{-1} R = \Phi R \succeq 0,$$

i.e. $\varphi_i^\top R \geq 0$ for all $i \in [N]$. Hence, any feasible reward function must be in $H_1 \cap \dots \cap H_N$ and thus element in $\mathrm{Aff}(R^*)$. So, we have shown that the feasible set of reward functions under π^1 with response $\pi^2 \equiv b_1$ is given by $\mathrm{Aff}(R^*)$.

In the special case of the constant reward function R^*, we have that the set $\mathrm{Aff}(R^*) = \{\lambda \mathbf{1} : \lambda \in \mathbb{R}\}$ becomes not a plane, but a line in $\mathbb{R}^N$. The proof for this case then progresses similarly to the proof above with the difference that we describe $\mathrm{Aff}(R^*)$ by N many half-spaces and that there is no need to consider the orthogonal projection of R^* as done before.

■

A.1.4 Proof of Corollary I.4.7

Proof. Recall that it follows from Lemma I.4.4 that $\mathrm{Aff}(R^*) \subseteq \mathcal{R}((\pi^1, \pi^2))$ for any policy π^1 with optimal response π^2. In other words, the positive affine transformations of the unknown reward function R^* are always feasible as R^* is always feasible. Now, let $R \in \mathbb{R}^{|S|}$ be some reward function and suppose that $\mathcal{A}_1$ plays the "ideal" policy π^1 with respect to R as it is constructed in the proof of Theorem I.4.5. Let π^2 be an optimal response to π^1. It follows from the combination of Lemma I.4.4 and Theorem I.4.5 that $\mathcal{R}((\pi^1, \pi^2)) = \mathrm{Aff}(R)$ if and only if $R \in \mathrm{Aff}(R^*)$. Now, using linear programming, we can check whether $\mathcal{R}((\pi^1, \pi^2)) = \mathrm{Aff}(R)$ holds true. If $\mathcal{R}((\pi^1, \pi^2)) = \mathrm{Aff}(R)$, we know that R must be a positive affine transformation of R^*. On the other hand, if we observe $\mathcal{R}((\pi^1, \pi^2)) \neq \mathrm{Aff}(R)$, then R cannot be element in $\mathrm{Aff}(R^*)$.

■

[5] This can, for instance, be verified using Theorem I.4.2.

A.1.5 Proof of Lemma I.4.8

Proof of Lemma I.4.8. Let $(\bar{\pi}^1, \bar{\pi}^2) \in \operatorname{argmax}_{\pi^1,\pi^2} \mathcal{V}_{\pi^1,\pi^2}$. Suppose $\mathcal{A}_1$ commits to $\bar{\pi}^1$. Then, $\mathcal{A}_2$ responds with $\pi^2(\bar{\pi}^1)$ such that $\mathcal{V}_{\bar{\pi}^1,\pi^2(\bar{\pi}^1)} \succeq \mathcal{V}_{\bar{\pi}^1,\pi^2}$ for all π^2 by optimality of $\mathcal{A}_2$. Now, since $\mathcal{V}_{\bar{\pi}^1,\bar{\pi}^2} \succeq \max_{\pi^1} \mathcal{V}_{\pi^1,\pi^2(\pi^1)}$ always, we also have

$$\max_{\pi^1} \mathcal{V}_{\pi^1,\pi^2(\pi^1)} \succeq \mathcal{V}_{\bar{\pi}^1,\pi^2(\bar{\pi}^1)} \succeq \mathcal{V}_{\bar{\pi}^1,\bar{\pi}^2} \succeq \max_{\pi^1} \mathcal{V}_{\pi^1,\pi^2(\pi^1)}.$$

Thus, $\max_{\pi^1} \mathcal{V}_{\pi^1,\pi^2(\pi^1)} = \mathcal{V}_{\bar{\pi}^1,\bar{\pi}^2} = \max_{\pi^1,\pi^2} \mathcal{V}_{\pi^1,\pi^2}$. In other words, Lemma I.4.8 states that the optimal joint policy yields an optimal commitment strategy for $\mathcal{A}_1$ when $\mathcal{A}_2$ responds optimally. ■

A.1.6 Proof of Proposition I.4.9

Proposition A.1.1. *Suppose that for any non-constant reward function $R \in \Delta(S)$ it holds that if an optimal joint policy (π^1, π^2) under R is suboptimal under R^*, then in return there exists an optimal response $\pi^2(\pi^1)$ under R^* that is suboptimal under R. Moreover, assume that $\mathcal{A}_2$ responds optimally and breaks ties between equally good policies uniformly at random. Then, the average regret suffered by Algorithm 1 converges to zero almost surely.*

For the proof of Proposition I.4.9, we will need the following sets: Let $\Pi^{\mathtt{opt}}(R)$ denote the set of optimal joint policies under reward function R, i.e. the set of optimal joint policies in the MDP $(S, A_1, A_2, \mathcal{P}, R, \gamma)$. Further, we denote the set of optimal responses under policy π^1 and reward function R by $\Pi_2^{\mathtt{opt}}(R, \pi^1)$. A key object of interest is the following set of reward functions. Let $\mathcal{O}$ be the set of reward functions in $\Delta(S)$ that always induce an optimal joint policy, i.e.

$$\mathcal{O} = \{R \in \Delta(S) : \Pi^{\mathtt{opt}}(R) \subseteq \Pi^{\mathtt{opt}}(R^*)\}.$$

Note that by Lemma I.4.8 any optimal joint policy yields an optimal commitment strategy for agent $\mathcal{A}_1$, i.e. any $R \in \mathcal{O}$ induces an optimal commitment strategy. We can easily check that $\mathcal{O}$ is a convex set.

Lemma A.1.2. *The set $\mathcal{O}$ is convex.*

Proof of Lemma A.1.2. Let $R_1, R_2 \in \mathcal{O}$. We show that $\lambda R_1 + (1-\lambda)R_2 \in \mathcal{O}$ for any $\lambda \in [0, 1]$. Recall that the value function $\mathcal{V}_\pi(R) = (I - \gamma \mathcal{P}_\pi)^{-1} R$ is linear in R and we therefore have $\mathcal{V}_\pi(\lambda R_1 + (1-\lambda)R_2) = \lambda \mathcal{V}_\pi(R_1) + (1-\lambda)\mathcal{V}_\pi(R_2)$. In a first step, we prove $\Pi^{\mathtt{opt}}(\lambda R_1 + (1-\lambda)R_2) \subseteq \Pi^{\mathtt{opt}}(R^*)$. Let $\pi \in \Pi^{\mathtt{opt}}(\lambda R_1 + (1-\lambda)R_2)$. Then, for all policies ν it must hold that

$$\mathcal{V}_\pi(\lambda R_1 + (1-\lambda)R_2) \succeq \mathcal{V}_\nu(\lambda R_1 + (1-\lambda)R_2), \tag{3}$$

where $\succ$ denotes element-wise inequality. Now, suppose that $\pi \notin \Pi^{\mathtt{opt}}(R^*)$. It follows that $\mathcal{V}_\pi(R_1) \preceq \mathcal{V}_\nu(R_1)$ and $\mathcal{V}_\pi(R_2) \preceq \mathcal{V}_\nu(R_2)$ for some $\nu \in \Pi^{\mathtt{opt}}(R^*) = \Pi^{\mathtt{opt}}(R_1) = \Pi^{\mathtt{opt}}(R_2)$ with strict inequality for at least one $s \in S$. This contradicts equation (3) and it follows that $\Pi^{\mathtt{opt}}(\lambda R_1 + (1-\lambda)R_2) \subseteq \Pi^{\mathtt{opt}}(R^*)$. We will now verify the relation $\Pi^{\mathtt{opt}}(R^*) \subseteq \Pi^{\mathtt{opt}}(\lambda R_1 + (1-\lambda)R_2)$. For

any $\pi \in \Pi^{\texttt{opt}}(R^*)$, we have $\mathcal{V}_\pi(R_1) \succeq \mathcal{V}_\nu(R_1)$ and $\mathcal{V}_\pi(R_2) \succeq \mathcal{V}_\nu(R_2)$ for all policies ν. It then directly follows that $\pi \in \Pi^{\texttt{opt}}(\lambda R_1 + (1-\lambda)R_2)$ and thus, $\Pi^{\texttt{opt}}(R^*) \subseteq \Pi^{\texttt{opt}}(\lambda R_1 + (1-\lambda)R_2)$, i.e. $\lambda R_1 + (1-\lambda)R_2 \in \mathcal{O}$. ■

Interestingly, Lemma A.1.2 implies that the set of reward functions that induce an optimal commitment strategy is a connected set. We will now prove Proposition I.4.9.

Proof of Proposition I.4.9. As Algorithm 1 only considers reward functions in the simplex $\Delta(S)$, we will simply write $\mathcal{R}_t$ instead of $\mathcal{R}_t \cap \Delta(S)$ for notational convenience.

In episode t, Algorithm 1 chooses a vertex of the set of feasible solutions of the linear program, i.e. a reward function $R_t \in \mathcal{R}_t$. Note that by construction of Algorithm 1 we never select the constant reward function in $\Delta(S)$. For any $R_t \in \mathcal{R}_t$ obtained from the LP (I.1) with uniformly random objective function c there are two possible cases: $R_t \in \mathcal{O}$ or $R_t \notin \mathcal{O}$. If $R_t \in \mathcal{O}$, then R_t induces an optimal joint policy, i.e. an optimal commitment strategy by Lemma I.4.8. Accordingly, Algorithm 1 commits to an optimal commitment strategy and thus suffers zero regret in episode $t+1$. We want to highlight that the proof does not require that the objective function in Algorithm 1 is being chosen in a randomised fashion. However, randomising the choice of the objective improved exploration in our experiments.

In the following, we show that for the case of $R_t \notin \mathcal{O}$, Algorithm 1 strictly decreases the set of feasible reward functions with positive probability. In order to show this, we first construct a finite cover of $\Delta(S)$. Let Π_1 and Π_2 denote the sets of deterministic policies for $\mathcal{A}_1$ and $\mathcal{A}_2$, respectively.[6] Note that both Π_1 and Π_2 are finite as we assumed finite action spaces A_1 and A_2. Let 2^{Π_2} denote the power set of Π_2. For $\pi^1 \in \Pi_1$ and $\bar{\Pi}_2 \in 2^{\Pi_2}$, we define

$$B(\pi^1, \bar{\Pi}_2) = \{R \in \Delta(S) : \bar{\Pi}_2 = \Pi_2^{\texttt{opt}}(R, \pi^1)\}.$$

The set $B(\pi^1, \bar{\Pi}_2)$ thus describes the reward functions that make the policies in $\bar{\Pi}_2$ optimal in response to π^1. Indeed, for any fixed $\pi^1 \in \Pi_1$, the collection $\mathcal{B}(\pi^1) = \{B(\pi^1, \bar{\Pi}_2) : \bar{\Pi}_2 \in 2^{\Pi_2}\}$ forms a finite partition of $\Delta(S)$

$$\bigcup\nolimits_{\bar{\Pi}_2 \in 2^{\Pi_2}} B(\pi^1, \bar{\Pi}_2) = \Delta(S),$$

as for any $R \in \Delta(S)$ there always exists at least one deterministic optimal policy in the MDP $(S, A_2, \mathcal{P}_{\pi^1}, R, \gamma)$ [Put14]. In other words, for any $\pi^1 \in \Pi_1$, we partition $\Delta(S)$ into sets that induce the same set of optimal responses to π^1. Naturally, due to $\mathcal{B}(\pi^1)$ being a *finite* partition of $\Delta(S)$ for any π^1, the Lebesgue-measure for all but finitely many $B(\pi^1, \bar{\Pi}_2)$ must be larger than some constant $\varepsilon > 0$.

[6] We assume here that $\mathcal{A}_2$ responds with deterministic policies in order to keep the proof as comprehensible as possible. However, this assumption can be dropped as we can still give a finite partition of $\Delta(S)$ when $\mathcal{A}_2$ also responds with optimal stochastic policies.

We now show that if $R_t \notin \mathcal{O}$, then with positive probability the set of feasible solutions is decreased by at least ε. If $R_t \notin \mathcal{O}$, then Algorithm 1 computes an optimal commitment strategy $\pi^1_{t+1} \in \Pi^{\mathtt{opt}}_1(R_t)$ (by computing the optimal joint policy under R_t, see Lemma I.4.8), which may be suboptimal under R^*, i.e. $\pi^1_{t+1} \notin \Pi^{\mathtt{opt}}_1(R^*)$.

Now, if π^1_{t+1} is suboptimal under R^*, then by assumption[7] there exists an optimal response $\pi^2_{t+1} \in \Pi^{\mathtt{opt}}_2(R^*, \pi^1_{t+1})$ that is suboptimal under R_t, i.e. $\pi^2_{t+1} \notin \Pi^{\mathtt{opt}}_2(R_t, \pi^1_{t+1})$. Recall that by our assumption $\mathcal{A}_2$ selects its response uniformly at random from $\Pi^{\mathtt{opt}}_2(R^*, \pi^1_{t+1})$. Since $\Pi^{\mathtt{opt}}_2(R^*, \pi^1_{t+1})$ is finite, $\mathcal{A}_2$ will respond with $\pi^2_{t+1} \notin \Pi^{\mathtt{opt}}_2(R_t, \pi^1_{t+1})$ with positive probability.

In that case, after observing π^2_{t+1} the reward function R_t cannot be feasible anymore, i.e. $R_t \notin \mathcal{R}_{t+1}$. In addition, we then also have that $B(\pi^1_{t+1}, \Pi^{\mathtt{opt}}_2(R_t, \pi^1_{t+1})) \cap \mathcal{R}_{t+1} = \emptyset$, as all reward functions in $B(\pi^1_{t+1}, \Pi^{\mathtt{opt}}_2(R_t, \pi^1_{t+1}))$ induce the same optimal responses $\Pi^{\mathtt{opt}}_2(R_t, \pi^1_{t+1})$ and π^2_{t+1} is not in $\Pi^{\mathtt{opt}}_2(R_t, \pi^1_{t+1})$. In other words, any $R \in B(\pi^1_{t+1}, \Pi^{\mathtt{opt}}_2(R_t, \pi^1_{t+1}))$ cannot satisfy the constraints of Corollary I.4.3.

As seen before, for all but finitely many $\bar{\Pi}_2 \in 2^{\Pi_2}$ we have $\lambda(B(\pi^1, \bar{\Pi}_2)) > \varepsilon$, where λ is the Lebesgue-measure. As a consequence, if $R_t \notin \mathcal{O}$, then we have for all but finitely many cases that $\lambda(\mathcal{R}_{t+1}) \leq \lambda(\mathcal{R} \setminus B(\pi^1_{t+1}, \Pi^{\mathtt{opt}}_2(R_t, \pi^1_{t+1})) \leq \lambda(\mathcal{R}_t) - \varepsilon$.

Therefore, every time when Algorithm 1 chooses a reward function $R_t \notin \mathcal{O}$[8] inducing a suboptimal commitment strategy, (with positive probability) R_t will not be feasible anymore and (except for finitely many times) we reduce the size of the feasible set by at least the constant amount ε. As a result, the feasible set of reward function $\mathcal{R}_t$ will eventually become smaller than or equal to $\mathcal{O}$, i.e. $\mathcal{R}_t \subseteq \mathcal{O}$. Consequently, Algorithm 1 will almost surely converge to choosing only reward function in $\mathcal{O}$ and will thus only play optimal commitment strategies. ■

A.1.7 Proof of Theorem I.5.2

Proof of Theorem I.5.2. We provide a problem instance for which there exists no dominating policy for any strictly increasing function $f : [0, \infty) \to [0, \infty)$. Consider the two-agent MDP in Figure A.1.5. We omitted consecutive transitions in Figure A.1.5, but assume that states s_1, s_3, and s_4 lead to the same (terminal) state with probability one.

We will show that the strictly optimal policy when in state s_0 is strictly suboptimal when in state s_2 for specific choices of $x > 0$ and $y > 0$. For simplicity, we omit the discount factor γ in the following.

$\mathcal{A}_1$ only influences transitions in state s_2 and thus there are essentially only two deterministic policies for $\mathcal{A}_1$, namely π^1 with $\pi^1(s_2) = a_1$ and $\bar{\pi}^1$ with $\bar{\pi}^1(s_2) = a_2$. Since $y > 0$, action a_1 is optimal in state s_2 and so π^1 is the

[7] Note that if π^1 is a suboptimal commitment strategy, then the joint policy (π^1, π^2) is suboptimal for any π^2.

[8] Recall that the special case of the constant reward function (which is not in $\mathcal{O}$) can be ignored.

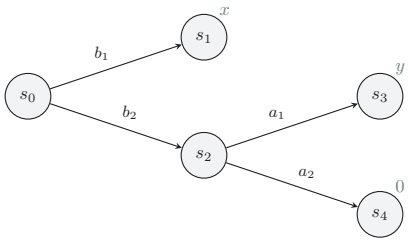


Figure A.1.5: Counterexample. All transitions are deterministic. The action of $\mathcal{A}_2$ alone determines the transitions from state s_0 to states s_1 and s_2, whereas in state s_2 only the action of $\mathcal{A}_1$ affects transitions. The green x, y and 0 denote the rewards obtained in states s_1, s_3, and s_4, respectively. States s_0 and s_2 yield zero reward.

optimal policy in state s_2. We now show that there exists $x, y > 0$ such that $\mathcal{V}_{\pi^1,\pi^2(\pi^1)}(s_0) < \mathcal{V}_{\bar{\pi}^1,\pi^2(\bar{\pi}^1)}(s_0)$, i.e. $\bar{\pi}^1$ is strictly better than π^1 when in state s_0.

Omitting the discount factor, we have $Q^*_{\pi^1}(s_0, b_1) = x$ and $Q^*_{\pi^1}(s_0, b_2) = y$ as well as $Q^*_{\bar{\pi}^1}(s_0, b_1) = x$ and $Q^*_{\bar{\pi}^1}(s_1, b_2) = 0$. We therefore want to show that there exist $x, y > 0$ such that

$$\begin{aligned}\mathcal{V}_{\pi^1,\pi^2(\pi^1)}(s_1) &= x\,\frac{f(x)}{f(x)+f(y)} + y\,\frac{f(y)}{f(x)+f(y)} \\ &< x\,\frac{f(x)}{f(x)+f(0)} = \mathcal{V}_{\bar{\pi}^1,\pi^2(\bar{\pi}^1)}(s_1).\end{aligned}$$

Suppose the contrary is true. Then, for all $x, y > 0$ it must hold that

$$\begin{aligned}& x\,\frac{f(x)}{f(x)+f(y)} + y\,\frac{f(y)}{f(x)+f(y)} \geq x\,\frac{f(x)}{f(x)+f(0)} \\ \Leftrightarrow\ & x\left(\frac{f(x)}{f(x)+f(0)} - \frac{f(x)}{f(x)+f(y)}\right) \leq y\,\frac{f(y)}{f(x)+f(y)} \\ \Leftrightarrow\ & xf(x)\left(\frac{f(x)+f(y)}{f(x)+f(0)} - 1\right) \leq yf(y) \\ \Leftrightarrow\ & xf(x)\,\frac{f(y)-f(0)}{f(x)+f(0)} \leq yf(y). \qquad (4)\end{aligned}$$

Note that $f(y) - f(0) > 0$, since f is strictly increasing. Now, for any fixed $y > 0$, we have that $f(x)\frac{f(y)-f(0)}{f(x)+f(0)} \to 1$ as $x \to \infty$, and the expression is therefore bounded from below by some positive value for x sufficiently large. Hence, for any fixed y there exists an $x > 0$ such that (4) does not hold. This shows that in fact for any $y > 0$ there exists $x > 0$ such that $\mathcal{V}_{\pi^1,\pi^2(\pi^1)}(s_0) < \mathcal{V}_{\bar{\pi}^1,\pi^2(\bar{\pi}^1)}(s_0)$, whereas we have seen before that $\mathcal{V}_{\pi^1,\pi^2(\pi^1)}(s_2) > \mathcal{V}_{\bar{\pi}^1,\pi^2(\bar{\pi}^1)}(s_2)$. Hence, no dominating commitment strategy exists for the MDP depicted in Figure A.1.5. ■

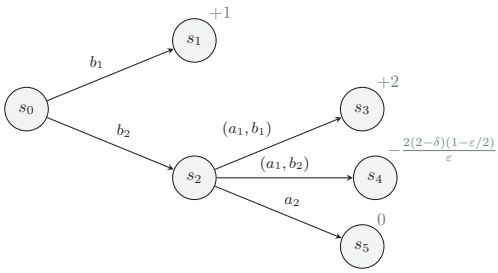


Figure A.1.6: Counterexample for ε-greedy responses. All transitions are deterministic. The actions from agent $\mathcal{A}_2$ alone determine the transitions from state s_0 to states s_1 and s_2. The green numbers denote the rewards obtained in the respective states. States s_0 and s_2 yield zero reward.

A.1.8 Proof of Lemma I.5.3

We define an ε-greedy response to a policy π^1 as the policy

$$\pi^2_\varepsilon(s, \pi^1) = \begin{cases} \pi^2_*(s, \pi^1) & w.p.\ 1-\varepsilon \\ \mathcal{U}(A_2) & w.p.\ \varepsilon, \end{cases}$$

where $\varepsilon \in [0, 1]$, $\pi^2_*(\pi^1)$ is an optimal response to π^1, and $\mathcal{U}(A_2)$ the uniform distribution over A_2.

Proof of Lemma I.5.3. We prove Lemma I.5.3 by means of the counterexample shown in Figure A.1.6. For convenience, we omit the discount factor here and assume that states s_1, s_3, s_4, and s_5 lead to some terminal state with probability one. There are two (deterministic) policies $\mathcal{A}_1$ can commit to: $\pi^1(s_2) = a_1$ and $\bar{\pi}^1(s_2) = a_2$.

For notational convenience, we write $\mathcal{V}_{a_1}(s) \triangleq \mathcal{V}_{\pi^1, \pi^2_\varepsilon(\pi^1)}(s)$ and $\mathcal{V}_{a_2}(s) \triangleq \mathcal{V}_{\bar{\pi}^1, \pi^2_\varepsilon(\bar{\pi}^1)}(s)$. Note that if $\mathcal{A}_1$ commits to π^1, the optimal action for $\mathcal{A}_2$ in state s_0 is to play b_2 followed by b_1 in state s_2. Recall that $\mathcal{A}_2$ is assumed to play ϵ-greedy, i.e. in any state, $\mathcal{A}_2$ plays the optimal response with probability $(1-\epsilon)$ and with probability ϵ selects an action uniformly at random. As a result, we have

$$\begin{aligned} \mathcal{V}_{a_1}(s_2) &= 2(1-\varepsilon/2) - (2-\delta)(1-\varepsilon/2) = \delta(1-\varepsilon/2) > 0 \\ \mathcal{V}_{a_1}(s_0) &= \delta(1-\varepsilon/2)^2 + \varepsilon/2. \end{aligned}$$

On the other hand, if $\mathcal{A}_1$ commits to $\bar{\pi}^1$, it is optimal for $\mathcal{A}_2$ to play b_1 in state s_0, i.e. $\mathcal{V}_{a_2}(s_0) = (1-\varepsilon/2)$. We observe that in state s_2, playing a_1 is optimal as $\mathcal{V}_{a_1}(s_2) > \mathcal{V}_{a_2}(s_2) = 0$. However, we also have $\mathcal{V}_{a_1}(s_0) - \mathcal{V}_{a_2}(s_0) = \varepsilon + \delta(1-\varepsilon/2)^2 - 1$. As we can choose δ arbitrarily close to 0, we then have $\mathcal{V}_{a_1}(s_0) < \mathcal{V}_{a_2}(s_0)$ for some $\delta > 0$. Thus, π^1 is strictly optimal in state s_2, whereas $\bar{\pi}^1$ is strictly optimal in state s_0. Therefore, there exists no dominating commitment strategy for the MDP in Figure A.1.6.

■

A.2 Approximate Algorithms for Cooperative Stackelberg Games with Suboptimal Followers

In this section, we first describe approximate value iteration algorithms for Boltzmann-rational policies as well as ε-greedy policies. We then evaluate both algorithms in the Maze-Maker and Random MDP environment for different levels of rationality (i.e. optimality) of agent $\mathcal{A}_2$.

A.2.1 $\mathcal{A}_2$ responds with Boltzmann-rational policies

Theorem I.5.2 states that no dominating commitment strategy may exist when agent $\mathcal{A}_2$ responds with Boltzmann-rational policies. In its essence, the approximate value iteration algorithm for Boltzmann-rational responses described in Algorithm 2 acts as if a dominating commitment strategy does exist and could therefore converge to suboptimal solutions. However, it aims to account for the suboptimality of agent $\mathcal{A}_2$ and keeps track of two sets of value functions: one value function corresponding to what $\mathcal{A}_1$ believes to be the actual value given that $\mathcal{A}_2$ plays Boltzmann, and one value function that aims to approximate the belief of agent $\mathcal{A}_2$ about the value of the game.

Algorithm 2 Approximate Value Iteration for Boltzmann-Rational Responses

initialise V and $\hat{V}$
repeat until V converges:
for $s \in S$ **do**
  for $(a, b) \in A_1 \times A_2$ **do**
    $\hat{Q}(s, a, b) = R(s) + \gamma \sum_{s'} \mathcal{P}(s'|s, a, b)\hat{V}(s')$
    $\pi^2(b \mid s, a) = \exp(\beta \hat{Q}(s, a, b))/Z$
  $\pi^1(s) = \operatorname{argmax}_a \sum_{s'} \mathbb{E}_{b\sim\pi^2}[\mathcal{P}(s'|s, a, b)]V(s')$
  $V(s) = R(s) + \gamma \sum_{s'} \mathbb{E}_{b\sim\pi^2}[\mathcal{P}(s'|s, \pi^1(s), b)]V(s')$
  $\hat{V}(s) = \max_b \hat{Q}(s, \pi^1(s), b)$

A.2.2 $\mathcal{A}_2$ responds with ε-greedy policies

The problem of planning with an agent that responds with ε-greedy policies is similar to the setting considered by [Dim+17] in the sense that $\mathcal{A}_2$ plans with the original transition kernel $\mathcal{P}$ (by computing an optimal response $\pi^2_*(\pi^1)$), whereas $\mathcal{A}_1$ plans (or should plan) with the "correct" transition kernel

$$\mathcal{P}_\varepsilon(\cdot \mid s, a, b) \equiv \varepsilon\mathcal{P}(\cdot \mid s, a, \mathcal{U}(A_2)) + (1 - \varepsilon)\mathcal{P}(\cdot \mid s, a, b).$$

In particular, note that $\varepsilon\mathcal{P}(s' \mid s, a, \mathcal{U}(A_2))$ is independent of the choice of b. Algorithm 3 approximately solves the planning problem. While Lemma I.5.3 states that a dominating commitment policy need not exist, Algorithm 3 simply acts as if one exists. Similarly to Algorithm 2, the idea is to maintain two value functions, one representing the value from the perspective of $\mathcal{A}_1$ and the other the value from the perspective of $\mathcal{A}_2$.

Algorithm 3 Approximate Value Iteration for ε-Greedy Responses

initialise V and $\hat{V}$
repeat until V converges:
for $s \in S$ **do**
  for $a \in A_1$ **do**
    $\pi^2(s,a) = \text{argmax}_b \sum_{s'} \mathcal{P}(s'|s,a,b)\hat{V}(s')$
  $\pi^1(s) = \text{argmax}_a \sum_{s'} \mathbb{E}_{b\sim\pi^2}[\mathcal{P}_\varepsilon(s'|s,a,b)]V(s')$
  $V(s) = R(s) + \gamma \sum_{s'} \mathbb{E}_{b\sim\pi^2}[\mathcal{P}_\varepsilon(s'|s,\pi^1(s),b)]V(s')$
  $\hat{V}(s) = R(s) + \gamma \sum_{s'} \mathbb{E}_{b\sim\pi^2}[\mathcal{P}(s'|s,\pi^1(s),b)]\hat{V}(s')$

A.2.3 Evaluation of Algorithm 2 and Algorithm 3

In this section, we empirically evaluate our approximate value iteration algorithms for Boltzmann-rational responses (Algorithm 2) and ε-greedy responses (Algorithm 3). We compare Algorithm 2 and Algorithm 3 in the Maze-Maker and Random MDP environment against committing $\mathcal{A}_1$'s part of the optimal joint policy. Note that by Lemma I.4.8, committing $\mathcal{A}_1$'s part of an optimal joint policy is optimal when $\mathcal{A}_2$ responds optimally.

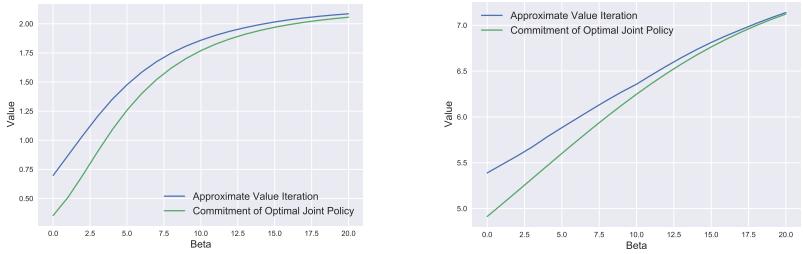
In both environments, we test the performance of our algorithms for different levels of rationality of $\mathcal{A}_2$. For the case of Boltzmann-rational responses (Figure A.2.7), we increase the inverse temperature of agent $\mathcal{A}_2$, which corresponds to the rationality (i.e. optimality) of $\mathcal{A}_2$. We see in Figure A.2.7 that Algorithm 2 consistently outperforms playing $\mathcal{A}_1's$ part of the optimal joint policy. In particular, the more suboptimal $\mathcal{A}_2$ is playing (lower values of β), the larger the advantage of Algorithm 2 is compared to playing $\mathcal{A}_1$'s part of the optimal joint policy. If $\mathcal{A}_2$ responds almost optimally ($\beta = 20$), the performance of both approaches is almost identical as expected.

For the case of ε-greedy responses (Figure A.2.8), we increase the rationality of $\mathcal{A}_2$ by decreasing the probability ε of random actions. Figure A.2.8 shows that Algorithm 3 outperforms playing the optimal joint policy for all values of ε in both environments. In particular, for $\varepsilon = 0$ agent $\mathcal{A}_2$ responds optimally and both approaches play an optimal commitment strategy.

A.3 Experimental Details

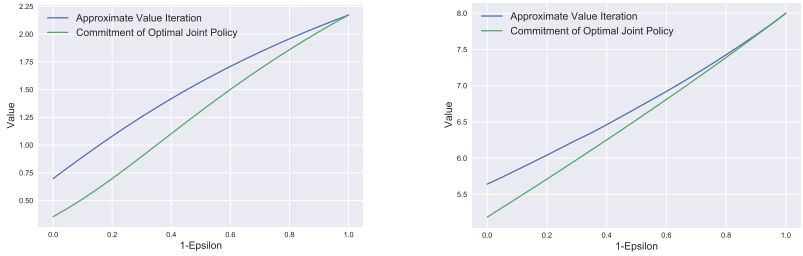
The experiments were carried out on a virtual machine with 32 CPUs, 60GB RAM, and CentOS Linux 8 operating system. The experiments were implemented in Python 3.7 and the libraries matplotlib 3.2.1, numpy 1.20.1, and scipy 1.6.2 (for the linear program) were used. The code is available at https://github.com/InteractiveIRL/src.

For the case of suboptimal responses and partial information, we assume that $\mathcal{A}_2$ responds with Boltzmann-rational policies with inverse temperature $\beta = 10$ in both environments. We assume that the inverse temperature, that is, the optimality of the second agent, is *unknown* to the learner and must therefore be



(a) Maze-Maker (b) Random MDPs

Figure A.2.7: Evaluation of Approximate Value Iteration for Boltzmann-Rational Responses (Algorithm 2) in the Maze-Maker and Random MDP environment for increasing values of β. The green line describes the return of playing $\mathcal{A}_1$'s part of an optimal joint policy.



(a) Maze-Maker (b) Random MDPs

Figure A.2.8: Evaluation of Approximate Value Iteration for ε-Greedy Responses (Algorithm 3) in the Maze-Maker and Random MDP environment for decreasing values of ε. The green line describes the return of playing $\mathcal{A}_1$'s part of an optimal joint policy.

inferred. We simulate the partial information setting by generating trajectories according to policies π_t^1 and π_t^2 in episode t, where the length of the episode is random. More precisely, we let an episode end with probability $1 - \gamma = 0.1$ each time step.[9]

A.3.1 Bayesian Interactive IRL

We employ a Bayesian approach using the Metropolis-Hastings algorithm to sample from the posterior, with a uniform prior on the reward function and an exponential prior on the inverse temperature. Our approach is specified in Algorithm 4. As a proposal distribution for the reward function, we consider a discretisation of the $|S|$-dimensional unit simplex $\Delta(S)$ with step size δ, similarly to [RA07]. The Metropolis-Hastings algorithm then generates a Markov chain

[9]We impose a minimal trajectory length of 2 time steps to prevent vacuous episodes.

Algorithm 4 Bayesian Interactive IRL via Simplex Walk

input: $(S, A_1, A_2, \mathcal{P}, \gamma)$, priors $\mathbb{P}(R)$, $\mathbb{P}(\beta)$, proposal distributions g_1, g_2, sample size K
initialise: choose π_1^1 uniformly at random, sample $R_0^0 \sim \mathbb{P}(R)$ and $\beta_0^0 \sim \mathbb{P}(\beta)$

for $t = 1, 2, \ldots$ **do**
  commit to policy π_t^1
  observe trajectory τ_t
  // sample from posterior via Metropolis-Hastings
  for $k = 1, \ldots, K$ **do**
    sample $R \sim g_1(\cdot \mid R_{k-1}^t)$
    sample $\beta \sim g_2(\cdot \mid \beta_{k-1}^t)$
    compute $p = \frac{\mathbb{P}((\pi_1^1,\tau_1),\ldots,(\pi_t^1,\tau_t)|R,\beta)\mathbb{P}(R)\mathbb{P}(\beta)}{g_1(R|R_{k-1}^t)g_2(\beta|\beta_{k-1}^t)}$
    w.p. $\min\{1, \frac{p}{p_{k-1}}\}$**:** $R_k^t = R$, $\beta_k^t = \beta$, $p_k^t = p$
    else: $R_k^t = R_{k-1}$, $\beta_k^t = \beta_{k-1}$, $p_k^t = p_{k-1}^t$
  set $R_0^{t+1} = R_K^t$, $\beta_0^{t+1} = \beta_K^t$, $p_0^{t+1} = p_K^t$
  calculate mean reward function $\bar{R}_t$ and beta $\bar{\beta}_t$
  compute π_{t+1}^1 under $\bar{R}_t$ and $\bar{\beta}_t$ via Algorithm 2

on the discretised simplex. To sample from the posterior given the last candidate R_{k-1}^t then means to choose a neighbour in the discretised simplex. This type of proposal distribution, which we refer to as Simplex Walk, proved to be a more efficient and robust sampling strategy as other proposal distributions (e.g. Dirichlet distributions). For the inverse temperature, we use a Gamma proposal distribution. Similarly to Algorithm 1, we play greedily with respect to our current estimate of the true reward function. After sampling K times from the posterior, we take the empirical means $\bar{R}_t$ and $\bar{\beta}_t$ and compute an approximately optimal commitment strategy under $\bar{R}_t$ and $\bar{\beta}_t$ by means of Algorithm 2. As a natural burn-in we use the last sampled reward and inverse temperature from episode t as the first candidate in episode $t + 1$.

A.3.2 Environments: Maze-Maker

In the Maze-Maker environment, agents $\mathcal{A}_1$ and $\mathcal{A}_2$ jointly control a cart in a 7×7 grid world. In this grid world, the doors leading from one cell to the neighbouring ones are locked. However, $\mathcal{A}_1$ can unlock exactly two doors at any time step before they fall shut again. $\mathcal{A}_2$ can attempt to move the cart through a door to a neighbouring cell. However, when the door is locked, the cart stays where it was. We assume that any attempted move of the cart succeeds with probability 0.8 and that with probability 0.2 the cart moves to a random neighbouring cell. Agents $\mathcal{A}_1$ and $\mathcal{A}_2$ are tasked with collecting three rewards of different value (+1, +2, +3), which are scattered in the grid world and disappear once collected. While $\mathcal{A}_2$ knows where the rewards are placed, $\mathcal{A}_1$ does not know

their location. An illustration of the environment is given by Figure I.2. We model this environment as a two-agent MDP with 392 states (49×8) and discount factor $\gamma = 0.9$, where $\mathcal{A}_1$ has six actions (unlocking two out of four doors) and $\mathcal{A}_2$ four actions (attempting to move the cart North, East, South, West). As we consider a Stackelberg game, $\mathcal{A}_2$ knows beforehand which doors $\mathcal{A}_1$ will unlock. Therefore, $\mathcal{A}_1$ essentially selects a maze layout, which is communicated to $\mathcal{A}_2$ and through which $\mathcal{A}_2$ can move the cart.

A.3.3 Details on Figure I.1

In Figure I.1b, we assumed that $\mathcal{A}_2$ plays a Boltzmann-rational policy with inverse temperature $\beta = 10$. For simplicity and proper comparison, we assume that we can observe the fully specified Boltzmann policy played by $\mathcal{A}_2$ in each of the mazes. We use an adaption if Bayesian IRL [RA07] and display the mean reward function in Figure I.1b, where the colour scale, i.e. colour transparency, is obtained from the mean reward function in a given cell. More precisely, we use the Metropolis-Hastings algorithm with uniform prior and a Dirichlet proposal to sample from the posterior distribution $\mathbb{P}(R \mid (\pi^1, \pi^2))$, where π^1 describes the maze layout.

A.4 Influence

Prior work on two-agent cooperation has considered measurements of how much one agent can influence the transition probabilities. [Dim+17] define the influence of agent $\mathcal{A}_1$ (analogously for $\mathcal{A}_2$) on the transition probabilities as

$$\mathcal{I}(\mathcal{A}_1) = \max_{s} \max_{a_1, a_2, b} \|\mathcal{P}(\cdot \mid s, a_1, b) - \mathcal{P}(\cdot \mid s, a_2, b)\|_1,$$

which has also been adopted by [Rad+19] and [Gho+19]. They use this definition of influence to bound the performance gap when the beliefs or the behaviour of the two agents are misaligned. In our setting, however, the influence of an agent also relates to the IRL problem and our capacity to solve it. In particular, if $\mathcal{I}(\mathcal{A}_1) = 0$, agent $\mathcal{A}_1$ does not influence the transition probabilities and it is therefore irrelevant what actions $\mathcal{A}_1$ takes. In terms of the IRL problem, we are then in the typical single-agent setting as $\mathcal{A}_2$ can ignore the presence of agent $\mathcal{A}_1$. On the other hand, if $\mathcal{I}(\mathcal{A}_2) = 0$, then $\mathcal{A}_2$ does not influence transitions at all and the IRL problem becomes intractable as $\mathcal{A}_2$'s actions yield no information about the underlying reward function.

Paper II

Environment Design for Inverse Reinforcement Learning

Thomas Kleine Buening, Christos Dimitrakakis

Presented in the *Human in the Loop Learning Workshop at NeurIPS*, 2022.

II

Abstract

The task of learning a reward function from expert demonstrations suffers from high sample complexity as well as inherent limitations to what can be learned from demonstrations in a given environment. As the samples used for reward learning require human input, which is generally expensive, much effort has been dedicated towards designing more sample-efficient algorithms. Moreover, even with abundant data, current methods can still fail to learn insightful reward functions that are robust to minor changes in the environment dynamics. We approach these challenges differently than prior work by improving the sample-efficiency as well as the robustness of learned rewards through adaptively designing a sequence of demonstration environments for the expert to act in. We formalise a framework for this environment design process in which learner and expert repeatedly interact, and construct algorithms that actively seek information about the rewards by carefully curating environments for the human to demonstrate the task in.

II.1 Introduction

Reinforcement Learning (RL) has proven to be a powerful framework for autonomous decision-making in games [Mni+15], continuous control problems [Lil+15], and robotics [Lev+16]. However, the challenge of specifying suitable reward functions remains one of the main barriers to the wider application of reinforcement learning in real-world settings. To this end, methods that allow us to communicate tasks without manually defining such reward functions could be of great practical value. One of such approaches is Inverse Reinforcement Learning (IRL), which aims to find a reward function that explains observed (human) behaviour [NR00; Rus98].

Much of the progress and recent efforts in IRL have been devoted to making existing methods more sample-efficient as well as robust to changes in the environment dynamics [AD21; FLL18]. Sample-efficiency is crucial for practical applications of IRL as the data used for learning requires human input, which

is typically expensive. Moreover, inferring robust estimates of the unknown reward function that induce near-optimal policies across slight variations of the original environment is paramount for ensuring the safeness and the success of autonomous agents in real-world scenarios.

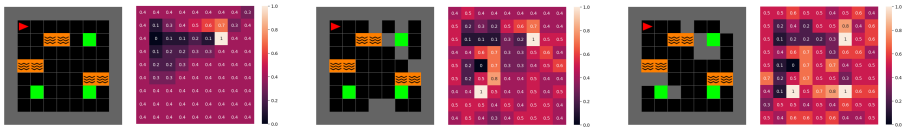
However, recent work has found that IRL methods tend to overfit to the specific transition dynamics under which the demonstration were provided, thereby failing to generalise even across minor changes in the environment [Toy+20]. More generally, even with unlimited access to expert demonstrations, we may still fail to learn suitable reward functions from a fixed environment. In particular, prior work has explored the identifiability problem in IRL [CCS21; Kim+21], illustrating the inherent limitations of IRL when learning from expert demonstrations in a single environment.

We address these challenges differently than prior work. Instead of trying to improve upon existing IRL methods directly, we aim to improve the data generation process by actively seeking information from the human expert by designing a sequence of demonstration environments. Our hypothesis is that intelligently choosing such demo environments will allow us to improve the sample-efficiency of IRL methods and the robustness of learned rewards against variations in the environment dynamics.

We consider the situation when there is a known set of demo environments in which the expert could potentially demonstrate the task in. Often this set is given by variants of some base environment. For example, when the task is to navigate to a goal state without crossing dangerous states, the set of demo environments could be given by the original world layout with obstacles being added, moved, or removed. We propose an environment design approach based on minimax Bayesian regret that aims to select demo environment so as to discover all performance-relevant aspects of the unknown reward function. An example of the environments generated by this approach is illustrated in Figure II.1.

Outline. After discussing related work in Section II.2, we will formally establish our framework of *Environment Design for Inverse Reinforcement Learning* in Section II.3. In Section II.4 we then propose an environment design approach based on a minimax Bayesian regret objective and explain how to compute demo environments efficiently when the set of environments exhibits useful structure. Section II.5 extends Bayesian IRL methods to the setting of learning from demonstrations in multiple environments. Finally, we perform a preliminary set of experiments in Section II.6 with the goal of evaluating the benefits of carefully curating the set of demo environments for reward learning.[1]

[1] In this preliminary version of this work, we will focus on the Bayesian formulation of the problem. We will briefly comment on how to extend this work to non-Bayesian IRL frameworks such as Maximum Entropy IRL in the Appendix. However, we defer extensive discussion and evaluation of this to a future version of this work.



(a) 1st round (b) 2nd round (c) 3rd round

Figure II.1: The expert navigates to three possible goal states while avoiding lava in adaptively designed maze environments. For three consecutive rounds (a)-(c), we display the mazes constructed by `ED-BIRL` as well as the estimated reward functions after observing an expert trajectory in the current and past mazes. By adaptively designing environments and combining the expert demonstrations, we can recover the locations of goal states and lava states. In contrast, from observations in a fixed environment, e.g. repeatedly observing the expert in maze (a), it would be impossible to recover all relevant aspects of the reward function, i.e. goal states, as only the closest goal state will be visited by the expert (repeatedly). Observing the human expert's actions in new and carefully constructed environments can thus lead to a more precise and robust estimate of the unknown reward function.

II.2 Related Work

(Active) IRL. The goal of IRL [NR00; Rus98] is to find a reward function that explains observed behaviour, which is assumed to be approximately optimal. Two of the most popular approaches to the IRL problem are Bayesian IRL [CK11; RA07; RD11] and Maximum Entropy IRL [FLA16; HE16; Zie+08]. In this work, we focus on extending the Bayesian IRL formulation to demonstrations in multiple environments as it provides a principled way to reason under reward uncertainty. This is also the typical IRL formulation under which Active IRL has been addressed in prior work.

In particular, the environment design problem that we consider can be viewed as one of active reward elicitation [LMM09]. Prior work on active reward learning has focused on querying the expert for additional demonstrations in specific states [BCN18; Lin+21; LMM09], mainly with the goal of resolving the uncertainty that is due to the expert's policy not being specified accurately in these states. In contrast, we consider the situation where we cannot directly query the expert for additional information in specific states, but instead sequentially choose demo environments for the expert to act in. Importantly, in our setting, the same state can be visited under different transition dynamics, which can be crucial to distinguish between two plausible reward functions. Hereto related, [AJS17] consider a repeated IRL setting in which the learner can choose *any* task for the expert to complete (with full information of the expert policy). Recently, [BGD22] also introduced Interactive IRL in which the learner interacts with a human in a collaborative Stackelberg game without knowledge of the joint reward function. This setting is similar to the framework presented here in that the leader in a Stackelberg game can be viewed as designing environments by

committing to specific policies.

Environment Design for Reinforcement Learning. Environment Design and Curriculum Learning for RL aim to design a sequence of environments with increasing difficulty to improve the training of an autonomous agent [Nar+20]. However, in contrast to our problem setup, observations in a generated training environments are cheap, as this only involves actions from an autonomous agent, not a human expert. As such, approaches like domain randomisation [Akk+19; Tob+17] can be practical for RL, whereas they can be extremely inefficient and wasteful in an IRL setting. Moreover, in IRL we typically work with a handful of rounds only, so that slowly improving the environment generation process over thousands of training episodes (i.e. rounds) is impractical [Den+20; Gur+21]. Finally, we also have to deal with the additional challenge of not knowing the true reward function according to which the expert is going to act, which makes reliably predicting the expert's behaviour in an environment difficult.

II.3 Problem Formulation

We now formally introduce the Environment Design for Inverse Reinforcement Learning framework. A Markov Decision Process (MDP) is a tuple $(S, \mathcal{A}, \mathcal{P}, R^*, \gamma, \omega)$, where S is a set of states, $\mathcal{A}$ is a set of actions, $\mathcal{P} : S \times \mathcal{A} \times S \to [0, 1]$ is a transition function, $R^* : S \to \mathbb{R}$ is a reward function, γ a discount factor, and ω an initial state distribution. We assume that there is a set transition functions $\mathcal{T}$ from which $\mathcal{P}$ can be selected. Similar models have been considered for the RL problem under the name of Underspecified MDPs [Den+20] or Configurable MDPs [MMR18; Ram+21].

We assume that the true reward function, denoted R^*, is unknown to the learner and consider the situation where the learning agent gets to interact with the human expert in a sequence of m rounds.[2] More precisely, every round $k \in [m]$, the learner gets to select a demo environment $\mathcal{P}_k \in \mathcal{T}$ for which an expert trajectory τ_k is observed. Our objective is to adaptively select a sequence of demo environments $\mathcal{P}_1, \ldots, \mathcal{P}_m$ so as to recover a robust estimate of the unknown reward function. We describe the general framework for this interaction between learner and human expert in Algorithm 5. To summarise, a problem-instance in our setting is given by $(S, \mathcal{A}, \mathcal{T}, R^*, \gamma, \omega, m)$, where $\mathcal{T}$ is a set of environments, R^* is the *unknown* reward function, and m the learner's budget.

[2]Typically, expert demonstrations are a limited resource as they involve expensive human input. We thus consider a limited budget of m expert trajectories that the learner is able to obtain.

Framework 5 Environment Design for Inverse Reinforcement Learning

input set of environments $\mathcal{T}$, resources $m \in \mathbb{N}$
for k = 1, . . . , m **do**
  Choose an environment $\mathcal{P}_k \in \mathcal{T}$ (Environment Design)
  Observe expert trajectory τ_k in environment $\mathcal{P}_k$
  Estimate rewards from observations up to round k (IRL)

From Framework 5 we see that the Environment Design for IRL problem has two main ingredients: a) choosing useful demo environments for the human to demonstrate the task in (Section II.4), and b) inferring the reward function from expert demonstration in multiple environments (Section II.5).

II.3.1 Preliminaries and Notation

Throughout the paper, note that R denotes a generic reward function, whereas R^* refers to the true (unknown) reward function. We let $\mathcal{V}^{\pi}_{R,\mathcal{P}}(s) \coloneqq \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R(s_t) \mid \pi, \mathcal{P}, s_0 = s]$ denote the expected discounted return, i.e. value function, of a policy π under some reward function R and transition function $\mathcal{P}$ in state s. For the value under the initial state distribution ω, we then merely write $\mathcal{V}^{\pi}_{R,\mathcal{P}} \coloneqq \mathbb{E}_{s\sim\omega}[\mathcal{V}^{\pi}_{R,\mathcal{P}}(s)]$ and denote its maximum by $\mathcal{V}^{*}_{R,\mathcal{P}} \coloneqq \max_{\pi} \mathcal{V}^{\pi}_{R,\mathcal{P}}$. We accordingly refer to the Q-values under a policy π by $Q^{\pi}_{R,\mathcal{P}}(s,a)$ and their optimal values by $Q^{*}_{R,\mathcal{P}}(s,a)$. In the following, we let $\pi_{R,\mathcal{P}}$ always denote the *optimal policy* w.r.t. R and $\mathcal{P}$, i.e. the policy maximising the expected discounted return in the MDP $(S, \mathcal{A}, \mathcal{P}, R, \gamma, \omega)$.

We generally write τ for expert trajectories. In particular, these expert trajectories are always observed with respect to a specific transition function $\mathcal{P}$. We therefore summarise the observation of an expert trajectory τ_k in an environment $\mathcal{P}_k$ by $\mathcal{D}_k = (\tau_k, \mathcal{P}_k)$ and write $\mathcal{D}_{1:k} = (\mathcal{D}_1, \ldots, \mathcal{D}_k)$ for all observations up to (and including) the k-th round. We let $\mathbb{P}(\cdot \mid \mathcal{D}_{1:k})$ denote the posterior over reward functions given observations $\mathcal{D}_{1:k}$. For the prior $\mathbb{P}(\cdot)$, we introduce the convention that $\mathbb{P}(\cdot) = \mathbb{P}(\cdot \mid \mathcal{D}_{1:0})$. Out of convenience, we sometimes refer to transition functions $\mathcal{P}$ as environments. In particular, when speaking of expert demonstrations in an environment $\mathcal{P}$, we refer to expert demonstrations in the MDP $(S, \mathcal{A}, \mathcal{P}, R^*, \omega, \gamma)$, where R^* denotes the true (unknown) reward function that the expert is maximising.

II.4 Environment Design via Minimax Bayesian Regret

Our goal is to adaptively select demo environments for the expert based on our current belief about the reward function. We consider the situation where at round $k+1$ we have access to a posterior belief $\mathbb{P}(\cdot \mid \mathcal{D}_{1:k})$ over reward functions, which in practice can be approximated using a Bayesian IRL approach whose discussion we postpone to Section II.5. In Section II.4.1, we will introduce a minimax Bayesian regret objective for the environment design process which aims to select demo environments so as to ensure that our reward estimate is

robust and risk-averse. Section II.4.2 then deals with the computation of such environments when the set of demo environments exhibits a useful structure.

II.4.1 Minimax Bayesian Regret

We begin by reflecting on the potential loss of an agent when deploying a policy π under transition function $\mathcal{P}$ and the true reward function R^*, given by the difference

$$\ell_{R^*}(\mathcal{P}, \pi) \coloneqq \mathcal{V}^*_{R^*,\mathcal{P}} - \mathcal{V}^\pi_{R^*,\mathcal{P}}.$$

The reward function R^* is unknown to us, so that we can instead use our belief $\mathbb{P}$ over reward functions and consider the Bayesian regret, i.e. loss, of a policy π under $\mathcal{P}$ and $\mathbb{P}$, i.e.

$$\mathrm{BR}_{\mathbb{P}}(\mathcal{P}, \pi) \coloneqq \mathbb{E}_{R\sim\mathbb{P}}\big[\ell_R(T, \pi)\big] = \mathbb{E}_{R\sim\mathbb{P}}\big[\mathcal{V}^*_{R,\mathcal{P}} - \mathcal{V}^\pi_{R,\mathcal{P}}\big].$$

The concept of Bayesian regret is well-known from, e.g. online optimisation and online learning [RV14] and has been utilised for IRL in a slightly different form by [BCN18]. The idea is that given a (prior) belief about some parameter, we evaluate our policy against an oracle that knows the true parameter. Typically, under such uncertainty about the true parameter (here, reward function) we are interested in risk-averse policies minimising the Bayesian regret, i.e.

$$\min_\pi \mathrm{BR}_{\mathbb{P}}(\mathcal{P}, \pi).$$

To derive an objective for the environment design problem, we then consider a minimax game where one player selects the environment and the other the policy:[3]

$$\max_{\mathcal{P}\in\mathcal{T}} \min_{\pi\in\Pi} \mathrm{BR}_{\mathbb{P}}(\mathcal{P}, \pi). \tag{II.1}$$

What this means is that we search for an environment $\mathcal{P} \in \mathcal{T}$ such that the regret-minimising policy w.r.t. $\mathbb{P}$ suffers maximal regret against the optimal policies w.r.t. reward candidates $R \sim \mathbb{P}$. Note that this objective has the advantage of generally selecting environments that the expert can solve, as the regret in degenerate or purely adversarial environments will be close to zero. Moreover, the minimax Bayesian regret objective is performance-based and not purely uncertainty-based (such as prior objectives based on entropy, e.g. [LMM09]). This is typically desired as reducing our uncertainty about the rewards in states that are not relevant under any transition function in $\mathcal{T}$ (e.g. states that are not being visited by any optimal policy) is unnecessary and generally a wasteful use of our budget. Finally, we also see that if the Bayesian regret objective becomes zero, the posterior mean is guaranteed to be optimal in every demonstration environment.

Lemma II.4.1. *If* $\max_{\mathcal{P}\in\mathcal{T}} \min_{\pi\in\Pi} \mathrm{BR}_{\mathbb{P}}(\mathcal{P}, \pi) = 0$ *for some posterior* $\mathbb{P}(\cdot \mid \mathcal{D})$*, then the posterior mean* $\bar{R} = \mathbb{E}_{\mathbb{P}}[R]$ *is optimal for every* $\mathcal{P} \in \mathcal{T}$*, i.e.* $\bar{R}$ *induces an optimal policy in every environment in* $\mathcal{T}$*.*

Algorithm 6 `ED-BIRL`: Environment Design for Bayesian IRL

1: **input** environments $\mathcal{T}$, prior distribution $\mathbb{P}$, resources $m \in \mathbb{N}$
2: **for** $k = 1, \dots, m$ **do**
3: Sample rewards from $\mathbb{P}(\cdot \mid \mathcal{D}_{1:k-1})$ using $\texttt{BIRL}(\mathcal{D}_{1:k-1})$ (Section II.4.2)
4: Construct empirical distribution $\hat{\mathbb{P}}_{k-1}$ from sampled rewards
5: Find $\mathcal{P}_k \in \operatorname{argmax}_{\mathcal{P}} \min_\pi \mathrm{BR}_{\hat{\mathbb{P}}_{k-1}}(\mathcal{P}, \pi)$ (Section II.5.1)
6: Observe expert trajectory τ_k in $\mathcal{P}_k$ and let $\mathcal{D}_k = (\tau_k, \mathcal{P}_k)$
7: **return** $\texttt{BIRL}(\mathcal{D}_{1:m})$

In our algorithm `ED-BIRL`, we sample from the posterior to construct an empirical distribution for which we then find the maximin transition function (II.1). To sample from the posterior, we use an extension of Bayesian IRL methods to the case where we observe expert demonstrations in multiple environments as described in Section II.5. The algorithm `ED-BIRL` is detailed in Algorithm 6. In the following, we will discuss how the maximin transition function $\operatorname{argmax}_{\mathcal{P}} \min_\pi \mathrm{BR}_{\hat{\mathbb{P}}}(\mathcal{P}, \pi)$ can be computed efficiently and consider the special case when the set of environments, $\mathcal{T}$, has a useful structure that we can exploit.

II.4.2 Environment Generation

Structured Environments. Often the set of environments has a useful structure that can be used to search the space of environments $\mathcal{T}$ efficiently. We begin by recalling that the value function is linear in the rewards, so that we can rewrite equation (II.1) as

$$\max_{\mathcal{P}} \min_{\pi} \mathrm{BR}_{\mathbb{P}}(\mathcal{P}, \pi) = \max_{\mathcal{P}} \left\{ \mathbb{E}_{R \sim \mathbb{P}}[\mathcal{V}^*_{R,\mathcal{P}}(s_0)] - \max_{\pi} \mathcal{V}^{\pi}_{\bar{R},\mathcal{P}}(s_0)] \right\},$$

where $\bar{R} = \mathbb{E}_{R \sim \mathbb{P}}[R]$ is the mean of $\mathbb{P}$. We now consider the special case where each environment $\mathcal{P} \in \mathcal{T}$ is build from a collection of transition matrices $\mathcal{P}_s$.

Let $\mathcal{P}_s \in \mathbb{R}^{S \times A}$ denote a state-transition matrix dictating the transition probabilities in state s. Clearly, we can identify any transition function $\mathcal{P}$ with a family of state-transition matrices $\{\mathcal{P}_s\}_{s \in S}$. We now say that an environment set $\mathcal{T}$ allows us to make *state-individual transition choices* if there exist sets $\mathcal{T}_s$ such that $\mathcal{T} = \{\{\mathcal{P}_s\}_{s \in S} \colon \mathcal{P}_s \in \mathcal{T}_s\}$. In other words, we can construct a new environment $\mathcal{P}$ by arbitrarily combining transition matrices for each state. Note that this of course allows for the case when the transitions in some state s are fixed, i.e. we have the singleton $\mathcal{T}_s = \{\mathcal{P}_s\}$. When we can make such state-individual transition choices, we can use an extended value iteration approach as detailed in Algorithm 7 that takes as input an empirical distribution $\hat{\mathbb{P}}$ as in Line 4 in Algorithm 6.

[3] Note that we here consider $\max_{\mathcal{P}} \min_\pi$ and not the reverse, as we are interested in finding the maximin demo environment (and not a minimax policy).

Algorithm 7 Extended Value Iteration for Structured Environments

input environments $\mathcal{T} = \{\mathcal{T}_s\}_{s\in S}$, empirical distr. $\hat{\mathbb{P}}$, mean $\bar{R} = \mathbb{E}_{R\sim\hat{\mathbb{P}}}[R]$
repeat until $\mathcal{V}_{\bar{R}}$ and $\mathcal{V}_R$ converge:
for $s \in S$ **do**
  $\mathcal{P}_s = \operatorname{argmax}_{\mathcal{P}_s\in\mathcal{T}_s} \left\{ \mathbb{E}_{R\sim\hat{\mathbb{P}}}\big[\max_{a\in\mathcal{A}} \mathcal{P}_{s,a}^\top \mathcal{V}_R\big] - \max_{b\in\mathcal{A}} \mathcal{P}_{s,b}^\top \mathcal{V}_{\bar{R}} \right\}$
  $\mathcal{V}_R(s) = \max_{a\in\mathcal{A}} R(s) + \gamma \mathcal{P}_{s,a}^\top \mathcal{V}_R$ for every $R \sim \hat{\mathbb{P}}$
  $\mathcal{V}_{\bar{R}}(s) = \max_{b\in\mathcal{A}} \bar{R}(s) + \gamma \mathcal{P}_{s,b}^\top \mathcal{V}_{\bar{R}}$
return environment $\mathcal{P} = \{\mathcal{P}_s\}_{s\in S}$

II.5 Inverse Reinforcement Learning with Multiple Environments

We now analyse how we can learn about the reward function from demonstrations that were provided under multiple, different environment dynamics. Recall that we consider the situation where the learner observes expert trajectories with respect to the same reward function under possibly different transition dynamics. In the following, we explain how to extend Bayesian IRL methods to this setting.

II.5.1 Bayesian IRL

The Bayesian perspective to the IRL problem provides a principled way to reason about reward uncertainty [RA07]. Typically, the human is modelled by a Boltzmann-rational policy [JMD20]. This means that for a given reward function R and transition function $\mathcal{P}$ the expert is acting according to a policy

$$\pi_{R,\mathcal{P}}^{\text{softmax}}(a \mid s) = \frac{\exp(cQ_{R,\mathcal{P}}^*(s,a))}{\sum_{a'} \exp(cQ_{R,\mathcal{P}}^*(s,a'))}, \tag{II.2}$$

where the parameter c relates to our judgement of the expert's optimality.[4] Given a prior distribution $\mathbb{P}(\cdot)$, the goal of Bayesian IRL is to recover the posterior distribution $\mathbb{P}(\cdot \mid \mathcal{D})$ and to either sample from the posterior using MCMC [RA07; RD11] or perform MAP estimation [CK11]. In our case, the data is given by the sequence $\mathcal{D}_{1:k} = (\mathcal{D}_1, \ldots, \mathcal{D}_k)$ with $\mathcal{D}_k = (\tau_k, \mathcal{P}_k)$. We see that this is no obstacle as the likelihood factorises as

$$\mathbb{P}(\mathcal{D}_{1:k} \mid R) = \prod_{i\le k} \mathbb{P}(\tau_i \mid R, \mathcal{P}_i),$$

since the expert trajectories (i.e. expert policies) are conditionally independent given the reward function and transition function. The likelihood of each expert demonstration is then given by $\mathbb{P}(\tau_i \mid R, \mathcal{P}_i) = \prod_{(s,a)\in\tau_i} \pi_{R,\mathcal{P}_i}^{\text{softmax}}(a \mid s)$, where

[4] Note that when using MCMC Bayesian IRL methods we can also perform inference over the parameter c and must not assume knowledge of the expert's optimality.

$\pi^{\text{softmax}}_{R,\mathcal{P}_i}$ is the Boltzmann-rational policy as defined in (II.2). As a result, we can, for instance, sample from the posterior using the Policy-Walk algorithm from [RA07] with minor modifications or the Metropolis-Hastings Simplex-Walk algorithm from [BGD22]. Other Bayesian approaches, e.g. those that model the reward function as a Gaussian process [LPK11] or take a variational inference approach [CS21], can similarly be adapted to demonstrations from multiple environments by using the factorisation of the likelihood. We generally denote any Bayesian IRL algorithm that is capable of sampling from the posterior by `BIRL`.

II.6 Experiments

We perform a preliminary set of experiments on a maze task as well as randomly generated MDPs. Our primary goal is to address the following two questions:

1. Can we *recover the true reward function* by adaptively designing demo environments?

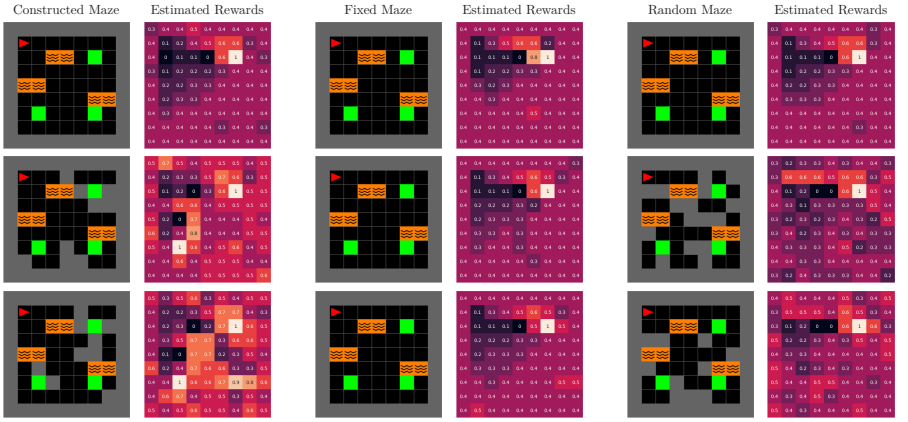
2. Can we learn *more robust reward functions* by adaptively designing demo environments?

II.6.1 Recovering the True Reward Function

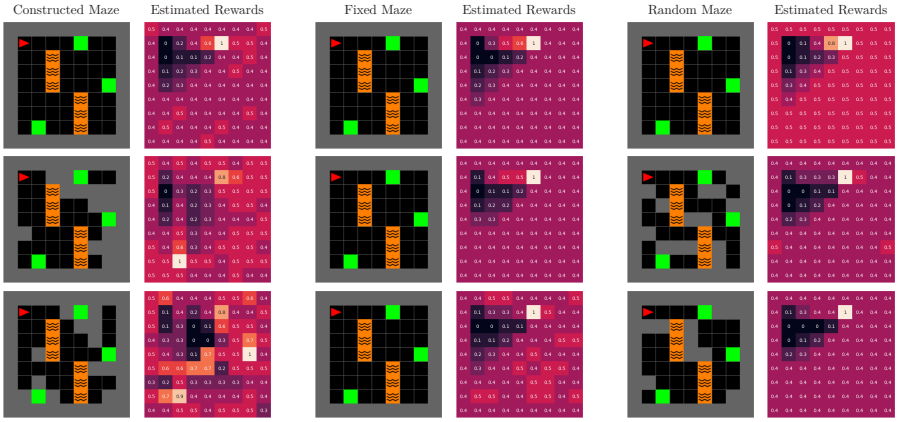
In this experiment, we consider a maze task in which the learner has the ability to add obstacles to a base layout of the maze. We visualise the designed mazes and estimated rewards and evaluate whether our approach can recover the true reward function by adaptively constructing these mazes.

Experimental Setup. We consider a maze task in which the goal is to reach one of three goal states while avoiding lava. Here, the learner is able to add obstacles to cells and observes two expert trajectories for each constructed maze, which is done to give a stronger learning signal to `BIRL` so as to require fewer samples. The true reward function, which is unknown to the learner, yields reward 1 in goal states and reward -1 in lava states. We consider two different base layouts: a basic layout with goal states and lava evenly spread out, Figure II.2 (a)-(c), and a second layout with vertical strips of lava which make it challenging to construct mazes so that the right side of the world is being visited, Figure II.2 (d)-(f). We compare our approach, `ED-BIRL`, with learning from a fixed maze, and learning from mazes that were randomly created. We randomly generate these mazes by adding an obstacle to a cell with probability 0.3.[5] The inference for all three approaches is done using `BIRL` and the computed reward estimates are scaled to $[0, 1]$ and rounded.

[5]Naturally, such randomly generated mazes can be very different every iteration and we can only display exemplary mazes for domain randomisation in Figure II.2. However, the presented examples can nevertheless serve as an illustration of the disadvantages of using domain randomisation for IRL.



(a) `ED-BIRL` (b) `Fixed Environment` (c) `Domain Randomisation`



(d) `ED-BIRL` (e) `Fixed Environment` (f) `Domain Randomisation`

Figure II.2: Comparison of `ED-BIRL`, `Fixed Environment`, and `Domain Randomisation` for two versions of the maze problem: (a)-(c) and (d)-(f). In each case, we display three consecutive rounds and the corresponding mazes and estimated rewards. We use the same colour scale as in Figure V.1, which ranges from black (0.0) to red (0.5) to white (1.0).

Results. In Figure II.2, we observe that `ED-BIRL` recovers the location of all three goal states after three rounds in both maze layouts. Moreover, the learner is able to identify the location of all lava strips in Figure II.2a, i.e. states with negative reward. In Figure II.2d, `ED-BIRL` also recovered the rewards of the upper lava region, whereas the estimates for the lower lava region are more imprecise (while they are also less performance-relevant). By adaptively designing a sequence of demo environments, `ED-BIRL` is thus capable of recovering (all performance-relevant aspects of) the unknown reward function.

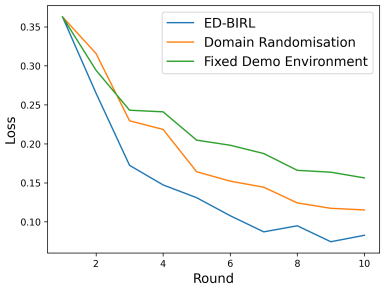
In contrast, learning from a fixed environment (Figure II.2b, II.2e) as well as domain randomisation (Figure II.2c, II.2f) fail to recover the location of all goal states, let alone lava. In a fixed maze, any near-optimal policy will visit the closest goal state only, which in this case is the top right corner in both versions of the maze. We also see that using domain randomisation is impractical for IRL, as we require carefully constructed mazes to recover the true reward function. Even worse, by obliviously randomising the maze layout, we may create unsolvable environments for the human expert, which yield no information at all (see e.g. Figure II.2c).

II.6.2 Learning Robust Reward Functions

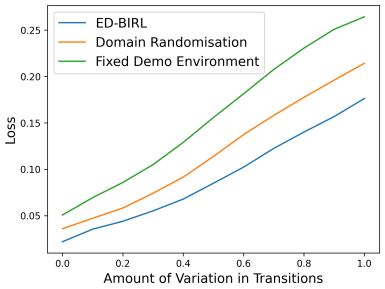
In this experiment, we provide the learner with a set of *demo* environments they can select for a demonstration. Afterwards, the agent is evaluated on a set of *test* environments. The performance in the test set captures the generalisation ability of the learned rewards to new dynamics.

Experimental Setup. We first randomly generate a base MDP $(S, \mathcal{A}, \mathcal{P}_{\texttt{base}}, R^*, \gamma, \omega)$ with base transition function $\mathcal{P}_{\texttt{base}}$. We then construct the set of possible demo environments, here denoted $\mathcal{T}_{\texttt{demo}}$ instead of $\mathcal{T}$ to clearly distinguish between demo and test environments, by sampling state-transition functions that differ from the base transitions $\mathcal{P}_{\texttt{base}}$ by at most some value $\rho_{\texttt{demo}}$ in terms of ℓ_∞-distance. In our experiments, we set the maximum amount of variation in the demo environments to $\rho_{\texttt{demo}} = 0.5$. Similarly, we create a set of test environments $\mathcal{T}_{\texttt{test}}$ with a maximum amount of perturbation $\rho_{\texttt{test}}$ on which we evaluate the learned reward functions. For all three approaches, we evaluate the posterior mean, which is computed using `BIRL`. For all $\mathcal{P} \in \mathcal{T}_{\texttt{test}}$, we optimise a policy w.r.t. the posterior mean and $\mathcal{P}$ and evaluate the computed policy under the true reward function R^* and transition function $\mathcal{P}$. Finally, we average the results over all environments in $\mathcal{T}_{\texttt{test}}$. We want to emphasise that the way we construct $\mathcal{T}_{\texttt{demo}}$ and $\mathcal{T}_{\texttt{test}}$, these sets are completely disjunct except for the base transition function, i.e. $\mathcal{T}_{\texttt{demo}} \cap \mathcal{T}_{\texttt{test}} = \{\mathcal{P}_{\texttt{base}}\}$. We therefore *do not* observe the expert in the environments that we evaluate our approaches on.

Results. In Figure II.3a, we observe that `ED-BIRL` outperforms domain randomisation and learning from a fixed environments over the course of all rounds. As expected, the loss of all three approaches increases the more diverse the test environments are and the more they differ from the base environment, which can be seen in Figure II.3b. Interestingly, even for $\rho_{\texttt{test}} = 0$, i.e. evaluation on the base environment only, `ED-BIRL` slightly outperforms learning directly from the fixed base environment suggesting a superior sample-efficiency of `ED-BIRL`.



(a) Average utility *loss* of `ED-BIRL`, `Domain Randomisation`, and `Fixed Environment IRL` over 10 rounds. The learned rewards are evaluated on a set of test environments that differ from the base environment by at most $\rho_{\texttt{test}} = 0.5$.



(b) Along the x-axis we increase $\rho_{\texttt{test}}$, i.e. the amount of variation in the test environments. We evaluate the learned reward functions after 10 rounds of interaction with the expert, i.e. the final reward estimate from Figure II.3a.

Figure II.3: On a randomly generated MDP task, we evaluate the robustness of reward estimates learned by `ED-BIRL`, `Domain Randomisation`, and `Fixed Environment IRL`, respectively.

II.7 Discussion

The presented work gives a first glance into Environment Design for Inverse Reinforcement Learning. In this paper, we focus on the Bayesian setting, where a belief about the reward function is computed using Bayesian IRL (with observations from multiple environments). This allowed us to reason about reward uncertainty in a principled way, guiding our environment design approach via a minimax Bayesian regret objective. A future version of this work will consider non-Bayesian IRL frameworks and explain how to perform environment design with point estimates of the reward function (instead of Bayesian beliefs). In future work it will also be interesting to consider a batch version of this setting, where the learner has to decide on a batch of demo environments every round.

References

[AD21] Arora, S. and Doshi, P. "A survey of inverse reinforcement learning: Challenges, methods and progress". In: *Artificial Intelligence* vol. 297 (2021), p. 103500.

[AJS17] Amin, K., Jiang, N., and Singh, S. "Repeated inverse reinforcement learning". In: *Advances in neural information processing systems* vol. 30 (2017).

[Akk+19] Akkaya, I. et al. "Solving rubik's cube with a robot hand". In: *arXiv preprint arXiv:1910.07113* (2019).

[BCN18] Brown, D. S., Cui, Y., and Niekum, S. “Risk-aware active inverse reinforcement learning”. In: *Conference on Robot Learning*. PMLR. 2018, pp. 362–372.

[BGD22] Büning, T. K., George, A.-M., and Dimitrakakis, C. “Interactive Inverse Reinforcement Learning for Cooperative Games”. In: *International Conference on Machine Learning*. PMLR. 2022, pp. 2393–2413.

[CCS21] Cao, H., Cohen, S., and Szpruch, L. “Identifiability in inverse reinforcement learning”. In: *Advances in Neural Information Processing Systems* vol. 34 (2021), pp. 12362–12373.

[CK11] Choi, J. and Kim, K.-e. “MAP Inference for Bayesian Inverse Reinforcement Learning”. In: *Advances in Neural Information Processing Systems*. Vol. 24. 2011.

[CS21] Chan, A. J. and Schaar, M. van der. “Scalable Bayesian Inverse Reinforcement Learning”. In: *arXiv preprint arXiv:2102.06483* (2021).

[Den+20] Dennis, M. et al. “Emergent complexity and zero-shot transfer via unsupervised environment design”. In: *Advances in Neural Information Processing Systems* vol. 33 (2020), pp. 13049–13061.

[Fin+16] Finn, C. et al. “A connection between generative adversarial networks, inverse reinforcement learning, and energy-based models”. In: *arXiv preprint arXiv:1611.03852* (2016).

[FLA16] Finn, C., Levine, S., and Abbeel, P. “Guided cost learning: Deep inverse optimal control via policy optimization”. In: *International conference on machine learning*. PMLR. 2016, pp. 49–58.

[FLL18] Fu, J., Luo, K., and Levine, S. “Learning Robust Rewards with Adverserial Inverse Reinforcement Learning”. In: *International Conference on Learning Representations*. 2018.

[Gur+21] Gur, I. et al. “Environment generation for zero-shot compositional reinforcement learning”. In: *Advances in Neural Information Processing Systems* vol. 34 (2021), pp. 4157–4169.

[HE16] Ho, J. and Ermon, S. “Generative adversarial imitation learning”. In: *Advances in neural information processing systems* vol. 29 (2016).

[JMD20] Jeon, H. J., Milli, S., and Dragan, A. “Reward-rational (implicit) choice: A unifying formalism for reward learning”. In: *Advances in Neural Information Processing Systems* vol. 33 (2020), pp. 4415–4426.

[Kim+21] Kim, K. et al. “Reward identification in inverse reinforcement learning”. In: *International Conference on Machine Learning*. PMLR. 2021, pp. 5496–5505.

[Lev+16] Levine, S. et al. “End-to-end training of deep visuomotor policies”. In: *The Journal of Machine Learning Research* vol. 17, no. 1 (2016), pp. 1334–1373.

[Lil+15] Lillicrap, T. P. et al. “Continuous control with deep reinforcement learning”. In: *arXiv preprint arXiv:1509.02971* (2015).

[Lin+21] Lindner, D. et al. “Information Directed Reward Learning for Reinforcement Learning”. In: *Advances in Neural Information Processing Systems* vol. 34 (2021), pp. 3850–3862.

[LMM09] Lopes, M., Melo, F., and Montesano, L. “Active learning for reward estimation in inverse reinforcement learning”. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2009, pp. 31–46.

[LPK11] Levine, S., Popovic, Z., and Koltun, V. “Nonlinear inverse reinforcement learning with gaussian processes”. In: *Advances in neural information processing systems* vol. 24 (2011).

[MMR18] Metelli, A. M., Mutti, M., and Restelli, M. “Configurable Markov decision processes”. In: *International Conference on Machine Learning*. PMLR. 2018, pp. 3491–3500.

[Mni+15] Mnih, V. et al. “Human-level control through deep reinforcement learning”. In: *nature* vol. 518, no. 7540 (2015), pp. 529–533.

[Nar+20] Narvekar, S. et al. “Curriculum learning for reinforcement learning domains: A framework and survey”. In: *arXiv preprint arXiv:2003.04960* (2020).

[NR00] Ng, A. Y. and Russell, S. J. “Algorithms for Inverse Reinforcement Learning”. In: *Proceedings of the Seventeenth International Conference on Machine Learning*. 2000, p. 2.

[RA07] Ramachandran, D. and Amir, E. “Bayesian Inverse Reinforcement Learning”. In: *Proceedings of the 20th International Joint Conference on Artifical Intelligence*. 2007, pp. 2586–2591.

[Ram+21] Ramponi, G. et al. “Learning in Non-Cooperative Configurable Markov Decision Processes”. In: *Advances in Neural Information Processing Systems* vol. 34 (2021).

[RD11] Rothkopf, C. A. and Dimitrakakis, C. “Preference elicitation and inverse reinforcement learning”. In: *Joint European conference on machine learning and knowledge discovery in databases*. 2011, pp. 34–48.

[Rus98] Russell, S. “Learning agents for uncertain environments”. In: *Proceedings of the eleventh annual conference on Computational learning theory*. 1998, pp. 101–103.

[RV14] Russo, D. and Van Roy, B. “Learning to optimize via information-directed sampling”. In: *Advances in Neural Information Processing Systems* vol. 27 (2014).

[Tob+17] Tobin, J. et al. “Domain randomization for transferring deep neural networks from simulation to the real world”. In: *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE. 2017, pp. 23–30.

[Toy+20] Toyer, S. et al. “The magical benchmark for robust imitation”. In: *Advances in Neural Information Processing Systems* vol. 33 (2020), pp. 18284–18295.

[Zie+08] Ziebart, B. D. et al. “Maximum entropy inverse reinforcement learning.” In: *Aaai*. Vol. 8. Chicago, IL, USA. 2008, pp. 1433–1438.

B.1 Proofs

Proof of Lemma II.4.1. For simplicity of exposition, we assume here that the posterior $\mathbb{P}$ is discrete. Now, as the value function is linear in rewards, we have

$$\min_{\pi} \mathrm{BR}_{\mathbb{P}}(\mathcal{P}, \pi) = \mathrm{BR}_{\mathbb{P}}(\mathcal{P}, \pi_{\bar{R},\mathcal{P}}),$$

where $\pi_{\bar{R},\mathcal{P}}$ is the optimal policy w.r.t. the posterior mean $\bar{R} = \mathbb{E}_{R\sim\mathbb{P}}[R]$ and the transition function $\mathcal{P}$. If $\max_{\mathcal{P}\in\mathcal{T}} \min_{\pi\in\Pi} \mathrm{BR}_{\mathbb{P}}(\mathcal{P}, \pi) = 0$, it then follows that $\max_{\mathcal{P}\in\mathcal{T}} \mathrm{BR}_{\mathbb{P}}(\mathcal{P}, \pi_{\bar{R},\mathcal{P}}) = 0$, i.e. $\mathcal{V}^{*}_{R,\mathcal{P}} = \mathcal{V}^{\pi_{\bar{R},\mathcal{P}}}_{R,\mathcal{P}}$ for all $R \in \mathrm{supp}(\mathbb{P})$ and $\mathcal{P} \in \mathcal{T}$. This must imply that $\mathcal{V}^{*}_{R^*,\mathcal{P}} = \mathcal{V}^{\pi_{\bar{R},\mathcal{P}}}_{R^*,\mathcal{P}}$ for all $\mathcal{P} \in \mathcal{T}$. In other words, $\bar{R}$ is optimal for all $\mathcal{P} \in \mathcal{T}$ (under the initial state distribution ω). ■

B.2 More Experimental Details

The `BIRL` method we used for the experiments is a straightforward extension of Algorithm 1 in [RD11] to multiple environments following our explanations in Section II.5.1.

Recovering the True Reward Function. For the experiments in Section II.6.1, we let the learner observe two trajectories for each maze. This was done in order to speed up the inference of `BIRL` and reduce the computational cost. The expert was modeled by a Boltzmann-rational policy and thus uniformly selected an optimal action when there were several optimal ones in a given state.

Learning Robust Reward Functions. For the experiments in Section II.6.2, we randomly generated an MDP with 50 states and 4 actions using a Dirichlet distribution for the transitions and a Beta distribution for the reward function. For each state we let the demo set of environments contain 15 choices. The size of the test environments was set to be $|\mathcal{T}_{\texttt{test}}| = 500$. Every round, the learner got to select a demo environment and observe a single expert trajectory in that environment. We limited the amount of deviation from the base transitions in our experiments according to $\rho_{\texttt{demo}}$ and $\rho_{\texttt{test}}$. In particular, note that any choice of $\rho_{\texttt{demo}}$ implies that $\|\mathcal{P}_{\texttt{base}} - \mathcal{P}\|_\infty = \max_{s,a} \|\mathcal{P}_{\texttt{base}}(\cdot \mid s,a) - \mathcal{P}(\cdot \mid s,a)\|_1 \leq \rho_{\texttt{demo}}$ for all $\mathcal{P} \in \mathcal{T}_{\texttt{demo}}$. The results were averaged over 5 complete runs, i.e. for 5 randomly generated problem instances.

B.2.1 Environment Design with Arbitrary Environments

In some situations, the set of demo environments $\mathcal{T}$ may not exhibit any useful structure. Moreover, we may not even have explicit knowledge of the transition functions in $\mathcal{T}$, but can only access a set of corresponding simulators. In this case, we are left with approximating the maximin environment (II.1) by sampling simulators from $\mathcal{T}$ and performing policy rollouts (see Algorithm 8).

Algorithm 8 Environment Design with Arbitrary Environments

input set of environments $\mathcal{T}$, rewards $\{R_1, \dots, R_k\}$, best guess $\bar{R}$
// if necessary, sample a subset $\mathcal{T}_\subset$ from $\mathcal{T}$
for $\mathcal{P} \in \mathcal{T}$ **do**
  calculate $\pi^* = \pi^*_{\bar{R},\mathcal{P}}$ (policy optimisation)
  for $R \in \{R_1, \dots, R_k\}$ **do**
    evaluate $\mathcal{V}^{\pi^*}_{R,\mathcal{P}}$ (policy evaluation)
    calculate $\mathcal{V}^*_{R,\mathcal{P}} = \max_\pi \mathcal{V}^\pi_{R,\mathcal{P}}$ (policy optimisation)
    $\ell(R) = \max_\pi \mathcal{V}^*_{R,\mathcal{P}} - \mathcal{V}^{\pi^*}_{R,\mathcal{P}}$
  $\mathrm{BR}(\mathcal{P}) = \sum_{R \in \{R_1,\dots,R_k\}} \ell(R)$
return $\mathcal{P}^* = \operatorname{argmax}_{\mathcal{P}\in\mathcal{T}} \mathrm{BR}(\mathcal{P})$

B.2.2 Maximum Entropy IRL with Multiple Environments

In the following, we give a brief outline of how Maximum Entropy (MaxEnt) IRL methods can be extended to multiple environments. For a practical algorithm we choose to extend the popular Adversarial IRL algorithm [FLL18].

In MaxEnt IRL, the reward function is assumed to be parameterised by some vector θ. While some work has considered non-linear parameterisation of the reward function, e.g. [FLA16], we can generally think of the reward function being linear in some feature vector $\mathbf{f}$, i.e. $R_\theta(s) = \theta^\top \mathbf{f}_s$. Under the MaxEnt model, the probability of trajectories is exponentially dependent on their value:

$$\mathbb{P}(\tau \mid \theta, \mathcal{P}) = \frac{e^{R_\theta(\tau)}}{Z(\theta,\mathcal{P})} \prod_{t=1}^{|\tau|} \mathcal{P}(s_{t+1} \mid s_t, a_t), \tag{3}$$

where $Z(\theta, \mathcal{P})$ is the partition function given by

$$Z(\theta,\mathcal{P}) = \sum_\tau e^{R_\theta(\tau)} \prod_{t=1}^{|\tau|} \mathcal{P}(s_{t+1} \mid s_t, a_t). \tag{4}$$

Note that here the sum over τ is over all possible trajectories. Our goal is then to solve the maximum likelihood problem

$$\operatorname*{argmax}_\theta \sum_{(\tau,\mathcal{P})\in\mathcal{D}} \log \mathbb{P}(\tau \mid \theta, \mathcal{P}). \tag{5}$$

We see that the only difference to the original MaxEnt IRL formulation is that we now sum over pairs $(\tau, \mathcal{P})$ instead of just τ. As a scalable solution to the MaxEnt IRL problem, Adversarial IRL [FLL18] as well as GAIL [Fin+16; HE16] cast the optimisation of (5) as a generative adversarial network (with different discriminators). To extend Adversarial IRL, we consider a set of policies $\pi_1, \dots, \pi_k$, used to generate trajectories in environments $\mathcal{P}_1, \dots, \mathcal{P}_k$, and discriminators $D_{1,\theta,\phi}, \dots D_{k,\theta,\phi}$ given by

$$D_{i,\theta,\phi}(s,a,s') = \frac{\exp(f_{\theta,\phi}(s,a,s'))}{\exp(f_{\theta,\phi}(s,a,s')) + \pi_i(a \mid s)} \tag{6}$$

with

$$f_{\theta,\phi}(s,a,s') = g_\theta(s) + \gamma h_\phi(s') - h_\phi(s), \tag{7}$$

where $g_\theta(s)$ is the reward approximator and h_ϕ a shaping term (see [FLL18]).

Algorithm 9 Adversarial IRL with Multiple Environments

1: **input** Observations $\mathcal{D} = (\mathcal{D}_1, \dots, \mathcal{D}_k)$ with $\mathcal{D}_i = (\tau_i, \mathcal{P}_i)$
2: Initialise policies $\pi_1, \dots, \pi_k$ and discriminators $D_{1,\theta,\phi}, \dots, D_{k,\theta,\phi}$
3: **for** $t = 0, 1, \dots$ **do**
4: Collect trajectories $\tau^G_{i,j} = (s_0, a_0, \dots, s_H, a_H)$ by executing π_i in $\mathcal{P}_k$ for $i \in [k]$.
5: Train discriminators $D_{1,\theta,\phi} \dots, D_{k,\theta,\phi}$ to classify expert data $\tau_1, \dots, \tau_k$ from samples $\{\tau^G_{1,j}\}_j, \dots, \{\tau^G_{k,j}\}_j$, respectively, via logistic regression with shared parameter θ.
6: Update reward $R_{\theta,\phi}(s,a,s') \leftarrow \sum_{i=1}^k \Big(\log D_{i,\theta,\phi}(s,a,s') - \log(1 - D_{i,\theta,\phi}(s,a,s')) \Big)$.
7: Update $\pi_1, \dots, \pi_k$ with respect to $R_{\theta,\phi}$ using any policy optimisation method.

With minor modifications, a justification of Algorithm 9 can be done analogous to that in [FLL18].

Paper III

ANACONDA: An Improved Dynamic Regret Algorithm for Adaptive Non-Stationary Dueling Bandits

Thomas Kleine Buening, Aadirupa Saha

Published in *26th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2023.

Abstract

We study the problem of non-stationary dueling bandits and provide the first adaptive dynamic regret algorithm for this problem. The only two existing attempts in this line of work fall short across multiple dimensions, including pessimistic measures of non-stationary complexity and non-adaptive parameter tuning that requires knowledge of the number of preference changes. We develop an elimination-based rescheduling algorithm to overcome these shortcomings and show a near-optimal $\tilde{O}(\sqrt{S^{\mathsf{CW}}T})$ dynamic regret bound, where S^{CW} is the number of times the Condorcet winner changes in T rounds. This yields the first near-optimal dynamic regret bound for unknown S^{CW}. We further study other related notions of non-stationarity for which we also prove near-optimal dynamic regret guarantees under additional assumptions on the preference model.

III.1 Introduction

Multi-Armed Bandits (MAB) [LS18; Rob52; Tho33] are a well-studied online learning framework, which can be used to model online decision-making under uncertainty. Due to its exploration-exploitation tradeoff, the MAB framework is able to model situations such as clinical trials or job scheduling, where the goal is to keep selecting the 'best item' in hindsight through sequentially querying one item at a time and subsequently observing a noisy reward feedback for the queried item [AB10; ACF02; AG12; BC+12].

The MAB framework has been studied and generalized to different settings, among which a popular variant is known as Dueling Bandits (DB) which has gained much attention in the machine learning community over the last two decades [WL16; Yue+12; Zog+14a; Zog+15]. DB are a preference-based variant of MAB in which every round the learner selects a pair of items (or arms)

whereupon a noisy preference between the two items is observed. Such a model is particularly useful in applications, where direct numerical feedback is unavailable, but observed feedback or behavior implies a preference of one item over the other. For instance, the DB framework can be used for search engine optimization through interleaved comparisons [HWD11; RC13].

In the classical stochastic dueling bandit problem, it is assumed that the underlying preferences between items remain fixed over time. However, this assumed stationarity of preferences is likely to be violated in many applications. For example, preferences over movies may change depending on the season or other external influences. Despite its strong practical motivation, regret minimization in non-stationary dueling bandits has only recently been studied for the fist time [GS22b; KBH22]. In contrast to the classical stochastic [Ben+21; Yue+12; Zog+14b] and adversarial [GUC15; SG22; SKM21] dueling bandit problem, which measures performance in terms of static regret w.r.t. a fixed benchmark (or best item in hindsight), in non-stationary dueling bandits we consider the stronger *dynamic regret*, which compares the algorithm's selection against a dynamic benchmark every round.

In general, the achievable dynamic regret depends on the amount of non-stationarity in the environment. Here, prior work [GS22b; KBH22] studied the number of changes in the preference matrix as a measure non-stationary complexity. While the number of such preference switches indeed relates to the hardness of the problem, it is, however, a pessimistic measure of non-stationarity. For example, a change in the preference between two widely suboptimal arms or a minor change in the preference matrix under which the optimal arm remains optimal should not significantly impact our ability to achieve low dynamic regret. To this end, one question that we aim to address in this paper for the paradigm of non-stationary dueling bandits is:

> **Q.1:** *Can we guarantee low dynamic regret for stronger and more meaningful notions of non-stationarity?*

Moreover, prior work in non-stationary dueling bandits [GS22b; KBH22] assumes knowledge of the non-stationary complexity, i.e. prior knowledge of the total number of preference switches (or total variation), which is a highly impractical assumption. The second question we thus address is:

> **Q.2:** *Can we achieve near-optimal dynamic regret in non-stationary dueling bandits adaptively, without the knowledge of the underlying non-stationary complexity?*

III.1.1 Our Contributions

We answer these two questions affirmatively. Our main contribution is a new algorithm `ANACONDA` that adaptively achieves near-optimal regret with respect to the number of 'best arm' switches—a measure that is sensitive only to the variations of the best arms in the preference sequence and indifferent to any other

'background noise' due to suboptimal arms. More precisely, our contributions can be listed as follows:

- **Connecting Different Notions of Non-Stationary Complexity in DB.** We first give an overview over different notions of non-stationarity measures for dueling bandits and analyze their interdependencies towards a better understanding of the implications of one to another (Section III.2.2).

- **Proposing Tighter Notions of Non-Stationarity (towards Q.1).** We propose three new notions of non-stationary complexity for dueling bandits: (i) S^{CW} which measures the number of Condorcet Winner Switches in the preference sequence, (ii) $\tilde{V}$ which measures the preference variation of the Condorcet arms, and (iii) $\tilde{S}^{\mathsf{CW}}$ that counts only the 'significant variations' in the Condorcet arms (Section III.2.2).

  The novelty of our proposed non-stationarity measures lies in capturing only the non-stationarity observed for the 'best arms' of the preference sequences. They remain unaffected by any changes in the suboptimal arms, which of course captures a stronger notion of non-stationarity than simply counting the number of preference shifts S^{P}, or total variation V, of the preference sequence $\{P_t\}_{t\in[T]}$, as studied in prior work [GS22b; KBH22]. In particular, we show that $\tilde{S}^{\mathsf{CW}} \leq S^{\mathsf{CW}} \leq S^{\mathsf{P}}$ and $\tilde{V} \leq V$ justifying the strength of our proposed non-stationarity measures.

- **Adaptive Algorithm (towards Q.2).** Besides using weaker notions of non-stationary complexity, another drawback of existing work on non-stationary dueling bandit is that, in order to optimize dynamic regret, their algorithms require exact knowledge of the non-stationary complexity (e.g. S^{P} or V), which is in practice of course expected to be unknown to the system/algorithm designed ahead of time. Our next main contribution lies in designing an adaptive algorithm (`ANACONDA`, Algorithm 10) that does not require knowledge of any underlying non-stationary complexity—it can adapt to any unknown number of best arm switches S^{CW} and yields a near-optimal regret bound of $\tilde{O}\big(\sqrt{S^{\mathsf{CW}}T}\big)$ (Theorem III.3.1, Section III.3).[1]

- **Improved and (Near-)Optimal Dynamic Regret Bounds.** Owing to the fact that $S^{\mathsf{CW}} \leq S^{\mathsf{P}}$, our dynamic regret bounds can be much tighter compared to the previous results by [GS22b; KBH22] which can only give a regret guarantee of $\tilde{O}\big(\sqrt{S^{\mathsf{P}}T}\big)$ (Remark III.2.2). Further our regret bound is also provably order optimal in T and S^{CW} as justified in Remark III.3.3.

- **Better Guarantees for Structured Preferences.** Moreover, in Section III.5 we discover a special class of preference matrices, those that respect a type of transitive property, for which we can prove even stronger dynamic regret guarantees of $\tilde{O}\big(\sqrt{\tilde{S}^{\mathsf{CW}}T}\big)$ in terms of Significant CW Switches $\tilde{S}^{\mathsf{CW}}$ and $\tilde{O}\big(\tilde{V}^{1/3}T^{2/3}\big)$ in terms of Condorcet Winner Variation $\tilde{V}$. The optimality of these bounds is discussed in Remark III.5.5 and Remark III.5.7.

[1] Here, $\tilde{O}$ notation hides logarithmic dependencies.

III.1.2 Related Works

The non-stationary MAB problem has been extensively studied for various non-stationarity measures, such as total variation [BGZ14; BGZ15], distribution switches [AFM17; AGO19; GM11], or best arm switches [AGL22; SK22b]. Moreover, its study has been extended to more complex setups including linear bandits [RCG20; RVC19] and contextual MAB [Che+19; Luo+18; WIW18]. We will particularly take inspiration from the recent advances of [AGL22; AGO19; SK22b] that were able to achieve near-optimal dynamic regret rates without knowledge of the number of distribution (or best arm) changes.

While the non-stationary MAB problem has seen much attention in recent years, its DB counterpart remains widely unexplored. The only two earlier works that address the non-stationary dueling bandit problem are [GS22b] and [KBH22]. However, these works are limited in a) the weakness of the analyzed non-stationarity measures, namely, general preference switches or total variation (see Section III.2.2), and b) in the fact that their algorithms require knowledge of the total amount of non-stationarity in advance, an unrealistic assumption. Here, we improve upon prior work by designing an adaptive algorithm `ANACONDA` that does not require knowledge of the amount of non-stationarity in the environment and achieves near-optimal dynamic regret w.r.t. the number of Condorcet winner switches, a stronger notion of non-stationarity than general preference switches. A more detailed review of previous work that is related to the non-stationary MAB and DB problem is provided in Appendix C.3.

III.2 Problem Setting

We consider preference matrices $P \in [0,1]^{K\times K}$ such that $P(a,b)$ indicates the probability of arm a being preferred over arm b. Here, P satisfies $P(a,b) = 1 - P(b,a)$ and $P(a,a) = 0.5$ for all $a,b \in [K]$. We say that a dominates b and write $a \succ b$ if $P(a,b) > 0.5$, i.e. arm a has a higher chance of winning than arm b in a duel (a,b). A well-studied concept of a *good benchmark arm* in dueling bandits is the *Condorcet Winner* (CW): Given any preference matrix $P \in [0,1]^{K\times K}$, an arm $a^* \in [K]$ is called a Condorcet winner of P if $P(a^*,b) > 0.5$ for all $b \in [K] \setminus \{a^*\}$ [Ben+21; Kom+15; SG22; WL16; Zog+14b].

Note that any preference matrix with a total ordering over arms invariably has a Condorcet winner. For example, assuming a total ordering $1 \succ 2 \succ \ldots \succ K$ implies that the Condorcet winner is arm 1. Any RUM-based preference matrix [SG19a; SG20a; SPX13], or more generally any P that satisfies stochastic transitivity [YJ09], always respects a total ordering. However, note that CW-based preference matrices consider a much bigger class of pairwise relations than total ordering. Despite this, in general a preference matrix might not have a Condorcet winner, which led to more general notions of benchmark arms in DB, such as the Borda winner [SKM21], the Copeland winner [Zog+15] or the von Neumann winner [Dud+15b; SK22a].

III.2.1 Non-Stationary Dueling Bandits (NSt-DB)

We consider a decision space of K arms denoted by $[K]$. At each round $t \in [T]$, the task of the learner is to select a pair of actions $(a_t, b_t) \in [K] \times [K]$, upon which a preference feedback $o_t(a_t, b_t) \sim \mathrm{Ber}(P_t(a_t, b_t))$ is revealed to the learner according to the underlying preference matrix $P_t \in [0,1]^{K \times K}$. The sequence of preferences $P_1, P_2, \ldots, P_T$ is generated adversarially and for any such preference matrix P_t we define

$$\delta_t(a, b) \coloneqq P_t(a, b) - 1/2$$

as the gap or preference-strength of arm a over arm b in round t. We here assume that every preference matrix P_t has a Condorcet winner, which we denote by a_t^*.

ParagraphStatic Regret in Dueling Bandits. In classical (stochastic) dueling bandits, where it is assumed that $P_1 = \ldots = P_T = P$ for some fixed preference matrix P, the performance of the learner is often measured w.r.t. the CW of P, described by the *static regret*

$$\mathrm{R}(T) := \sum_{t=1}^{T} \frac{\delta_t(a^*, a_t) + \delta_t(a^*, b_t)}{2},$$

where a^* is the CW of P [Ben+21; SG21; Sui+18; YJ09]. Note that here $\delta_t(a^*, a) = P_t(a^*, a) - 1/2$ essentially quantifies the net loss of arm a against the fixed benchmark arm a^*.

However, regret with respect to any fixed benchmark (comparator arm) soon becomes meaningless when the underlying preference matrices are changing over time, since no single fixed arm may represent a reasonably good benchmark over T rounds. Consider the following simple motivating example:

Example III.2.1. Let $K = 2$ and define

$$P_1 = \begin{bmatrix} 0.5 & 1 \\ 0 & 0.5 \end{bmatrix}, \qquad P_2 = \begin{bmatrix} 0.5 & 0 \\ 1 & 0.5 \end{bmatrix}.$$

Now, assume a preference sequence such that $P_t = P_1$ for the first $\lfloor T/2 \rfloor$ rounds and $P_t = P_2$ for the last $\lceil T/2 \rceil$ rounds. We see that a policy that plays any of the two arms all T rounds, e.g. $Pi_t = 1$ for all $t \in [T]$, has regret $O(1)$ against any fixed benchmark arm, since $\delta_t(1,2) = 1/2$ for the first $T/2$ rounds and $\delta_t(1,2) = -1/2$ for last $T/2$ rounds. However, against a *dynamic benchmark*, e.g. arm 1 for $t < T/2$ and arm 2 for $t \geq T/2$, any policy that plays a fixed arm all T rounds suffers $O(T/2)$ regret (while suffering only constant regret against any fixed benchmark).

Dynamic Regret in Dueling Bandits. Drawing motivation from the above, we seek to formulate a stronger and more meaningful notion of dueling bandit regret, where the benchmark in every round is chosen dynamically based

on P_t. More precisely, letting a_t^* be the CW of P_t, we define *dynamic regret* as

$$\mathrm{DR}(T) \coloneqq \sum_{t=1}^{T} \frac{\delta_t(a_t^*, a_t) + \delta_t(a_t^*, b_t)}{2}.$$

III.2.2 Measures of Non-Stationarity

Clearly, without any control over the amount of non-stationarity in the sequence $\{P_t\}_{t\in[T]}$, it is impossible for any learner to achieve sublinear $o(T)$ dynamic regret in the worst case. To see this, consider the matrices from Example III.2.1 and note that for any choice of arms (a_t, b_t), the adversary can choose a matrix so as to guarantee instantaneous regret of at least $1/2$. This consequently leads to linear regret for the learner, implying that to achieve sublinear dynamic regret, we need to restrict the adversary in terms of the total amount of non-stationarity it can induce in the sequence $P_1, \ldots, P_T$. But what could be a good measure of non-stationarity? In this paper, we study several of these measures, which we will now formally introduce and put in relation to one another.

Paragraph1. Pv. A non-stationarity measure that has been studied in the previous work on `NSt-DB` is the number of times P_t changes [GS22b; KBH22]:

$$S^{\mathsf{P}} \coloneqq \sum_{t=2}^{T} \mathbf{1}\{P_t \neq P_{t-1}\}.$$

However, S^{P} can be a quite pessimistic measure of non-stationarity, as changes in the preference between two suboptimal arms or minor preference shifts that do not change the CW are counted toward S^{P}, whereas they should not significantly affect the performance of a good learning algorithm.

2. Condorcet Winner Switches. A naturally stronger measure of non-stationarity is the total number of Condorcet Winner Switches, i.e. the number of times the identity of a_t^* changes:

$$S^{\mathsf{CW}} \coloneqq \sum_{t=2}^{T} \mathbf{1}\{a_t^* \neq a_{t-1}^*\}.$$

Remark III.2.2 (S^{P} vs S^{CW}). Of course, we always have $S^{\mathsf{CW}} \leq S^{\mathsf{P}}$. In fact, it is easy to construct a simple scenario where $S^{\mathsf{CW}} \ll S^{\mathsf{P}}$: Assume $K = 3$ and consider the following two preference matrices

$$P_1 = \begin{bmatrix} 0.5 & 0.55 & 0.55 \\ 0.45 & 0.5 & 1 \\ 0.45 & 0 & 0.5 \end{bmatrix}, \qquad P_2 = \begin{bmatrix} 0.5 & 0.55 & 0.55 \\ 0.45 & 0.5 & 0 \\ 0.45 & 1 & 0.5 \end{bmatrix},$$

and a preference sequence such that $P_t = P_1$ when t is odd and $P_t = P_2$ otherwise. We then find that $S^{\mathsf{CW}} = 0$ (since 1 is the CW in all rounds t), whereas $S^{\mathsf{P}} = T$.

3. Significant Condorcet Winner Switches. Recently, [SK22b] proposed a new (and strong) notion of non-stationarity for multi-armed bandits, called *Significant Shifts*, that aims to account only for severe distribution shifts and comprises previous complexity measures. We can define a similar concept for dueling bandits: Let $\nu_0 \coloneqq 1$ and define ν_{i+1} recursively as the first round in $[\nu_i, T)$ such that for all arms $a \in [K]$ there exist rounds $\nu_i \leq s_1 < s_2 < \nu_{i+1}$ such that

$$\sum_{t=s_1}^{s_2} \delta_t(a_t^*, a) \geq \sqrt{K(s_2 - s_1)}.$$

Let $\tilde{S}^{\mathsf{CW}}$ denote the number of such *Significant CW Switches* $\nu_1, \ldots, \nu_{\tilde{S}^{\mathsf{CW}}}$. We immediately see that we have $\tilde{S}^{\mathsf{CW}} \leq S^{\mathsf{CW}}$, since not all CW Switches are also Significant CW Switches. For example, a 'non-severe' and quickly reverted change of the Condorcet winner may not be counted towards $\tilde{S}^{\mathsf{CW}}$.

4. Total Variation. Another common notion of non-stationarity studied in the multi-armed bandits literature is the total variation in the rewards [BGZ14; Luo+18]. Its analogue in dueling bandits can be defined as

$$V \coloneqq \sum_{t=2}^{T} \max_{a,b \in [K]} |P_t(a,b) - P_{t-1}(a,b)|,$$

which has been previously studied in [GS22b]. However, V can also be a pessimistic measure of complexity, as it can be of order $O(T)$ even though the Condorcet winner remains fixed throughout all rounds.

5. Condorcet Winner Variation. We can then formulate a more refined version of total variation by accounting only for the maximal drift in the winning probabilities of the current Condorcet winner:

$$\tilde{V} \coloneqq \sum_{t=2}^{T} \max_{a \in [K]} |P_t(a_t^*, a) - P_{t-1}(a_t^*, a)|.$$

Remark III.2.3 (V vs $\tilde{V}$). It is clear from the definition that $\tilde{V} \leq V$. Moreover, we again see that the Condorcet Winner Variation can be much smaller than the Total Variation in the preference sequence, i.e. $\tilde{V} \ll V$. For example, in the problem instance of Remark III.2.2, we find that $\tilde{V} = 0$, whereas $V = T$. Thus, a regret bound in terms of the Condorcet Winner Variation $\tilde{V}$ can potentially be much stronger.

III.3 Proposed Algorithm: ANACONDA

Following recent advances in non-stationary multi-armed bandits [AGL22; AGO19; Che+19] and especially [SK22b], we construct an episode-based algorithm with a carefully chosen replay schedule, called ANACONDA.

Algorithm 10 ANACONDA: Adaptive Non-stationAry CONdorcet Dueling Algorithm

input: horizon T
$t \leftarrow 1$
while $t \leq T$ **do**
  $t_\ell \leftarrow t$ // start of the ℓ-th episode
  $\mathcal{A}_{\text{good}} \leftarrow [K]$
  for $m \in \{2, \dots, 2^{\lceil \log(T) \rceil}\}$ and $s \in \{t_\ell + 1, \dots, T\}$ **do**
    Sample $B_{s,m} \sim \text{Bern}\Big(\frac{1}{\sqrt{m(s-t_\ell)}}\Big)$ // set replay schedule
  Run CondaLet$(t_\ell, T + 1 - t_\ell)$ // root replay in ℓ-th episode

Algorithm 11 CondaLet(t_0, m_0)

input: scheduled time t_0, duration m_0, replay schedule $\{B_{s,m}\}_{s,m}$
initialize: $t \leftarrow t_0$, $\mathcal{A}_t \leftarrow [K]$
while $t \leq T$ and $t \leq t_0 + m_0$ and $\mathcal{A}_{\text{good}} \neq \emptyset$ **do**
  Play arm-pair $(a_t, b_t) \in \mathcal{A}_t$ with each arm being selected w.p. $1/|\mathcal{A}_t|$
  $\mathcal{A}_{\text{good}} \leftarrow \mathcal{A}_{\text{good}} \setminus \{a \in [K] \colon \exists [s_1, s_2] \subseteq [t_\ell, t)$ s.t. (III.2) holds$\}$
  $\mathcal{A}_{\text{local}} \leftarrow \mathcal{A}_t$ // save active set of arms locally
  $t \leftarrow t + 1$
  if $\exists m$ such that $B_{t,m} = 1$ **then** // check for scheduled child replays
    Run CondaLet(t, m) with $m = \max\{m \in \{2, \dots, 2^{\lceil \log(T) \rceil}\} \colon B_{t,m} = 1\}$
  $\mathcal{A}_t \leftarrow \mathcal{A}_{\text{local}} \setminus \{a \in [K] \colon \exists [s_1, s_2] \subseteq [t_0, t)$ s.t. (III.2) holds$\}$

Recall that our goal is to minimize dynamic regret w.r.t. a changing benchmark a_t^*. However, we quickly notice that we cannot reliably track the dynamic regret of some arm $a \in [K]$, i.e. $\sum_t \delta_t(a_t^*, a)$, as the identity of the benchmark, a_t^*, changes at unknown times. As a resolution to this, we aim to detect relevant changes in the preference matrix by tracking the *static regret* $\max_{a' \in [K]} \sum_{t=s_1}^{s_2} \delta_t(a', a)$ instead. It will be the main challenge of our analysis to ensure that properly timed replays will occur (and not too many of these) so that it is in fact sufficient to track the static regret to guarantee low dynamic regret.

In the following, we explain our algorithmic approach in more detail. The algorithm is organized in episodes, denoted ℓ. Similar to recent approaches to non-stationary multi-armed bandits [AGL22; AGO19; SK22b], the algorithm maintains a set of good arms, $\mathcal{A}_{\text{good}}$, and a replay schedule, $\{B_{s,m}\}_{s,m}$, within each episode. When no good arms are left in $\mathcal{A}_{\text{good}}$, a new episode begins and the set of good arms and the replay schedule are being reset. Here, ANACONDA (Algorithm 10) is the meta procedure that initializes each episode by resetting the set of good arms to $[K]$, sampling a new replay schedule, and triggering the root call of CondaLet$(t_\ell, T + 1 - t_\ell)$.

When active in round t, a run of CondaLet(t_0, m_0) (Algorithm 11) samples two arms uniformly at random from the active set of arms at round t, denoted $\mathcal{A}_t$. The set $\mathcal{A}_t$ is globally maintained by all calls of CondaLet and reset to

$[K]$ at the beginning of each replay, i.e. call of `CondaLet`. When a child replay `CondaLet`(t, m) is scheduled in round t, i.e. $B_{t,m} = 1$ for some m, the parent algorithm, say `CondaLet`(t_0, m_0), is interrupted (before eventually resuming if $t \leq t_0 + m_0$ and $\mathcal{A}_{\text{good}} \neq \emptyset$). To not overwrite arm eliminations of a parent by resetting $\mathcal{A}_t$ to $[K]$ in interrupting calls of `CondaLet`, each version of `CondaLet` saves a local set of arms, $\mathcal{A}_{\text{local}}$, before checking for children.

Gap Estimates. Recall the definition of the gap between two arms as $\delta_t(a, b) = P_t(a, b) - 1/2$. Based on observed outcomes of duels, ANACONDA maintains the following importance weighted estimates of $\delta_t(a, b)$:

$$\hat{\delta}_t(a, b) = |\mathcal{A}_t|^2 \mathbf{1}_{\{a_t = a, b_t = b\}} o_t(a, b) - 1/2. \tag{III.1}$$

Wee see that whenever $a, b \in \mathcal{A}_t$, i.e. both arms are in the active set in round t, the estimator $\hat{\delta}_t(a, b)$ is an unbiased estimate of $\delta_t(a, b)$, as we select a pair of arms uniformly at random from $\mathcal{A}_t$ every round (see Line 4 in Algorithm 11).

Elimination Rule. In Line 5 and Line 10 of Algorithm 11, we eliminate an arm $a \in [K]$ in round t if there exist rounds $0 \leq s_1 < s_2 \leq t$ such that

$$\max_{a' \in [K]} \sum_{t=s_1}^{s_2} \hat{\delta}_t(a', a) > C \log(T) K \sqrt{(s_2 - s_1) \vee K^2}, \tag{III.2}$$

where $C > 0$ is some universal constant that does not depend on T, K, or S^{CW}, and can be derived from the regret analysis.

III.3.1 Main Result

The main result of this paper is a $\tilde{O}(\sqrt{S^{\mathsf{CW}} T})$ dynamic regret bound of ANACONDA without knowledge of the number of CW Switches S^{CW}. When $S^{\mathsf{CW}} \ll S^{\mathsf{P}}$, this bound substantially improves upon the *non-adaptive* $\tilde{O}(\sqrt{S^{\mathsf{P}} T})$ rates in [GS22b] and [KBH22]. In particular, as previously mentioned, the number of preference switches S^{P} can be a very pessimistic measure of complexity. For example, a change in the preference between two suboptimal arms, or a minor change of the winning probabilities of the Condorcet winner under which it remains optimal, should not substantially affect our performance (see Remark III.2.2).

Theorem III.3.1 (Dynamic Regret of ANACONDA)**.** *Let S^{CW} denote the unknown number of Condorcet Winner Switches. Let $\tau_1, \ldots, \tau_{S^{\mathsf{CW}}}$ be the unknown times of these switches and let $\tau_0 \coloneqq 1$ and $\tau_{S^{\mathsf{CW}}+1} \coloneqq T$. For some constant $c > 0$, the dynamic regret of ANACONDA is bounded as*

$$\mathrm{DR}(T) \leq c \log^3(T) K \sum_{i=0}^{S^{\mathsf{CW}}} \sqrt{\tau_{i+1} - \tau_i}.$$

An application of Jensen's inequality shows that this implies a dynamic regret bound of order $\tilde{O}(K\sqrt{S^{\mathsf{CW}} T})$, stated in the following corollary.

Corollary III.3.2 (Dynamic Regret w.r.t. S^{CW})**.** *For some constant $c > 0$, the dynamic regret of* ANACONDA *is bounded as*

$$\mathrm{DR}(T) \leq c \log^3(T) K \sqrt{(S^{\mathsf{CW}} + 1)T}.$$

Remark III.3.3 (Regret Lower Bound and Tightness of Theorem III.3.1). Note that a lower bound of $\Omega(\sqrt{KS^{\mathsf{P}}T})$ has recently been shown by [GS22b], which can also be seen to give a lower bound $\Omega(\sqrt{KS^{\mathsf{CW}}T})$ in terms of CW Switches S^{CW} as $S^{\mathsf{CW}} \leq S^{\mathsf{P}}$ (in particular, the lower bound problem instance used in [GS22b] is precisely such that $S^{\mathsf{CW}} = S^{\mathsf{P}}$). As a result, we find that the above bound is optimal up to logarithmic factors in its dependence on S^{CW} and T, whereas its dependence on K may not be tight.

III.4 Regret Analysis of ANACONDA

We build on recent advances in non-stationary multi-armed bandits, which are able to achieve near-optimal dynamic guarantees [AGL22; AGO19; SK22b] without knowledge of the non-stationary complexity. A common basis of the regret analysis in these works is a decomposition of the dynamic regret using the notion of good arms.

Challenges in the Dueling Setting. More precisely, within each episode ℓ, prior work in multi-armed bandits [AGL22; AGO19; SK22b] decomposes the regret of their algorithm's selection, say, a_t into its relative regret against the last good arm $a_\ell^g \in \mathcal{A}_{\text{good}}$, and the relative regret of a_ℓ^g against the best arm, say, a_t^*. A key advantage of this decomposition is that estimating the relative regret of some arm a w.r.t. a_ℓ^g instead of a_t^* is much easier. In particular, since a_ℓ^g is by definition considered good throughout the episode, it is always actively played, which guarantees unbiased estimates of the difference in rewards between any played arm a and the last good arm a_ℓ^g.

However, pairwise preferences are generally not transitive, let alone linear, so that a triangle inequality does not hold, i.e. $\delta_t(a_t^*, a) \not\leq \delta_t(a_t^*, a_\ell^g) + \delta_t(a_\ell^g, a)$. In NSt-DB, we can thus generally not utilize a_ℓ^g, or any other temporarily fixed arm, as a benchmark to detect large regret. Instead, in contrast to prior work in multi-armed bandits, we face the difficulty of having to argue directly that we can guarantee low dynamic regret $\sum_t \delta_t(a_t^*, a)$ without a proxy benchmark such as a_ℓ^g.

Key Ideas to Overcome these Challenges. To overcome these challenges, we consider every fixed arm $a \in [K]$ in isolation and split each episode ℓ into the rounds before arm a gets eliminated from $\mathcal{A}_{\text{good}}$ and the rounds after it gets eliminated from $\mathcal{A}_{\text{good}}$. Letting t_ℓ^a be the elimination round of arm a, we will then argue that t_ℓ^a will occur sufficiently early to guarantee low regret (in episode ℓ) before round t_ℓ^a. For the rounds after elimination from $\mathcal{A}_{\text{good}}$, it will be key to dissect each possible replay of the eliminated arm and obtain replay-specific

regret bounds, where we distinguish between 'confined' and 'unconfined' replays of arms. We now give an outline of our regret analysis.

III.4.1 Proof Sketch of Theorem III.3.1

In the following, we let $\tilde{c} > 0$ denote a positive constant that does not dependent on T, K, or S^{CW}, but may change from line to line. To begin our analysis, we state a concentration bound on the martingale difference sequence $\hat{\delta}_t(a,b) - \mathbb{E}[\hat{\delta}_t(a,b) \mid \mathcal{F}_{t-1}]$ as it can be found in similar form in [Bey+11] and [SK22b].

Lemma III.4.1. *Let $\mathcal{E}$ be the event that for all rounds $1 \leq s_1 < s_2 \leq T$ and all arms $a, b \in [K]$:*

$$\left|\sum_{t=s_1}^{s_2} \hat{\delta}_t(a,b) - \sum_{t=s_1}^{s_2} \mathbb{E}\left[\hat{\delta}_t(a,b) \mid \mathcal{F}_{t-1}\right]\right| \leq \tilde{c}\log(T)\left(K\sqrt{(s_2 - s_1)} + K^2\right) \quad \text{(III.3)}$$

for a sufficiently large constant $\tilde{c} > 0$ and where $\mathcal{F} = \{\mathcal{F}_t\}_{t\in\mathbb{N}_0}$ denotes the canonical filtration. Then, event $\mathcal{E}$ occurs with probability at least $1 - 1/T^2$.

Note that our elimination rule (III.2) has been chosen in accordance with the above concentration bound. In particular, let t_ℓ^a denote the round in episode ℓ in which arm a is eliminated from $\mathcal{A}_{\text{good}}$. Then, on the concentration event $\mathcal{E}$, if $a' \in \mathcal{A}_{\text{good}}$ for all $t_\ell \leq t < t_\ell^a$, we must have

$$\sum_{t=t_\ell}^{t_\ell^a - 1} \delta_t(a', a) = \sum_{t=t_\ell}^{t_\ell^a - 1} \mathbb{E}\left[\hat{\delta}_t(a', a) \mid \mathcal{F}_{t-1}\right] \leq \tilde{c}\log(T) K \sqrt{(t_\ell^a - t_\ell) \vee K^2},$$

where the initial identity holds as $\hat{\delta}_t(a', a)$ is unbiased when $a, a' \in \mathcal{A}_t$ and the inequality follows from the elimination rule (III.2) and the concentration bound (III.3). However, note that the above crucially used that both a and a' are actively played throughout the interval $[t_\ell, t_\ell^a)$, as we are otherwise not able to accurately estimate $\sum_t \delta_t(a', a)$. It will be the primary challenge of our analysis to ensure that through properly timed replays, i.e. calls of `CondaLet`, we can obtain unbiased estimates w.r.t. the changing CW that allow us to eliminate bad arms before they amass large regret.

Bounding Regret Within Episodes. We proceed by bounding regret within each episode separately. Recall that we let $\tau_1 < \ldots < \tau_{S^{\mathsf{CW}}}$ denote the (unknown) rounds in which the Condorcet winner changes. We then refer to the interval $[\tau_i, \tau_{i+1})$ as the i-th phase, i.e. the interval for which $a_t^* = a_{\tau_i}^*$ for all $t \in [\tau_i, \tau_{i+1})$. Let $\text{Phases}(t_1, t_2) = \{i : [\tau_i, \tau_{i+1}) \cap [t_1, t_2) \neq \emptyset\}$ be the set of phases i such that $[\tau_i, \tau_{i+1})$ intersects with the interval $[t_1, t_2)$. Our main claim is the following

upper bound on the dynamic regret within each episode:

$$\mathbb{E}\left[\sum_{t=t_\ell}^{t_{\ell+1}-1} \frac{\delta_t(a_t^*, a_t) + \delta_t(a_t^*, b_t)}{2}\right] \leq \tilde{c}K\log^3(T)\,\mathbb{E}\left[\sum_{i\in\text{Phases}(t_\ell, t_{\ell+1})} \sqrt{\tau_{i+1} - \tau_i}\right]. \tag{III.4}$$

By conditioning on t_ℓ and carefully applying the tower property, we can rewrite the expected dynamic regret within an episode in terms of fixed arms $a \in [K]$:

Lemma III.4.2. *We have*

$$\mathbb{E}\left[\sum_{t=t_\ell}^{t_{\ell+1}-1} \frac{\delta_t(a_t^*, a_t) + \delta_t(a_t^*, b_t)}{2}\right] = \mathbb{E}\left[\sum_{a=1}^{K}\sum_{t=t_\ell}^{t_{\ell+1}-1} \frac{\delta_t(a_t^*, a)}{|\mathcal{A}_t|}\mathbf{1}_{\{a\in\mathcal{A}_t\}}\right].$$

In a next step, we split the RHS into the rounds before a fixed arm $a \in [K]$ has been eliminated from the good set, and the rounds after its elimination. Recall t_ℓ^a to be the round in episode ℓ in which arm a is eliminated from $\mathcal{A}_{\text{good}}$ and consider

$$\underbrace{\mathbb{E}\left[\sum_{a=1}^{K}\sum_{t=t_\ell}^{t_\ell^a-1} \frac{\delta_t(a_t^*, a)}{|\mathcal{A}_t|}\right]}_{R_1(\ell)} + \underbrace{\mathbb{E}\left[\sum_{a=1}^{K}\sum_{t=t_\ell^a}^{t_{\ell+1}-1} \frac{\delta_t(a_t^*, a)}{|\mathcal{A}_t|}\mathbf{1}_{\{a\in\mathcal{A}_t\}}\right]}_{R_2(\ell)},$$

where we could drop the indicator in $R_1(\ell)$, since $\mathcal{A}_{\text{good}} \subseteq \mathcal{A}_t$ by construction of these sets. The remainder of our analysis is mostly concerned with showing that both, $R_1(\ell)$ and $R_2(\ell)$, are upper bounded by the RHS in (III.4).

Regret Before Elimination. The main difficulty in bounding $R_1(\ell)$ lies in the fact that some arm could have been eliminated due to being suboptimal, only to become the Condorcet winner shortly after. As a result, large regret could go undetected, as the current Condorcet winner is not being actively played anymore. To this end, we have to argue that with high probability there will always be a replay scheduled that eliminates any bad arm from $\mathcal{A}_{\text{good}}$ in a timely manner, thereby eventually triggering a restart.

Here, we specifically consider calls of `CondaLet`(s, m) that provably eliminate bad arms from $\mathcal{A}_{\text{good}}$. Importantly, by construction of our elimination rule (III.2), we can guarantee on the concentration event $\mathcal{E}$ that any run of `CondaLet` that is scheduled within some phase i will actively play the Condorcet winner of said phase.

Lemma III.4.3. *On event $\mathcal{E}$, no call of* `CondaLet`(s, m) *with $\tau_i \leq s < \tau_{i+1}$ eliminates arm a_i^* before round τ_{i+1}.*

Roughly speaking, we can then argue that a replay that eliminates arm a will be scheduled with high probability before the smallest round $s(a) > t_\ell$ such that

$\sum_{t=t_\ell}^{s(a)} \delta_t(a_t^*, a) \gtrsim \sqrt{s(a) - t_\ell}$. In other words, arm a is going to be eliminated from $\mathcal{A}_{\text{good}}$ before it suffers too much regret. Since t_ℓ^a is defined as the round in episode ℓ in which a is eliminated from $\mathcal{A}_{\text{good}}$, we must have $t_\ell^a < s(a)$, which implies that the inner sum in $R_1(\ell)$ is at most of order $\sqrt{t_\ell^a - t_\ell}$ for every fixed arm $a \in [K]$. Finally, using that

$$\sqrt{t_\ell^a - t_\ell} \leq \sum_{i \in \text{Phases}(t_\ell, t_\ell^a)} \sqrt{\tau_{i+1} - \tau_i}$$

and summing over all arms, we obtain the desired bound of (III.4). Note that here summing over arms can be seen to account for a $\log(K)$ factor which we coarsely upper bound by $\log(T)$.

Regret After Elimination. $R_2(\ell)$ can be viewed as the regret due to replaying arms after they have been eliminated from the good set $\mathcal{A}_{\text{good}}$. We here distinguish between two types of replays, i.e. calls of `CondaLet`:

Definition III.4.4. We call `CondaLet`(s, m) *confined* if there exists $i \in \text{Phases}(t_\ell, T)$ s.t. $[s, s+m] \subseteq [\tau_i, \tau_{i+1})$. In turn, we say that `CondaLet`(s, m) is *unconfined* if for all $i \in \text{Phases}(t_\ell, T)$, we have $[s, s+m] \subseteq [\tau_i, \tau_{i+1})$.

To bound the regret within a confined replay, we recall that according to Lemma III.4.3, on the concentration event $\mathcal{E}$, no call of `CondaLet` will eliminate the Condorcet winner within the phase it is scheduled in. Thus, whenever some arm a is being played by a confined replay, we obtain unbiased estimates of $\delta_t(a_t^*, a)$. It is then straightforward to show that for any confined `CondaLet`(s, m), we have that $\sum_{t=s}^{s+m} \delta_t(a_t^*, a)$ is at most of order $\sqrt{m}$.

A similar line of argument does not work for unconfined replays, as they intersect with several phases. We then face a similar difficulty as when bounding $R_1(\ell)$, where the Condorcet winner of the current phase could have been eliminated (from the replay) in an earlier phase. Using similar arguments than for bounding $R_1(\ell)$, we show that for any unconfined `CondaLet`(s, m), we have that $\sum_{t=s}^{s+m} \delta_t(a_t^*, a)$ is at most of order $\sqrt{s - t_\ell} + \sqrt{m}$.

Lastly, recall that in episode ℓ a replay `CondaLet`(s, m) is scheduled with probability $1/\sqrt{m(s - t_\ell)}$. Crucially, any unconfined `CondaLet` scheduled in $[\tau_i, \tau_{i+1})$ must have duration at least $m \geq \tau_{i+1} - s$ (otherwise it is not unconfined). Careful summation over confined and unconfined `CondaLet` then yields the desired upper bound (III.4).

Counting Episodes. Lastly, we show that ANACONDA only restarts if there has been a CW switch.

Lemma III.4.5. *On event $\mathcal{E}$, for all episodes ℓ but the last there exists a change of the CW $t_\ell \leq \tau_i < t_{\ell+1}$.*

This follows directly from the fact that on the concentration event within a single phase the CW will never be eliminated from $\mathcal{A}_{\text{good}}$. Thus, if there is a restart, i.e. every arm has been eliminated from $\mathcal{A}_{\text{good}}$, there must have been

a change of CW. Lemma III.4.5 thus tells us that any phase intersects with at most two episodes. Summing the RHS of (III.4) over episodes then gives the claimed upper bound of

$$\mathbb{E}\left[\sum_{t=1}^{T}\frac{\delta_t(a_t^*,a_t)+\delta_t(a_t^*,b_t)}{2}\right] \leq 2\tilde{c}K\log^3(T)\mathbb{E}\left[\sum_{i=1}^{S^{\mathsf{CW}}}\sqrt{\tau_{i+1}-\tau_i}\right].$$

A detailed proof of Theorem III.3.1 is given in Appendix C.1.

III.5 Tighter Bounds Under SST and STI

We show that ANACONDA can in fact yield a stronger regret guarantee in terms of a more refined notion of non-stationarity, Significant Condorcet Winner Switches (see Section III.2.2), under additional assumptions on the preference sequence $P_1,\dots,P_T$: Strong Stochastic Transitivity (SST) and Stochastic Triangle Inequality (STI) [YJ09; YJ11; Yue+12]. Let $a,b,c \in [K]$ and let $a \succ_t b$ denote that a is preferred over b in round t.

Assumption III.5.1 (Strong Stochastic Transitivity)**.** Every preference matrix P_t satisfies that if $a \succ_t b \succ_t c$, we have $\delta_t(a,c) \geq \delta_t(a,b) \vee \delta_t(b,c)$.

Assumption III.5.2 (Stochastic Triangle Inequality)**.** Every preference matrix P_t satisfies that if $a \succ_t b \succ_t c$, we have $\delta_t(a,c) \leq \delta_t(a,b) + \delta_t(b,c)$.

Remark III.5.3 (Example of SST & STI). Among the preference models that satisfy Assumption III.5.1 and Assumption III.5.2, are utility-based models with a symmetric and monotonically increasing link function σ. In these models, every arm a has an associated (time-dependent) utility $u_t(a)$ and the probability of arm a winning a duel against arm b is given by $P_t(a \succ b) = \sigma(u_t(a) - u_t(b))$, where σ is an increasing function satisfying $\sigma(x) = 1 - \sigma(-x)$ and $\sigma(0) = 1/2$ that maps utility differences to probabilities [Ben+21; Yue+12].

III.5.1 Improved Dynamic Regret Analysis

We now show that ANACONDA achieves strong regret guarantees in terms of Significant CW Switches and CW Variation under SST and STI.

Significant Condorcet Winner Switches. Under Assumption III.5.1 and Assumption III.5.2, we are able to obtain the following adaptive dynamic regret bound in terms of $\tilde{S}^{\mathsf{CW}}$.

Theorem III.5.4. *Let $\tilde{S}^{\mathsf{CW}}$ be the unknown number of Significant Condorcet Winner Switches. Under Assumption 1 and Assumption 2,* ANACONDA *has dynamic regret $\tilde{O}\big(K\sqrt{\tilde{S}^{\mathsf{CW}}T}\big)$.*

Remark III.5.5. Recall from Section III.2.2, since $\tilde{S}^{\mathsf{CW}} \leq S^{\mathsf{CW}}$ (as not all CW Switches are also Significant CW Switches), Theorem III.5.4 gives a tighter

dynamic regret guarantee for the class of non-stationary preference sequences with SST and STI. Also note that this bound does not violate the $\Omega(\sqrt{KS^{\mathsf{P}}T})$ lower bound from III.3.3, as the lower bound is shown for a worst-case preference sequence $P_1, \dots, P_T$ where $\tilde{S}^{\mathsf{CW}} = S^{\mathsf{CW}} = S^{\mathsf{P}}$.

Proof Overview. With some additional effort, Assumption III.5.1 and Assumption III.5.2 allow us to utilize a dynamic regret decomposition similar to prior work in non-stationary multi-armed bandits [AGL22; AGO19; SK22b]. Roughly speaking, this allows us to reuse the regret analysis for CW Switches (Theorem III.3.1) in the analysis under Significant CW Switches. ■

We want to give a brief intuition about why additional assumptions are necessary when bounding dynamic regret w.r.t. Significant CW Switches $\tilde{S}^{\mathsf{CW}}$ opposed to CW Switches S^{CW}.[2] Consider a phase $[\nu_i, \nu_{i+1})$ in the sense of Significant CW Switches as defined in Section III.2.2. As previously mentioned, the definition of a Significant CW Switch allows for several (non-severe) CW changes within each phase $[\nu_i, \nu_{i+1})$. As a result, we cannot guarantee that there will be any intervals during which the CW remains fixed, which would enable us to accurately estimate the relative regret $\sum_t \delta_t(a_t^*, a)$ so as to eliminate bad arms. Broadly speaking, assuming a sort of transitivity (i.e. SST and STI) enables us to identify bad arms based on knowledge of $\sum_t \delta_t(a', a)$ for some temporarily fixed benchmark a'. More details and a complete proof can be found in Appendix C.2.

Condorcet Winner Variation. Recall the definition of the Condorcet Winner Variation $\tilde{V}$ from Section III.2.2. As a consequence of Theorem III.5.4, we can show that ANACONDA also achieves near-optimal dynamic regret w.r.t. $\tilde{V}$.

Corollary III.5.6. *Let $\tilde{V}$ be the unknown Condorcet Winner Variation. Under Assumption III.5.1 and Assumption III.5.2, ANACONDA has dynamic regret $\tilde{O}\big(K\sqrt{T} + \tilde{V}^{1/3}(KT)^{2/3}\big)$.*

Remark III.5.7. By definition, we have $\tilde{V} \leq V$, which means that Corollary III.5.6 may yield a tighter dynamic regret bound than the (non-adaptive) guarantee w.r.t. V in [GS22b]. In view of the lower bound of $\Omega\big((KV)^{1/3}T^{2/3}\big)$ shown in [GS22b], the regret guarantee of ANACONDA is also tight up to logarithmic factors and a factor of $K^{1/3}$. Note once again that the lower bound in [GS22b] is not violated as their lower bound uses a worst-case preference sequence $P_1, \dots, P_T$ where $\tilde{V} = V$.

III.6 Discussion

We studied the problem of dynamic regret minimization in non-stationary dueling bandits and proposed an adaptive algorithm that yields provably optimal regret

[2] Note that this is a limitation of our regret analysis. It is an open question whether it is possible to achieve $O(\sqrt{\tilde{S}^{\mathsf{CW}}T})$ dynamic regret in NSt-DB with general preference models.

guarantees in terms of strong notions of non-stationary complexity. Our proposed algorithm is the first to achieve optimal dynamic dueling bandit regret without prior knowledge of the underlying non-stationary complexity. While our results certainly close some of the practical open problems in preference elicitation in time-varying preference models, it also leads to plethora of new questions along the line. We provide an outlook to future directions and open problems in the supplementary material.

Future Work. While our results certainly address some of the practical open problems for preference elicitation in time-varying preference models, it also leads to plethora of new questions along the line. In particular, as an extension to this work, one obvious question would be to understand non-stationary dueling bandits for more general preference matrices: What happens if the preference sequences do not have a Condorcet winner in each round? What could be a good dynamic benchmark in that case? Hereto related, another open question is whether it is possible to obtain dynamic regret bounds in terms of Significant CW Switches ($\tilde{S}^{\mathsf{CW}}$) for general preference sequences (without transitivity assumptions). Extending the considered pairwise preference setting to more general subsetwise feedback [GS22a; SG18; SG19b; SG20b] would be another interesting direction from a practical point of view.

References

[AB10] Audibert, J.-Y. and Bubeck, S. "Best arm identification in multi-armed bandits". In: *COLT-23th Conference on Learning Theory-2010*. 2010, 13–p.

[ACF02] Auer, P., Cesa-Bianchi, N., and Fischer, P. "Finite-time analysis of the multiarmed bandit problem". In: *Machine learning* vol. 47, no. 2-3 (2002), pp. 235–256.

[AFM17] Allesiardo, R., Féraud, R., and Maillard, O.-A. "The non-stationary stochastic multi-armed bandit problem". In: *International Journal of Data Science and Analytics* vol. 3 (2017), pp. 267–283.

[AG12] Agrawal, S. and Goyal, N. "Analysis of Thompson sampling for the multi-armed bandit problem". In: *Conference on Learning Theory*. 2012, pp. 39–1.

[AGL22] Abbasi-Yadkori, Y., Gyorgy, A., and Lazic, N. "A New Look at Dynamic Regret for Non-Stationary Stochastic Bandits". In: *arXiv preprint arXiv:2201.06532* (2022).

[AGO19] Auer, P., Gajane, P., and Ortner, R. "Adaptively tracking the best bandit arm with an unknown number of distribution changes". In: *In Proceedings of the 32nd International Conference on Learning Theory* vol. 99 (2019), pp. 138–158.

[AKJ14] Ailon, N., Karnin, Z. S., and Joachims, T. "Reducing Dueling Bandits to Cardinal Bandits." In: *ICML*. Vol. 32. 2014, pp. 856–864.

[BC+12] Bubeck, S., Cesa-Bianchi, N., et al. "Regret analysis of stochastic and nonstochastic multi-armed bandit problems". In: *Foundations and Trends® in Machine Learning* vol. 5, no. 1 (2012), pp. 1–122.

[Ben+21] Bengs, V. et al. "Preference-based Online Learning with Dueling Bandits: A Survey." In: *Journal of Machine Learning Research* (2021).

[Bey+11] Beygelzimer, A. et al. "Contextual bandit algorithms with supervised learning guarantees". In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. JMLR Workshop and Conference Proceedings. 2011, pp. 19–26.

[BGZ14] Besbes, O., Gur, Y., and Zeevi, A. "Stochastic multi-armed-bandit problem with non-stationary rewards". In: *Advances in Neural Information Processing Systems* vol. 27 (2014), pp. 199–207.

[BGZ15] Besbes, O., Gur, Y., and Zeevi, A. "Non-stationary stochastic optimization". In: *Operations research* vol. 63, no. 5 (2015), pp. 1227–1244.

[Che+19] Chen, Y. et al. "A new algorithm for non-stationary contextual bandits: Efficient, optimal, and parameter-free". In: *In Proceedings of the 32nd Conference on Learning Theory* vol. 99 (2019), pp. 1–30.

[Dud+15a] Dudik, M. et al. "Contextual Dueling Bandits". In: *Conference on Learning Theory* (2015), pp. 563–587.

[Dud+15b] Dudík, M. et al. "Contextual Dueling Bandits". In: *Conference on Learning Theory*. 2015, pp. 563–587.

[GM11] Garivier, A. and Moulines, E. "On upper-confidence bound policies for switching bandit problems". In: *International Conference on Algorithmic Learning Theory*. Springer. 2011, pp. 174–188.

[GS22a] Ghoshal, S. and Saha, A. "Exploiting Correlation to Achieve Faster Learning Rates in Low-Rank Preference Bandits". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2022, pp. 456–482.

[GS22b] Gupta, S. and Saha, A. "Optimal and efficient dynamic regret algorithms for non-stationary dueling bandits". In: *International Conference on Machine Learning*. PMLR. 2022, pp. 19027–19049.

[GUC15] Gajane, P., Urvoy, T., and Clérot, F. "A Relative Exponential Weighing Algorithm for Adversarial Utility-based Dueling Bandits". In: *Proceedings of the 32nd International Conference on Machine Learning*. 2015, pp. 218–227.

[HWD11] Hofmann, K., Whiteson, S., and De Rijke, M. "A probabilistic method for inferring preferences from clicks". In: *Proceedings of the 20th ACM international conference on Information and knowledge management*. 2011, pp. 249–258.

[KBH22] Kolpaczki, P., Bengs, V., and Hüllermeier, E. "Non-Stationary Dueling Bandits". In: *arXiv preprint arXiv:2202.00935* (2022).

[Kom+15] Komiyama, J. et al. "Regret Lower Bound and Optimal Algorithm in Dueling Bandit Problem." In: *COLT*. 2015, pp. 1141–1154.

[LS18] Lattimore, T. and Szepesvári, C. "Bandit Algorithms". In: *preprint* (2018).

[Luo+18] Luo, H. et al. "Efficient contextual bandits in non-stationary worlds". In: *In Proceedings of the 31st Conference On Learning Theory* vol. 75 (2018), pp. 1739–1776.

[RC13] Radlinski, F. and Craswell, N. "Optimized interleaving for online retrieval evaluation". In: *Proceedings of the sixth ACM international conference on Web search and data mining*. 2013, pp. 245–254.

[RCG20] Russac, Y., Cappé, O., and Garivier, A. "Algorithms for non-stationary generalized linear bandits". In: *arXiv preprint arXiv:2003.10113* (2020).

[Rob52] Robbins, H. "Some aspects of the sequential design of experiments". In: *Bulletin of the American Mathematical Society* vol. 58, no. 5 (1952), pp. 527–535.

[RVC19] Russac, Y., Vernade, C., and Cappé, O. "Weighted linear bandits for non-stationary environments". In: *Advances in Neural Information Processing Systems* vol. 32 (2019).

[SG18] Saha, A. and Gopalan, A. "Battle of Bandits". In: *Uncertainty in Artificial Intelligence*. 2018.

[SG19a] Saha, A. and Gopalan, A. "Combinatorial bandits with relative feedback". In: *Advances in Neural Information Processing Systems*. 2019.

[SG19b] Saha, A. and Gopalan, A. "PAC Battling Bandits in the Plackett-Luce Model". In: *Algorithmic Learning Theory*. 2019, pp. 700–737.

[SG20a] Saha, A. and Gopalan, A. "Best-item learning in random utility models with subset choices". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2020, pp. 4281–4291.

[SG20b] Saha, A. and Gopalan, A. "From PAC to instance-optimal sample complexity in the Plackett-Luce model". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 8367–8376.

[SG21] Saha, A. and Gaillard, P. "Dueling Bandits with Adversarial Sleeping". In: *Advances in Neural Information Processing Systems* vol. 34 (2021), pp. 27761–27771.

[SG22] Saha, A. and Gaillard, P. “Versatile Dueling Bandits: Best-of-both-World Analyses for Online Learning from Preferences”. In: *International Conference on Machine Learning*. PMLR. 2022.

[SK22a] Saha, A. and Krishnamurthy, A. “Efficient and Optimal Algorithms for Contextual Dueling Bandits under Realizability”. In: *International Conference on Algorithmic Learning Theory*. PMLR. 2022, pp. 968–994.

[SK22b] Suk, J. and Kpotufe, S. “Tracking Most Significant Arm Switches in Bandits”. In: *Conference on Learning Theory*. PMLR. 2022, pp. 2160–2182.

[SKM21] Saha, A., Koren, T., and Mansour, Y. “Adversarial Dueling Bandits”. In: *International Conference on Machine Learning*. PMLR. 2021, pp. 9235–9244.

[SPX13] Soufiani, H. A., Parkes, D. C., and Xia, L. “Preference Elicitation For General Random Utility Models”. In: *Uncertainty in Artificial Intelligence*. Citeseer. 2013, p. 596.

[Sui+17] Sui, Y. et al. “Multi-dueling bandits with dependent arms”. In: *Conference on Uncertainty in Artificial Intelligence*. UAI’17. 2017.

[Sui+18] Sui, Y. et al. “Advancements in Dueling Bandits.” In: *IJCAI*. 2018, pp. 5502–5510.

[Tho33] Thompson, W. R. “On the likelihood that one unknown probability exceeds another in view of the evidence of two samples”. In: *Biometrika* vol. 25, no. 3-4 (1933), pp. 285–294.

[WIW18] Wu, Q., Iyer, N., and Wang, H. “Learning contextual bandits in a non-stationary environment”. In: *In Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval* (2018), pp. 495–504.

[WL16] Wu, H. and Liu, X. “Double Thompson sampling for dueling bandits”. In: *Advances in Neural Information Processing Systems*. 2016, pp. 649–657.

[YJ09] Yue, Y. and Joachims, T. “Interactively optimizing information retrieval systems as a dueling bandits problem”. In: *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM. 2009, pp. 1201–1208.

[YJ11] Yue, Y. and Joachims, T. “Beat the mean bandit”. In: *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*. 2011, pp. 241–248.

[Yue+12] Yue, Y. et al. “The k-armed dueling bandits problem”. In: *Journal of Computer and System Sciences* vol. 78, no. 5 (2012), pp. 1538–1556.

[Zog+14a] Zoghi, M. et al. "Relative confidence sampling for efficient on-line ranker evaluation". In: *Proceedings of the 7th ACM international conference on Web search and data mining*. ACM. 2014, pp. 73–82.

[Zog+14b] Zoghi, M. et al. "Relative upper confidence bound for the k-armed dueling bandit problem". In: *JMLR Workshop and Conference Proceedings*. 32. JMLR. 2014, pp. 10–18.

[Zog+15] Zoghi, M. et al. "Copeland dueling bandits". In: *Advances in Neural Information Processing Systems*. 2015, pp. 307–315.

Notation

a_t, b_t	Arms selected by the algorithm in round t
a, a', b	Generic fixed arms in $[K]$
$\delta_t(a,b)$	Gap between arm a and arm b
$\hat{\delta}_t(a,b)$	Importance weighted gap estimate
a_t^*	Condorcet winner in round t
t_ℓ	First round in the ℓ-th episode
t_ℓ^a	Round in the ℓ-th episode in which a is eliminated from $\mathcal{A}_{\text{good}}$
S^{CW}	Number of Condorcet Winner Switches
$\tau_1, \dots, \tau_{S^{\mathsf{CW}}}$	Rounds in which the Condorcet winner changes
a_i^*	Condorcet winner in phase $i \in [S^{\mathsf{CW}}]$, i.e. $a_t^* = a_i^*$ for $t \in [\tau_i, \tau_{i+1})$
$\tilde{S}^{\mathsf{CW}}$	Number of Significant Condorcet Winner Switches
$\nu_1, \dots, \nu_{\tilde{S}^{\mathsf{CW}}}$	Rounds of Significant CW Switches
a_i^s	Last safe arm in phase $[\nu_i, \nu_{i+1})$, i.e. last arm to satisfy (29)
$\tilde{V}$	Condorcet Winner Variation

C.1 Proof of Theorem III.3.1

We organize the proof of Theorem III.3.1 as follows. Section C.1.1 contains basic preliminary facts that will be the foundation of the upcoming proof. Section C.1.2 then bounds the regret any fixed arm suffers within each episode *before* being eliminated from the good set. Complementary to this, Section C.1.3 then deals with the regret an arm suffers *after* being eliminated.

C.1.1 Preliminaries

In this preliminary section, we introduce a concentration bound on the sum of our estimates $\hat{\delta}_t$ in Section C.1.1.1. We then show in Section C.1.1.2 that the beginning of a new episode implies that the Condorcet winner has changed (on the concentration event), which will be useful later. Finally, Section C.1.1.3

decomposes the regret in terms episodes, arms, and rounds, which will form the basis of our analysis.

C.1.1.1 Martingale Concentration Bound

We will rely on a similar martingale tail bound as [Bey+11] and [SK22b], which is based on a version of Freedman's inequality given below.

Lemma C.1.1 (Theorem 1 in [Bey+11])**.** *Let $(X_t)_{t\in\mathbb{N}}$ be a martingale difference sequence w.r.t. some filtration $(\mathcal{F}_t)_{t\in\mathbb{N}_0}$. Assume that is X_t is almost surely uniformly bounded, i.e. $X_t \leq R$ a.s. for some constant R. Moreover, suppose that $\sum_{s=1}^{t} \mathbb{E}[X_s^2 \mid \mathcal{F}_{s-1}] \leq V_t$ a.s. for some sequence of constants $(V_t)_{t\in\mathbb{N}}$. Then, for any $\delta \in (0,1)$, with probability at least $1-\delta$, we have*

$$\sum_{s=1}^{t} X_s \leq (e-1)\left(\sqrt{V_t \log(1/\delta)} + R\log(1/\delta)\right). \tag{5}$$

Proof. See Theorem 1 in [Bey+11] and Lemma 1 in [SK22b]. ■

We now apply the above concentration bound to the martingale difference sequence $\hat{\delta}_t(a,b) - \mathbb{E}[\hat{\delta}_t(a,b) \mid \mathcal{F}_{t-1}]$.

Lemma C.1.2. *Let $\mathcal{E}$ be the event that for all rounds $s_1 < s_2$ and all arms $a, b \in [K]$:*

$$\left|\sum_{t=s_1}^{s_2} \hat{\delta}_t(a,b) - \sum_{t=s_1}^{s_2} \mathbb{E}\left[\hat{\delta}_t(a,b) \mid \mathcal{F}_{t-1}\right]\right| \leq c_1 \log(T)\left(K\sqrt{(s_2-s_1)} + K^2\right) \tag{6}$$

for an appropriately large constant $c_1 > 0$ and where $\mathcal{F} = \{\mathcal{F}_t\}_{t\in\mathbb{N}_0}$ is the canonical filtration generated by observations in past rounds. Then, event $\mathcal{E}$ occurs with probability at least $1 - 1/T^2$.

Proof. Note that $\hat{\delta}_t(a,b) - \mathbb{E}[\hat{\delta}_t(a,b) \mid \mathcal{F}_{t-1}]$ is naturally a martingale difference, since $\mathbb{E}\big[\hat{\delta}_t(a,b) - \mathbb{E}[\hat{\delta}_t(a,b) \mid \mathcal{F}_{t-1}] \mid \mathcal{F}_{t-1}\big] = 0$ a.s. Using that $|\mathcal{A}_t| \leq K$, we have that $X_t \leq 2K^2$ a.s. for all rounds t. Moreover, we get that

$$\begin{aligned}\sum_{t=s_1}^{s_2} \mathbb{E}\left[\hat{\delta}_t^2(a,b) \mid \mathcal{F}_{t-1}\right] &\leq \sum_{t=s_1}^{s_2} |\mathcal{A}_t|^4 \mathbb{E}\left[\mathbf{1}_{\{a_t=a, b_t=b\}} \mid \mathcal{F}_{t-1}\right] \\ &= \sum_{t=s_1}^{s_2} |\mathcal{A}_t|^2 \leq K^2(s_2-s_1).\end{aligned}$$

We can thus apply Lemma C.1.1 with $R = K^2$ and $V_t = 2K^2 t$. Using $|x-y| \leq |x| + |y|$ and taking union bounds over a, b and s_1, s_2, we then obtain Lemma C.1.2. ■

C.1.1.2 Episodes and Condorcet Winner Switches

Lemma C.1.3. *On event $\mathcal{E}$, for each episode $[t_\ell, t_{\ell+1})$ with $t_{\ell+1} \leq T$, there exists a change of the CW $\tau_i \in [t_\ell, t_{\ell+1})$.*

This implies that any phase $[\tau_i, \tau_{i+1})$ will intersect with at most two episodes.

Proof. The start of a new episode means that every arm $a \in [K]$ has been eliminated from $\mathcal{A}_{\text{good}}$ at some round in $t_\ell^a \in [t_\ell, t_{\ell+1})$. As a result, there must exist an interval $[s_1, s_2] \subseteq [t_\ell, t_\ell^a)$ and some arm $a' \in [K]$ so that the elimination rule (III.2) holds. Using Lemma C.1.2, we then find that for some constant $c_2 > 0$:

$$\sum_{t=s_1}^{s_2} \mathbb{E}\left[\hat{\delta}_t(a', a) \mid \mathcal{F}_{t-1}\right] > c_2 \log(T) K \sqrt{(s_2 - s_1) \vee K^2}. \tag{7}$$

Note that by construction of $\hat{\delta}_t(a', a)$, we always have $\delta_t(a', a) \geq \mathbb{E}[\hat{\delta}_t(a', a) \mid \mathcal{F}_{t-1}]$ since

$$\mathbb{E}[\hat{\delta}_t(a', a) \mid \mathcal{F}_{t-1}] = \begin{cases} \delta_t(a', a) & a', a \in \mathcal{A}_t \\ -1/2 & \text{otherwise.} \end{cases} \tag{8}$$

Thus, in view of inequality (7), there exists no arm $a \in [K]$ such that $\max_{a'} \delta_t(a', a) = 0$ for all $t \in [t_\ell, t_{\ell+1})$, i.e. no fixed arm is optimal throughout the episode and there must have been a change of Condorcet winner. ■

C.1.1.3 Decomposing Regret across Episodes and Arms

We will bound regret of the algorithm withing each episode separately, i.e. we consider

$$\mathbb{E}\left[\sum_{t=t_\ell}^{t_{\ell+1}-1} \frac{\delta_t(a_t^*, a_t) + \delta_t(a_t^*, b_t)}{2}\right], \tag{9}$$

where t_ℓ is the first round in episode ℓ and a_t^* is the Condorcet winner in round $t \in [T]$.

Recall that, every round $t \in [T]$, the algorithm selects an arm a uniformly at random from the active set $\mathcal{A}_t$. It will then be useful to rewrite (11) in terms of fixed arms $a \in [K]$.

Lemma C.1.4. *We can write* (11) *in terms of the regret suffered by fixed arms:*

$$\mathbb{E}\left[\sum_{t=t_\ell}^{t_{\ell+1}-1} \frac{\delta_t(a_t^*, a_t) + \delta_t(a_t^*, b_t)}{2}\right] = \mathbb{E}\left[\sum_{a=1}^{K} \sum_{t=t_\ell}^{t_{\ell+1}} \frac{\delta_t(a_t^*, a)}{|\mathcal{A}_t|} \mathbf{1}_{\{a \in \mathcal{A}_t\}}\right] \tag{10}$$

Proof. As the algorithm independently and symmetrically selects two arms (a_t, b_t) in each round (Line 4 in Algorithm 11), we can focus on bounding regret for one of the two arms, say a_t, by writing

$$\mathbb{E}\left[\sum_{t=t_\ell}^{t_{\ell+1}-1} \frac{\delta_t(a_t^*, a_t) + \delta_t(a_t^*, b_t)}{2}\right] = \mathbb{E}\left[\sum_{t=t_\ell}^{t_{\ell+1}-1} \delta_t(a_t^*, a_t)\right]. \tag{11}$$

Conditioning on t_ℓ and using the tower property, we then further find that

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=t_\ell}^{t_{\ell+1}} \delta_t(a_t^*, a_t)\right] &= \mathbb{E}\left[\mathbb{E}\left[\sum_{t=t_\ell}^{t_{\ell+1}} \delta_t(a_t^*, a_t) \mid t_\ell\right]\right] \\
&= \mathbb{E}\left[\sum_{t=t_\ell}^{T} \mathbb{E}\left[\mathbf{1}_{\{t<t_{\ell+1}\}} \mathbb{E}\left[\delta_t(a_t^*, a_t) \mid \mathcal{F}_{t-1}\right] \mid t_\ell\right]\right] \\
&= \mathbb{E}\left[\sum_{t=t_\ell}^{T} \sum_{a\in\mathcal{A}_t} \mathbb{E}\left[\mathbf{1}_{\{t<t_{\ell+1}\}} \mid t_\ell\right] \frac{\delta_t(a_t^*, a)}{|\mathcal{A}_t|}\right] \\
&= \mathbb{E}\left[\sum_{t=t_\ell}^{t_{\ell+1}} \sum_{a\in\mathcal{A}_t} \frac{\delta_t(a_t^*, a)}{|\mathcal{A}_t|}\right],
\end{aligned}$$

where we used that $\mathbf{1}_{\{t<t_{\ell+1}\}}$ is $\mathcal{F}_{t-1}$-measurable and

$$\mathbb{E}\left[\delta_t(a_t^*, a_t) \mid \mathcal{F}_{t-1}\right] = \sum_{a\in\mathcal{A}_t} \frac{\delta_t(a_t^*, a)}{|\mathcal{A}_t|}.$$

Lastly, Lemma C.1.4 then follows from rewriting the sum over $a \in \mathcal{A}_t$ using the indicator $\mathbf{1}_{\{a\in\mathcal{A}_t\}}$ and swapping the order of the sums. ■

In an important next step, we split the dynamic regret for *each fixed arm* $a \in [K]$ into:

(i) the regret we suffer from playing arm a in the ℓ-th episode before its elimination from $\mathcal{A}_{\text{good}}$,

(ii) the regret we suffer from (re)playing arm a in the ℓ-th episode after its elimination from $\mathcal{A}_{\text{good}}$.

Recall that $t_\ell^a \in [t_\ell, t_{\ell+1})$ denotes the time that arm a is eliminated from $\mathcal{A}_{\text{good}}$ in episode ℓ. Using Lemma C.1.4, we then decompose the dynamic regret in episode ℓ as

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=t_\ell}^{t_{\ell+1}-1} \frac{\delta_t(a_t^*, a_t) + \delta_t(a_t^*, b_t)}{2}\right] \\
&\qquad = \underbrace{\mathbb{E}\left[\sum_{a=1}^{K} \sum_{t=t_\ell}^{t_\ell^a-1} \frac{\delta_t(a_t^*, a)}{|\mathcal{A}_t|}\right]}_{R_1(\ell)} + \underbrace{\mathbb{E}\left[\sum_{a=1}^{K} \sum_{t=t_\ell^a}^{t_{\ell+1}-1} \frac{\delta_t(a_t^*, a)}{|\mathcal{A}_t|} \mathbf{1}_{\{a\in\mathcal{A}_t\}}\right]}_{R_2(\ell)},
\end{aligned}$$

where for $R_1(\ell)$ we used that $a \in \mathcal{A}_{\text{good}}$ implies $a \in \mathcal{A}_t$ by construction of these sets. For every fixed arm, $R_1(\ell)$ corresponds to the regret suffered before said arm is eliminated from the master set. Accordingly, $R_2(\ell)$ is the regret due to replaying an arm after its elimination from the master set. The remainder of the proof is mainly concerned with bounding $R_1(\ell)$ and $R_2(\ell)$ appropriately.

C.1.2 Bounding $R_1(\ell)$: Regret Before Elimination

We begin by assuming w.l.o.g. that $t_\ell^1 \leq \cdots \leq t_\ell^K$ so that for each round $t < t_\ell^a$ all arms $a' \geq a$ are element in $\mathcal{A}_{\text{good}} \subseteq \mathcal{A}_t$. As a result, we have $|\mathcal{A}_t| \geq K+1-a$ for all $t \leq t_\ell^a$, and thus

$$\mathbb{E}\left[\sum_{a=1}^{K}\sum_{t=t_\ell}^{t_\ell^a-1}\frac{\delta_t(a_t^*,a)}{|\mathcal{A}_t|}\right] \leq \mathbb{E}\left[\sum_{a=1}^{K}\sum_{t=t_\ell}^{t_\ell^a-1}\frac{\delta_t(a_t^*,a)}{K+1-a}\right]. \tag{12}$$

As we can see, the denominator will eventually account for a factor of $\log(K) \approx \sum_{a=1}^{K} 1/a$. We now concentrate on bounding the inner sum in (12), i.e. the regret of any fixed arm before being eliminated in the ℓ-th episode.

C.1.2.1 Bounding $\mathbb{E}[\sum_{t=t_\ell}^{t_\ell^a-1}\delta_t(a_t^*,a)]$ for any fixed arm $a \in [K]$

This section is devoted to proving the following upper bound.

Lemma C.1.5. *For some constant $c > 0$:*

$$\mathbb{E}\left[\sum_{t=t_\ell}^{t_\ell^a-1}\delta_t(a_t^*,a)\right] \leq c\log^2(T)K\,\mathbb{E}\left[\sum_{i\in\text{Phases}(t_\ell,t_\ell^a)}\sqrt{\tau_{i+1}-\tau_i}\right] + \frac{K}{T^2} + \frac{1}{T}. \tag{13}$$

To prove Lemma C.1.5, we will divide the interval $[t_\ell, t_\ell^a)$ into segments over the course of which arm a suffers large regret and show that not too many of such segments will occur in interval $[t_\ell, t_\ell^a)$, i.e. until arm a is being eliminated from $\mathcal{A}_{\text{good}}$. The definition of such bad segments is analogous to their construction in [AGL22] and [SK22b]. Whereas prior work utilizes such segments to bound the regret of the last arm considered good in an episode, i.e. the last arm in $\mathcal{A}_{\text{good}}$, we will instead derive a regret bound for *any fixed arm a*. While the according regret bound will be in some sense weaker, it will still be sufficiently tight for our purposes. We here follow the notation in [SK22b].

Definition C.1.6 (Bad Segments)**.** Fix t_ℓ and let $[\tau_i, \tau_{i+1})$ be any phase intersecting $[t_\ell, T)$. For an arm a, define rounds $s_{i,j}(a) \in [t_\ell \vee \tau_i, \tau_{i+1})$ recursively as follows: let $s_{i,0}(a) = t_\ell \vee \tau_i$ and define $s_{i,j+1}(a)$ as the smallest round in $(s_{i,j}(a), \tau_{i+1})$ such that arm a satisfies for some constant $c_3 > 0$:

$$\sum_{t=s_{i,j}(a)}^{s_{i,j+1}(a)}\delta_t(a_i^*,a) > c_3\log(T)K\sqrt{s_{i,j+1}(a)-s_{i,j}(a)}, \tag{14}$$

if such round $s_{i,j+1}(a)$ exists. Otherwise, we let $s_{i,j+1}(a) = \tau_{i+1} - 1$. We refer to the intervals $[s_{i,j}, s_{i,j+1})$ as bad segments if (14) is satisfied. If a segment does not satisfy (14), we refer to them as non-bad segments.[3]

[3]Note that by definition every segment but the last segment in a given phase must always satisfy (14)

Note that the concept of bad segments will become useful later as, for a fixed t_ℓ, by definition of the bad segments, we can always upper bound the dynamic regret on an interval $[s_{i,j}(a), s_{i,j+1}(a))$ by

$$\sum_{t=s_{i,j}(a)}^{s_{i,j+1}(a)-1} \delta_t(a_t^*, a) \leq c_3 \log(T) K \sqrt{s_{i,j+1}(a) - s_{i,j}(a)}. \tag{15}$$

We now define the *bad round* for an arm a as the smallest round when the aggregated regret of bad segments exceeds $\sqrt{\text{interval length}}$ regret.

Definition C.1.7 (Bad Round)**.** Fix t_ℓ and some arm a. The bad round $s(a) > t_\ell$ is defined as the smallest round which satisfies for some universally fixed constant $c_4 > 0$:

$$\sum_{(i,j)\colon\, s_{i,j+1}(a) < s(a)} \sqrt{s_{i,j+1}(a) - s_{i,j}(a)} > c_4 \log(T) \sqrt{s(a) - t_\ell}, \tag{16}$$

where the sum is over all bad segments with $s_{i,j+1}(a) < s(a)$.

For a given episode ℓ, we will show that arm a is eliminated with high probability by the time the bad round $s(a)$ occurs. To this end, we will introduce perfect replays, i.e. those runs of `CondaLet` which are properly timed and eliminate arm a before it aggregates large regret.

C.1.2.2 Perfect Replays

The following result will become very useful and makes the intuition precise that on the concentration event the Condorcet winner will not be eliminated. More precisely, any run of `CondaLet`(s, m) scheduled in phase i will never eliminate a_i^* inside phase i as long as our concentration bound holds.

Lemma C.1.8. *On event $\mathcal{E}$, no run of* `CondaLet`(s, m) *with $s \in [\tau_i, \tau_{i+1})$ ever eliminates arm a_i^* before round τ_{i+1}.*

Proof. Suppose the contrary that some `CondaLet`(s, m) with $s \in [\tau_i, \tau_{i+1})$ eliminates arm a_i^* before round τ_{i+1}. Then, we must have for some arm $a \in [K]$ and interval $[s_1, s_2] \subseteq [s, \tau_{i+1})$ that

$$C \log(T) K \sqrt{(s_2 - s_1) \vee K^2} < \sum_{t=s_1}^{s_2} \hat{\delta}_t(a, a_i^*), \tag{17}$$

which using the concentration bound (6) implies on event $\mathcal{E}$ that

$$c_2 \log(T) K \sqrt{(s_2 - s_1) \vee K^2} < \sum_{t=s_1}^{s_2} \mathbb{E}\left[\hat{\delta}_t(a, a_i^*) \mid \mathcal{F}_{t-1}\right] \leq \sum_{t=s_1}^{s_2} \delta_t(a, a_i^*), \tag{18}$$

where the last inequality holds by merit of (8). Now, by the definition of arm a_i^* as the Condorcet winner in phase i, we must have $\delta_t(a, a_i^*) \leq 0$ for all $t \in [\tau_i, \tau_{i+1})$ and all $a \in [K]$. Lemma C.1.8 then follows from contradiction. ■

This leads to the following important property of CondaLet that states that properly timed replays of sufficient length will eliminate arms from $\mathcal{A}_{\text{good}}$ in the course of their bad segments. We call such calls of CondaLet *perfect replays*.

Proposition C.1.9 (Perfect Replay)**.** *Suppose that event $\mathcal{E}$ holds. Let $[s_{i,j}(a), s_{i,j+1}(a))$ be a bad segment w.r.t. arm a and let $\tilde{s}_{i,j}(a) = \left\lceil \frac{s_{i,j}(a)+s_{i,j+1}(a)}{2} \right\rceil$ be the midpoint of the interval. It holds that any run of CondaLet(s,m) with $s \in [s_{i,j}(a), \tilde{s}_{i,j}(a)]$ and $m \geq s_{i,j+1}(a) - s_{i,j}(a)$ will eliminate arm a from $\mathcal{A}_{good}$. We refer to such calls of CondaLet as* perfect replays *w.r.t. arm a.*

Proof. Let CondaLet(s,m) be a replay such that $s \in [s_{i,j}(a), \tilde{s}_{i,j}(a)]$ and $m \geq s_{i,j+1}(a) - s_{i,j}(a)$. As any bad segment is by definition contained inside a phase, Lemma C.1.8 tells us that $a_i^* \in \mathcal{A}_t$ for all $t \in [\tilde{s}_{i,j}(a), s_{i,j+1}(a)]$. Recall that the estimates $\hat{\delta}_t(a_i^*, a)$ are unbiased if $a, a_i^* \in \mathcal{A}_t$ and we are thus able to obtain unbiased estimates of $\sum_{t=\tilde{s}_{i,j}(a)}^{s_{i,j+1}(a)} \delta_t(a_i^*, a)$. What is left to show is that in fact arm a suffers sufficiently large regret to cause its elimination on this interval. To this end, by definition of the bad segments and basic algebraic manipulation, we find that

$$\begin{aligned}
\sum_{t=\tilde{s}_{i,j}(a)}^{s_{i,j+1}(a)} \delta_t(a_i^*, a) &= \sum_{t=s_{i,j}(a)}^{s_{i,j+1}(a)} \delta_t(a_i^*, a) - \sum_{t=s_{i,j}(a)}^{\tilde{s}_{i,j}(a)-1} \delta_t(a_i^*, a) \\
&\overset{(14)}{\geq} c_3 \log(T) K \left(\sqrt{s_{i,j+1}(a) - s_{i,j}(a)} - \sqrt{\tilde{s}_{i,j}(a) - 1 - s_{i,j}(a)} \right) \\
&\geq \frac{c_3}{4} \log(T) K \sqrt{s_{i,j+1}(a) - \tilde{s}_{i,j}(a)}.
\end{aligned}$$

Using that $\sum_{t=\tilde{s}_{i,j}(a)}^{s_{i,j+1}(a)} \hat{\delta}_t(a_i^*, a)$ is an unbiased estimate of $\sum_{t=\tilde{s}_{i,j}(a)}^{s_{i,j+1}(a)} \delta_t(a_i^*, a)$ and applying the concentration bound (6), this shows that arm a satisfies the elimination rule (III.2) over interval $[\tilde{s}_{i,j}(a), s_{i,j+1}(a)]$ and will thus be eliminated by CondaLet(s,m). ■

C.1.2.3 Perfect replays are scheduled w.h.p.

Following [SK22b], we will now show that a perfect replay that eliminates arm a is scheduled before round $s(a)$ with high probability. A replay CondaLet(s,m) is scheduled if $B_{s,m} = 1$ and the random variables $B_{s,m}$ with $s \geq t_\ell$ are conditionally independent on t_ℓ (see Line 7 in Algorithm 10). We are thus interested in perfect replays CondaLet(s,m) such that for any bad segment $[s_{i,j}(a), s_{i,j+1}(a))$ with $s_{i,j+1}(a) < s(a)$, we have $s \in [s_{i,j}(a), \tilde{s}_{i,j}(a)]$ and $m \geq s_{i,j+1}(a) - s_{i,j}(a)$. Moreover, we define $m_{i,j}$ as the smallest element in $\{2, \dots, 2^{\lceil \log(T) \rceil}\}$ such that $m_{i,j} \geq s_{i,j+1}(a) - s_{i,j}(a)$, which implies that $s_{i,j+1}(a) - s_{i,j}(a) \geq \frac{m_{i,j}}{2}$. We will obtain the high probability guarantee via concentration on the sum

$$X(t_\ell, s(a)) = \sum_{(i,j)\colon\, s_{i,j+1}(a) < s(a)} \sum_{s=s_{i,j}(a)}^{\tilde{s}_{i,j}(a)} B_{s,m_{i,j}}. \tag{19}$$

Lemma C.1.10. *Let $\mathcal{E}'(t_\ell)$ denote the event that $X(t_\ell, s(a)) \geq 1$ for all arms a, i.e. a perfect replay is scheduled before round $s(a)$. We have $\mathbb{P}(\mathcal{E}'(t_\ell) \mid t_\ell) \geq 1 - K/T^3$.*

Proof. Recalling that $B_{s,m} \mid t_\ell \sim \text{Bernoulli}\left(\frac{1}{\sqrt{m(s-t_\ell)}}\right)$, we find that

$$\begin{aligned}\mathbb{E}[X(t_\ell, s(a)) \mid t_\ell] &\geq \frac{1}{\sqrt{2}} \sum_{\substack{(i,j):\\ s_{i,j+1}(a)<s(a)}} \frac{\tilde{s}_{i,j}(a) - s_{i,j}(a)}{\sqrt{s_{i,j+1}(a) - s_{i,j}(a)}\sqrt{s(a) - t_\ell}} \\ &\geq \frac{1}{4} \sum_{\substack{(i,j):\\ s_{i,j+1}(a)<s(a)}} \sqrt{\frac{s_{i,j+1}(a) - s_{i,j}(a)}{s(a) - t_\ell}} \overset{(16)}{\geq} \frac{c_4}{4}\log(T)\end{aligned}$$

For c_4 sufficiently large the standard Chernoff bound tells us that

$$\mathbb{P}\left(X(t_\ell, s(a)) \leq \frac{\mathbb{E}[X(t_\ell, s(a)) \mid t_\ell]}{2} \mid t_\ell\right) \leq \exp\left(-\frac{\mathbb{E}[X(t_\ell, s(a)) \mid t_\ell]}{8}\right) \leq \frac{1}{T^3}.$$

The desired bound then follows from taking a union bound over all arms in $[K]$. ∎

Now, on event $\mathcal{E} \cap \mathcal{E}'(t_\ell)$, it must hold that $t_\ell^a < s(a)$ for all arms $a \in [K]$, since otherwise a would have been eliminated by some perfect replay before round t_ℓ^a (by definition of event $\mathcal{E}'(t_\ell)$). As the bad round $s(a)$ is defined as the *smallest* round satisfying (16), we then have

$$\sum_{(i,j):\ s_{i,j+1}(a)<t_\ell^a} \sqrt{s_{i,j+1}(a) - s_{i,j}(a)} \leq c_4 \log(T) K \sqrt{t_\ell^a - t_\ell}. \tag{20}$$

Hence, in view of equation (15), over the bad segments, the regret of arm a is at most of order $\log^2(T)\sqrt{t_\ell^a - t_\ell}$. Moreover, for every last segment in some phase i, $[s_{i,j}, s_{i,j+1}(a))$, as well as the final segment $[s_{i,j}(a), t_\ell^a)$, we know that the regret suffered from playing a is upper bounded by $c_3 \log(T)\sqrt{\tau_{i+1} - \tau_i}$ by definition of non-bad segments (Definition C.1.6). Therefore, on event $\mathcal{E} \cap \mathcal{E}'(t_\ell)$, it follows from equation (20) and the above that

$$\sum_{t=t_\ell}^{t_\ell^a - 1} \delta_t(a_t^*, a) \leq c_5 K \log^2(T) \sum_{i \in \text{Phases}(t_\ell, t_\ell^a)} \sqrt{\tau_{i+1} - \tau_i}, \tag{21}$$

where we used that $\sqrt{t_\ell^a - t_\ell} \leq \sum_{i \in \text{Phases}(t_\ell, t_\ell^a)} \sqrt{\tau_{i+1} - \tau_i}$. Finally, we obtain Lemma C.1.5 by taking expectation and using that $\mathcal{E} \cap \mathcal{E}'(t_\ell)$ holds with high

probability,

$$\mathbb{E}\left[\sum_{t=t_\ell}^{t_\ell^a-1}\delta_t(a_t^*,a)\right] \le \mathbb{E}\left[\left[\mathbf{1}_{\{\mathcal{E}\cap\mathcal{E}'(t_\ell)\}}\sum_{t=t_\ell}^{t_\ell^a-1}\delta_t(a_t^*,a)\mid t_\ell\right]\right] + T\big(\mathbb{P}(\mathcal{E}^c)+\mathbb{P}(\mathcal{E}'(t_\ell)^c\mid t_\ell)\big)$$
$$\le c_5 K\log^2(T)\,\mathbb{E}\left[\sum_{i\in\text{Phases}(t_\ell,t_\ell^a)}\sqrt{\tau_{i+1}-\tau_i}\right]+\frac{1}{T}+\frac{K}{T^2}.$$

C.1.2.4 Summing Over Arms

Note that $t_\ell^a \le t_{\ell+1}$ for all $a\in[K]$ by definition of t_ℓ^a. Then, summing over all arms, it follows from Lemma C.1.5 and (12) that for some constant $c_6>0$:

$$\mathbb{E}\left[\sum_{a=1}^{K}\sum_{t=t_\ell}^{t_\ell^a-1}\frac{\delta_t(a_t^*,a)}{|\mathcal{A}_t|}\right] \le c_6 K\log^3(T)\,\mathbb{E}\left[\sum_{i\in\text{Phases}(t_\ell,t_{\ell+1})}\sqrt{\tau_{i+1}-\tau_i}\right], \quad (22)$$

where we loosely upper bound $\log(K)$ by $\log(T)$.

C.1.3 Bounding $R_2(\ell)$: Regret After Elimination

Before we can begin, we will have to lay some groundwork to simplify the analysis in later steps. Recall the definition of bad segments from Section C.1.2 and define for every phase $[\tau_i,\tau_{i+1})$ intersecting with $[t_\ell^a,t_{\ell+1})$, i.e. $i\in\text{Phases}(t_\ell^a,t_{\ell+1})$, the segments $[s_{i,j}(a),s_{i,j+1})$ as in Definition C.1.6.

We will split the regret due to bad segments, i.e. those that satisfy (14), from the regret due to non-bad segments, i.e. the last segments in a phase that do no satisfy (14). For a fixed arm $a\in[K]$, we let $\text{bad}(a)$ denote the rounds $t\in[t_\ell,t_{\ell+1})$ such that $t\in[s_{i,j}(a),s_{i,j+1}(a))$ for any *bad* segment $[s_{i,j}(a),s_{i,j+1}(a))$.

By the definition of a non-bad segment (w.r.t. arm a), we know that that there is at most one such segment in every phase and that the regret of arm a in each segment is upper bounded by $c_3\log(T)\sqrt{\tau_{i+1}-\tau_i}$, where $[\tau_i,\tau_{i+1})$ is the phase that contains the segment. To take care of the denominator $|\mathcal{A}_t|$, assume w.l.o.g. that there is a run of $\texttt{CondaLet}(t_\ell^a,m)$ that remains active and uninterrupted until the final round T.[4] We can then reorder arms $a\in[K]$ according to the round that they are being eliminated by $\texttt{CondaLet}(t_\ell^a,m)$, which gives $|\mathcal{A}_t|\ge K+1-a$ whenever $a\in\mathcal{A}_t$. As before, this yields a factor of $\log(K)$ when summing over all arms. We then bound $R_2(\ell)$ over non-bad segments as

$$\mathbb{E}\left[\sum_{a=1}^{K}\sum_{t=t_\ell^a}^{t_{\ell+1}-1}\frac{\delta_t(a_t^*,a)}{|\mathcal{A}_t|}\mathbf{1}_{\{a\in\mathcal{A}_t,t\notin\text{bad}(a)\}}\right] \quad (23)$$

[4] Note that this is w.l.o.g. when bounding $1/|\mathcal{A}_t|$ as any interrupting call of `CondaLet` would only increase $|\mathcal{A}_t|$ by resetting it to $[K]$.

$$\leq c_3 K \log(K) \log(T) \mathbb{E}\left[\sum_{i \in \text{Phases}(t_\ell^a, t_{\ell+1})} \sqrt{\tau_{i+1} - \tau_i}\right].$$

The more challenging task is now to bound $R_2(\ell)$ for rounds in bad segments. Recall that, for a fixed arm $a \in [K]$, the sum in question relates to the expected regret suffered within an episode from replaying arm a after it has been eliminated from $\mathcal{A}_{\text{good}}$, i.e. after time t_ℓ^a. We begin by a straightforward upper bound. To this end, for a given replay $\texttt{CondaLet}(s, m)$, let $M(s, m, a)$ be the last round in $[s, s+m]$, where arm a is active in $\texttt{CondaLet}(s, m)$ and all of its children. Then,

$$\mathbb{E}\left[\sum_{a=1}^{K} \sum_{t=t_\ell^a}^{t_{\ell+1}-1} \frac{\delta_t(a_t^*, a)}{|\mathcal{A}_t|} \mathbf{1}_{\{a \in \mathcal{A}_t, t \in \text{bad}(a)\}}\right] \tag{24}$$

$$\leq \mathbb{E}\left[\sum_{a=1}^{K} \sum_{s=t_\ell+1}^{t_{\ell+1}-1} \sum_{m} \mathbf{1}_{\{B_{s,m}=1\}} \sum_{t=s \vee t_\ell^a}^{M(s,m,a)} \frac{\delta_t(a_t^*, a)}{|\mathcal{A}_t|} \mathbf{1}_{\{t \in \text{bad}(a)\}}\right],$$

where the most inner sum on the right hand side is for $m \in \{2, \ldots, 2^{\lceil \log(T) \rceil}\}$. We will keep the convention that whenever a sum over m is not further specified, it will be over the above set. Note that (24) is a loose upper bound. While of course only a single call of `CondaLet` can be active at any point in time, we here sum over every possible replay and ignore the potential nesting and interleaving of replays. In particular, this upper bound is justified as each $\delta_t(a_t^*, a)$ is non-negative by definition of the CW a_t^*. The looseness of (24) will pose no obstacle, as the remainder of our upper bounds will be sufficiently tight as we will see.

Again, we first take care of the dependence on K due to the denominator on the right hand side of (24). Note that for a fixed $\texttt{CondaLet}(s, m)$ if a_k is the k-th arm to be eliminated by $\texttt{CondaLet}(s, m)$, then $\min_{t \in [s, M(s,m,a_k)]} |\mathcal{A}_t| \geq K + 1 - k$. Similarly to before, this will result in a multiplicative $\log(K)$ term when eventually switching the order of the sums and summing over all arms. For now, we therefore focus on the expression

$$\mathbb{E}\left[\sum_{s=t_\ell+1}^{t_{\ell+1}-1} \sum_{m} \mathbf{1}_{\{B_{s,m}=1\}} \sum_{t=s \vee t_\ell^a}^{M(s,m,a)} \delta_t(a_t^*, a) \mathbf{1}_{\{t \in \text{bad}(a)\}}\right] \tag{25}$$

for any fixed arm $a \in [K]$. To deal with this quantity, it will be helpful to distinguish between two types of replays, i.e. calls of `CondaLet`, which we refer to as confined and unconfined replays.

Definition C.1.11 (Confined and Unconfined Replays)**.** For a fixed t_ℓ, we call $\texttt{CondaLet}(s, m)$ *confined* if there exists $i \in \text{Phases}(t_\ell, T)$ such that $[s, s+m] \subseteq [\tau_i, \tau_{i+1})$, i.e. the replay intersects with a single phase only. In turn, we say that $\texttt{CondaLet}(s, m)$ is *unconfined* if for all $i \in \text{Phases}(t_\ell, T)$, we have $[s, s+m] \not\subseteq [\tau_i, \tau_{i+1})$.

An illustration of confined and unconfined replays is given in Figure .1.

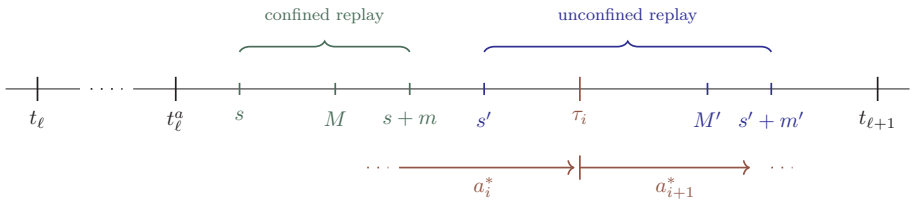


Figure .1: For some episode $[t_\ell, t_{\ell+1})$ and arm $a \in [K]$, an example of a confined replay and a unconfined replay, where $M = M(s, m, a)$ and $M' = M(s', m', a)$. When a replay $\texttt{CondaLet}(s', m')$ intersects with more than one phase, the CW in the next phase $[\tau_i, \tau_{i+1})$, denoted a^*_{i+1}, could be evicted before the beginning of that phase, i.e. in the interval $[s', \tau_i)$.

We proceed by upper bounding the inner sum $\sum_{t=s\vee t_\ell^a}^{M(s,m,a)} \delta_t(a_t^*, a)\mathbf{1}_{\{t\in \mathrm{bad}(a)\}}$ for confined and unconfined replays separately. The bound for confined replays comes with no major intricacies, whereas bounding the regret due to unconfined replays is slightly more involved.

C.1.3.1 Bounding Regret for Confined Replays

We begin by bounding, the inner sum $\sum_{t=s\vee t_\ell^a}^{M(s,m,a)} \delta_t(a_t^*, a)$ for any confined replay in terms of the replay duration m.

Lemma C.1.12. *On event $\mathcal{E}$, for any fixed arm a and confined replay (s, m), it holds that*

$$\sum_{t=s\vee t_\ell^a}^{M(s,m,a)} \delta_t(a_t^*, a) \le c_2 \log(T) K \sqrt{m}.$$

Proof of Lemma C.1.12. Consider any confined replay $\texttt{CondaLet}(s, m)$ with $[s, s+m] \subseteq [\tau_i, \tau_{i+1})$ for some phase i. This implies that on interval $[s, s+m]$ the Condorcet winner remains the same, i.e. $a_t^* = a_i^*$ for all $t \in [s, s+m]$. Now, recall from Lemma C.1.8 that, on event $\mathcal{E}$, arm a_i^* will not be eliminated inside of $[s, s+m]$ as it is a subset of phase $[\tau_i, \tau_{i+1})$. As a result, we must have $a, a_i^* \in \mathcal{A}_t$ for all $t \in [s \vee t_\ell^a, M(s, m, a)]$ and our estimate $\hat{\delta}_t(a_i^*, a)$ is thus unbiased. Since $M(s, m, a)$ is the last round that arm a is retained by $\texttt{CondaLet}(s, m)$ (and its children), it follows from the elimination rule (III.2) and the concentration bound (6) that

$$\sum_{t=s\vee t_\ell^a}^{M(s,m,a)} \delta_t(a_i^*, a) \le c_2 \log(T) K \sqrt{M(s, m, a) - s \vee t_\ell^a} \le c_2 \log(T) K \sqrt{m},$$

where the last inequality uses that $M(s, m, a) \le s + m$. ∎

C.1.3.2 Bounding Regret for Unconfined Replays

Lemma C.1.13. *On event $\mathcal{E} \cap \mathcal{E}''(t_\ell)$, for any fixed arm a and unconfined replay (s, m), it holds that*

$$\sum_{t=s\vee t_\ell^a}^{M(s,m,a)} \delta_t(a_t^*, a)\mathbf{1}_{\{t\in bad(a)\}} \leq c_5 \log^2(T)K\big(\sqrt{s-t_\ell} + 2\sqrt{m}\big).$$

Here, the event $\mathcal{E}''(t_\ell)$ is a concentration event similar to that in Lemma C.1.10 and will be defined in the following.

Proof of Lemma C.1.13. Consider any unconfined replay $\texttt{CondaLet}(s, m)$ with $s \in [t_\ell, t_{\ell+1})$. Let i be the phase so that $s \in [\tau_{i-1}, \tau_i)$. We can then split the sum over $t \in [s \vee t_\ell^a, M(s, m, a)]$ into the rounds before the Condorcet winner changes for the first time within $[s, s + m]$ and the remaining rounds, i.e.

$$\sum_{t=s\vee t_\ell^a}^{M(s,m,a)} \delta_t(a_t^*, a) = \sum_{t=s\vee t_\ell^a}^{\tau_i-1} \delta_t(a_i^*, a) + \sum_{t=\tau_i}^{M(s,m,a)} \delta_t(a_t^*, a). \tag{26}$$

Note that the interval $[\tau_i, M(s, m, a)]$ can itself span over several phases. The first sum on the right hand side can be bounded as in Lemma C.1.12. Using Lemma C.1.8, the elimination rule, and the concentration bound, we get

$$\sum_{t=s\vee t_\ell^a}^{\tau_i-1} \delta_t(a_i^*, a) \leq c_2 \log(T)K\sqrt{m}.$$

The second sum cannot be bounded in a similar way, as we cannot guarantee that the Condorcet winner in some round $t \in [\tau_i, M(s, m, a)]$ has not been eliminated in prior rounds $[s \vee t_\ell^a, \tau_i)$. For example in Figure .1, the unconfined replay $\texttt{CondaLet}(s', m')$ could have eliminated a_{i+1}^* on interval $[s', \tau_i)$ before it became the Condorcet winner. We may therefore fail to detect that a suffers large regret without additional replays.

To resolve this difficulty, we can reuse part of the arguments from Section C.1.2. Define the bad segments $[s_{k,j}(a), s_{k,j+1}(a))$ for $k \geq i$ as in Definition C.1.6. Similarly to before, we now define the bad round $s'(a)$ as the smallest round $s'(a) > \tau_i$ such that for the same constant $c_4 > 0$ as in (16)

$$\sum_{(k,j)\colon s_{k,j+1}(a)<s'(a)} \sqrt{s_{k,j+1}(a) - s_{k,j}(a)} > c_4 \log(T)\sqrt{s'(a) - t_\ell}, \tag{27}$$

where the sum is over all bad segments with $k \geq i$ and $s_{k,j+1}(a) < s'(a)$.

Importantly, for this definition of $s'(a)$ and with the sum $X(t_\ell, s'(a))$ defined accordingly, the high probability guarantee of Lemma C.1.10 still holds. This implies that a perfect replay (see Proposition C.1.9) that eliminates arm a (from the unconfined replay $\texttt{CondaLet}(s, m)$) is scheduled w.h.p. before the bad round $s'(a)$ occurs. Let the corresponding event denote $\mathcal{E}''(t_\ell)$ as in Lemma C.1.10.

The round $M(s,m,a)$ was defined as the last round for which a is retained in $\texttt{CondaLet}(s,m)$ and all of its children. Hence, on event $\mathcal{E}\cap\mathcal{E}''(t_\ell)$, we must have $M(s,m,a)<s'(a)$ as otherwise a would have been eliminated from $\texttt{CondaLet}(s,m)$ (or one of its children) before round $M(s,m,a)$, a contradiction. By merit of (15), this yields

$$\sum_{(k,j)\colon s_{k,j+1}(a)<M(s,m,a)}\sqrt{s_{k,j+1}(a)-s_{k,j}(a)}\le c_4\log(T)K\sqrt{M(s,m,a)-t_\ell}$$

The regret on the final segment $[s_{k,j}(a),M(s,m,a)]$ can trivially be bounded by $c_3\log(T)K\sqrt{m}$, as it must be a non-bad segment and $M(s,m,a)-s_{k,j}(a)\le m$. Finally, in view of (15), it follows that

$$\begin{aligned}\sum_{t=s\vee t_\ell^a}^{M(s,m,a)}\delta_t(a_t^*,a)\mathbf{1}_{\{t\in\mathrm{bad}(a)\}}&\le c_5\log^2(T)K(\sqrt{M(s,m,a)-t_\ell}+\sqrt{m})\\&\le c_5\log^2(T)K(\sqrt{s-t_\ell}+2\sqrt{m}),\end{aligned}$$

where the second inequality uses $\sqrt{M(s,m,a)-t_\ell}\le\sqrt{s-t_\ell}+\sqrt{m}$, since $M(s,m,a)\le s+m$ and $s\ge t_\ell$.

■

C.1.3.3 Combining Confined and Unconfined Replays

We will now conclude the bound on $R_2(\ell)$. To this end, recall that the replay schedule is chosen according to $B_{s,m}\mid t_\ell\sim\mathrm{Bern}\big(1/\sqrt{m(s-t_\ell)}\big)$. Then, conditioning on t_ℓ, we have

$$\begin{aligned}\mathbb{E}\left[\sum_{s=t_\ell+1}^{t_{\ell+1}}\sum_m\mathbf{1}_{\{B_{s,m}\}}\right]&=\mathbb{E}\left[\sum_{s=t_\ell+1}^{T}\sum_m\mathbb{E}\left[\mathbf{1}_{\{B_{s,m}\}}\mid t_\ell\right]\mathbb{E}\left[\mathbf{1}_{\{s<t_{\ell+1}\}}\mid t_\ell\right]\right]\\&=\mathbb{E}\left[\sum_{s=t_{\ell+1}}^{t_{\ell+1}-1}\frac{1}{\sqrt{m(s-t_\ell)}}\right].\end{aligned}$$

Moreover, note that we can rewrite a sum over $s\in[t_\ell+1,t_{\ell+1})$ as a double sum over $i\in\mathrm{Phases}(t_\ell,t_{\ell+1})$ and $s\in[\tau_i\vee(t_\ell+1),\tau_{i+1}\wedge t_{\ell+1})$. For unconfined replays, we notice that when $\texttt{CondaLet}(s,m)$ is scheduled with $s\in[\tau_i,\tau_{i+1})$, it must hold that $m\ge\tau_{i+1}-s$, as $\texttt{CondaLet}(s,m)$ would otherwise not be unconfined.

Now, combining Lemma C.1.12 and Lemma C.1.13, we obtain

$$\begin{aligned}&\mathbb{E}\left[\mathbf{1}_{\{\mathcal{E}\cap\mathcal{E}''(t_\ell)\}}\sum_{a=1}^{K}\sum_{t=t_\ell^a}^{t_{\ell+1}-1}\frac{\delta_t(a_t^*,a)}{|\mathcal{A}_t|}\mathbf{1}_{\{a\in\mathcal{A}_t,t\in\mathrm{bad}(a)\}}\right]\\&\le\mathbb{E}\left[\mathbf{1}_{\{\mathcal{E}\cap\mathcal{E}''(t_\ell)\}}\sum_{s=t_\ell+1}^{t_{\ell+1}-1}\sum_m\mathbf{1}_{\{B_{s,m}=1\}}\sum_{t=s\vee t_\ell^a}^{M(s,m,a)}\frac{\delta_t(a_t^*,a)}{|\mathcal{A}_t|}\mathbf{1}_{\{t\in\mathrm{bad}(a)\}}\right]\end{aligned}$$

$$
\begin{aligned}
&\leq c_2 K \log(K) \log(T)\, \mathbb{E}\left[\sum_{s=t_\ell}^{t_{\ell+1}-1} \sum_{m} \frac{\sqrt{m}}{\sqrt{m(s-t_\ell)}}\right] \\
&\quad + c_5 K \log(K) \log^2(T)\, \mathbb{E}\left[\sum_{i \in \text{Phases}(t_\ell, t_{\ell+1})} \sum_{s=\tau_i}^{\tau_{i+1}-1} \sum_{m \geq \tau_{i+1}-s} \frac{\sqrt{s-t_\ell} + 2\sqrt{m}}{\sqrt{m(s-t_\ell)}}\right] \\
&\leq c_2 K \log^3(T)\, \mathbb{E}\left[\sqrt{t_{\ell+1}-t_\ell}\right] \\
&\quad + c_5 K \log^3(T)\, \mathbb{E}\left[\sum_{i \in \text{Phases}(t_\ell, t_{\ell+1})} \sum_{s=\tau_i}^{\tau_{i+1}-1} \frac{1}{\sqrt{\tau_{i+1}-s}} + 2\sqrt{t_{\ell+1}-t_\ell}\right] \\
&\leq c_7 K \log^3(T)\, \mathbb{E}\left[2 \sum_{i \in \text{Phases}(t_\ell, t_{\ell+1})} \sqrt{\tau_{i+1}-\tau_i} + 5 \sum_{i \in \text{Phases}(t_\ell, t_{\ell+1})} \sqrt{\tau_{i+1}-\tau_i}\right] \\
&\leq 7 c_7 K \log^3(T)\, \mathbb{E}\left[\sum_{i \in \text{Phases}(t_\ell, t_{\ell+1})} \sqrt{\tau_{i+1}-\tau_i}\right].
\end{aligned}
$$

We here repeatedly used that $\sum_{k=1}^{n} 1/\sqrt{k} \leq 2\sqrt{n}$ in the third and fourth inequality. In particular, the fourth inequality holds as $\sqrt{t_{\ell+1}-t_\ell} \leq \sum_{i \in \text{Phases}(t_\ell, t_{\ell+1})} \sqrt{\tau_{i+1}-\tau_i}$ and

$$
\sum_{s=\tau_i}^{\tau_{i+1}-1} \frac{1}{\sqrt{\tau_{i+1}-s}} = \sum_{s=1}^{\tau_{i+1}-\tau_i-1} \frac{1}{\sqrt{s}} \leq \sqrt{\tau_{i+1}-\tau_i}.
$$

Further note that, as explained before, the denominator $|\mathcal{A}_t|$ can be seen to account for a factor of $\log(K)$, which we loosely upper bounded by $\log(T)$. Together with (23), we then obtain for some constant $c_8 > 0$ the desired bound of

$$
E\left[\sum_{a=1}^{K} \sum_{t=t_\ell^a}^{t_{\ell+1}-1} \frac{\delta_t(a_t^*, a)}{|\mathcal{A}_t|} \mathbf{1}_{\{a \in \mathcal{A}_t\}}\right] \leq c_8 K \log^3(T)\, \mathbb{E}\left[\sum_{i \in \text{Phases}(t_\ell, t_{\ell+1})} \sqrt{\tau_{i+1}-\tau_i}\right]. \tag{28}
$$

C.1.4 Summing Over Episodes

In Section C.1.2 and Section C.1.3, we bounded the regret of arms within an episode before and after their elimination, respectively. Combining (22) and

(28), and summing over episodes, we then obtain

$$\mathbb{E}\left[\sum_{t=1}^{T}\frac{\delta_t(a_t^*,a_t)+\delta_t(a_t^*,b_t)}{2}\right] \leq c_9 K\log^3(T)\,\mathbb{E}\left[\mathbf{1}_{\{\mathcal{E}\}}\sum_{\ell=1}^{L}\sum_{i\in\text{Phases}(t_\ell,t_{\ell+1})}\sqrt{\tau_{i+1}-\tau_i}\right]+\frac{1}{T}.$$

Now, on the concentration event $\mathcal{E}$, Lemma C.1.3 tells us that any phase $[\tau_i,\tau_{i+1})$ intersects with at most two episodes. Recall that $\tau_0 \coloneqq 1$ and $\tau_{S^{\mathsf{CW}}+1} \coloneqq T$. It then follows from the above that

$$\mathbb{E}\left[\sum_{t=1}^{T}\frac{\delta_t(a_t^*,a_t)+\delta_t(a_t^*,b_t)}{2}\right] \leq 2c_9 K\log^3(T)\sum_{i=0}^{S^{\mathsf{CW}}}\sqrt{\tau_{i+1}-\tau_i}+\frac{1}{T}.$$

C.2 Missing Details from Section III.5

C.2.1 Significant CW Switches

Let us first recall the definition of Significant Condorcet Winner Switches from Section III.2.2. Let $\nu_0 \coloneqq 1$ and define ν_{i+1} recursively as the first round in $[\nu_i, T)$ such that for all arms $a \in [K]$ there exist rounds $\nu_i \leq s_1 < s_2 < \nu_{i+1}$ such that

$$\sum_{t=s_1}^{s_2}\delta_t(a_t^*,a) \geq \sqrt{K(s_2-s_1)}. \tag{29}$$

Let $\tilde{S}^{\mathsf{CW}}$ denote the number of such Significant CW Switches $\nu_1,\ldots,\nu_{\tilde{S}^{\mathsf{CW}}}$. The key idea of [SK22b] when developing this notion of non-stationarity (for multi-armed bandits) is that a restart in exploration is only warranted if there are no *safe* arms left to play, i.e. there is no arm left that does not suffer regret (29) on some interval $[s_1,s_2]$. For every phase $[\nu_i,\nu_{i+1})$, we denote by a_i^s the last safe arm in phase i, i.e. the last arm to satisfy (29) in phase i. Moreover. we define the sequence of safe arms as $a_t^s = a_i^s$ for $t \in [\nu_i,\nu_{i+1})$.

Significant CW Switches are able to reconcile switch-based non-stationarity measures such as CW Switches S^{CW} and variation-based non-stationarity measures such as the CW Variation $\tilde{V}$. More specifically, it naturally holds that $\tilde{S}^{\mathsf{CW}} \leq S^{\mathsf{CW}}$ and Corollary III.5.6 shows that near-optimal dynamic regret w.r.t. $\tilde{S}^{\mathsf{CW}}$ also implies near-optimal dynamic regret w.r.t. $\tilde{V}$.

C.2.2 Proof of Theorem III.5.4

For convenience, we recall the assumptions of Theorem III.5.4.

Assumption C.2.1 (Strong Stochastic Transitivity)**.** Every preference matrix P_t satisfies that if $a \succ_t b \succ_t c$, we have $\delta_t(a,c) \geq \delta_t(a,b) \vee \delta_t(b,c)$.

Assumption C.2.2 (Stochastic Triangle Inequality)**.** Every preference matrix P_t satisfies that if $a \succ_t b \succ_t c$, we have $\delta_t(a,c) \leq \delta_t(a,b) + \delta_t(b,c)$.

We see that together Assumption III.5.1 and Assumption III.5.2 imply a more general type of triangle inequality for any triplet $a, b, c \in [K]$ with $a \succ b$ and $a \succ c$.

Lemma C.2.3. *Under Assumption 1 and Assumption 2, for any triplet $a, b, c \in [K]$ with $a \succ_t b$ and $a \succ_t c$, it holds that*

$$\delta_t(a,c) \leq 2\delta_t(a,b) + \delta_t(b,c).$$

Proof. Suppose that $b \succ_t c$. Then, the claim follows directly from the stochastic triangle inequality, since $\delta_t(a,c) \leq \delta_t(a,b) + \delta_t(b,c)$. Suppose that $c \succ_t b$. Leveraging strong stochastic transitivity of the triplet $a \succ_t c \succ_t b$, we have

$$\delta_t(a,b) \geq \delta_t(a,c) \vee \delta_t(c,b).$$

This implies that $\delta_t(a,c) \leq \delta_t(a,b)$ as well as $\delta_t(c,b) \leq \delta_t(a,b)$. By definition of the gaps, this also yields $|\delta_t(b,c)| \leq \delta_t(a,b)$, since $c \succ_t b$. Consequently, it holds that $\delta_t(a,c) \leq 2\delta_t(a,b) + \delta_t(b,c)$. ■

As briefly discussed in Section III.5, these assumptions on the preference sequence $P_1, \ldots, P_T$ allow us to decompose the dynamic regret so that we can compare against a temporarily fixed benchmark.

We can w.l.o.g. assume that $a_t^* \succ_t a_t$ and $a_t^* \succ_t a_t^s$. To see that this assumption is valid, note that a_t^* is the Condorcet winner in round t and it is then easy to verify that Lemma C.2.3 also holds if a_t^* equals one (or both) of a_t and a_t^s. Applying Lemma C.2.3 to a_t^*, a_t^s and a_t, we have

$$\delta_t(a_t^*, a_t) \leq 2\delta_t(a_t^*, a_t^s) + \delta_t(a_t^s, a_t).$$

Recalling equation (11) from Section C.1, we then get the following decomposition of the dynamic regret within each episode as

$$\mathbb{E}\left[\sum_{t=t_\ell}^{t_{\ell+1}-1} \frac{\delta_t(a_t^*, a_t) + \delta_t(a_t^*, b_t)}{2}\right] \leq 2\underbrace{\mathbb{E}\left[\sum_{t=t_\ell}^{t_{\ell+1}-1} \delta_t(a_t^*, a_t^s)\right]}_{\tilde{R}_1(\ell)} + \underbrace{\mathbb{E}\left[\sum_{t=t_\ell}^{t_{\ell+1}-1} \delta_t(a_t^s, a_t)\right]}_{\tilde{R}_2(\ell)}.$$

C.2.2.1 Bounding $\tilde{R}_1(\ell)$

We can bound $\tilde{R}_1(\ell)$ directly using the definition of Significant CW Switches. By definition of a_i^s as the last safe arm in phase $[\nu_i, \nu_{i+1})$, i.e. the last arm to satisfy (29) for some interval $[s_1, s_2] \subseteq [\nu_i, \nu_{i+1})$, it holds that

$$\sum_{t=\nu_i}^{\nu_{i+1}} \delta_t(a_t^*, a_i^s) \leq \sqrt{K(\nu_{i+1} - \nu_i)}.$$

We can then sum over all phases $i \in [\tilde{S}^{\mathsf{CW}}]$ to obtain

$$\sum_{t=1}^{T} \delta_t(a_t^*, a_t^s) \leq \sum_{i=1}^{\tilde{S}^{\mathsf{CW}}} \sqrt{K(\nu_{i+1} - \nu_i)}.$$

C.2.2.2 Bounding $\tilde{R}_2(\ell)$

As briefly mentioned in the main text, the difficulty in bounding $\sum_{t=t_\ell}^{t_{\ell+1}-1} \delta_t(a_t^*, a_t)$ for Significant CW Switches is that the identity of the Condorcet winner, i.e. a_t^*, may change several times within each significant phase $i \in [\tilde{S}^{\mathsf{CW}}]$. This makes accurately tracking $\delta_t(a_t^*, a)$ (nearly) impossible even across small intervals and the arguments that we used to prove Theorem III.3.1 fail.

In contrast, when we consider the relative regret of a_t against the last safe arm a_t^s (or sequence thereof), this difficulty can be resolved. Considering a_t^s (instead of a_t^*) as a benchmark ensures that within each phase $i \in [\tilde{S}^{\mathsf{CW}}]$ the comparator arm is fixed, since $a_t^s = a_i^s$ for all $t \in [\nu_i, \nu_{i+1})$. Hence, the relative regret w.r.t. a_t^s can still be dealt with. In particular, the proof of Theorem III.3.1 from Section C.1 can be seen to hold with minor changes when swapping a_t^* for a_t^s and considering significant phases $\nu_1, \dots, \nu_{\tilde{S}^{\mathsf{CW}}}$. For completeness, we reformulate and prove two important lemmas from Section C.1 that relied on properties of a_t^* and $\tau_1, \dots, \tau_{S^{\mathsf{CW}}}$. We want to emphasise that we here again rely on Assumption III.5.1 and Assumption III.5.2.

The following lemma shows that the beginning of a new episode implies a Significant CW Switch, i.e. every arm suffers at least (29) much regret over some interval within the episode.

Lemma C.2.4 (Lemma C.1.3 for $\tilde{S}^{\mathsf{CW}}$)**.** *On event $\mathcal{E}$, for each episode $[t_\ell, t_{\ell+1})$ with $t_{\ell+1} \leq T$, there exists a Significant CW Switch $\nu_i \in [t_\ell, t_{\ell+1})$.*

Proof. The start of a new episode means that every arm $a \in [K]$ has been eliminated from $\mathcal{A}_{\text{good}}$ at some round in $t_\ell^a \in [t_\ell, t_{\ell+1})$. As a result, there must exist an interval $[s_1, s_2] \subseteq [t_\ell, t_\ell^a)$ and some arm $a' \in [K]$ so that the elimination rule (III.2) holds. Using Lemma C.1.2, we then find that for some constant $c_2 > 0$:

$$\sum_{t=s_1}^{s_2} \mathbb{E}\left[\hat{\delta}_t(a', a) \mid \mathcal{F}_{t-1}\right] > c_2 \log(T) K \sqrt{(s_2 - s_1) \vee K^2}. \tag{30}$$

Note that by construction of $\hat{\delta}_t(a', a)$, we always have $\delta_t(a', a) \geq \mathbb{E}[\hat{\delta}_t(a', a) \mid \mathcal{F}_{t-1}]$ since

$$\mathbb{E}[\hat{\delta}_t(a', a) \mid \mathcal{F}_{t-1}] = \begin{cases} \delta_t(a', a) & a', a \in \mathcal{A}_t \\ -1/2 & \text{otherwise.} \end{cases} \tag{31}$$

Applying Lemma C.2.3 to the triplet (a_t^*, a', a), we get that $\delta_t(a_t^*, a) \geq 2\delta_t(a_t^*, a') + \delta_t(a', a) \geq \delta_t(a', a)$. Thus, (30) tells us that there exists no arm

$a \in [K]$ such that for all $[s_1, s_2] \subseteq [t_\ell, t_{\ell+1})$

$$\sum_{t=s_1}^{s_2} \delta_t(a_t^*, a) < \sqrt{K(s_2 - s_1)}.$$

In other words, there is no arm that remains safe to play throughout the episode and there must have been a Significant CW Switch $\nu_i \in [t_\ell, t_{\ell+1})$. ■

The following lemma ensures that the last safe arm a_i^s within phase i is not being eliminated before round ν_{i+1} by any replay $\texttt{CondaLet}(s, m)$ that is scheduled in said phase.

Lemma C.2.5 (Lemma C.1.8 for a_i^s)**.** *On event $\mathcal{E}$, no run of $\texttt{CondaLet}(s, m)$ with $s \in [\nu_i, \nu_{i+1})$ ever eliminates arm a_i^s before round ν_{i+1}.*

Proof. Suppose on the contrary that some $\texttt{CondaLet}(s, m)$ with $s \in [\nu_i, \nu_{i+1})$ eliminates arm a_i^s before round ν_{i+1}. Then, we must have for some arm $a \in [K]$ and interval $[s_1, s_2] \subseteq [s, \nu_{i+1})$ that

$$C \log(T) K \sqrt{(s_2 - s_1) \vee K^2} < \sum_{t=s_1}^{s_2} \hat{\delta}_t(a, a_i^s), \tag{32}$$

In view of the concentration bound (6), this implies on event $\mathcal{E}$ that

$$c_2 \log(T) K \sqrt{(s_2 - s_1) \vee K^2} < \sum_{t=s_1}^{s_2} \mathbb{E}\left[\hat{\delta}_t(a, a_i^s) \mid \mathcal{F}_{t-1}\right] \leq \sum_{t=s_1}^{s_2} \delta_t(a, a_i^s), \tag{33}$$

where the last inequality holds by merit of (31). Now, by the definition of a_i^s as the last safe arm in phase i, it must hold that $\delta_t(a, a_i^s) < \sqrt{K(s_2 - s_1)}$ for all $t \in [\nu_i, \nu_{i+1})$ and all $a \in [K]$. This stands in contradiction to the above which proves Lemma C.2.5. ■

Now, following the same steps as in the proof of Theorem III.3.1 in Section C.1, we obtain for some constant $\tilde{c} > 0$

$$\tilde{R}_2(\ell) \leq \tilde{c} K \log^3(T) \mathbb{E}\left[\sum_{i \in \mathrm{Phases}_{\tilde{S}^{\mathsf{CW}}}(t_\ell, t_{\ell+1})} \sqrt{\nu_{i+1} - \nu_i}\right],$$

where $\nu_{\tilde{S}^{\mathsf{CW}}+1} \coloneqq T$ and $\mathrm{Phases}_{\tilde{S}^{\mathsf{CW}}}(t_1, t_2) \coloneqq \{i \in [\tilde{S}^{\mathsf{CW}}] \colon [\nu_i, \nu_{i+1}) \cap [t_1, t_2) \neq \emptyset\}$. Lastly, in view of the modified Lemma C.2.4, it follows that (cf. Section C.1.4)

$$\mathrm{DR}(T) = \mathbb{E}\left[\sum_{t=1}^{T} \frac{\delta_t(a_t^*, a_t) + \delta_t(a_t^*, b_t)}{2}\right] \leq \tilde{2} c K \log^3(T) \sum_{i=0}^{\tilde{S}^{\mathsf{CW}}} \sqrt{\nu_{i+1} - \nu_i}. \tag{34}$$

An application of Jensen's inequality shows that $\mathrm{DR}(T) \leq \tilde{O}(K\sqrt{\tilde{S}^{\mathsf{CW}} T})$.

C.2.3 Proof of Corollary III.5.6

Recall the definition of the Condorcet Winner Variation from Section III.2.2:

$$\tilde{V} \coloneqq \sum_{t=2}^{T} \max_{a\in[K]} |P_t(a_t^*, a) - P_{t-1}(a_t^*, a)|.$$

We define the CW Variation over phase $[\nu_i, \nu_{i+1})$ as $\tilde{V}_{[\nu_i,\nu_{i+1})} \coloneqq \sum_{t=\nu_i+1}^{\nu_{i+1}} \max_{a\in[K]} |P_t(a_t^*, a) - P_{t-1}(a_t^*, a)|$. Note that in view of the bound in (34), it suffices to show that $\sum_{i=0}^{\tilde{S}^{\mathsf{CW}}} K\sqrt{\nu_{i+1}-\nu_i} \le K\sqrt{T} + \tilde{V}^{1/3}(KT)^{2/3}$.

Consider a phase $[\nu_i, \nu_{i+1})$ with $0 \le i < \tilde{S}^{\mathsf{CW}}$. By definition of Significant CW Switches, every arm $a \in [K]$ must satisfy on some interval $[s_1, s_2] \subseteq [\nu_i, \nu_{i+1})$ that

$$\sum_{t=s_1}^{s_2} \delta_t(a_t^*, a) \ge \sqrt{K(s_2 - s_1)}.$$

In particular, this is also the case for the Condorcet winner $a^*_{\nu_{i+1}}$ in round ν_{i+1}. Then, since $\sqrt{s_2 - s_1} > \sum_{t=s_1}^{s_2} \frac{1}{\nu_{i+1}-\nu_i}$, there exists a round $t \in [s_1, s_2]$ such that $\delta_t(a_t^*, a^*_{\nu_{i+1}}) \ge \sqrt{\frac{K}{\nu_{i+1}-\nu_i}}$. We then have

$$\begin{aligned}
\sqrt{\frac{K}{\nu_{i+1}-\nu_i}} &\le \delta_t(a_t^*, a^*_{\nu_{i+1}}) \\
&\le \delta_t(a_t^*, a^*_{\nu_{i+1}}) + \delta_{\nu_{i+1}}(a^*_{\nu_{i+1}}, a_t^*) \\
&\le \delta_t(a_t^*, a^*_{\nu_{i+1}}) - \delta_{\nu_{i+1}}(a_t^*, a^*_{\nu_{i+1}}) \\
&\le |\delta_t(a_t^*, a^*_{\nu_{i+1}}) - \delta_{\nu_{i+1}}(a_t^*, a^*_{\nu_{i+1}})| \\
&= |P_t(a_t^*, a^*_{\nu_{i+1}}) - P_{\nu_{i+1}}(a_t^*, a^*_{\nu_{i+1}})| \\
&\le \sum_{s=t+1}^{\nu_{i+1}} \max_{a\in[K]} |P_t(a_t^*, a) - P_{t-1}(a_t^*, a)| \le \tilde{V}_{[\nu_i,\nu_{i+1}]},
\end{aligned}$$

where we used that $\delta_{\nu_{i+1}}(a^*_{\nu_{i+1}}, a_t^*) \ge 0$ and $\delta_{\nu_{i+1}}(a^*_{\nu_{i+1}}, a_t^*) = -\delta_t(a_t^*, a^*_{\nu_{i+1}})$ in the second and third inequality, respectively. We can now apply Hölder's inequality to obtain

$$\begin{aligned}
\sum_{i=0}^{\tilde{S}^{\mathsf{CW}}} K\sqrt{\nu_{i+1}-\nu_i} &\le K\sqrt{T} + \sum_{i=0}^{\tilde{S}^{\mathsf{CW}}-1} K\sqrt{\nu_{i+1}-\nu_i} \\
&\le K\sqrt{T} + \left(\sum_{i=0}^{\tilde{S}^{\mathsf{CW}}} \sqrt{\frac{K}{\nu_{i+1}-\nu_i}}\right)^{1/3} \left(\sum_{i=0}^{\tilde{S}^{\mathsf{CW}}} K^{5/4}(\nu_{i+1}-\nu_i)\right)^{2/3} \\
&\le K\sqrt{T} + \left(\sum_{i=0}^{\tilde{S}^{\mathsf{CW}}} \tilde{V}_{[\nu_i,\nu_{i+1})}\right)^{1/3} K^{5/6}\, T^{2/3}
\end{aligned}$$

$$= K\sqrt{T} + \tilde{V}^{1/3} K^{5/6} T^{2/3}.$$

The above dependence on K can be improved to $K^{4/9}$ (which is even smaller than the $K^{2/3}$ dependence in Corollary III.5.6) by modifying the definition of Significant CW Switches so that ν_{i+1} is the first round in $[\nu_i, T)$ such that for all arms $a \in [K]$ there exist rounds $\nu_i \leq s_1 < s_2 < \nu_{i+1}$ with

$$\sum_{t=s_1}^{s_2} \delta_t(a_t^*, a) \geq K\sqrt{s_2 - s_1}.$$

It is straightforward to check that Theorem III.5.4 holds true also for this definition of Significant CW Switches.

C.3 More Related Work

Related to the non-stationary dueling bandit problem studied in this paper are adversarial dueling bandits [AKJ14; GUC15; SKM21; Sui+17]. Here, [AKJ14] was the first to study the dueling bandit problem in an adversarial setup and introduced a popular sparring idea, which has been picked up by many follow-up works [Dud+15a; GS22b; GUC15; SKM21]. The settings in [AKJ14] and [GUC15] are restricted to utility-based preference models, where each arm has an associated utility in each round. This entails a complete ordering over the arms in each round, which only covers a small subclass of $[K] \times [K]$ preference matrices. Moreover, [GUC15] assume that the feedback includes not only the winner but also the difference in the utilities between the winning and losing arm, which is more similar to MAB feedback and than the 0/1 one bit preference feedback considered by us. [SKM21] consider the dueling bandit setup for general adversarial preferences, but they measure performance in terms of (static) regret w.r.t. *Borda-scores*. This measure of regret is very different from our preference-based regret objective. In general, the adversarial dueling bandit problem aims to minimize *static regret* w.r.t. some fixed benchmark a^*, whereas we study *dynamic regret* w.r.t. a time-varying benchmark a_t^*. As discussed in Section III.2, static regret can be an undesirable measure of performance when no single fixed arm represents a reasonably good benchmark over all rounds (see Example III.2.1).

Another somewhat related line of work considers the sleeping dueling bandit problem, where the action space is non-stationary (as opposed to the preference sequence). The objective here is to be competitive w.r.t. the best active arm at each round. [SG21] studies the setup for adversarial sleeping but assumes a fixed preference matrix across all rounds.

Paper IV

Minimax-Bayes Reinforcement Learning

Thomas Kleine Buening, Christos Dimitrakakis, Hannes Eriksson, Divya Grover, Emilio Jorge

Published in *26th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2023.

Abstract

While the Bayesian decision-theoretic framework offers an elegant solution to the problem of decision making under uncertainty, one question is how to appropriately select the prior distribution. One idea is to employ a worst-case prior. However, this is not as easy to specify in sequential decision making as in simple statistical estimation problems. This paper studies (sometimes approximate) minimax-Bayes solutions for various reinforcement learning problems to gain insights into the properties of the corresponding priors and policies. We find that while the worst-case prior depends on the setting, the corresponding minimax policies are more robust than those that assume a standard (i.e. uniform) prior.

IV.1 Introduction

Reinforcement learning is the problem of an agent learning how to act in an unknown environment through interaction and reinforcement. In the standard setting, the learning agent acts in an unknown Markov Decision Process μ, within some class of MDPs $\mathcal{M}$. The agent observes the state $s_t \in S$ of the MDP and selects an action $a_t \in A$ using a policy π. It then observes a reward $r_t \in \mathbb{R}$ and the next state s_{t+1}. The agent's goal is to maximise utility, defined as the sum of rewards to some horizon T, $u = \sum_{t=1}^{T} r_t$, in expectation, i.e., $\mathbb{E}_\mu^\pi[u]$, where $\mathbb{E}_\mu^\pi$ is the expectation under the MDP and policy. Since the true μ is unknown, this optimisation problem is ill-posed. In the Bayesian setting, this conundrum is solved by selecting some *subjective* prior distribution β over MDPs and maximising $\mathbb{E}_\beta^\pi[u] = \int_{\mathcal{M}} \mathbb{E}_\mu^\pi[u] \, \mathrm{d}\beta(\mu)$. Then it remains to compute the optimal adaptive (i.e., history-dependent) policy, something that can be only done approximately in general, due to the fact that the number of adaptive policies increases exponentially with the problem horizon.

The above discussion assumes that the agent has *somehow* chosen a prior. However, it is not clear how such a prior can be selected from first principles, if we have no domain knowledge, but still want to be robust. The minimax-Bayes idea [Ber85] is to assume that nature selects the *worst* possible prior β^* for the agent, but *without* knowledge of the agent's policy. This can be formalised by having nature play the minimising player in a simultaneous-move zero-sum game defined by the expected utility $\mathbb{E}^\pi_\beta[u]$, where the agent (who maximises) chooses π, and nature (who minimises) chooses β. In simple Bayesian decision problems (e.g. linear regression) the minimax-Bayes problem is well-studied and β^* sometimes corresponds to a maximum entropy prior. However, in an interactive setting, results are limited to one-shot experiment design [GD04], which shows that maximum entropy priors are not the worst-case priors generally.

In reinforcement learning, which can be seen as a sequential generalisation of one-shot experiment design, this problem has not received much attention in the past. Sometimes, the concept of maximum entropy has been used in reinforcement learning as a penalty term on the policy [e.g. EL21; Haa+18; Tod06] as well as in the context of inverse reinforcement learning [Zie10], but an explicit connection to the minimax-Bayes literature has not been made. In preliminary work, [AD14] analysed variants of the weighted majority algorithm for finding minimax priors in a restricted version of this setting.

Contributions. In this paper, we study the basic theoretical and algorithmic properties of minimax-Bayes reinforcement learning. This includes (a) characterising the existence of solutions under different assumptions on the policy and MDP space (b) defining algorithms, together with convergence guarantees when possible, and (c) performing numerical experiments to illustrate the behaviour of (approximate) minimax-Bayes algorithms and contrast them with Bayesian RL algorithms that assume a standard maximum-entropy (e.g. uniform) prior.

The paper is organised as follows. In Section IV.2, we formally introduce the setting. In Section IV.3, we introduce regret definitions and prove some basic properties of the regret as well as relations between Bayesian regret and Bayes-optimal regret. Section IV.4 discusses the existence of a value for the game between a Bayesian agent and Nature, which selects the prior. Section IV.5 develops algorithms for finding approximately minimax policies in certain policy classes. In particular, we consider (a) finite-horizon Bayes-optimal policies (b) posterior sampling policies, and (c) parametrised adaptive policies. Our results indicate that, not only is an approximate minimax solution achievable in many settings but that they are much more robust than Bayes-adaptive policies under common priors. Finally, Section IV.7 contains the related work and conclusions.

IV.2 Setting

A Markov Decision Process (MDP) is a tuple $\mu = \langle S, A, \mathcal{P}, \rho, T \rangle$, where S is a set of states, A is a set of actions, $\mathcal{P} : S \times A \to \Delta(S)$ is a transition function,

$\rho : S \times A \to [0, 1]$ is a reward function, and T is a (potentially random) horizon. Let $\mathcal{M}$ denote the space of MDPs.

For simplicity, in our theoretical development, we focus on the setting where the agent is acting in a finite state space S with a finite set of actions A, the reward function ρ is known, and the horizon T is fixed and finite, although many of our results could be more generally applicable. In each round t, the agent observes state $s_t \in S$, chooses an action $a_t \in S$ and receives a reward $r_t = \rho(s_t, a_t)$. We write $s^t = (s_1, \dots, s_t)$ and $a^t = (a_1, \dots, a_t)$ for the sequence of states and actions up to round t. Given the reward function, the history $h_t = (s^t, a^{t-1})$ describes the information available to the agent before choosing an action in round t. The agent's utility u is an additive function of individual rewards $u \triangleq \sum_{t=1}^{T} r_t$. The agent is acting in an MDP through a policy $\pi \in \Pi$, where we let Π denote a generic policy space. For a fixed MDP $\mu \in \mathcal{M}$ and policy $\pi \in \Pi$, the expected utility is given by $U(\pi, \mu) \triangleq \mathbb{E}_\mu^\pi[u]$ with maximal utility denoted by $U^*(\mu) \triangleq \max_{\pi \in \Pi} U(\pi, \mu)$.

When the MDP is unknown, as in the reinforcement learning problem, the policy is adaptive and the agent's actions can depend on what it has been observed in the past, as we explain below.

IV.2.1 Policies.

Let $\mathcal{H}$ be the set of all histories. A (stochastic) policy π is a set of probability measures $\{ \pi(\cdot \mid h) \mid h \in \mathcal{H} \}$ on the set of actions A. We denote the set of all behavioural[1] policies by Π^{S}. A policy is *deterministic* if, for each history $h_t = (s^t, a^{t-1})$, there exists an action $a \in A$ such that $\pi(a_t = a \mid h_t) = 1$. We denote the set of deterministic policies by Π^{D}. A policy is *memoryless* (or reactive) if, for all histories h_t with $s_t = s$, we have $\pi(a_t = a \mid h_t) = \pi(a_t = a \mid s_t = s)$. We denote the set of memoryless (stochastic) policies by Π_1^{S}. The set of memoryless deterministic policies is denoted by Π_1^{D}. Obviously, $\Pi_1^{\mathrm{D}} \subset \Pi^{\mathrm{D}} \subset \Pi^{\mathrm{S}}$ and $\Pi_1^{\mathrm{D}} \subset \Pi_1^{\mathrm{S}} \subset \Pi^{\mathrm{S}}$. Finally, for any MDP μ there exists a deterministic, memoryless policy that is optimal, i.e., $U^*(\mu) = \max_{\pi \in \Pi} U(\pi, \mu) = \max_{\pi \in \Pi_1^{\mathrm{D}}} U(\pi, \mu)$ [Put14].

Strategies. Typically, minimax results rely on the notion of mixed strategies. Here, we let $\sigma \in \Delta(\Pi)$ denote a probability measure over a set of base policies Π.

Fact IV.2.1. For any strategy $\sigma \in \Delta(\Pi^{\mathrm{D}})$ there exists an equivalent stochastic policy $\pi \in \Pi^{\mathrm{S}}$ such that $\sigma(a_t|h_t) = \pi(a_t \mid h_t)$ for all histories h_t with positive probability.

[1] That is, history-dependent and stochastic policies.

IV.2.2 Utility and Beliefs

In the following, we overload the $U(\pi, \beta)$ to also mean the expected utility of π with respect to a distribution β over MDPs:

$$U(\pi, \beta) \triangleq \mathbb{E}^{\pi}_{\beta}[u] = \int_{\mathcal{M}} U(\pi, \mu) \, \mathrm{d}\beta(\mu), \tag{IV.1}$$

under appropriate measurability assumptions.

There are two possible ways to interpret the distribution β, depending on how it is chosen. If β is chosen by the agent selecting π, it corresponds to the subjective belief of the decision maker about which is the most likely MDP *a priori*. Then, $U(\pi, \beta)$ corresponds to the expected utility of a particular policy under this belief. Let

$$U^*(\beta) \triangleq \max_{\pi \in \Pi} U(\pi, \beta)$$

denote the Bayes-optimal utility for a belief. We recall the fact that this is a convex function [c.f. DeG70]. By definition, the following bounds hold:

$$U(\pi, \beta) \leq U^*(\beta) \leq \int_{\mathcal{M}} U^*(\mu) \, \mathrm{d}\beta(\mu), \qquad \forall \pi \in \Pi,$$

so that $U^*(\beta)$ is convex with respect to β. In the above, the left-hand side is the utility of an arbitrary policy, while the right side can be seen as the expected utility we would obtain if the true MDP was revealed to us.

The second view of β is to assume that the MDP is *actually* drawn randomly from the distribution β. If this is known, then the subjective value of a policy is equal to its true expected value. However, it is more interesting to consider the case where nature arbitrarily selects β from a set of possible priors $\mathcal{B}$. Then we wish to find a policy π^* achieving:

$$\max_{\pi \in \Pi} \min_{\beta \in \mathcal{B}} U(\pi, \beta). \tag{IV.2}$$

A minimax solution exists if the game *has a value*, i.e., $\max_{\pi \in \Pi} \min_{\beta \in \mathcal{B}} U(\pi, \beta) = \min_{\beta \in \mathcal{B}} \max_{\pi \in \Pi} U(\pi, \beta)$. Then there exists a maximin policy π^* which is optimal in response to some minimax belief β^*, and vice versa. A sufficient condition for this to occur is for $U^*(\beta)$ to be convex and differentiable everywhere [c.f. GD04]. In particular, a maximin *strategy* (i.e., a distribution over policies) can always be found when Π is finite. On the other hand, for any fixed prior β, there is always an optimal deterministic policy. Note that this is only a *best-response* policy and not a solution to the maximin problem (IV.2).

Fact IV.2.2. For any distribution β over MDPs, there exists a deterministic, history-dependent policy that is optimal, i.e. $U^*(\beta) = \max_{\pi \in \Pi} U(\pi, \beta) = \max_{\pi \in \Pi^{\mathrm{D}}} U(\pi, \beta)$.

Unfortunately, looking at the problem from the point of view of utility maximisation is somewhat problematic. This is because an unrestricted set of priors for nature may lead to absurd solutions: nature could pick a prior so that

all rewards are zero, thus trivially achieving minimal utility. For that reason, we actually focus on the problem of minimax *regret*, i.e., the gap between the agent's policy and that of an oracle. We give the appropriate definitions in the next section.

IV.3 Properties of the regret

We generally write $R(\pi, \mathcal{I})$ to mean the regret of some algorithmic policy π relative to an oracle with information $\mathcal{I}$.

Let us start with the regret of a policy relative to an oracle that knows the underlying MDP:

Definition IV.3.1 (Regret)**.** The regret of a policy π for an MDP μ is $R(\pi, \mu) \triangleq U^*(\mu) - U(\pi, \mu)$.

Since this regret notion may be too strong, it is also interesting to define the regret of a policy with respect to the oracle that knows β. This allows us to take into account oracles which have less knowledge than the actual MDP.

Definition IV.3.2 (Bayes-optimal Regret)**.** This is the regret of a policy π with respect to the Bayes-optimal policy[2] for β: $R(\pi, \beta) \triangleq U^*(\beta) - U(\pi, \beta) = \int_{\mathcal{M}} \mathrm{d}\beta(\mu)[U(\pi^*(\beta), \mu) - U(\pi, \mu)]$, where $\pi^*(\beta) = \arg\max_\pi U(\pi, \beta)$.

This notion of regret tells us how much we lose relative to a computationally unbounded oracle that knows the prior. We can use it to measure the loss both due to a misspecified prior, by fixing $\pi^*(\beta_0)$ to some prior β_0 and examining $R(\pi^*(\beta_0), \beta)$ as the actual prior β varies, and due to computational approximations, by measuring $R(\pi^*_\epsilon(\beta), \beta)$ for policies calculated with some approximate algorithm.

Finally, we may wish to subjectively calculate our expected regret under an oracle that knows the underlying MDP. Since the agent does not know the underlying MDP, it necessarily measures regret under a Bayesian prior.

Definition IV.3.3 (Bayesian regret)**.** The Bayesian regret of a policy π under a prior β is $L(\pi, \beta) \triangleq \mathbb{E}_{\mu\sim\beta}[R(\pi, \mu)] = \sum_\mu \beta(\mu) R(\pi, \mu) = \sum_\mu \beta(\mu)[U^*(\mu) - U(\pi, \mu)]$.

These definitions of regret are closely related, as we shall show in the remainder. It will be illuminating to look at the difference between the regret the agent subjectively expects to suffer with respect to some prior distribution β, relative to the regret of the same policy compared to the Bayes-optimal policy for the same prior.

Remark IV.3.4. The Bayesian regret of a policy π is greater than the Bayes-optimal regret, i.e., $R(\pi, \beta) \leq L(\pi, \beta)$.

[2]Generally this policy will belong to the set of history-dependent policies, but in some cases, it makes sense to restrict them to e.g. a subset of parametrised policies.

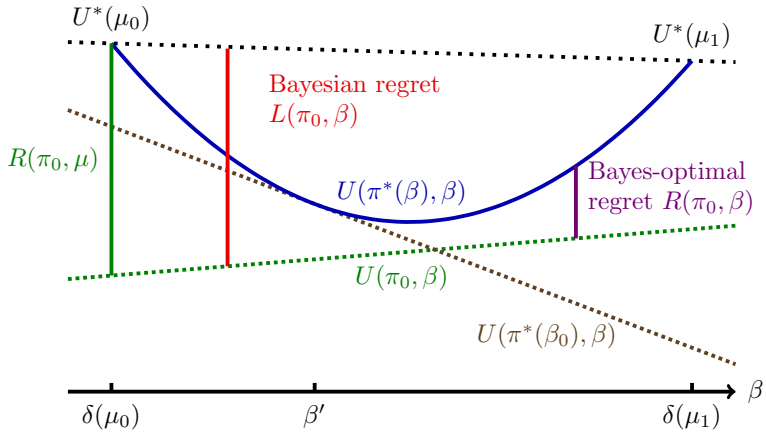


Figure IV.1: Illustration of the notions of regret for different policies with a belief β over two MDPs μ_1 and μ_2, where $\delta(\mu)$ denotes the Dirac belief on μ. Any *fixed* policy π_0 will have a utility that is a linear function of the belief (green dotted line). The blue curve shows the utility of the Bayes-optimal policy $\pi^*(\beta) = \operatorname{argmax}_\pi U(\pi, \beta)$. This policy is prior-aware, and hence not fixed, but depends on the prior β. Note that by definition, $U(\pi^*(\beta), \beta)$ is convex. However, if we fix a Bayes-optimal policy for a specific prior β_0, we obtain a tangent $U(\pi^*(\beta_0), \beta)$ to the Bayes-optimal curve at β_0. The Bayesian regret (of π_0) (red line) is the expected regret of a policy compared against an oracle that knows the MDP (black dotted line). The Bayes-optimal regret (of π_0) is the difference in performance to the Bayes-optimal policy (purple line).

Proof. Note that $R(\pi, \beta) = \int_{\mathcal{M}} \mathrm{d}\beta(\mu)[U(\pi^*(\beta), \mu) - U(\pi, \mu)] \leq \int_{\mathcal{M}} \mathrm{d}\beta(\mu)[U^*(\mu) - U(\pi, \mu)] = L(\pi, \beta)$, since $U(\pi^*(\beta), \mu) \leq U^*(\mu)$ by definition of $U^*(\mu)$. ■

The above also follows from the fact that for any policy π and prior β, the Bayesian regret of π equals the Bayesian regret of the Bayes-optimal policy[3] plus the Bayes-optimal regret of π, that is, $L(\pi, \beta) = L(\pi^*(\beta), \beta) + R(\pi, \beta)$. Geometrically, this follows from the fact that the utility of any fixed policy is lower bounding the convex Bayes-optimal utility curve, as can be seen in Figure IV.1. The following fact also follows from a simple geometrical argument:

Remark IV.3.5. $R(\pi, \beta)$ is convex in β.

Proof. By definition of the Bayesian-optimal regret, we have $R(\pi, \beta) = U^*(\beta) - \mathbb{E}_{\mu\sim\beta}[U(\pi, \mu)]$. As $U^*(\beta)$ is convex in β and $\mathbb{E}_{\mu\sim\beta}[U(\pi, \mu)]$ is linear in β, their difference is also convex. ■

Of course, the game where nature sees the agent's policy π first before selecting a prior is strictly determined and nature can simply select a single

[3]This is equal to the difference between the Bayes-optimal value and the upper bound.

MDP (Dirac distribution) as its best response to π. In this particular case, this follows directly from the convexity of the Bayes-optimal regret.

Following the steps of the proof by [Lat21] for the bandit case, we can show that the maximum regret is attained in Dirac beliefs. Here, we let $\mathcal{B}$ denote the set of beliefs and we work under the assumption that the degenerate beliefs are contained in the belief space.

Lemma IV.3.6 ([Lat21])**.** *If for each MDP $\mu \in \mathcal{M}$ there exists an associated Dirac belief $\beta_\mu \in \mathcal{B}$, then for any policy π we have* $\max_{\mu\in\mathcal{M}} R(\pi,\mu) = \max_{\beta\in\mathcal{B}} R(\pi,\beta)$.

This immediately implies that the minimax regret is the same over both beliefs and MDPs:

$$\min_{\pi\in\Pi}\max_{\mu\in\mathcal{M}} R(\pi,\mu) = \min_{\pi\in\Pi}\max_{\beta\in\mathcal{B}} R(\pi,\beta) \tag{IV.3}$$

We find a similar result for the Bayesian regret.

Lemma IV.3.7. *If for each MDP $\mu \in \mathcal{M}$ there exists an associated Dirac belief $\beta_\mu \in \mathcal{B}$, then for any π:*

$$\max_{\mu\in\mathcal{M}} R(\pi,\mu) = \max_{\beta\in\mathcal{B}} L(\pi,\beta). \tag{IV.4}$$

Proof. For any $\beta \in \mathcal{B}$, we have

$$\begin{aligned}
\max_{\mu\in\mathcal{M}} R(\pi,\mu) &\geq \max_{\mu\in\mathrm{supp}(\beta)} R(\pi,\mu)\\
&= \max_{\mu\in\mathrm{supp}(\beta)} U(\pi^*(\mu),\mu) - U(\pi,\mu)\\
&\geq \int_{\mathrm{supp}(\beta)} \mathrm{d}\beta(\mu)[U(\pi^*(\mu),\mu) - U(\pi,\mu)]\\
&= L(\pi,\beta).
\end{aligned}$$

Consequently $\max_\mu R(\pi,\mu) \geq \max_\beta L(\pi,\beta)$. Using $\delta(\mathcal{M})$ to denote the set of Dirac beliefs over $\mathcal{M}$, we have: $\max_\beta L(\pi,\beta) \geq \max_{\beta\in\delta(\mathcal{M})} L(\pi,\beta) = \max_{\mu\in\mathcal{M}} R(\pi,\mu)$, due to the fact that $R(\pi,\mu) = L(\pi,\beta_\mu)$ for the singular belief β_μ on MDP μ. As a result, it must hold that $\max_{\mu\in\mathcal{M}} R(\pi,\mu) \geq \max_{\beta\in\mathcal{B}} L(\pi,\beta) \geq \max_{\mu\in\mathcal{M}} R(\pi,\mu)$. ■

[LS19] show that for the problem of prediction with partial information, the minimax regret equals the minimax Bayesian regret. We show that this also holds in a general setting, as an immediate consequence of Lemma IV.3.7.

Corollary IV.3.8. *If for each MDP $\mu \in \mathcal{M}$ there exists an associated Dirac belief $\beta_\mu \in \mathcal{B}$, then for any π:*

$$\min_{\pi\in\Pi}\max_{\mu\in\mathcal{M}} R(\pi,\mu) = \min_{\pi\in\Pi}\max_{\beta\in\mathcal{B}} L(\pi,\beta) \tag{IV.5}$$

Equations (IV.3) and (IV.5) can be made intuitive through a simple geometric argument. Due to the linearity of the expected regret with respect to the belief for any fixed policy, the best response for nature always includes singular beliefs.

IV.4 Minimax theorems

The above results merely make precise the intuition that when playing second, nature does not need to randomise: it can simply pick the worst-case MDP for the policy we have chosen. However, we typically want to model a worst-case setting by assuming nature picks its distribution without knowing which policy the decision maker will pick. For that reason, it is important to investigate whether the normal form game against nature, where nature and the agent play without seeing each other's move, has a value. We would expect this to be the case if the regret was a bilinear function of the policy and prior. Consequently, the answer is positive with respect to both the Bayesian regret and the utility in the finite setting. However, this is not the case for the Bayes-optimal regret.

Corollary IV.4.1. *For a finite set of MDPs in a finite state-action space, with a known reward function and a finite horizon, the utility and Bayesian regret satisfy:*

$$\min_{\beta\in\mathcal{B}} \max_{\pi\in\Pi} U(\pi,\beta) = \max_{\pi\in\Pi} \min_{\beta\in\mathcal{B}} U(\pi,\beta), \tag{IV.6}$$

$$\max_{\beta\in\mathcal{B}} \min_{\pi\in\Pi} L(\pi,\beta) = \min_{\pi\in\Pi} \max_{\beta\in\mathcal{B}} L(\pi,\beta) \tag{IV.7}$$

Proof. First note that, due to Fact IV.2.1, the stochastic policy π can always be written as a distribution σ over deterministic behavioural policies $d \in \Pi^{\mathrm{D}}$ so that $U(\pi,\beta) = \sum_\mu \sum_d \beta(\mu) U(d,\mu)\sigma(d)$. The result follows from the standard minimax theorem. Similarly for regret, we use $L(\pi,\beta) = \sum_\mu \sum_d \beta(\mu) R(d,\mu)\sigma(d)$. ■

The same does not hold for the Bayes-optimal regret, since for arbitrary policy spaces the agent's Bayes-optimal policy has zero Bayes-optimal regret, as it is aware of the prior distribution. However, the minimax value is generally greater than zero.

Lemma IV.4.2. *The game $R(\pi,\beta)$ does not have a value when $\mathcal{M}$ contains at least two MDPs μ,μ' whose optimal policy sets have an empty intersection.*

Proof. For $\pi \in \Pi^{\mathrm{D}}$, we have $\max_\beta \min_\pi R(\pi,\beta) = 0$, so that $\min_\pi \max_\beta R(\pi,\beta) \geq \max_\beta \min_\pi R(\pi,\beta) = 0$. From (IV.3), it then follows that $\min_\pi \max_\mu R(\pi,\mu) = \min_\pi \max_\beta R(\pi,\beta) \geq \max_\beta \min_\pi R(\pi,\beta) = 0$. It remains to show that $\min_\pi \max_\mu R(\pi,\mu) > 0$. Assume the contrary. Then there is some policy π^* for which $\max_\mu R(\pi^*,\mu) = 0$. However, there exists at least one μ' whose optimal policy does not coincide with π^*, hence $R(\pi^*,\mu') > 0$. ■

Finally, it is interesting to consider the Bayesian regret of the Bayes-optimal policy. For the worst-case Bayesian regret of the Bayes-optimal policy, we find that it is equal to the minimax Bayesian regret.

Lemma IV.4.3. *For finite $\mathcal{M}$, the worst-case Bayesian regret of the Bayes-optimal policy equals the minimax Bayesian regret, i.e.,*

$$\max_{\beta\in\mathcal{B}} L(\pi^*(\beta),\beta) = \max_{\beta\in\mathcal{B}} \min_{\pi\in\Pi} L(\pi,\beta) = \min_{\pi\in\Pi} \max_{\beta\in\mathcal{B}} L(\pi,\beta).$$

Proof. By definition of the Bayes-optimal policy, we have $U(\pi^*(\beta), \beta) = \max_\pi U(\pi, \beta)$. Thus,

$$\begin{aligned}\max_\beta L(\pi^*(\beta), \beta) &= \max_\beta \sum_\mu \beta(\mu)[U^*(\mu) - U(\pi^*(\beta), \mu)] \\ &= \max_\beta \min_\pi \sum_\mu \beta(\mu)[U^*(\mu) - U(\pi, \mu)] \\ &= \max_\beta \min_\pi L(\pi, \beta).\end{aligned}$$

While the above holds for arbitrary $\mathcal{M}$, for the second equality we need to use Corollary IV.4.1, which states that the game has a value when $\mathcal{M}$ is finite, so that $\max_\beta \min_\pi L(\pi, \beta) = \min_\pi \max_\beta L(\pi, \beta)$. ■

It is important to emphasise that this does not imply that $\pi^*(\beta^*)$ is a minimax policy, but merely that its value at the worst-case belief β^* is equal to the value of the game. As we shall see in Section IV.6.2, in settings with a finite number of policies, β^* is located at a vertex with at least two best response policies π^*, where the minimax policy must be a mixture between those.

Open questions. This concludes our preliminary discussion of minimax values for Bayesian games on MDPs. While it is clear that standard minimax theorems apply in the discrete case when we consider stochastic policies, it is an open question whether those can be extended to a more general setting. In particular, do the utility and Bayesian regret game have a value with an uncountable family of priors such as the Dirichlet-product prior? It is also an open question whether a value for the game exists when we are restricted to deterministic policies in some cases. We conjecture that this is generally not the case. For example in discrete, finite horizon problems, the set of policies pure deterministic policies is finite, and so it is unlikely that one of them is maximin. We explore these questions experimentally, after we first develop some algorithms in the following section.

IV.5 Algorithms

In this section, we attempt to answer some of the above questions empirically. In particular, does there exist an equilibrium for bandit problems, where the Bayes-optimal policy can be efficiently approximated through Gittins indices? What about settings where we must restrict the policy space to parametrised or tree policies? Does solving the minimax problem approximately lead to robust policies? Are the worst-case priors we obtain through optimisation actually preferable in some way to standard priors such as the uniform one? For example, do they lead to more robust policies?

For the infinite horizon case, we cannot consider the Bayes-optimal regret, as it requires us to compute the Bayes-optimal policy. However, we can always target the Bayesian regret, which is an upper bound on the Bayes-optimal regret.

(And since the former is usually the same as the minimax regret, it gives us a minimax policy).

Section IV.5.1 describes a stochastic gradient descent-ascent algorithm for finding an approximate minimax regret pair. For the finite horizon case, we can obtain the Bayes-optimal response to any prior distribution. More specifically, when the set of possible MDPs is finite, and we have an optimal policy oracle, we can employ a cutting plane algorithm, described in Section IV.5.2. This allows us to obtain the set of all best response policies to the worst-case prior, and hence the minimax policy.

IV.5.1 Gradient descent ascent

We want to calculate the minimax pair (π^*, β^*) for the Bayesian regret. This can be done through gradient descent-ascent (GDA) [LJJ20], which alternates performing a gradient step for the prior and performing a gradient step for the policy. We show convergence guarantees for GDA in the finite MDP setting, for certain parametrisations of the policy. To calculate the minimax solution for the Bayesian regret, we need the gradient with respect to the policy and the prior.

$$\nabla_\pi L(\pi, \beta) = -\int_{\mathcal{M}} d\beta(\mu) \nabla_\pi U(\pi, \mu) \tag{IV.8}$$

$$\nabla_\beta L(\pi, \beta) = \int_{\mathcal{M}} R(\pi, \mu) \nabla_\beta d\beta(\mu). \tag{IV.9}$$

Intuitively, Algorithm 12 works as follows: First, we sample M MDPs from the current prior β_{t-1}. We use those to do a policy gradient step and obtain a new policy π_t using standard policy gradient algorithms, as well as a gradient step in the prior space to obtain a new prior β_t. Since each gradient may not be exact, we use $G_\pi(\pi, \beta)$ and $G_\beta(\pi, \beta)$ to denote the approximate gradient with respect to the policy and prior respectively. Appendix D.1 describes how we obtain those in detail. Since gradient steps may lead us outside the feasible prior space $\mathcal{B}$, we use a projection $\mathcal{P}_\mathcal{B}$ to ensure we have a valid prior distribution. Finally, we return a randomly selected policy-prior pair from the ones generated during the algorithm's run.

IV.5.1.1 Convergence guarantees for finite set of MDPs

In the MDP setting with n MDPs, we have $\mathcal{B}$ as the probability simplex which has the diameter $D = \sqrt{2}$. Additionally, the gradient

$$\nabla_\beta L(\pi, \beta) = \sum_i^n R(\pi, \mu_i) \nabla_\beta P(\mu_i | \beta) \tag{IV.10}$$

$$\nabla_{\beta_i} L(\pi, \beta) = R(\pi, \mu_i) \tag{IV.11}$$

is constant and therefore convex.

Algorithm 12 Stochastic GDA

input: policy π_0, belief β_0, learning rates (η_π, η_β) and stochastic gradient estimators G_π, G_β for $\nabla_\pi L, \nabla_\beta L$
for $t = 1, \ldots, T$ **do**
Using M i.i.d. samples, get directions

$$g_\beta = \frac{1}{M} \sum_i G_\beta^{(i)}(\pi_{t-1}, \beta_{t-1}) \quad \text{and} \quad g_\pi = \frac{1}{M} \sum_i G_\pi^{(i)}(\pi_{t-1}, \beta_{t-1}).$$

$\pi_t \leftarrow \pi_{t-1} - \eta_\pi g_\pi$
$\beta_t \leftarrow \mathcal{P}_\mathcal{B}\Big(\beta_{t-1} + \eta_\beta g_\beta\Big)$
return: β^*, π^* uniformly at random from $\{(\beta_1, \pi_1), \ldots, (\beta_T, \pi_T)\}$

Lemma IV.5.1. *If the policy π is parameterised as a softmax over actions, independently for each h_t and the horizon T is fixed. Then $L(\pi, \beta)$ is $T^2(|A|+1)$-smooth and $L(\cdot, \beta)$ is T^2-Lipschitz*

With these properties, and a batch size $M = 1$, the requirements of Theorem 4.9 of [LJJ20] are fulfilled and Algorithm 12 will find a $\epsilon-$stationary point in terms of Moreau envelopes, given appropriate step sizes, with an iteration complexity of

$$\mathcal{O}\left(|A|^3 T^6 \left(\frac{\left(T^4 + \sigma^2\right)\widehat{\Delta}_\Phi}{\epsilon^6} + \frac{\widehat{\Delta}_0}{\epsilon^4}\right) \max\left\{1, \frac{\sigma^2}{\epsilon^2}\right\}\right), \tag{IV.12}$$

as long as $\mathbb{E}_G\left[\|G(\pi, \beta) - \nabla L(\pi, \beta)\|^2\right] \leq \sigma^2$. Note that no guarantees exist for general non-convex non-concave Bayesian regret L, as is the case for Dirichlet beliefs and parametric policies.

Here the stationarity is defined as $\|\nabla \Phi_{1/2l}(\pi)\|_2 \leq \epsilon$ as in [LJJ20]. We have $\Phi(\cdot) = \max_{\beta \in \mathcal{B}} L(\cdot, \beta)$ and $\Phi_\lambda(\pi) = \min_{w \in \Pi} \Phi(w) + (1/2\lambda)\|\omega - \pi\|_2^2$ is the Moreau envelope of Φ. Finally we obtain $\widehat{\Delta}_\Phi = \Phi_{1/2l}(\pi_0) - \min_\pi \Phi_{1/2l}(\pi)$ and $\widehat{\Delta}_0 = \Phi(\pi_0) - L(\pi_0, \beta_0)$.

IV.5.2 Cutting planes

In this section we demonstrate an efficient method for localising the minimax pair (π^*, β^*) for beliefs over a finite set of MDPs, given that an oracle for the Bayes-optimal policy for a given belief is available. This could for example be obtained in finite horizon tasks with a sufficiently small horizon such that a tree-policy is tractable. An example of this can be found in [Duf02, Section 1.5].

We use the approximate centroid cutting plane algorithm from [BV04], which can be seen as a high dimensional extension of the bisection algorithm. The goal here is to find a way to repeatedly obtain a plane where we can reject one side of the half-plane, quickly shrinking the plausible set of beliefs.

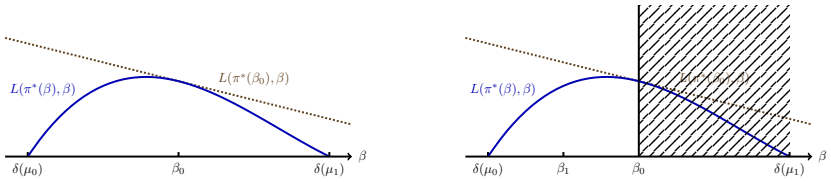


Figure IV.2: Illustration of cutting plane algorithm for two dimensions. The left image illustrates the Bayesian regret plane obtained for queried belief β_0 while the right image shows how the cut obtained by the plane discards the right side of the belief space and a new queried belief β_1 is obtained.

Each policy π has a corresponding regret plane[4] $L(\pi, \beta)$ over β. Since $L(\pi^*(\beta), \beta)) \leq \max_{\beta \in \mathcal{B}} L(\pi^*(\beta), \beta))$, any $\beta : L(\pi^*(\beta'), \beta')) > L(\pi^*(\beta'), \beta))$ can not be the minimax β and can be discarded. This is the same as discarding the half-plane given by the descent direction of the Bayesian regret plane. An illustration of this principle in two dimensions can be found in Figure IV.2.

Selecting a new approximate centroid as the next β to query guarantees fast convergence in the volume of the plausible set of beliefs given the following lemma.

Lemma IV.5.2 (Lemma 5 [BV04])**.** *Each cut in Algorithm 13 will reduce the volume of the set K_t by at least 1/3 with high probability.*

The full procedure is described in Algorithm 13. Here β_t is the approximate centroid (through one of the methods in [BV04], such as hit-and-run sampling) of the set K_t. K_t contains the plausible beliefs that could be the minimax belief, at step t of the algorithm. The cut is given by C_t which is the normal to the Bayes regret plane at β_t where each element $C_t^{(i)} = R(\pi^*(\beta_t), \beta = \delta_{\mu_i})$.

Algorithm 13 Cutting plane algorithm for finding minimax belief

input: Initial belief set of constraints K_0, optimal policy oracle, policy evaluation oracle
for $t \in 0, \ldots, T-1$ **do**
  Obtain $\beta_t \approx \mathbb{E}_{K_t}[x]$
  Obtain optimal policy $\pi^*_{\beta_t}$ and $C_t^{(i)} = R(\pi^*(\beta_t), \beta = \delta_{\mu_i})$.
  $K_{t+1} = K_t \cap \{\beta : C_t^T(\beta - \beta_t) > 0\}$
return: $\beta^* \in K_T$ that has $\frac{\text{VOL}(K_T)}{\text{VOL}(K_0)} < \left(\frac{2}{3}\right)^T$ w.h.p. and corresponding $\pi^*(\beta^*)$.

This method is also applicable when the policy space is a set of ϵ-optimal policies $\Pi^\epsilon \subset \Pi$, i.e., such that $\max_{\pi \in \Pi^\epsilon} U(\pi, \beta) \geq \max_{\pi \in \Pi} U(\pi, \beta) - \epsilon$ for any $\beta \in \mathcal{B}$. It is natural to look at such a policy space, because policies obtained through look-ahead tree search or neural network may be adaptive, but they can only be ϵ-optimal in general.

[4] Due to the Bayesian regret being an expectation over MDPs and hence is linear.

Lemma IV.5.3. *If* $\max_{\pi\in\Pi^\epsilon} L(\pi,\beta) \le \max_{\pi\in\Pi} L(\pi,\beta) + \epsilon$ *for all* $\beta \in \mathcal{B}$ *then*

$$\min_{\pi\in\Pi} L(\pi,\beta^{\epsilon,*}) \ge \max_{\beta\in\mathcal{B}} \min_{\pi\in\Pi} L(\pi,\beta) - \epsilon \tag{IV.13}$$

where $\beta^{\epsilon,*} = \operatorname{argmax}_{\beta\in\mathcal{B}} \min_{\pi\in\Pi^\epsilon} L(\pi,\beta)$. *Additionally, if* $\min_{\pi\in\Pi} L(\pi,\beta)$ *is c-concave in* β *then* $||\beta^{\epsilon,*} - \beta^*||_2 < \sqrt{\epsilon/c}$.

A proof is provided in the appendix.

IV.6 Experiments

We perform three experiments to see how minimax priors differ from common uniform priors, and examine the relative robustness of the corresponding policies. The first characterises worst-case priors for Bernoulli bandits. The second experiment is on finite MDP sets with a finite horizon. Here we verify the feasibility of the cutting plane algorithm for finding minimax solutions. We also illustrate the regret of posterior sampling. The final experiment is for the general case of discrete MDPs and parametric adaptive policies, where a value may not exist.[5]

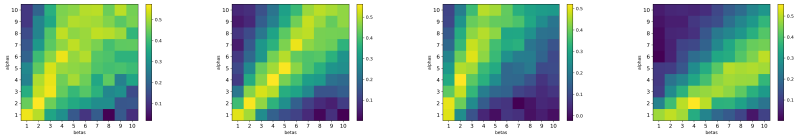
IV.6.1 Illustrations of Worst-Case Priors for Bernoulli Bandits

We are interested in analysing the worst-case priors when the Bayesian agent is responding to nature's prior with a Bayes-optimal policy. In general, computing the Bayes-optimal policy is intractable. However, for Bernoulli bandits with infinite horizon and geometrically discounted rewards, so that the utility is defined as $u = \sum_t \gamma^t r_t$, Gittins [GGW11; Git79] showed that an index policy, the so-called Gittins index, yields a Bayes-optimal policy.

For K-armed Bernoulli bandits $\theta = (\theta_1, \dots, \theta_K)$ with $\theta_k \in [0,1]$, we then consider Beta product priors such that $\beta(\theta) = \prod_{k=1}^K Beta(a_k, b_k)\{\theta_k\}$. To illustrate how the Bayes-expected regret of the Bayes-optimal policy changes with respect to the prior, we consider a two-armed Bernoulli bandit, where the first arm's prior is fixed to some distribution $Beta(a_1, b_1)$ and the second arm's prior $Beta(a_2, b_2)$ is set to different values. Figure IV.3 shows the Bayesian regret for different fixed priors for arm 1 and varying prior for arm 2.

We observe that high Bayesian regret is typically suffered when the second prior's mean approximately matches the mean of the first arm's prior, i.e. $\mathbb{E}[Beta(a_1, b_1)] = \mathbb{E}[Beta(a_2, b_2)]$. Moreover, it seems that maximal Bayesian regret is achieved at a completely symmetric prior, i.e. $Beta(a_1, b_1) = Beta(a_2, b_2)$, irrespective of how the first arm's prior is chosen. More generally, we can observe that lower values of a and b yield higher Bayesian regret, making the intuition precise that the Bayes-optimal policy suffers higher Bayesian regret when the prior provides less information. Based on this, a worst-case prior can

[5]The code is made available at https://github.com/minimaxBRL/minimax-bayes-rl.



(a) $Beta(1, 1)$ (b) $Beta(3, 3)$ (c) $Beta(4, 2)$ (d) $Beta(2, 4)$

Figure IV.3: The Bayesian regret of the Bayes-optimal policy in two-armed Bernoulli bandits, where the first arm's prior is fixed. The x- and y-axis denote the two parameters of the second arm's prior.

be conjectured to make arms maximally indistinguishable a priori; as one may expect.

We also allowed all priors to vary to discover the actual worst-case prior. We found this depends heavily on the discount factor γ and the number of arms K. For $K = 2$ and $\gamma = 0.9$, we found it is approximately $Beta(0.8, 0.8)$ for both arms. In general, the worst-case prior is symmetric with parameters increasing in the number of arms and the discount factor, i.e. moving towards short-tailed priors.

IV.6.2 Finite Set of MDPs

In this section, we study the properties of minimax problems where we have a belief over a finite set of MDPs. The transition matrix is randomly sampled from an exponential distribution before being normalised. The agent starts in state 1, and the reward is 1 for taking the first action in state N, and zero elsewhere. We use a finite horizon $T = 5$ to allow exact computation of the optimal policies and Bayesian regret. Additionally we use $\gamma = 1$.

Figure IV.4 show the Bayesian regret for a two-MDP task. This helps us visualise that the Bayes-optimal value is a piecewise linear function consisting of the minimum over locally optimal policies. We also compare with the Bayesian regret of the PSRL policy [Str00], which for every episode acts optimally with respect to a sampled MDP from the belief. The quadratic curve for PSRL is due to the fact that we allow the policy to change with the belief.

In additional experiments in Appendix D.3, we study the Bayesian regret landscape for a three MDP setup (see Figure .6). We also compare the worst case Bayesian regret of the minimax policy and of the Bayes optimal policy for the uniform belief for a few different setups with 16 different MDPs in Table .1 and can see that the minimax policy significantly outperforms the uniform best response policy.

IV.6.3 Infinite Set of MDPs

In the following experiments, we study priors over an infinite space of MDPs. The main prior of interest is Dirichlet product-priors. We use the minimax policy gradient algorithm to simultaneously update the parameters of the belief

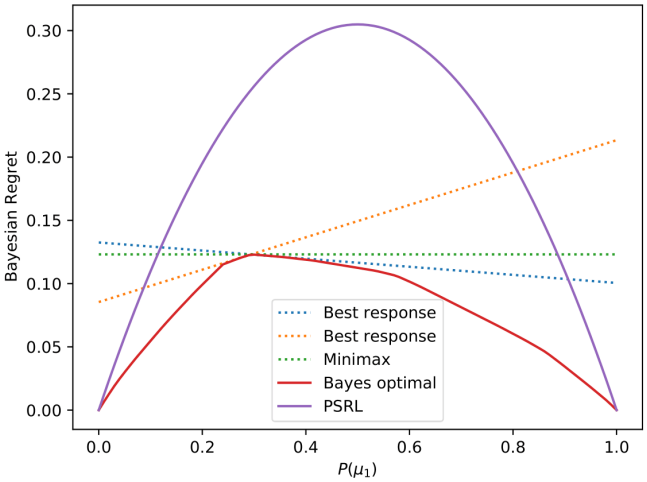


Figure IV.4: This figure shows the Bayesian regret of different policies. The dashed lines show the value of three adaptive policies optimal for the maximin-regret prior. Two of them are best responses, which are also optimal on either side of the maximin point. The minimax-regret policy is shown in green, and it has a uniform regret no matter what the actual prior is. The solid lines show policies which have knowledge of the MDP prior: the Bayes-optimal policy and the best PSRL policy for that specific prior. Their dependency on the prior makes their regret a concave function.

β and the parameters of the policy π. We choose a history-dependent policy parametrisation using a softmax rule. In these experiments we study MDPs with 5 states and two actions. Further, we consider problems with horizon $T = 1000$.

In Figure IV.5 we investigate the performance of the minimax policy π^* compared to the baseline *best response* adaptive policies, $\pi^*(\beta^1), \pi^*(\beta^*)$, to the uniform prior β^1 and the maximin prior β^*, respectively. The three policies are evaluated on six different priors. These are, the uniform prior β^1, the maximin prior β^*, two priors interpolated between the uniform and the maximin prior, a uniform prior over deterministic MDPs β^D and a delta distribution over the parameters of the Chain environment [Str00], β^{Chain}.

In this setting we can only expect to find approximate minimax solutions. Thus, there is no guarantee the obtained minimax solution is globally robust to changes in belief. However, in Figure IV.5 we observe the minimax policy π^* to be the most robust taking all priors into account.

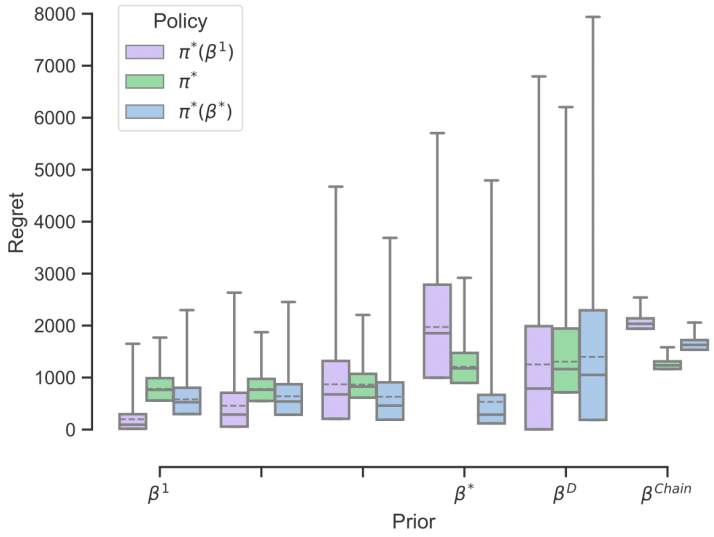


Figure IV.5: β^D is approximately uniform over deterministic MDPs. β^{Chain} is a delta distribution over the Chain MDP. The MDPs in between β^1 (Uniform) and β^* (Maximin) are interpolated. The mean is depicted with a dashed line, the solid line is the median and the upper whisker is the 99.9% percentile.

IV.7 Discussion and Conclusion

Related work We studied the problem of minimax-Bayes reinforcement learning. Although minimax-Bayes problems are well-known in statistical inference [c.f. Ber85], they have received little attention in sequential problems. Older work such as [ABG49] is interested in minimax and Bayes optimal solutions to decision making tasks but without combining them. Similarly, [HL52] relaxes the property of minimax risk to restricted Bayes solutions where the maximal risk is bounded while also changing the objective to an interpolation between the expected and maximal risk. While this is work in the same spirit as ours it is fundamentally different. [GD04] studied the problem of one-shot experiment design prior to estimation. In the partial monitoring setting, [LS19] made connections between the Bayesian minimax regret and the minimax regret.

There have been a variety of work interested in using meta learning to create Bayes-(adaptive) optimal agents such as [HYC01; Mik+20; Wan+16; Zin+21]. They use recurrent neural networks to encode an episode's history so as to adapt optimally in a new episode in a new MDP. As they are interested in optimising for specific MDP distribution, β is considered fixed and they solve $\max_\pi \mathbb{E}_{\mu\sim\beta} U(\mu, \pi)$ without studying β's impact on the utility or regret.

Work on Bayesian robust reinforcement learning [Der+20; PR19] is related in the manner that they search for policies that are robust against interference from nature. The difference is that they wish to find policies that are good

against the worst MDP from the set of MDPs that are plausible with respect to a specific posterior, rather than against an adversarial prior.

Conclusion In this work we study the computation of minimax-Bayes policies, which have not been previously considered. We also include conditions for when the solutions can be guaranteed to be found efficiently. Experimentally we find that these policies not only appear to be feasible, but also that such policies can be significantly more robust than those based on standard uninformative priors. Finally, we make exposition of many important properties of minimax-Bayes solutions for reinforcement learning to make a basis for future work in this area.

References

[ABG49] Arrow, K. J., Blackwell, D., and Girshick, M. A. "Bayes and minimax solutions of sequential decision problems". In: *Econometrica, Journal of the Econometric Society* (1949), pp. 213–244.

[AD14] Androulakis, E. G. and Dimitrakakis, C. "Generalised entropy MDPs and minimax regret". In: *arXiv preprint arXiv:1412.3276* (2014).

[Ber85] Berger, J. O. *Statistical decision theory and Bayesian analysis*. Springer, 1985.

[BV04] Bertsimas, D. and Vempala, S. "Solving Convex Programs by Random Walks". In: *J. ACM* vol. 51, no. 4 (July 2004), pp. 540–556.

[DeG70] DeGroot, M. H. *Optimal Statistical Decisions*. John Wiley & Sons, 1970.

[Der+20] Derman, E. et al. "A bayesian approach to robust reinforcement learning". In: *Uncertainty in Artificial Intelligence*. PMLR. 2020, pp. 648–658.

[DO18] Dimitrakakis, C. and Ortner, R. "Decision making under uncertainty and reinforcement learning". In: *Book available at http://www. cse. chalmers. se* (2018).

[Duf02] Duff, M. O. "Optimal Learning Computational Procedures for Bayes-adaptive Markov Decision Processes". PhD thesis. University of Massachusetts at Amherst, 2002.

[EL21] Eysenbach, B. and Levine, S. "Maximum entropy rl (provably) solves some robust rl problems". In: *arXiv preprint arXiv:2103.06257* (2021).

[GD04] Grünwald, P. D. and Dawid, A. P. "Game theory, Maximum Entropy, Minimum Discrepancy and Robust Bayesian decision Theory". In: *Annals of Statistics* (2004).

[GGW11] Gittins, J., Glazebrook, K., and Weber, R. *Multi-armed bandit allocation indices*. John Wiley & Sons, 2011.

[Git79] Gittins, J. C. "Bandit processes and dynamic allocation indices". In: *Journal of the Royal Statistical Society: Series B (Methodological)* vol. 41, no. 2 (1979), pp. 148–164.

[HL52] Hodges Jr, J. L. and Lehmann, E. L. "The use of previous experience in reaching statistical decisions". In: *The Annals of Mathematical Statistics* (1952), pp. 396–407.

[HYC01] Hochreiter, S., Younger, A. S., and Conwell, P. R. "Learning to learn using gradient descent". In: *International conference on artificial neural networks*. Springer. 2001, pp. 87–94.

[Haa+18] Haarnoja, T. et al. "Soft actor-critic algorithms and applications". In: *arXiv preprint arXiv:1812.05905* (2018).

[Lat21] Lattimore, T. Personal Communication. Mar. 2021.

[LJJ20] Lin, T., Jin, C., and Jordan, M. "On gradient descent ascent for nonconvex-concave minimax problems". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 6083–6093.

[LS19] Lattimore, T. and Szepesvári, C. "An information-theoretic approach to minimax regret in partial monitoring". In: *Conference on Learning Theory*. PMLR. 2019, pp. 2111–2139.

[Mik+20] Mikulik, V. et al. "Meta-trained agents implement bayes-optimal agents". In: *Advances in neural information processing systems* vol. 33 (2020), pp. 18691–18703.

[PR19] Petrik, M. and Russel, R. H. "Beyond confidence regions: Tight bayesian ambiguity sets for robust mdps". In: *Advances in neural information processing systems* vol. 32 (2019).

[Put14] Puterman, M. L. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[Str00] Strens, M. "A Bayesian framework for reinforcement learning". In: *ICML*. Vol. 2000. 2000, pp. 943–950.

[Tod06] Todorov, E. "Linearly-solvable Markov decision problems". In: *Advances in neural information processing systems* vol. 19 (2006).

[Wan+16] Wang, J. X. et al. "Learning to reinforcement learn". In: *arXiv preprint arXiv:1611.05763* (2016).

[Zie10] Ziebart, B. D. *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. Carnegie Mellon University, 2010.

[Zin+21] Zintgraf, L. et al. "VariBAD: variational Bayes-adaptive deep RL via meta-learning". In: *The Journal of Machine Learning Research* vol. 22, no. 1 (2021), pp. 13198–13236.

D.1 Gradient calculations

For solving the minimax problem either for the expected utility or the expected regret, we need to calculate the appropriate gradient for both the policy and the prior. The gradients for the expected utility are as follows:

$$\nabla_\pi U(\pi, \beta) = \int_{\mathcal{M}} d\beta(\mu) \nabla_\pi U(\pi, \mu), \qquad \nabla_\beta U(\pi, \beta) = \int_{\mathcal{M}} U(\pi, \mu) \nabla_\beta d\beta(\mu).$$

The Bayesian regret gradient is similarly obtained:

$$\nabla_\pi L(\pi, \beta) = -\int_{\mathcal{M}} d\beta(\mu) \nabla_\pi R(\pi, \mu) \qquad \nabla_\beta L(\pi, \beta) = \int_{\mathcal{M}} R(\pi, \mu) \nabla_\beta d\beta(\mu).$$

Since in the minimax regret scenario, the agent is minimising rather than maximising, the policy update is identical. However, the prior gradient is scaled with respect to the regret rather than the utility. Let us now look at how to calculate those gradients in more detail.

D.1.1 Policy gradient

Here we look at two classes of policies. The first occurs when there is a finite number of bases (possibly stochastic and behavioural) policies from which the agent chooses one randomly. The second is a class of parametrised stochastic behavioural policies.

Finite policy distributions. For a strategy $\sigma = (\sigma_1, \dots, \sigma_n)$ over a finite set of n policies $\Pi \subset \Pi^{\mathrm{S}}$, we can write

$$U(\sigma, \beta) = \sum_{\pi, \mu} \sigma(\pi) U(\pi, \mu) \beta(\mu).$$

We then obtain

$$\frac{\partial}{\partial \sigma_i} U(\sigma, \beta) = \sum_{\mu} U(\pi_i, \mu) \beta(\mu).$$

We do not use this setting in practice in the paper, but it is an interesting special case.

Stochastic policies. Stochastic policies π in a parametrised policy space $\Pi_W \subset \Pi^{\mathrm{S}}$ can be an arbitrary neural network policy. For a finite set of MDPs, the gradient is:

$$\nabla_\pi U(\pi, \beta) = \sum_{\mu} \nabla_\pi U(\pi, \mu) \beta(\mu).$$

For an infinite set of MDPs, we have

$$\nabla_\pi U(\pi, \beta) = \int_{\mathcal{M}} \nabla_\pi U(\pi, \mu) \, \mathrm{d}\beta(\mu) \approx \frac{1}{M} \sum_{k=1}^{M} \nabla_\pi U(\pi, \mu^{(k)}), \; \mu_k \sim \beta(\mu)$$

So it is only necessary to compute

$$\begin{aligned}\nabla_\pi U(\pi,\mu) &= \sum_h U(h)\mathbb{P}^\pi_\mu(h)\sum_t \nabla_\pi \ln \pi(a_t \mid h_t)\\ &= \sum_h U(h)\mathbb{P}^\pi_\mu(h)\sum_t \frac{\nabla_\pi \pi(a_t \mid h_t)}{\pi(a_t \mid h_t)},\end{aligned}$$

where for a given history $h = (s_1, r_1, a_1, \ldots, s_T, r_T)$, $h_t = (s_1, r_1, a_1, \ldots, s_t, r_t)$. It remains to compute $\nabla_\pi \pi(a_t \mid h_t)$, which can be done automatically using auto-grad software.

However, one particular case is when the policy is parametrised with $\mathbf{w}_a = (w_{a,i})_{i=1}^n$ vectors combined with a statistic $\phi : \mathcal{H} \to \mathbb{R}^n_+$ so that

$$\begin{aligned}\pi(a_t = a \mid h_t) &= \frac{\mathbf{w}_a^\top \phi(h_t)}{\sum_b \mathbf{w}_b^\top \phi(h_t)} = \frac{\sum_i w_{a,i}\phi_i(h_t)}{\sum_b \sum_i w_{b,i}\phi_i(h_t)}\\ \frac{\partial}{\partial w_{a,i}}\pi_{\mathbf{w}}(a_t = a \mid h_t) &= \frac{\phi_i(h_t)[\sum_{(b,j)\neq(a,i)} w_{b,j}\phi_j(h_t)]}{[\sum_b \sum_j w_{b,j}\phi_j(h_t)]^2}\\ \frac{\partial}{\partial w_{b,i}}\pi_{\mathbf{w}}(a_t = a \mid h_t) &= -\frac{\phi_i(h_t)\sum_j w_{a,j}\phi_j(h_t)}{[\sum_b \sum_j w_{b,j}\phi_j(h_t)]^2}.\end{aligned}$$

With a feature representation $\phi : \mathcal{H} \times A \to \mathbb{R}^n$ and a softmax policy then

$$\begin{aligned}\pi(a_t \mid h_t) &= \frac{e^{\mathbf{w}^\top \phi(h_t,a_t)}}{\sum_b e^{\mathbf{w}^\top \phi(h_t,b)}}\\ \nabla_{\mathbf{w}} \ln \pi(a_t \mid h_t) &= \phi(h_t, a_t) - \sum_{a\in A} \pi(a_t = a \mid h_t)\phi(h_t, a).\end{aligned}$$

For the case where $\phi(h_t, a)$ simply partitions the history, so that $\mathbf{w}^\top \phi(h, a) = w_{h,a}$, the above becomes

$$\frac{\partial}{\partial w_{h,a}} \ln \pi(a_t \mid h_t) = \begin{cases} 1 - \pi(a|h), & a_t = a, h_t = h\\ -\pi(a|h), & a_t \neq a, h_t = h\\ 0, & h_t \neq h\end{cases} \tag{14}$$

D.1.2 Prior gradient

The steps above were all standard policy gradient steps, which can be implemented with sampled MDPs from the current prior. However, we also need to update the prior distribution with a gradient step. Here we distinguish two cases: a belief over a finite number of MPDs and a Dirichlet belief.

Finite $\mathcal{M}$. Now let us represent the belief as a finite-dimensional vector $\beta = (\beta_i)$ on the simplex. The partial derivative is then:

$$\frac{\partial}{\partial \beta_i} U(\pi, \beta) = \sum_j U(\pi, \mu_j)\frac{\partial}{\partial \beta_i}\beta_j = U(\pi, \mu_j)$$

Dirichlet $\mathcal{M}$. Let us first consider the general case of an infinite MDP space. Then we can approximate the gradient of the expected utility through sampling:

$$\nabla_\beta U(\pi, \beta) = \int_{\mathcal{M}} U(\pi, \mu) \nabla_\beta \ln[\beta(\mu)] d\beta(\mu) \approx \frac{1}{M} \sum_{k=1}^{M} U(\pi, \mu^{(k)}) \nabla_\beta \ln[\beta(\mu^{(k)})],$$

where $\mu^{(k)} \sim \beta$ are samples from the current prior.

For discrete state-action MDPs for a certain number of states and actions, we can use a Dirichlet-product distribution. This means that for each state-action's (s, a) transition distribution, we define a separate Dirichlet distribution $\beta(\mu_{s,a})$ with parameter vector $\alpha_{s,a} \in \mathbb{R}_+^{|S|}$:

$$\beta(\mu) = \prod_{(s,a)} \beta(\mu_{s,a}), \qquad \beta(\mu_{s,a}) = \frac{1}{B(\alpha_{s,a})} \prod_i \mu_{s,a,i}^{\alpha_{s,a,i}-1}$$

where $\mu_{s,a,i} = \mathbb{P}(s_{t+1} = i | s_t = s, a_t = a)$. For the sequel, it is notationally convenient to ignore the s, a subscript and focus only on the next state distribution i

$$\begin{aligned}
\frac{\partial}{\partial \alpha_j} \ln \beta(\mu) &= \frac{\partial}{\partial \alpha_j} \ln \left\{ \frac{1}{B(\alpha)} \prod_i \mu_i^{\alpha_i - 1} \right\} \\
&= \frac{\partial}{\partial \alpha_j} \left\{ \ln \frac{1}{B(\alpha)} + \sum_i (\alpha_i - 1) \ln \mu_i \right\} \\
&= \frac{\partial}{\partial \alpha_j} \ln \frac{1}{B(\alpha)} + \ln \mu_j
\end{aligned}$$

Note that

$$\begin{aligned}
\ln 1/B(\alpha) &= \ln \frac{\Gamma(\sum_i \alpha_i)}{\prod_i \Gamma(\alpha_i)} \\
&= \ln \Gamma(\sum_i \alpha_i) - \sum_i \log \Gamma(\alpha_i)
\end{aligned}$$

So that

$$\begin{aligned}
\frac{\partial}{\partial \alpha_j} \ln 1/B(\alpha) &= \frac{\partial}{\partial \alpha_j} \ln \Gamma(\sum_i \alpha_i) - \frac{\partial}{\partial \alpha_j} \ln \Gamma(\alpha_j) \\
&= \frac{1}{\Gamma(\sum_i \alpha_i)} \frac{\partial}{\partial \alpha_j} \Gamma(\sum_i \alpha_i) - \frac{1}{\Gamma(\alpha_j)} \frac{\partial}{\partial \alpha_j} \Gamma(\alpha_j) \\
&= \psi(\sum_i \alpha_i) - \psi(\alpha_j)
\end{aligned}$$

where ψ is the digamma function.

This means that the overall derivative is

$$\begin{aligned}\frac{\partial}{\partial \mu_{s,a,i}} \ln \beta(\mu) &= \frac{\partial}{\partial \mu_{s,a,i}} \ln \prod_{(s',a')} \beta(\mu_{s',a'}) \\ &= \frac{\partial}{\partial \mu_{s,a,i}} \sum_{s',a'} \ln \beta(\mu_{s',a'}) \\ &= \frac{\partial}{\partial \mu_{s,a,i}} \ln \beta(\mu_{s,a}) \\ &= \psi(\sum_j \alpha_{s,a,j}) - \psi(\alpha_{s,a,i}) + \ln(\mu_{s,a,i})\end{aligned}$$

Combining the above, we get

$$\alpha_{s,a,i}^{(k)} = \alpha_{s,a,i}^{(k-1)} - \delta^{(k)} U(\pi, \mu^{(k)}) \left[\psi(\sum_j \alpha_{s,a,j}) - \psi(\alpha_{s,a,i}) + \ln(\mu_{s,a,i}^{(k)}) \right],$$

where $\delta^{(k)}$ is the step-size.

Reward prior. We can derive a similar update for Beta-distributed rewards, with

$$\begin{aligned}\alpha_s^{(k)} &= \alpha_s^{(k-1)} - \delta^{(k)} U(\pi, \mu^{(k)}) \left[\psi(\alpha_s + \beta_s) - \psi(\alpha_s) + \ln(\rho_s^{(k)}) \right] \\ \beta_s^{(k)} &= \beta_s^{(k-1)} - \delta^{(k)} U(\pi, \mu^{(k)}) \left[\psi(\alpha_s + \beta_s) - \psi(\beta_s) + \ln(1 - \rho_s^{(k)}) \right].\end{aligned}$$

We can also define the Beta-distribution with alternate parametrisation: $p_s = \alpha_s/(\alpha_s + \beta_s)$, $n_s = \alpha_s + \beta_s$ which implies $\alpha_s = p_s n_s, \beta_s = n_s(1 - p_s)$. We then obtain

$$\begin{aligned}\frac{\partial}{\partial p_s} \ln \beta(\mu) &= n_s \frac{\partial}{\partial \alpha_s} \ln \beta(\mu) - n_s \frac{\partial}{\partial \beta_s} \ln \beta(\mu) \\ &= n_s \left[-\psi(\alpha_s) + \psi(\beta_s) + \ln(\rho_s^{(k)}) - \ln(1 - \rho_s^{(k)}) \right] \\ &= n_s \left[-\psi(\alpha_s) + \psi(\beta_s) + \ln \left(\frac{\rho_s^{(k)}}{1 - \rho_s^{(k)}} \right) \right]\end{aligned}$$

$$\begin{aligned}\frac{\partial}{\partial n_s} \ln \beta(\mu) &= p \frac{\partial}{\partial \alpha_s} \ln \beta(\mu) + (1 - p) \frac{\partial}{\partial \beta_s} \ln \beta(\mu) \\ &= p \left[-\psi(\alpha_s) + \psi(\beta_s) + \ln(\rho_s^{(k)}) - \ln(1 - \rho_s^{(k)}) \right] \\ &\quad + \left[\psi(\alpha_s + \beta_s) - \psi(\beta_s) + \ln(1 - \rho_s^{(k)}) \right] \\ &= p \left[-\psi(\alpha_s) + \psi(\beta_s) + \ln \left(\frac{\rho_s^{(k)}}{1 - \rho_s^{(k)}} \right) \right] \\ &\quad + \left[\psi(\alpha_s + \beta_s) - \psi(\beta_s) + \ln(1 - \rho_s^{(k)}) \right]\end{aligned}$$

D.2 Omitted proofs

Proof of Lemma IV.3.6. For any β

$$\begin{aligned}\max_{\mu\in\mathcal{M}} R(\pi,\mu) &\geq \max_{\mu\in\text{supp}(\beta)} R(\pi,\mu)\\ &= \max_{\mu\in\text{supp}(\beta)} U(\pi^*(\mu),\mu) - U(\pi,\mu)\\ &\geq \max_{\mu\in\text{supp}(\beta)} U(\pi^*(\beta),\mu) - U(\pi,\mu)\\ &\geq \sum_{\mu\in\text{supp}(\beta)} \beta(\mu)[U(\pi^*(\beta),\mu) - U(\pi,\mu)]\\ &= U(\pi^*(\beta),\beta) - U(\pi,\beta) = R(\pi,\beta).\end{aligned}$$

Since the above holds for any β, $\max_\mu R(\pi,\mu) \geq \max_\beta R(\pi,\beta)$. Letting $\delta(\mathcal{M})$ denote the degenerate distributions on individual members of $\mathcal{M}$, we have:

$$\max_\beta R(\pi,\beta) \geq \max_{\beta\in\delta(\mathcal{M})} R(\pi,\mu) = \max_{\mu\in\mathcal{M}} R(\pi,\mu)$$

■

Proof of Lemma IV.5.1. Let $\pi,\pi',\pi'' \in \Pi$. To verify that $L(\pi,\beta)$ is l-smooth we study if

$$||\nabla L(\pi,\beta) - \nabla L(\pi',\beta')|| \leq l||(\pi,\beta) - (\pi',\beta')||.$$

$$\begin{aligned}&||\nabla L(\pi'',\beta'') - \nabla L(\pi',\beta')||_2^2\\ \leq&||(\pi'',\beta'') - (\pi',\beta')||_2^2(\sup_{\pi,\beta}||\nabla^2 L(\pi,\beta)||_2^2)\\ =&||(\pi'',\beta'') - (\pi',\beta')||_2^2(\sup_{\pi,\beta}||\nabla_\pi^2 L(\pi,\beta)||_2^2)\\ \leq&||(\pi'',\beta'') - (\pi',\beta')||_2^2(\sup_{\pi,\beta}||\nabla_\pi^2 L(\pi,\beta)||_F^2)\end{aligned}$$

Here the second transformation is due to the fact that any derivative with respect to β is constant, and therefore the second order derivatives are zero except for ∇_π^2. $||.||_F$ denotes the Frobenius norm.

For stochastic policies π in a parametrised policy space $\Pi_W \subset \Pi^{\text{S}}$, we can write (cf. [DO18]):

$$\nabla_\pi L(\pi,\beta) = \nabla_\pi U(\pi,\beta) = \sum_\beta \nabla_\pi U(\pi,\mu)\beta(\mu).$$

Similarly, we obtain, for the Hessian:

$$\nabla_\pi^2 L(\pi,\beta) = \nabla_\pi^2 U(\pi,\beta) = \sum_\beta \nabla_\pi^2 U(\pi,\mu)\beta(\mu).$$

So it is only necessary to compute

$$\begin{aligned}
\nabla_\pi^2 U(\pi,\mu) &= \sum_h U(h)\nabla_\pi(\mathbb{P}_\mu^\pi(h)\sum_t \nabla_\pi \ln \pi(a_t \mid h_t)) \\
&= \sum_h U(h)(\nabla_\pi(\mathbb{P}_\mu^\pi(h))\sum_t \nabla_\pi \ln \pi(a_t \mid h_t)) \\
&\quad + \mathbb{P}_\mu^\pi(h)\sum_t \nabla_\pi^2 \ln \pi(a_t \mid h_t)) \\
&= \sum_h U(h)(\mathbb{P}_\mu^\pi(h)\sum_t \nabla_\pi \ln \pi(a_t \mid h_t)\sum_t \nabla_\pi \ln \pi(a_t \mid h_t)^T \\
&\quad + \mathbb{P}_\mu^\pi(h)\sum_t \nabla_\pi^2 \ln \pi(a_t \mid h_t)) \\
&= \sum_h U(h)(\mathbb{P}_\mu^\pi(h)\sum_t \nabla_\pi \ln \pi(a_t \mid h_t)\nabla_\pi \ln \pi(a_t \mid h_t)^T \\
&\quad + \mathbb{P}_\mu^\pi(h)\sum_t \nabla_\pi^2 \ln \pi(a_t \mid h_t))
\end{aligned}$$

where for a given history $h = (s_1, r_1, a_1, \ldots, s_T, r_T)$, $h_t = (s_1, r_1, a_1, \ldots, s_t, r_t)$.

From the setting of a softmax policy and a partitioned history in Eq (14).

$$\frac{\partial}{\partial w_{h,a}} \ln \pi(a_t \mid h_t) = \begin{cases} 1 - \pi(a|h), & a_t = a, h_t = h \\ -\pi(a|h), & a_t \neq a, h_t = h \\ 0, & h_t \neq h \end{cases} \tag{15}$$

$$\frac{\partial\partial}{\partial w_{h,a}\partial w_{h,a'}} \ln \pi(a_t \mid h_t) = \begin{cases} \pi(a|h)(\pi(a|h) - 1), & a = a', h_t = h \\ \pi(a|h)\pi(a'|h), & a \neq a', h_t = h \\ 0, & h_t \neq h. \end{cases} \tag{16}$$

We then get Let $\nabla_\pi^2 U(\pi,\mu) = G_1 + G_2$ where

$$\begin{aligned}
G_1 &= \sum_h U(h)\mathbb{P}_\mu^\pi(h)\sum_t \nabla_\pi \ln \pi(a_t \mid h_t)\nabla_\pi \ln \pi(a_t \mid h_t)^T \\
G_2 &= \sum_h U(h)\mathbb{P}_\mu^\pi(h)\sum_t \nabla_\pi^2 \ln \pi(a_t \mid h_t).
\end{aligned}$$

$$||G_1||_F = ||\sum_h U(h)\mathbb{P}_\mu^\pi(h)\sum_t \nabla_\pi \ln \pi(a_t \mid h_t)\nabla_\pi \ln \pi(a_t \mid h_t)^T||_F \tag{17}$$

$$\leq \max_h |U(h)|||\sum_h \mathbb{P}_\mu^\pi(h)\sum_t \nabla_\pi \ln \pi(a_t \mid h_t)\nabla_\pi \ln \pi(a_t \mid h_t)^T||_F \tag{18}$$

$$\leq T||\sum_h \mathbb{P}_\mu^\pi(h)\sum_t \nabla_\pi \ln \pi(a_t \mid h_t)\nabla_\pi \ln \pi(a_t \mid h_t)^T||_F \tag{19}$$

$$= T\sqrt{\sum_{h_t}\sum_{a\in\mathcal{A}}\sum_{a'\in\mathcal{A}}\left(\mathbb{P}_\mu^\pi(h_t)T\frac{\partial \ln \pi(a_t \mid h_t)}{\partial \omega_{h_t,a}}\frac{\partial \ln \pi(a_t \mid h_t)}{\partial \omega_{h_t,a'}}\right)^2} \tag{20}$$

$$\leq T\sqrt{\sum_{h_t} T^2 \mathbb{P}_\mu^\pi(h_t)^2 \sum_{a\in\mathcal{A}}\sum_{a'\in\mathcal{A}} 1^2} \tag{21}$$

$$\leq T\sqrt{T^2 \sum_{h_t} \mathbb{P}_\mu^\pi(h_t) \sum_{a\in\mathcal{A}}\sum_{a'\in\mathcal{A}} 1} \tag{22}$$

$$\leq T\sqrt{T^2|\mathcal{A}|^2} \tag{23}$$

$$\leq |\mathcal{A}|T^2 \tag{24}$$

Here equation (20) comes from the definition of the Frobenius norm and the fact that every element (h_t, a, a') in the matrix corresponds to $\sum_h \mathbb{I}_{h_t\in h}\mathbb{P}_\mu^\pi(h)\frac{\partial \ln \pi(a_t|h_t)}{\partial \omega_{h_t,a}}\frac{\partial \ln \pi(a_t|h_t)}{\partial \omega_{h_t,a'}}$ and that $\mathbb{P}_\mu^\pi(h_t) = \sum_h \mathbb{P}_\mu^\pi(h_t|h)\mathbb{P}_\mu^\pi(h) = \sum_h \mathbb{I}_{h_t\in h} 1/T\mathbb{P}_\mu^\pi(h)$. Equation (21) follows from the absolute value of equation (15) being bounded by one.

$$\begin{aligned}
||G_2||_F &= ||\sum_h U(h)\mathbb{P}_\mu^\pi(h)\sum_t \nabla_\pi^2 \ln \pi(a_t \mid h_t)||_F \\
&\leq T||\sum_h \mathbb{P}_\mu^\pi(h)\sum_t \nabla_\pi^2 \ln \pi(a_t \mid h_t)||_F \\
&\leq T||\sum_{h_t} T\mathbb{P}_\mu^\pi(h_t)\nabla_\pi^2 \ln \pi(a_t \mid h_t)||_F \\
&\leq T\sqrt{\sum_{h_t} T^2 \mathbb{P}_\mu^\pi(h_t)^2 1} \\
&\leq T\sqrt{T^2 \sum_{h_t} \mathbb{P}_\mu^\pi(h_t) 1} \\
&\leq T^2
\end{aligned}$$

Similarly to the case for G_1, the steps follow the definition of the Frobenius norm, the observation that each element is weighted by $\mathbb{P}_\mu^\pi(h_t)T$, and that the absolute value of the partial derivatives is bounded by 1.

Finally this yields

$$l \leq ||\nabla_\pi^2 U(\pi,\mu)||_F \leq ||G_1||_F + ||G_2||_F \leq T^2(|\mathcal{A}|+1).$$

$L(.,\beta)$ is $\mathcal{L}$-Lipschitz if $||\nabla_\pi U(\pi,\mu)||_2 \leq \mathcal{L}$.

$$\begin{aligned}
||\nabla_\pi U(\pi,\mu)||_2 &= ||\sum_h U(h)\mathbb{P}_\mu^\pi(h)\sum_t \nabla_\pi \ln \pi(a_t \mid h_t)||_2 \\
&\leq ||\sum_h U(h)\mathbb{P}_\mu^\pi(h)\sum_t \nabla_\pi \ln \pi(a_t \mid h_t)||_F \\
&\leq T\sqrt{T^2 \sum_{h_t} \mathbb{P}_\mu^\pi(h_t)^2 1^2}
\end{aligned}$$

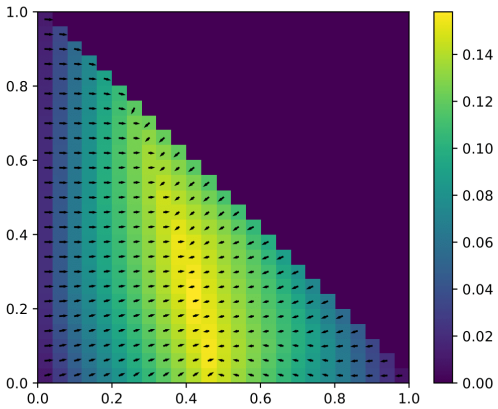


Figure .6: Visualisation of Bayesian regret for three finite-horizon MDPs. The arrows show the gradients of the Bayesian regret for the corresponding Bayes-optimal policy. The axes represent the belief of two of the MDPs while the belief of the final MDP is given by 1-x-y.

$$\leq \max_h(|U(h)|)T.$$

This then gives $\mathcal{L} \leq T^2$.

■

Proof of Lemma IV.5.3. Firstly,

$$\min_{\pi\in\Pi} L(\pi, \beta^{\epsilon,*}) \geq \min_{\pi\in\Pi^\epsilon} L(\pi, \beta^{\epsilon,*}) - \epsilon \geq \min_{\pi\in\Pi^\epsilon} L(\pi, \beta^*) - \epsilon \geq \min_{\pi\in\Pi} L(\pi, \beta^*) - \epsilon$$

which completes the first part of the proof.

Secondly from the definition of c-convexity, and the fact that $\nabla_\beta \min_{\pi\in\Pi} L(\pi, \beta^*)^T(\beta - \beta^*)$ must be zero since the gradient must be zero in any direction that does not move out of $\mathcal{B}$, we have

$$\min_{\pi\in\Pi} L(\pi, \beta) \leq \min_{\pi\in\Pi} L(\pi, \beta^*) - c||\beta^* - \beta||_2^2.$$

Rearranging and setting $\beta = \beta^{\epsilon,*}$ finishes the proof. ■

D.3 Additional results for finite MDPs

In this section we generate MDPs as in the same way as in Section IV.6.2, with the difference that Table .1 uses $\gamma = 0.9$.

Figure .6 gives an example of what the Bayesian regret landscape looks like for a task with three MDPs. The change in Bayesian regret for the fixed optimal policy of a certain belief is visualised with arrows.

Table .1: Comparison of worst-case Bayesian regret for optimal policies at minimax and uniform belief for 16 MDP tasks.

Seed	1	2	3	4	5
Minimax	0.247	0.314	0.348	0.342	0.363
Uniform	0.640	0.554	0.484	0.646	0.850

In Table .1 we have some additional results comparing the performance of the uniform-prior and worst-case prior policies. In particular, we generate 5 sets of 16 MDPs. For each set, we calculate the minimax policy and the best response to the uniform prior. We then calculate the worst-case Bayesian regret for each policy. As we can expect, the minimax policy significantly outperforms the uniform best response policy.

Paper V

Bandits Meet Mechanism Design to Combat Clickbait in Online Recommendation

Thomas Kleine Buening, Aadirupa Saha, Christos Dimitrakakis, Haifeng Xu

To appear at the *International Conference on Learning Representations*, 2024.

Abstract

We study a strategic variant of the multi-armed bandit problem, which we coin the strategic click-bandit. This model is motivated by applications in online recommendation where the choice of recommended items depends on both the click-through rates and the post-click rewards. Like in classical bandits, rewards follow a fixed unknown distribution. However, we assume that the click-rate of each arm is chosen strategically by the arm (e.g., a host on Airbnb) in order to maximize the number of times it gets clicked. The algorithm designer does not know the post-click rewards nor the arms' actions (i.e., strategically chosen click-rates) in advance, and must learn both values over time. To solve this problem, we design an incentive-aware learning algorithm, UCB-S, which achieves two goals simultaneously: (a) incentivizing desirable arm behavior under uncertainty; (b) minimizing regret by learning unknown parameters. We characterize all approximate Nash equilibria among arms under UCB-S and show a $\tilde{O}(\sqrt{KT})$ regret bound uniformly in every equilibrium. We also show that incentive-unaware algorithms generally fail to achieve low regret in the strategic click-bandit. Finally, we support our theoretical results by simulations of strategic arm behavior which confirm the effectiveness and robustness of our proposed incentive design.

V.1 Introduction

Recommendation platforms act as intermediaries between *vendors* and *users* so as to recommend *items* from the former to the latter. On Amazon, vendors sell physical items, while on Youtube the recommended items are videos. The recommendation problem is how to select one or more items to present to each user so that they are most likely to click on at least one of them.

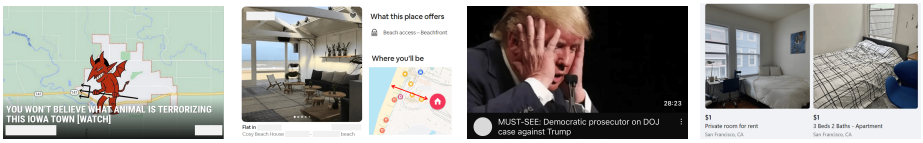


Figure V.1: Examples of unrepresentative or clickbait headlines and thumbnails on Bing News, Airbnb, Youtube, and Facebook Marketplace (identifying information partly redacted).

However, vendor-chosen *item descriptions* are an essential aspect of the problem that is often ignored. These invite vendors to exaggerate their true value in the descriptions in order to increase their Click-Through-Rates (CTRs). As a consequence, even though online learning algorithms can generally identify relevant items, the existence of unrepresentative or exaggerated item descriptions remains a challenge [HBR12; YPR10]. These include thumbnails or headlines that do not truly reflect the underlying item (see Figure V.1)—a well-known internet phenomenon called the *clickbait* [Wan+21]. While moderately increasing user click-rates through attractive descriptions is often encouraged since it helps to increase the overall user activity, clickbait can be harmful to a platform as it leads to bad recommendation outcomes and damage to the platform's reputation which may exceed the value of any additional clicks. A key reason for such dishonest or exaggerated item deceptions is the *strategic behavior* of vendors driven by their incentive to increase their item's exposure and click probability. Thus naturally, vendors are better off carefully choosing descriptions so as to increase click-rates, which leads to phenomena such as clickbait.[1]

To address this issue, we take an approach that marries *mechanism design* without payments with *online learning*, which are two celebrated research areas, however, mostly studied as separate streams. Since clickbait is fundamentally driven by vendor incentives, we believe that the novel design of online learning policies *that can carefully align vendor incentives with the platform's overall objective* may help to resolve this issue from its root.

To incorporate vendor-chosen item descriptions in this setting, we propose and study a natural strategic variant of the classical Multi-Armed Bandit (MAB) problem, which we call the *strategic click-bandit* in order to emphasize the strategic role that clicks and CTRs play in our setup.[2] Concretely, in strategic click-bandits, each arm i is characterized by (a) a reward distribution with mean μ_i, inherent to the arm; and (b) a click probability $s_i \in [0, 1]$, chosen freely by the arm at the beginning. Since the learner (i.e., the recommendation system) knows neither of these values in advance, it must learn them through interaction. The learner's objective is represented through a general utility function $u(s_i, \mu_i)$ that depends on both click-rate and post-click rewards.

[1] This is possible because most platforms rely on vendors to provide descriptions about their items. For instance, the images of restaurants on Yelp, rentals on Airbnb, hotels on Expedia, title and thumbnails of Youtube videos, and descriptions of products on Amazon are all provided by the vendors.

[2] We use the terms click-through-rate, click-rate, and click probability interchangeably.

We highlight two fundamental differences between strategic click-bandits and standard MABs. First, each arm in the strategic click-bandit is a *self-interested agent* whose objective is to maximize the number of times it gets clicked. This captures the strategic behavior of many vendors in online recommendations, especially those who are rewarded based on user clicks (e.g., [You23]). Second, s_i is a freely chosen *action* by arm i, rather than a fixed parameter of arm i. We believe these modeling adjustments more realistically capture vendor behaviors in real applications. They also lead to intriguing mechanism design questions since the bandit algorithm not only needs to learn the unknown parameters, but also has to carefully align incentives to avoid undesired arm behavior. In summary, our contributions are:

1. We introduce the strategic click-bandit problem, which involves strategic arms manipulating click-rates so as to maximize their own utility, and show that *incentive-unaware* algorithms generally fail to achieve low regret in the strategic click-bandit (Section V.3, Proposition V.4.1).

2. We design an *incentive-aware* learning algorithm, UCB-S, that combines mechanism design and online learning techniques and effectively incentivizes desirable arm strategies while minimizing regret by making credible and justified threats to arms under uncertainty (Section V.5).

3. We characterize the set of Nash equilibria for the arms under the UCB-S mechanism and show that every arm i's strategy is $\tilde{\mathcal{O}}\big(\max\{\Delta_i, \sqrt{K/T}\}\big)$ close to the desired strategy in equilibrium (Theorem V.5.2). We then show that UCB-S achieves $\tilde{\mathcal{O}}\big(\sqrt{KT}\big)$ strong strategic regret (Theorem V.5.3) and complement this with an almost matching lower bound of $\Omega\big(\sqrt{KT}\big)$ for weak strategic regret (Theorem V.5.5).

4. We simulate strategic arm behavior through repeated interaction and gradient ascent and empirically demonstrate the effectiveness of the proposed UCB-S mechanism (Section V.6).

V.2 Related Work

The MAB problem is a well-studied online learning framework, which can be used to model decision-making under uncertainty [Aue02; LR85]. Since it inherently involves sequential actions and the exploration-exploitation trade-off, the MAB framework has been applied to online recommendations [Li+10; WWW17; Zon+16] as well as a myriad of other domains [BRA20]. While there is much work studying strategic machine learning [e.g., Fre+20; Har+16; ZC21], we here wish to highlight related work that connects online learning (and specifically the MAB formalism) to mechanism design [NR99]. Additional related work is discussed in Appendix E.8.

To the best of our knowledge, [Bra+19] are the first to study a strategic variant of the MAB problem. In their model, when an arm is pulled, it receives a privately observed reward ν and chooses to pass on a portion x of it to the

principal, keeping $\nu - x$ for itself. The goal of the principal is then to incentivize arms to share as much reward with the principal as possible. In contrast to our work, the principal must not learn the underlying reward distribution or the arm strategies, but instead design an auction among arms based on the shared rewards. [FPX20] and [Don+22] study the robustness of bandit algorithms to strategic reward manipulations. However, neither work attempts to align incentives by designing mechanisms, but instead assume a limited manipulation budget. [SLO22] study MABs with strategic replication in which agents can submit several arms with replicas to the platform. They design an algorithm, which separately explores the arms submitted by each agent and in doing so discourages agents from creating additional arms and replicas. Another line of work studies auction-design in MAB formalisms, often motivated by applications in ad auctions [BKS15; BSS09; DK09]. In these models, in every round the auctioneer selects one advertiser's item, which is subsequently clicked or not, and the goal of the auctioneer is to incentivize advertisers to truthfully bid their value-per-click by constructing selection and payment rules.

To the best of our knowledge, our work is the first to study the situation where the arms' strategies (as well as other parameters) are initially unobserved, and must be learned from interaction while simultaneously incentivizing arms under uncertainty without payments. As a result, while other work is usually able to precisely incentivize certain arm strategies, our mechanism design and characterization of the Nash equilibria are *approximate*.

V.3 The Strategic Click-Bandit Problem

We consider a natural strategic variant of the classical MAB, motivated by applications in online recommendation. Unlike classical MABs, strategic click-bandits feature decentralized interactions with the learner and multiple self-interested arms.

Let $[K] := \{1, \dots, K\}$ denote the set of arms, each being viewed as a strategic *agent*. The strategic click-bandit proceeds in two phases. In the first phase, the learner commits to an online learning policy M, upon which each arm i chooses a description, which results in a corresponding click-rate $s_i \in [0, 1]$. The second phase proceeds in rounds. At each round t: (1) the algorithm M pulls/recommends an arm i_t based on observed past data; (2) arm i_t is clicked with probability s_{i_t}; (3) if i_t is clicked, arm i_t receives utility 1 (whereas all other arms i receive utility 0) and the learner observes a post-click reward $r_{t,i_t} \in [0, 1]$ drawn from i_t's reward distribution with mean $\mu_{i_t} \in [0, 1]$. If i_t is *not* clicked, all arms receive 0 utility and the learner does not observe any post-click rewards. The post-click mean μ_i is fixed for each arm i and captures the *true value* of the arm. From the learner's perspective, *both* s_i and μ_i of each arm are unknown but can be learned from online bandit feedback, that is, whether the recommended arm is clicked and, if so, what its realized reward is. In the following, we will also refer to the online learning policy M as a *mechanism* to emphasize its dual role in learning and incentive design. We summarize the interaction in Model 14.

Model 14 The Strategic Click-Bandit Problem

Learner commits to algorithm M, which is shared with all arms
Arms choose strategies $(s_1, \dots, s_K) \in [0,1]^K$ (unknown to M)
for $t = 1, \dots, T$ **do**
  Algorithm M selects arm $i_t \in [K]$
  Arm i_t is clicked with probability s_{i_t}, i.e., $c_{t,i_t} \sim \text{Bern}(s_{i_t})$
  if i_t was clicked ($c_{t,i_t} = 1$) **then**
    Arm i_t receives utility 1 from the click
    M observes post-click reward r_{t,i_t} drawn from a distr. with mean μ_{i_t}

V.3.1 Learner's Utility

The learner's utility of selecting an arm i with CTR s_i and post-click value μ_i is denoted $u(s_i, \mu_i)$. One example of this utility function is $u(s, \mu) = s\mu$. In this case, the learner monotonically prefers large s and does not care about how much the click-rate s differs from the post-click value μ. However, we believe that the learner (e.g., a platform like Youtube or Airbnb) usually values consistency between the click-rates and the post-click values of arms. This could be captured by a penalty term for how much s_i differs from μ_i; for instance, a natural choice is $u(s, \mu) = s\mu - \lambda(s - \mu)^2$ for some weight $\lambda > 0$. Such *non-monotonicity* of the learner's utility $u(s_i, \mu_i)$ in s_i *versus* arm i's monotonic preference of larger click-rates forms the fundamental tension in the strategic click-bandit model and is also the reason that mechanism design is needed. We keep the above utility functions in mind as running examples, but derive our results for a much more general class of functions satisfying the following mild regularity assumptions:

(A1) $u\colon [0,1] \times [0,1] \to \mathbb{R}$ is L-Lipschitz w.r.t. the ℓ_1-norm.

(A2) $u^*(\mu) := \max_{s \in [0,1]} u(s, \mu)$ is monotonically increasing.

(A3) $s^*(\mu) := \operatorname{argmax}_{s \in [0,1]} u(s, \mu)$ is H-Lipschitz and is bounded away from zero.

Assumption (A1) bounds the loss of selecting a suboptimal arm. (A2) states that, in the (idealized) situation when the arms choose click-rates so as to maximize the learner's utility u, then arms with larger post-click rewards μ are always preferred. (A3) then ensures that from the perspective of the learner most desired strategy $s^*(\mu)$ does not change abruptly w.r.t. μ and the learner wishes to incentivize non-zero click-rates. In what follows, the function $s^*(\mu)$ will play a central role as it describes the arm strategy that maximizes the learner's utility. For instance, in the case of $u(s, \mu) = s\mu - \lambda(s - \mu)^2$ it is given by $s^*(\mu) = (1 + \frac{1}{2\lambda})\mu$. As such, the learner will typically try to incentivize an arm with post-click reward μ_i to choose strategy $s^*(\mu_i)$.

V.3.2 Arms' Utility and Nash Equilibria Among Arms

The mean post-click reward μ_i of each arm i is fixed, whereas arm i can freely choose the CTR s_i. In the strategic click-bandit, the objective of each arm i is to maximize the number of times it gets clicked $\sum_{t=1}^{T} \mathbb{1}_{\{i_t=i\}} c_{t,i}$, which captures the objectives of vendors on internet platforms for whom user traffic typically proportionally converts to revenue.[3] We now introduce the solution concept for the game among arms defined by a mechanism M and post-click rewards $\mu_1, \dots, \mu_K$, often referred to as an *equilibrium*. Let s_{-i} denote the $K-1$ strategies of all arms except i. Each arm i chooses s_i to maximize their *expected* number of clicks $v_i(M, s_i, s_{-i})$, which is a function of the mechanism M, their own action s_i as well as all other arms' actions s_{-i}. Concretely,

$$v_i(M, s_i, s_{-i}) := \mathbb{E}_M \left[\sum_{t=1}^{T} \mathbb{1}_{\{i_t=i\}} c_{t,i} \right] \tag{V.1}$$

where the expectation is taken over the mechanism's decisions and the environment's randomness. We generally write $\boldsymbol{s} := (s_1, \dots, s_K)$ to summarize a strategy profile of the arms. Let Σ denote the set of probability measures over $[0, 1]$. Given a *mixed* strategy profile $\boldsymbol{\sigma} = (\sigma_i, \sigma_{-i}) \in \Sigma^K$, i.e., a distribution over $[0, 1]^K$, arm i's utility is then defined as $v_i(M, \sigma_i, \sigma_{-i}) := \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\sigma}}[v_i(M, s_i, s_{-i})]$.

Definition V.3.1 (Nash Equilibrium)**.** We say that $\boldsymbol{\sigma} = (\sigma_1, \dots, \sigma_K) \in \Sigma^K$ is a Nash equilibrium (NE) under mechanism M if $v_i(M, \sigma_i, \sigma_{-i}) \geq v_i(M, \sigma_i', \sigma_{-i})$ for all $i \in [K]$ and strategies $\sigma_i' \in \Sigma$.

In other words, $\boldsymbol{\sigma}$ is in NE if no arm can increase its utility by *unilaterally* deviating to some other strategy. If some NE $\boldsymbol{\sigma} \in \Sigma^K$ has weight one on a pure strategy profile $\boldsymbol{s} \in [0, 1]^K$, this equilibrium is said to be in pure-strategies. Let $\mathrm{NE}(M) := \{\boldsymbol{\sigma} \in \Sigma^K : \boldsymbol{\sigma} \text{ is a NE under } M\}$ denote the set of all (possibly mixed) NE under mechanism M. Following conventions in standard economic analysis, we assume that the arms will form a NE in $\mathrm{NE}(M)$ in response to an algorithm M.[4]

Remark V.3.2 (Existence of Nash Equilibrium). In general, the arms' utility functions $v_i(M, s_i, s_{-i})$ may be discontinuous in the arms' strategies due to their intricate dependence on the learning algorithm M. It is well-known that in games with discontinuous utilities, a NE may not exist [Ren99]. However, for all subsequently considered algorithms we will prove the existence of a NE by either explicitly describing the equilibrium or implicitly proving its existence.

[3] More generally, different arms i may have a different value-per-click ν_i that could as well depend on μ_i so that $v_i(M, s_i, s_{-i}) = \mathbb{E}_M[\sum_t \mathbb{1}_{\{i_t=i\}} c_{t,i} \nu_i]$. This can easily be accommodated for by our model and our results readily extend to this case since each arm's goal still boils down to maximizing the number of clicks.

[4] For instance, a sufficient condition for the arms to find a NE is their knowledge about how far away they are from the best arm, i.e., their optimality gap in post-click rewards $\Delta_i := \max_{j\in[K]} \mu_j - \mu_i$.

V.3.3 Strategic Regret

The learner's goal is to maximize $\sum_{t=1}^{T} u(s_{i_t}, \mu_{i_t})$ which naturally depends on the arm strategies $s_1, \ldots, s_K$. For given post-click values $\mu_1, \ldots, \mu_K$, the maximal utility $u(s^*, \mu^*)$ is then achieved for $\mu^* := \max_{i \in [K]} \mu_i$ and $s^* := s^*(\mu^*)$, that is, $u(s^*, \mu^*) = \max_{i \in [K]} \max_{s \in [0,1]} u(s, \mu_i)$. With $u(s^*, \mu^*)$ as a benchmark, we can define the *strategic regret* of a mechanism M under a pure-strategy equilibrium $\boldsymbol{s} \in \mathrm{NE}(M)$ as

$$R_T(M, \boldsymbol{s}) := \mathbb{E}\left[\sum_{t=1}^{T} u(s^*, \mu^*) - u(s_{i_t}, \mu_{i_t})\right]. \tag{V.2}$$

For some mixed-strategy equilibrium $\boldsymbol{\sigma} \in \mathrm{NE}(M)$, we then accordingly define strategic regret as $R_T(M, \boldsymbol{\sigma}) := \mathbb{E}_{\boldsymbol{s} \sim \boldsymbol{\sigma}}[R_T(M, \boldsymbol{s})]$. In general, there may exist several Nash equilibria for the arms under a given mechanism M. We can then consider the *strong strategic regret* of M given by the regret under the worst-case equilibrium:

$$R_T^+(M) := \max_{\boldsymbol{\sigma} \in \mathrm{NE}(M)} R_T(M, \boldsymbol{\sigma}),$$

or the *weak strategic regret* given by the regret under the most favorable equilibrium:

$$R_T^-(M) := \min_{\boldsymbol{\sigma} \in \mathrm{NE}(M)} R_T(M, \boldsymbol{\sigma}),$$

where $R_T^-(M) \leq R_T^+(M)$. The regret upper bound of our proposed algorithm, UCB-S, holds under any equilibrium in NE(UCB-S), thereby bounding *strong strategic regret* (Theorem V.5.3). On the other hand, the proven lower bounds (Proposition V.4.1 and Theorem V.5.5) hold for *weak strategic regret* and thus also apply to its strong counterpart.

V.4 Limitations of Incentive-Unaware Algorithms

We start our analysis of the strategic click-bandit problem by showing that simply finding the arm with the largest post-click reward, $\mathrm{argmax}_i \mu_i$, or largest utility, $\mathrm{argmax}_i u(s_i, \mu_i)$, is insufficient to achieve $o(T)$ *weak* strategic regret. In fact, we find that even with oracle knowledge of $\mu_1, \ldots, \mu_K$ and $s_1, \ldots, s_K$, an algorithm may suffer linear weak strategic regret if it fails to account for the arms' strategic nature. For such incentive-*unaware* oracle algorithms, we show a $\Omega(T)$ lower bound for weak strategic regret on any non-trivial problem instance.

Recall that $\mu^* := \max_{i \in [K]} \mu_i$ and $s^* := s^*(\mu^*)$ and suppose that the arm $i^* = \mathrm{argmax}_{i \in [K]} \mu_i$ with maximal post-click rewards is unique. Our negative results rely on the following problem-dependent gaps in terms of utility:

$$\beta := u(s^*, \mu^*) - u(1, \mu^*) \quad \text{and} \quad \eta := u(s^*, \mu^*) - \max_{i \in [K] \setminus \{i^*\}} u^*(\mu_i).$$

Here, β denotes the cost of the optimal arm i^* deviating from the desired strategy $s^* = s^*(\mu^*)$ by playing $s_{i^*} = 1$. The quantity η denotes the gap between the maximally achievable utility $u(s^*, \mu^*)$ and the utility of the second best arm.

Proposition V.4.1. *Let μ-Oracle be the algorithm with oracle knowledge of $\mu_1, \dots, \mu_K$ that plays $i_t = \operatorname{argmax}_{i\in[K]} \mu_i$ in every round t, whereas (s,μ)-Oracle is the algorithm with oracle knowledge of $\mu_1, \dots, \mu_K$ and $s_1, \dots, s_K$ that always plays $i_t = \operatorname{argmax}_{i\in[K]} u(s_i, \mu_i)$ with ties broken in favor of the larger μ. We then have*

(i) *Under every equilibrium $\boldsymbol{\sigma} \in \mathrm{NE}(\mu\text{-Oracle})$, the μ-Oracle suffers regret $\Omega\big(\beta T\big)$, i.e.,*

$$R_T^-(\mu\text{-Oracle}) = \Omega\big(\beta T\big).$$

(ii) *Under every $\boldsymbol{\sigma} \in \mathrm{NE}((s,\mu)\text{-Oracle})$, the (s,μ)-Oracle suffers regret $\Omega\big(\min\{\beta,\eta\}T\big)$, i.e.,*

$$R_T^-((s,\mu)\text{-Oracle}) = \Omega\big(\min\{\beta,\eta\}T\big).$$

Proof Sketch. (i): We show that $s = 1$ is a strictly dominant strategy for arm i^* under the μ-Oracle. This implies that arm i^* plays $s_{i^*} = 1$ with probability one in every NE under the μ-Oracle. The claimed lower bound then follows from bounding the instantaneous regret per round from below by β. *(ii)*: Let $j^* \in \operatorname{argmax}_{i\neq i^*} \mu_i$. It can be seen that in any NE, arm i^* will play the largest $s \in [0,1]$ such that $u(s, \mu_{i^*}) \geq u(s_{j^*}, \mu_{j^*})$. We then show that either $s_{i^*} = 1$ or $u(s_{i^*}, \mu_{i^*}) = u(s^*(\mu_{j^*}), \mu_{j^*})$. Once again this allows us to lower bound the regret per round by $\min\{\beta,\eta\}$. ■

As a concrete example of the failure of the μ-Oracle and the (s,μ)-Oracle, let us consider the running example of $u(s,\mu) = s\mu - \lambda(s-\mu)^2$. In this case, letting $\lambda = 5$ and $\mu_{i^*} = 0.8$ and $\mu_i \leq 0.7$ for $i \neq i^*$, we get $\beta \geq 0.1$ and $\eta \geq 0.1$ so that both oracles suffer $\Omega(T)$ regret in every equilibrium.

V.5 No-Regret Incentive-Aware Learning: UCB-S

The results of Proposition V.4.1 suggest that any incentive-unaware learning algorithm that is oblivious to the strategic nature of the arms will generally fail to achieve low regret. In particular, "unconditional" selection of any arm will likely result in undesirable equilibria among arms. For these reasons, we deploy a conceptually simple screening idea, which threatens arms with elimination when deviating from the desired strategies.

Let denote $n_t(i)$ be the number of times up to (and including) round t that arm i was selected, and let $m_t(i)$ denote the number of times post-click rewards were observed for arm i up to (and including) round t. Let $\hat{s}_i^t$ be the average observed click-rate and $\hat{\mu}_i^t$ the average observed post-click reward for arm i. We then define the pessimistic and optimistic estimates of s_i and μ_i as

$$\underline{s}_i^t = \hat{s}_i^t - \sqrt{2\log(T)/n_t(i)}, \qquad \overline{s}_i^t = \hat{s}_i^t + \sqrt{2\log(T)/n_t(i)},$$
$$\underline{\mu}_i^t = \hat{\mu}_i^t - \sqrt{2\log(T)/m_t(i)}, \qquad \overline{\mu}_i^t = \hat{\mu}_i^t + \sqrt{2\log(T)/m_t(i)}.$$

Mechanism 15 UCB with Screening (UCB-S)

1: **initialize:** $A_0 = [K]$
2: **for** $t = 1, \ldots, T$ **do**
3: **if** $A_{t-1} \neq \emptyset$ **then**
4: Select $i_t \in \operatorname{argmax}_{i \in A_{t-1}} \overline{\mu}_i^{t-1}$
5: **else**
6: Select i_t uniformly at random from $[K]$
7: Arm i_t is clicked with probability s_{i_t}, i.e., $c_{t,i_t} \sim \text{Bern}(s_{i_t})$
8: **if** i_t was clicked ($c_{t,i_t} = 1$) **then**
9: Observe post-click reward r_{t,i_t}
10: **if** $\overline{s}_{i_t}^t < \min_{\mu \in [\underline{\mu}_{i_t}^t, \overline{\mu}_{i_t}^t]} s^*(\mu)$ or $\underline{s}_{i_t}^t > \max_{\mu \in [\underline{\mu}_{i_t}^t, \overline{\mu}_{i_t}^t]} s^*(\mu)$ **then**
11: Ignore arm i_t in future rounds: $A_t \leftarrow A_{t-1} \setminus \{i_t\}$

where $\underline{s}_i^t = -\infty$ and $\overline{s}_i^t = +\infty$ for $n_t(i) = 0$ as well as $\underline{\mu}_i^t = -\infty$ and $\overline{\mu}_i^t = +\infty$ for $m_t(i) = 0$.

In every round, UCB-S (Mechanism 15) selects arms optimistically according to their post-click rewards and subsequently observes if the arm is clicked, i.e., c_{t,i_t}, and, if so, a post-click reward r_{t,i_t}. However, if an arm's click-rate s_i is detected to be different from the learner's desired arm strategy $s^*(\mu_i)$, the arm is eliminated forever, expressed by the screening rule in line 10:

$$\overline{s}_{i_t}^t < \min_{\mu \in [\underline{\mu}_{i_t}^t, \overline{\mu}_{i_t}^t]} s^*(\mu) \quad \text{or} \quad \underline{s}_{i_t}^t > \max_{\mu \in [\underline{\mu}_{i_t}^t, \overline{\mu}_{i_t}^t]} s^*(\mu).$$

The only exception is when all arms have been eliminated. Then, UCB-S plays them all uniformly for the remaining rounds. To ensure that the elimination of an arm is credible and justified with high probability, we leverage confidence bounds on s_i and μ_i. More precisely, if an arm is truthful and chooses $s_i = s^*(\mu_i)$, then with probability $1 - 1/T^2$ it will not be eliminated by the screening rule.

As a prelude to the analysis of the UCB-S mechanism, we begin by showing that there always exists a NE among the arms under UCB-S. As mentioned briefly in Section V.3, the existence of a NE among the arms is not guaranteed under an arbitrary mechanism due to the arms' continuous strategy space and possibly discontinuous utility function.

Lemma V.5.1. *For any post-click rewards $\mu_1, \ldots, \mu_K$, there always exists a (possibly mixed) Nash equilibrium for the arms under the UCB-S mechanism.*

V.5.1 Characterizing the Nash Equilibria under UCB-S

We now approximately characterize all NE for the arms under the UCB-S mechanism. In order to prove a regret upper bound for UCB-S, it will be key to ensure that each arm i plays a strategy s_i which is sufficiently close to the desired strategy $s^*(\mu_i)$ (i.e., the strategy that maximizes the learner's utility). This is particularly important for arms i^* with maximal post-click rewards

$\mu_{i^*} = \max_{i \in [K]} \mu_i$. If such arms i^* were to deviate substantially from $s^*(\mu_{i^*})$, e.g., by a constant amount, the learner would be forced to suffer constant regret even when selecting arms with maximal post-click rewards, making it impossible to achieve sublinear regret.

In the following, we show that under the UCB-S mechanism every NE is such that the strategies of arms with maximal post-click rewards deviate from the desired strategies by at most $\tilde{\mathcal{O}}(\sqrt{K/T})$. We then also show that for suboptimal arms the difference between each arm i's strategy s_i and the desired strategy $s^*(\mu_i)$ is governed by their optimality gap in post-click rewards, given by $\Delta_i := \mu^* - \mu_i$. Recall that H denotes the Lipschitz constant of $s^*(\mu)$.

Theorem V.5.2. *For all $\boldsymbol{s} \in \operatorname{supp}(\boldsymbol{\sigma})$ with $\boldsymbol{\sigma} \in \mathrm{NE}(\text{UCB-S})$ and all $i \in [K]$:*

$$s_i = s^*(\mu_i) + \mathcal{O}\left(H \cdot \max\left\{\Delta_i, \sqrt{\frac{K \log(T)}{T}}\right\}\right).$$

In particular, for all arms $i^* \in [K]$ with $\Delta_{i^*} = 0$, i.e., maximal post-click rewards:

$$s_{i^*} = s^*(\mu_{i^*}) + \mathcal{O}\left(H\sqrt{\frac{K \log(T)}{T}}\right).$$

The derivation of Theorem V.5.2 can be best understood by noting that the estimates of each arm's strategy roughly concentrate at a rate of $1/\sqrt{t}$. Then, depending on how often an arm expects to be selected by UCB-S, it can exploit our uncertainty about its strategy and safely increase its click-rates to match our confidence. Generally, optimal arms expect at least T/K allocations while preventing elimination, which can be seen to imply NE strategies that deviate by at most $\sqrt{K/T}$. On the other hand, suboptimal arms can expect roughly $\log(T)/\Delta_i^2$ allocations as long as they can prevent elimination and all other arms act rationally, which results in the linear dependence on Δ_i. Hence, interestingly UCB-S' selection policy directly impacts the truthfulness of the arms, as arms that are selected more frequently are forced to choose strategies closer to $s^*(\mu_i)$. We thus observe a trade-off between incentivizing *all* arms to be truthful and recommending only the best arms. The proof of Theorem V.5.2 (Appendix E.3) then relies on the above observation and careful and repeated application of the best response property of the Nash equilibrium.

V.5.2 Upper Bound of the Strong Strategic Regret of UCB-S

With the approximate NE characterization from Theorem V.5.2 at our disposal, we are ready to prove a regret upper bound for UCB-S. We show that the *strong strategic regret* of the UCB-S mechanism is upper bounded by $\tilde{\mathcal{O}}(\sqrt{KT})$, that is, for any $\boldsymbol{\sigma} \in \mathrm{NE}(\text{UCB-S})$ the regret guarantee holds.

Theorem V.5.3. *Let $\Delta_i := \mu^* - \mu_i$ and let L and H denote the Lipschitz constants of $u(s, \mu)$ and $s^*(\mu)$, respectively. The strong strategic regret of UCB-S is bounded*

as

$$R_T^+(\text{UCB-S}) = LH \cdot \mathcal{O}\left(\sqrt{KT\log(T)} + \sum_{i:\Delta_i>0} \frac{\log(T)}{\Delta_i}\right). \tag{V.3}$$

In other words, the above regret bound is achieved under any equilibrium $\boldsymbol{\sigma} \in \text{NE(UCB-S)}$.

Proof Sketch. As suggested by the regret bound there are two sources of regret. Broadly speaking, the first term on the right hand side of (V.3) corresponds to the regret UCB-S suffers due to arms with maximal post-click rewards (i.e., $\Delta_i = 0$) deviating from the utility-maximizing strategy $s^*(\mu^*)$. For such arms Theorem V.5.2 bounded the deviation by a term of order $\sqrt{K/T}$, thereby leading to at most order $\sqrt{KT}$ regret. The second term in (V.3) corresponds to the regret suffered from playing arms with suboptimal post-click rewards, i.e., $\Delta_i > 0$. Using a typical UCB argument, the Lipschitzness of $u(s,\mu)$ and $s^*(\mu)$, and again Theorem V.5.2 applied to $|s^*(\mu^*) - s_i| \leq |s^*(\mu^*) - s^*(\mu_i)| + \mathcal{O}(H\Delta_i) \leq H\Delta_i + \mathcal{O}(H\Delta_i)$ we obtain the claimed upper bound. ■

Similarly to classical MABs we can state a regret bound independent of the instance-dependent quantities Δ_i and translate Theorem V.5.3 into a minimax-type guarantee.

Corollary V.5.4. *The strong strategic regret of UCB-S is bounded as*

$$R_T^+(\text{UCB-S}) = \mathcal{O}\left(LH\sqrt{KT\log(T)}\right).$$

In other words, the above regret bound is achieved under any equilibrium $\boldsymbol{\sigma} \in \text{NE(UCB-S)}$.

Theorem V.5.3 nicely shows that the additional cost of the incentive design and the strategic behavior of the arms is of order $\sqrt{KT}$ which primarily stems from arms with maximal post-click rewards deviating by roughly $\sqrt{K/T}$ from the desired strategy (see Theorem V.5.2). The dishonesty of suboptimal arms does not notably contribute to the regret and is contained in the $\log(T)/\Delta_i$ expressions as we can bound the number of times suboptimal arms are played sufficiently well. As a result, the total cost of incentive design and strategic behavior matches the minimax learning complexity of MABs so that we obtain an overall $\tilde{\mathcal{O}}(\sqrt{KT})$ strategic regret bound under every equilibrium.

V.5.3 Lower Bound for Weak Strategic Regret

Complementing our regret analysis, we prove a lower bound on *weak strategic regret* in the strategic click-bandit. By definition, weak strategic regret lower bounds its strong counterpart, i.e., $R_T^-(M) \leq R_T^+(M)$, so that the shown lower bound directly applies to strong strategic regret as well, which implies that UCB-S is near-optimal.

Theorem V.5.5. *Let M be any mechanism with $\mathrm{NE}(M) \neq \emptyset$. There exists a utility function u satisfying (A1)-(A3) and post-click rewards $\mu_1, \dots, \mu_K$ such that for all Nash equilibria $\boldsymbol{\sigma} \in \mathrm{NE}(M)$:*

$$R_T(M, \boldsymbol{\sigma}) = \Omega\big(\sqrt{KT}\big).$$

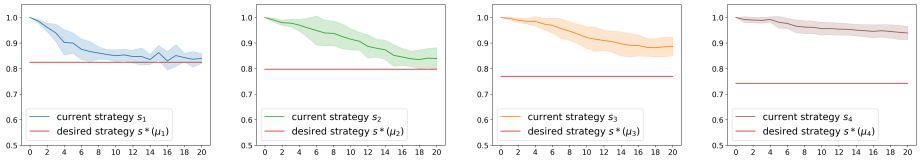
In other words, $R_T^-(M) = \Omega\big(\sqrt{KT}\big)$.

Proof Sketch. Consider the utility function $u(s, \mu) = s\mu$. Intuitively, for any low regret mechanism M the NE for the arms will be in $(s_1, \dots, s_K) = (1, \dots, 1)$ as these strategies maximize the learner's utility u and are to the advantage of the arms. In this case, the learning problem reduces to a classical MAB and we inherit the well-known minimax $\sqrt{KT}$ lower bound. However, it is not directly clear that there exists no better mechanism that would, e.g., incentivize arm strategies $(s_1, \dots, s_{i^*}, \dots, s_K) = (0, \dots, 1, \dots, 0)$ under which $i^* = \mathrm{argmax}_i\, \mu_i$ becomes easier to distinguish from $i \neq i^*$. For this reason, we argue via the arms' utilities and lower bound the minimal utility a suboptimal arm must receive in any NE. This directly implies a lower bound on the number of times we must play any suboptimal arm in equilibrium, which yields the claimed result. ■

V.6 Simulating Strategic Arm Behavior via Repeated Interaction

Goal of the experiments is to analyze the effect of the proposed incentive-aware learning algorithm UCB-S on strategically responding arms. Strategic arm behavior is here modeled through decentralized gradient ascent and repeated interaction with the mechanism. Contrary to the assumption of arms playing in NE, arms follow a simple gradient ascent strategy to adapt to the mechanism, which serves as a realistic and natural model of strategic behavior. This requires no prior knowledge from the point of view of the arms and all learning is performed through sequential interaction with the mechanism. For this reason, the final strategies in our experiments may not necessarily be in NE. Despite this, we want to see whether the mechanism is still able to incentivize arms to behave in the desired manner which will also provide insight into the robustness of the proposed incentive design.

Experimental Setup. We consider the earlier introduced utility function defined as $u(s, \mu) = s\mu - \lambda(s - \mu)^2$ such that the desired (learner's utility-maximizing) strategy given μ is $s^*(\mu) = (1 + \frac{1}{2\lambda})\mu$. We let $\lambda = 5$. To model the strategic behavior of arms in response to UCB-S, we let the strategic arms interact with the mechanism over the course of 20 epochs (x-axis) and model each arm's strategic behavior via gradient ascent w.r.t. its utility v_i. More precisely, after every epoch (i.e., interaction over $T = 50$k rounds), each arm performs an approximated gradient step with respect to its utility v_i. We initialized the arm strategies to $s_i = 1$, however, our experiments show that other initialization,



(a) Optimal arm with $\mu_1 = 0.75$.

(b) Suboptimal arm with $\mu_2 = 0.725$.

(c) Suboptimal arm with $\mu_3 = 0.7$.

(d) Suboptimal arm with $\mu_4 = 0.675$.

Figure V.2: The strategic behavior of $K = 4$ arms when each arm uses gradient ascent to maximize their utility v_i in response to the UCB-S mechanism. In red, the desired strategy $s^*(\mu_i)$ for each arm i, respectively. As suggested by Theorem V.5.2, the truthfulness, i.e., distance to $s^*(\mu_i)$, of a suboptimal arm i is governed by the arm's optimality gap Δ_i. We see this confirmed as the distance $s_i - s^*(\mu_i)$ increases as Δ_i increases. In accordance with our theoretical results, the optimal arm 1 has the largest incentive to play close to the desired strategy (as it loses the most when eliminated).

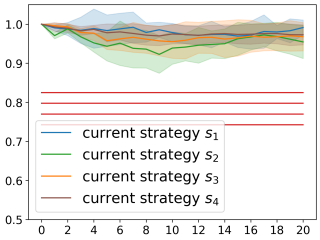


Figure V.3: Strategic arm behavior in response to the incentive-*unaware* standard UCB algorithm. UCB fails to incentivize desirable arm strategies. The strategies are plotted jointly and all 4 arms exhibit similar behavior.

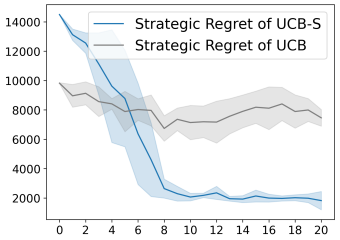


Figure V.4: Strategic regret of UCB-S and standard UCB as arms adapt their strategies in response to the respective algorithm. The more the arms have interacted with the UCB-S mechanism, the less regret UCB-S suffers.

such as $s_i = 0$ or $s_i = 0.5$, yield similar results. All results are averaged over 10 complete runs and the standard deviation shown in shaded color.

Results. In Section V.5 we have theoretically shown that our mechanism incentivizes desirable NE among arms. The conducted simulations show that under natural greedy behavior as modeled by gradient ascent, the incentive design of UCB-S is still effective and desirable arm strategies incentivized (Figure V.2). Most notably, the optimal arm (having the largest incentive to be truthful) converges to a strategy close to the desired strategy $s^*(\mu_1)$. The suboptimal arms do not converge to a strategy close to the desired strategy and we observe that the distance to $s^*(\mu_i)$ depends on the optimality gap Δ_i, which mirrors our theoretical results (Theorem V.5.2). In addition, Figure V.4 shows that as the arms interact with UCB-S and adapt their strategies, the regret of UCB-S

improves substantially. In contrast, incentive-unaware algorithms like UCB fail to incentivize desirable strategies (all arm strategies remain close to 1, see Figure V.3) and UCB accordingly suffers large regret (Figure V.4) throughout all epochs. The observation that UCB-S initially suffer larger regret than UCB can be explained by the elimination rule causing UCB-S to select arms uniformly at random when arms are notably untruthful. This threat of elimination, however, incentivizes the arms to adapt their strategies in the next epoch and eventually leads to smaller regret for UCB-S.

V.7 Discussion

We study the strategic click-bandit problem in which each arm is associated with a click-rate, chosen strategically by the arms, and an immutable post-click reward. We show the necessity of incentive design in this model and design an incentive-aware online learning algorithm that incentivizes desirable arm strategies under uncertainty. As the learner has no prior knowledge of the arm strategies and the post-click rewards, the mechanism design is approximate and leaves room for arms to exploit the learner's uncertainty. This leads to an interesting regret bound which makes the intuition precise that arms can exploit the learner's uncertainty about their strategies. In our simulations we then observe that our incentive design is robust and still effective under natural greedy arm behavior and that the design of incentive-aware learning algorithms is necessary to achieve low regret under strategic arm behavior. Some interesting open questions which we leave for future work include whether the proposed incentive design remains effective under adaptive arm strategies and whether we can construct a mechanism under which there exists a desirable NE in dominant strategies.

References

[Aue02] Auer, P. "Using Confidence Bounds for Exploitation-Exploration Trade-offs". In: *Journal of Machine Learning Research* vol. 3 (2002), pp. 397–422.

[BKS15] Babaioff, M., Kleinberg, R. D., and Slivkins, A. "Truthful mechanisms with implicit payment computation". In: *Journal of the ACM (JACM)* vol. 62, no. 2 (2015), pp. 1–37.

[Bra+19] Braverman, M. et al. "Multi-armed bandit problems with strategic arms". In: *Conference on Learning Theory*. PMLR. 2019, pp. 383–416.

[BRA20] Bouneffouf, D., Rish, I., and Aggarwal, C. "Survey on applications of multi-armed and contextual bandits". In: *2020 IEEE Congress on Evolutionary Computation (CEC)*. IEEE. 2020, pp. 1–8.

[BSS09] Babaioff, M., Sharma, Y., and Slivkins, A. "Characterizing truthful multi-armed bandit mechanisms". In: *Proceedings of the 10th ACM conference on Electronic commerce*. 2009, pp. 79–88.

[BV19] Bergemann, D. and Välimäki, J. "Dynamic mechanism design: An introduction". In: *Journal of Economic Literature* vol. 57, no. 2 (2019), pp. 235–274.

[DK09] Devanur, N. R. and Kakade, S. M. "The price of truthfulness for pay-per-click auctions". In: *Proceedings of the 10th ACM conference on Electronic commerce*. 2009, pp. 99–106.

[Don+22] Dong, J. et al. "Combinatorial Bandits under Strategic Manipulations". In: *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 2022, pp. 219–229.

[FPX20] Feng, Z., Parkes, D., and Xu, H. "The intrinsic robustness of stochastic bandits to strategic manipulation". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 3092–3101.

[Fre+20] Freeman, R. et al. "No-regret and incentive-compatible prediction with expert advice". In: *arXiv preprint arXiv:2002.08837* (2020).

[Gao+21] Gao, G. et al. "Auction-based combinatorial multi-armed bandit mechanisms with strategic arms". In: *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE. 2021, pp. 1–10.

[GH13] Ghosh, A. and Hummel, P. "Learning and incentives in user-generated content: Multi-armed bandits with endogenous arms". In: *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*. 2013, pp. 233–246.

[Gli52] Glicksberg, I. L. "A further generalization of the Kakutani fixed point theorem, with application to Nash equilibrium points". In: *Proceedings of the American Mathematical Society* vol. 3, no. 1 (1952), pp. 170–174.

[GLT12] Gatti, N., Lazaric, A., and Trovò, F. "A truthful learning mechanism for contextual multi-slot sponsored search auctions with externalities". In: *Proceedings of the 13th ACM Conference on Electronic Commerce*. 2012, pp. 605–622.

[GMS19] Garivier, A., Ménard, P., and Stoltz, G. "Explore first, exploit next: The true shape of regret in bandit problems". In: *Mathematics of Operations Research* vol. 44, no. 2 (2019), pp. 377–399.

[Har+16] Hardt, M. et al. "Strategic classification". In: *Proceedings of the 2016 ACM conference on innovations in theoretical computer science*. 2016, pp. 111–122.

[HBR12] Hofmann, K., Behr, F., and Radlinski, F. "On caption bias in interleaving experiments". In: *Proceedings of the 21st ACM international conference on Information and knowledge management*. 2012, pp. 115–124.

[Hro+22] Hron, J. et al. “Modeling content creator incentives on algorithm-curated platforms”. In: *arXiv preprint arXiv:2206.13102* (2022).

[Hu+23] Hu, X. et al. “Incentivizing High-Quality Content in Online Recommender Systems”. In: *arXiv preprint arXiv:2306.07479* (2023).

[Kan+23] Kandasamy, K. et al. “VCG Mechanism Design with Unknown Agent Values under Stochastic Bandit Feedback”. In: *Journal of Machine Learning Research* vol. 24, no. 53 (2023), pp. 1–45.

[LH18] Liu, Y. and Ho, C.-J. “Incentivizing high quality user contributions: New arm generation in bandit learning”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018.

[Li+10] Li, L. et al. “A contextual-bandit approach to personalized news article recommendation”. In: *Proceedings of the 19th international conference on World wide web*. 2010, pp. 661–670.

[LR85] Lai, T. L. and Robbins, H. “Asymptotically efficient adaptive allocation rules”. In: *Advances in applied mathematics* vol. 6, no. 1 (1985), pp. 4–22.

[LS20] Lattimore, T. and Szepesvári, C. *Bandit algorithms*. Cambridge University Press, 2020.

[MT04] Mannor, S. and Tsitsiklis, J. N. “The sample complexity of exploration in the multi-armed bandit problem”. In: *Journal of Machine Learning Research* vol. 5, no. Jun (2004), pp. 623–648.

[Naz+16] Nazerzadeh, H. et al. “Where to sell: Simulating auctions from learning algorithms”. In: *Proceedings of the 2016 ACM Conference on Economics and Computation*. 2016, pp. 597–598.

[NR99] Nisan, N. and Ronen, A. “Algorithmic mechanism design”. In: *Proceedings of the thirty-first annual ACM symposium on Theory of computing*. 1999, pp. 129–140.

[Par07] Parkes, D. C. “Online mechanisms”. In: (2007).

[PST14] Pavan, A., Segal, I., and Toikka, J. “Dynamic mechanism design: A myersonian approach”. In: *Econometrica* vol. 82, no. 2 (2014), pp. 601–653.

[Ren99] Reny, P. J. “On the existence of pure and mixed strategy Nash equilibria in discontinuous games”. In: *Econometrica* vol. 67, no. 5 (1999), pp. 1029–1056.

[Sli+19] Slivkins, A. et al. “Introduction to multi-armed bandits”. In: *Foundations and Trends® in Machine Learning* vol. 12, no. 1-2 (2019), pp. 1–286.

[SLO22] Shin, S., Lee, S., and Ok, J. “Multi-armed Bandit Algorithm against Strategic Replication”. In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2022, pp. 403–431.

[Wan+21] Wang, W. et al. “Clicks can be cheating: Counterfactual recommendation for mitigating clickbait issue”. In: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2021, pp. 1288–1297.

[WWW17] Wang, H., Wu, Q., and Wang, H. “Factorization bandits for interactive recommendation”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 31. 1. 2017.

[You23] Youtube. *How to earn money on YouTube*. 2023.

[YPR10] Yue, Y., Patel, R., and Roehrig, H. “Beyond position bias: Examining result attractiveness as a source of presentation bias in clickthrough data”. In: *Proceedings of the 19th international conference on World wide web*. 2010, pp. 1011–1018.

[ZC21] Zhang, H. and Conitzer, V. “Incentive-aware PAC learning”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 6. 2021, pp. 5797–5804.

[Zon+16] Zong, S. et al. “Cascading bandits for large-scale recommendation problems”. In: *arXiv preprint arXiv:1603.05359* (2016).

The appendix is arranged as follows:

- Section E.1 contains the proof of Proposition V.4.1.
- Section E.2 proves the existence of a NE under UCB-S (Lemma V.5.1).
- Section E.3 contains the proof of the NE characterization (Theorem V.5.2).
- Section E.4 contains the regret upper bound of UCB-S (Theorem V.5.3).
- Section E.5 contains the proof of Corollary V.5.4.
- Section E.6 contains the proof of the lower bound (Theorem V.5.5).
- Section E.7 contains basic technical lemmas that are used in the proofs.
- Section E.8 discusses additional related work.

E.1 Proof of Proposition V.4.1

Proof of Proposition V.4.1. (i): Under any strategy profile $\boldsymbol{s} = (s_1, \dots, s_K)$, arm $i \neq i^*$ has utility $v_i(\mu\text{-Oracle}, s_i, s_{-i}) = 0$, while arm i^* has utility

$$v_{i^*}(\mu\text{-Oracle}, s_{i^*}, s_{-i^*}) = Ts_{i^*}.$$

Hence, the pure strategy $s = 1$ is a strictly dominant strategy for arm i^*, which implies that i^* plays $s_{i^*} = 1$ with probability one in every Nash equilibrium. Now,

$$u(s^*, \mu_{i^*}) - u(s_{i^*}, \mu_{i^*}) = u(s^*, \mu_{i^*}) - u(1, \mu_{i^*}) = \beta$$

and the μ-Oracle thus suffers regret β every round, which implies the claimed $\Omega(\beta T)$ lower bound in every equilibrium.

(ii): Let $j^* \in \operatorname{argmax}_{i \neq i^*} \mu_i$ be the arm with second largest post-click value and define $u^*_{j^*} := u(s^*(\mu_{j^*}), \mu_{j^*})$. Let s' be the largest $s \in [0,1]$ such that $u(s', \mu_{i^*}) \geq u^*_{j^*}$. We distinguish between two cases:

Case 1. Suppose that $u(s', \mu_{i^*}) > u^*_{j^*}$. From the continuity of u it then follows that $s' = 1$. To see this, suppose the contrary is true. Then, for all $s'' > s'$ with $s'' \in [0,1]$ it must hold that $u(s'', \mu_{i^*}) < u^*_{j^*}$ by definition of s' as the largest $s \in [0,1]$ such that $u(s, \mu_{i^*}) \geq u_{j^*}$. However, this contradicts the continuity of $u(s, \mu)$ in s, since we have just shown that $u(s'', \mu_{i^*}) < u^*_{j^*} < u(s', \mu_{i^*})$ for all $s'' > s'$. We have thus shown by contradiction that $s' = 1$.

Then, if arm i^* chooses strategy $s_{i^*} = 1$, arm i^* is pulled every round by (s, μ)-Oracle for all $s_{-i^*} \in [0,1]^{K-1}$ so that $v_{i^*}((s, \mu)\text{-Oracle}, 1, s_{-i^*}) = T$. This immediately implies that $s_{i^*} = 1$ is a strictly dominant strategy for i^*, since $v_{i^*}((s, \mu)\text{-Oracle}, s, s_{-i^*}) \leq Ts < T$ for all $s \in [0, 1)$. Thus, arm i^* plays $s_{i^*} = 1$ in every Nash equilibrium of the arms. Analogous to the proof of (i), this yields $|u(s^*, \mu^*) - u(s_{i^*}, \mu_{i^*})| = \beta$, which implies that the (s, μ)-Oracle suffers $\Omega(\beta T)$ under any Nash equilibrium of the arms.

Case 2. Suppose that $u(s', \mu_{i^*}) = u^*_{j^*}$. In a first step, we show that arm i^* plays s' with probability one in every Nash equilibrium. We begin by noting that if arm i^* plays $s_{i^*} = s'$, then for any opponent strategies $s_{-i^*} \in [0,1]^{K-1}$ arm i^* is played all T rounds so that $v_{i^*}((s,\mu)\text{-Oracle}, s', s_{-i^*}) = Ts'$. Naturally, s' thus strictly dominates any other strategy $s'' < s'$, since $v_i((s,\mu)\text{-Oracle}, s'', s_{i^*}) \leq Ts''$.

Next, suppose that arm i^* plays some strategy $s'' > s'$ with probability one.[5] Then, by definition of s', we have $u(s'', \mu_{i^*}) < u^*_{j^*} := u(s^*(\mu_{j^*}), \mu_{j^*})$. As a result, arm j^*'s best response s_{j^*} to s'' will be such that $u(s'', \mu_{i^*}) < u(s_{j^*}, \mu_{j^*})$, thereby obtaining utility $v_{j^*}((s,\mu)\text{-Oracle}, s_{j^*}, s_{-j^*}) \geq Ts^*(\mu_{j^*})$. As a result, if j^* plays a best response, arm i^* receives utility 0 when playing s'', whereas arm i^* receives utility Ts' when playing s'. Hence, any $s'' > s'$ cannot be part of an equilibrium for arm i^* and we have shown that arm i^* plays s' with probability one in every equilibrium. Finally, by definition of s', we have

$$u(s^*, \mu^*) - u(s', \mu_{i^*}) \geq u(s^*, \mu^*) - u(s^*(\mu_{j^*}), \mu_{j^*}) = u(s^*, \mu^*) - u^*(\mu_{j^*}) = \eta$$

which implies that (s,μ)-Oracle suffers $\Omega(\eta T)$ regret under any Nash equilibrium of the arms. Hence, we obtain the claimed lower bound of $\Omega\big(\min\{\beta, \eta\}T\big)$. ■

Remark E.1.1. Interestingly, when the (s,μ)-Oracle from Proposition V.4.1 *(ii)* does not break ties in favor of the larger μ but instead uniformly at random, it can be shown that in all but a few problem instances no Nash equilibrium for the arms exists. However, for any $\varepsilon > 0$ we can explicitly construct an ε-Nash equilibrium for the arms under which the algorithm suffers $\Omega(\min\{\beta, \eta\, T)$ strategic regret.

Before proving the statement of Remark E.1.1, we formally introduce the concept of an ε-Nash equilibrium among the arms here.

Definition E.1.2 (ε-Nash Equilibrium)**.** For $\varepsilon > 0$, we say that strategies $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_K)$ form an ε-Nash equilibrium under M if $v_i(M, \sigma_i, \sigma_{-i}) \geq v_i(M, \sigma'_i, \sigma_{-i}) - \varepsilon$ for all $i \in [K]$ and $\sigma'_i \in \Sigma$.

For Remark E.1.1, we will show that there exists an ε-Nash equilibrium in pure-strategies $\boldsymbol{s} \in [0,1]^K$ such that the oracle algorithm that breaks ties uniformly suffers linear strategic regret.

Proof of Remark E.1.1. As in the proof of Proposition V.4.1 (ii), let $j^* \in \operatorname{argmax}_{i \neq i^*} \mu_i$ be the arm with second largest post-click value and define $u^*_{j^*} := u(s^*(\mu_{j^*}), \mu_{j^*})$. Now, let s' be the largest $s \in [0,1]$ such that

$$u(s', \mu_{i^*}) \geq u^*_{j^*} \text{ and } u(s' - \varepsilon', \mu_{i^*}) > u^*_{j^*} \text{ for all } \varepsilon' > 0.$$

Note that such s' exists since u is continuous and $u(s^*(\mu_{i^*}), \mu_{i^*}) > u^*_{j^*}$. We again distinguish between two cases, similarly to the proof of Proposition V.4.1.

[5] For simplicity, we assume that arm i^* plays the strategy with probability one. The case where i^* plays $s'' > s'$ with some positive probability can be treated analogously.

Case 1. If $u(s', \mu_{i^*}) > u^*_{j^*}$, it follows that $s' = 1$. This means that $\boldsymbol{s} = (s_{i^*}, s_{-i^*})$ with $s_{i^*} = 1$ and arbitrary $s_{-i^*} \in [0,1]^{K-1}$ form a pure strategy Nash equilibrium for the arms. As in the proof of (i), we then obtain

$$u(s^*, \mu^*) - u(s_{i^*}, \mu_{i^*}) = \beta$$

which implies order $\Omega(\beta T)$ regret under (s_{i^*}, s_{-i^*}).

Case 2. Now, suppose that $u(s', \mu_{i^*}) = u^*_{j^*}$. Let $s_{i^*} = s' - \varepsilon'$ and $s_i = s^*(\mu_i)$ for all $i \neq i^*$. We see that (s_{i^*}, s_{-i^*}) is a $(T\varepsilon')$-Nash equilibrium under the oracle algorithm. Hence, for any $\varepsilon > 0$, the strategy profile $\boldsymbol{s}_{\varepsilon'} := (s' - \varepsilon', s_{-i^*})$ is a ε-Nash equilibrium for all $\varepsilon' < \frac{\varepsilon}{T}$. Using that u is L-Lipschitz, we have

$$|u(s' - \varepsilon', \mu_{i^*}) - u(s_{j^*}, \mu_{j^*})| = |u(s' - \varepsilon', \mu_{i^*}) - u(s', \mu_{i^*})| \leq L\varepsilon',$$

and it follows that

$$|u(s^*, \mu^*) - u(s' - \varepsilon', \mu_{i^*})| \geq |u(s^*, \mu^*) - u(s_{j^*}, \mu_{j^*})| - L\varepsilon' \geq \eta - L\varepsilon',$$

We can choose $\varepsilon' < \frac{\varepsilon}{T}$ sufficiently small so that $L\varepsilon' < 1/T$. Hence, over T rounds the oracle algorithm suffers $\Omega(\eta T)$ regret under the ε-Nash equilibrium given by $\boldsymbol{s}_{\varepsilon'}$. This yields the claimed lower bound.

■

E.2 Proof of Lemma V.5.1

Proof of Lemma V.5.1. We use Glicksberg's theorem [Gli52], which guarantees the existence of a Nash equilibrium in continuous games with compact strategy space and continuous utility functions v_i. The strategy space $[0,1]$ is compact and we are left with proving the continuity of $v_i(\text{UCB-S}, \boldsymbol{s})$ in $\boldsymbol{s} \in [0,1]^K$. Since $v_i(\text{UCB-S}, \boldsymbol{s}) = \mathbb{E}_{\boldsymbol{s}}[n_T(i)]s_i$, the question is whether $\mathbb{E}_{\boldsymbol{s}}[n_T(i)]$ is continuous in $\boldsymbol{s}$ under UCB-S. The choice of $\boldsymbol{s}$ influences the actions of UCB-S when through the screening rule in line 10, but also the UCB-type selection in line 4, since post-click rewards are only observed when the arm is clicked.

Let $\mathcal{H}_t$ denote the history of the mechanism's selections and observations up to round t, consisting of tuples $(i_t, c_{t,i_t}, r_{t,i_t})$. Even though r_{t,i_t} is sometimes not observed, we include it here and note that it will not matter as the realizations of r_{t,i_t} are independent of $\boldsymbol{s}$. We let $\mathscr{H}_t$ up round t denote the set of all possible histories.

While we are interested in $\mathbb{E}_{\boldsymbol{s}}[n_T(i)]$, for technical reasons, it will be more convenient to prove the continuity of $\mathbb{P}_{\boldsymbol{s}}(\mathcal{H}_t \in \cdot)$ as a function of $\boldsymbol{s}$. We will do so by induction over $t \in [T]$. Naturally, $\mathbb{P}_{\boldsymbol{s}}(\mathcal{H}_1 \in \cdot)$ is continuous in $\boldsymbol{s}$, since $\mathbb{P}_{\boldsymbol{s}}(c_{1,i_1} = 1) = s_{i_1}$ and we break ties in line 4 independent of $\boldsymbol{s}$. For the proof by induction, let us now assume that $\mathbb{P}_{\boldsymbol{s}}(\mathcal{H}_t \in \cdot)$ is continuous in $\boldsymbol{s}$. Then, for $t+1$ we find that again $\mathbb{P}_{\boldsymbol{s}}(c_{t+1,i_{t+1}} = 1) = 1 - \mathbb{P}_{\boldsymbol{s}}(c_{t+1,i_{t+1}} = 0) = s_{i_{t+1}}$ is continuous in $\boldsymbol{s}$.[6] The interesting part is then whether $\mathbb{P}_{\boldsymbol{s}}(i_{t+1} = i)$ is continuous in $\boldsymbol{s}$.

[6] Note that $r_{t+1,i_{t+1}}$ is independent of $\boldsymbol{s}$.

Lemma E.2.1. *For any event A, if $\mathbb{P}_{\boldsymbol{s}}(A \mid \mathcal{H}_t)$ and $\mathbb{P}_{\boldsymbol{s}}(\mathcal{H}_t)$ are continuous in $\boldsymbol{s}$ for all $\mathcal{H}_t \in \mathscr{H}_t$, then $\mathbb{P}_{\boldsymbol{s}}(A)$ is also continuous in $\boldsymbol{s}$.*

Proof. This follows from the law of total probability. ■

We begin by analyzing the dependence of the screening rule in line 10 on $\boldsymbol{s}$. First of all, note that for all $i \in A_{t-1} \setminus \{i_t\}$, we always have $i \in A_t$, i.e., no other arm than i_t will ever be eliminated at the end of round t. Moreover, since $\mathbb{P}_{\boldsymbol{s}}(c_{t,i_t} = 1) = s_{i_t}$ is continuous in $\boldsymbol{s}$, it follows that $\mathbb{P}_{\boldsymbol{s}}(\underline{s}^t_{i_t} > a)$ and $\mathbb{P}_{\boldsymbol{s}}(\underline{s}^t_{i_t} > a \mid \mathcal{H}_t)$ are also continuous in $\boldsymbol{s}$ for all $a \in \mathbb{R}$. Consequently, the probability that arm i_t is eliminated in line 10 at the end of round t, i.e., $\mathbb{P}_{\boldsymbol{s}}(i_t \notin A_t)$, must be continuous in $\boldsymbol{s}$.

Let us assume that $A_t \neq \emptyset$, since $\mathbb{P}_{\boldsymbol{s}}(i_{t+1} = i)$ is always continuous in $\boldsymbol{s}$ if $A_t = \emptyset$. If $i \notin A_t$, we have $\mathbb{P}_{\boldsymbol{s}}(i_{t+1} = i) = 0$. Note that for all $i \neq i_t$, we have $\overline{\mu}^t_i = \overline{\mu}^{t-1}_i$. We will now first consider any $i \neq i_t$. If $i \in A_t$, we then have

$$
\begin{aligned}
&\mathbb{P}_{\boldsymbol{s}}(i_{t+1} = i \mid \mathcal{H}_t) \\
&= \mathbb{P}_{\boldsymbol{s}}\big(\overline{\mu}^t_i > \max_{j \in A_t \setminus \{i, i_t\}} \overline{\mu}^t_j \mid i_t \notin A_t, \mathcal{H}_t\big) \cdot \mathbb{P}_{\boldsymbol{s}}(i_t \notin A_t \mid \mathcal{H}_t) \\
&\quad + \mathbb{P}_{\boldsymbol{s}}\big(\overline{\mu}^t_i > \max_{j \in A_t \setminus \{i\}} \overline{\mu}^t_j \mid i_t \in A_t, \mathcal{H}_t\big) \cdot \mathbb{P}_{\boldsymbol{s}}(i_t \in A_t \mid \mathcal{H}_t) \qquad (4) \\
&= \mathbb{P}\big(\overline{\mu}^{t-1}_i > \max_{j \in A_t \setminus \{i, i_t\}} \overline{\mu}^{t-1}_j \mid i_t \notin A_t, \mathcal{H}_t\big) \cdot \mathbb{P}_{\boldsymbol{s}}(i_t \notin A_t \mid \mathcal{H}_t) \\
&\quad + \mathbb{P}\big(\overline{\mu}^{t-1}_i > \max_{j \in A_t \setminus \{i, i_t\}} \overline{\mu}^{t-1}_j \mid i_{t+1} \neq i_t, \mathcal{H}_t\big) \qquad (5) \\
&\quad \cdot \mathbb{P}_{\boldsymbol{s}}(i_{t+1} \neq i_t \mid i_t \in A_t, \mathcal{H}_t) \cdot \mathbb{P}_{\boldsymbol{s}}(i_t \in A_t \mid \mathcal{H}_t).
\end{aligned}
$$

The leading factors are independent of $\boldsymbol{s}$ and we have already shown that $\mathbb{P}_{\boldsymbol{s}}(i_t \notin A_{t+1})$ is continuous in $\boldsymbol{s}$. We are thus left with proving the continuity of $\mathbb{P}_{\boldsymbol{s}}(i_{t+1} \neq i_t \mid i_t \in A_{t+1}, \mathcal{H}_t)$.

It holds that $\mathbb{P}_{\boldsymbol{s}}(\overline{\mu}^t_{i_t} \in \cdot \mid \mathcal{H}_t) = s_{i_t}\mathbb{P}(\overline{\mu}^t_{i_t} \in \cdot \mid c_{t,i_t} = 1, \mathcal{H}_t) + (1 - s_{i_t})\mathbb{P}(\overline{\mu}^t_{i_t} \in \cdot \mid c_{t,i_t} = 0, \mathcal{H}_t)$, where we used that $\mathbb{P}_{\boldsymbol{s}}(\overline{\mu}^t_{i_t} \in \cdot \mid c_{t,i_t}, \mathcal{H}_t) = \mathbb{P}(\overline{\mu}^t_{i_t} \in \cdot \mid c_{t,i_t}, \mathcal{H}_t)$ is independent of $\boldsymbol{s}$ (conditional on the click-event c_{t,i_t}). Hence, as a sum and product of continuous functions $\mathbb{P}_{\boldsymbol{s}}(\overline{\mu}^t_{i_t} \in \cdot \mid \mathcal{H}_t)$ is continuous in $\boldsymbol{s}$ and we get that

$$
\mathbb{P}_{\boldsymbol{s}}(i_{t+1} = i_t \mid \mathcal{H}_t) = \mathbb{P}_{\boldsymbol{s}}(\overline{\mu}^t_{i_t} > \max_{j \neq i_t} \overline{\mu}^{t-1}_j \mid \mathcal{H}_t)\, \mathbb{P}_{\boldsymbol{s}}(i_t \in A_{t+1} \mid \mathcal{H}_t)
$$

is continuous in $\boldsymbol{s}$, where we used that $\overline{\mu}^t_j = \overline{\mu}^{t-1}_j$ for all $j \neq i_t$ independent of $\boldsymbol{s}$.[7] Then, since $\mathbb{P}_{\boldsymbol{s}}(i_{t+1} = i_t \mid i_t \notin A_t) = 0$, we have

$$
\mathbb{P}_{\boldsymbol{s}}(i_{t+1} = i_t \mid \mathcal{H}_t) = \mathbb{P}_{\boldsymbol{s}}(i_{t+1} = i_t \mid i_t \in A_t, \mathcal{H}_t)\, \mathbb{P}_{\boldsymbol{s}}(i_t \in A_t \mid \mathcal{H}_t),
$$

which shows the continuity of $\mathbb{P}_{\boldsymbol{s}}(i_{t+1} = i_t \mid i_t \in A_t, \mathcal{H}_t)$. Hence, in view of equation (4), we obtain that $\mathbb{P}_{\boldsymbol{s}}(i_{t+1} = i \mid \mathcal{H}_t)$ is continuous in $\boldsymbol{s}$. Finally,

[7] Note that $\overline{\mu}^{t-1}_i$ is $\mathcal{H}_t$-measurable for all i.

Lemma E.2.1 tells us that, since $\mathbb{P}_{\boldsymbol{s}}(\mathcal{H}_t)$ is assumed to be continuous, $\mathbb{P}_{\boldsymbol{s}}(i_{t+1} = i)$ is continuous as well. Hence, $\mathbb{P}_{\boldsymbol{s}}(\mathcal{H}_{t+1})$ is continuous and by induction we get that $\mathbb{P}_{\boldsymbol{s}}(\mathcal{H}_T)$ is continuous, which implies the continuity of $\mathbb{E}_{\boldsymbol{s}}[n_T(i)]$ in $\boldsymbol{s}$ for all i.

■

E.3 Proof of Theorem V.5.2

Proof of Theorem V.5.2. In the following let $\boldsymbol{\sigma} = (\sigma_1, \dots, \sigma_K) \in \text{NE(UCB-S)}$ and $\boldsymbol{s} \in \text{supp}(\boldsymbol{\sigma})$. We start of with some preliminaries. Recall that the arm i's utility function given algorithm UCB-S and strategies $\boldsymbol{s} = (s_i, s_{-i})$ can be expressed as

$$v_i(\text{UCB-S}, s_i, s_{-i}) = \mathbb{E}_{(\text{UCB-S}, s_i, s_{-i})}[n_T(i)]s_i,$$

and $v_i(\text{UCB-S}, s_i, \sigma_{-i}) = \mathbb{E}_{s_{-i} \sim \sigma_{-i}}[v_i(\text{UCB-S}, s_i, s_{-i})] = \mathbb{E}_{(s_i, \sigma_{-i})}[n_T(i)]s_i$. For convenience, we omit the argument UCB-S in the following, as every probability and expectation will be w.r.t. UCB-S. The following variables will prove useful. Let τ_i be the first round that arm i is not in the active set A_t anymore,

$$\tau_i := \min\{t \in [T] \colon i \notin A_t\},$$

and let τ be the first rounds in which A_t is empty,

$$\tau := \min\{t \in [T] \colon A_t = \emptyset\}.$$

Here, we introduce the convention that $\tau_i = T$ if $i \in A_T$ and $\tau = T$ if $A_T \neq \emptyset$.

To characterize the strategy profiles in the support of any Nash equilibrium under UCB-S, we are going to rely on the best response property of the Nash equilibrium. More precisely, for any $\boldsymbol{s} \in \text{supp}(\boldsymbol{\sigma})$ with $\boldsymbol{\sigma} \in \text{NE(UCB-S)}$ arm i's strategy, s_i, must be a best response to σ_{-i}, i.e., for all $s_i' \in [0, 1]$:

$$v_i(s_i, \sigma_{-i}) \geq v_i(s_i', \sigma_{-i}).$$

In a first step, we show that UCB-S incentivizes arms to choose strategies s_i at least as large as the desired strategy $s^*(\mu_i)$. While this seems obvious at first since each arm i's utility includes a linear factor of s_i, we notice that in the click-bandit model arms can prevent the principal from learning about their true post-click value μ by choosing low click-rates s. This could in theory be a viable strategy for suboptimal arms, i.e., $\mu_i < \mu^*$, since it would delay the principal from detecting that the arm is suboptimal. However, we quickly notice that delaying UCB-S from learning about $\mu_1, \dots, \mu_K$ is to each arm's disadvantage as any delay simply delays the round in which it receives utility. Moreover, while an arm may delay the learning of μ_i, UCB-S still improves its estimate of s_i and the threat of elimination becomes more imminent.

Lemma E.3.1. *For all $\boldsymbol{s} \in \text{supp}(\boldsymbol{\sigma})$ with $\boldsymbol{\sigma} \in \text{NE(UCB-S)}$ and all $i \in [K]$:*

$$s_i \geq s^*(\mu_i).$$

Proof. Let $\boldsymbol{\sigma} \in \mathrm{NE}(\text{UCB-S})$. We begin by making some fundamental observations about UCB-S in the click-bandit model. Let $t < T$. If $c_{t,i_t} = 0$ and $i_t \in A_t$, then $i_{t+1} = i_t$.[8] To see this, note that the estimates of $\mu_1, \dots, \mu_K$ and their confidence bounds do not change from t to $t+1$ if $c_{t,i_t} = 0$, since no post-click reward was observed for any of the arms. Hence, given that $i_t \in A_t$, we have

$$i_{t+1} = \operatorname*{argmax}_{i \in A_t} \overline{\mu}_i^t = \operatorname*{argmax}_{i \in A_{t-1}} \overline{\mu}_i^{t-1} = i_t.$$

Thus, given that i_t is not eliminated in the mean time, UCB-S plays arm i_t until arm i_t is clicked, i.e., until the arm receives utility 1, or we've reached round T. Hence, whenever $c_{t,i_t} = 0$, it simply delays the UCB selection rule by one round as the estimates and confidences of $\mu_1, \dots, \mu_K$ do not change. At the same time, arm i with $i = i_t$ still only receives utility 1 for this sequence of selections by UCB-S, since the UCB selection rule "progresses" once $c_{t,i_t} = 1$.

More formally, we can define the phases of the UCB selection rule recursively by $\eta_k := \min\{t > \eta_{k-1} \colon c_{t,i_t} = 1\}$ with $\eta_0 := 0$ and $\eta_k = \infty$ if round T is exceeded without a click. We define the number of such rounds as $N := \max\{k \colon \eta_k < \infty\}$ and remark that $N \leq T$ always.

We first note that conditional on $A_{\eta_{k-1}}$ the identity of i_{η_k} is independent of s_i (and σ_{-i}), but only depends on $\mu_1, \dots, \mu_K$ and their realization at rounds $\eta_1, \dots, \eta_{k-1}$, i.e., $\mathbb{P}_{(s_i,\sigma_{-i})}(i_{\eta_k} = i \mid A_{\eta_{k-1}}) = \mathbb{P}(i_{\eta_k} = i \mid A_{\eta_{k-1}})$. Moreover, we also see that $\mathbb{P}_{(s_i,\sigma_{-i})}(A_{\eta_k} = \cdot \mid i \in A_{\eta_k})$ is independent of s_i.[9] Then, since

$$\begin{aligned}
&\mathbb{P}_{(s_i,\sigma_{-i})}(i_{\eta_k} = i \mid i \in A_{\eta_{k-1}}) \\
&\qquad = \sum_A \mathbb{P}_{(s_i,\sigma_{-i})}(i_{\eta_k} = i \mid A_{\eta_{k-1}} = A \cup \{i\})\, \mathbb{P}(A_{\eta_{k-1}} = A \mid i \in A_{\eta_{k-1}}),
\end{aligned}$$

this implies that $\mathbb{P}_{(s_i,\sigma_{-i})}(i_{\eta_k} = i \mid i \in A_{\eta_{k-1}})$ is independent of s_i. Using the shown independence, let us then write

$$\begin{aligned}
\mathbb{P}_{(s_i,\sigma_{-i})}(i_{\eta_k} = i) = {}& \mathbb{P}(i_{\eta_k} = i \mid i \in A_{\eta_{k-1}})\, \mathbb{P}_{(s_i,\sigma_{-i})}(i \in A_{\eta_{k-1}}) \\
&+ \mathbb{P}_{(s_i,\sigma_{-i})}(i_{\eta_k} = i \mid i \notin A_{\eta_{k-1}})\, \mathbb{P}_{(s_i,\sigma_{-i})}(i \notin A_{\eta_{k-1}}).
\end{aligned} \tag{6}$$

Now, it holds that $\mathbb{P}(i_{\eta_k} = i \mid i \in A_{\eta_{k-1}}) \geq \mathbb{P}_{(s_i,\sigma_{-i})}(i_{\eta_k} = i \mid i \notin A_{\eta_{k-1}})$ always. Naturally, for $s_i < s^*(\mu_i)$ we have $\mathbb{P}_{(s_i,\sigma_{-i})}(i \in A_{\eta_{k-1}}) \leq \mathbb{P}_{(s^*(\mu_i),\sigma_{-i})}(i \in A_{\eta_{k-1}})$ so that from equation (6) it follows that

$$\mathbb{P}_{(s_i,\sigma_{-i})}(i_{\eta_k} = i) \leq \mathbb{P}_{(s^*(\mu_i),\sigma_{-i})}(i_{\eta_k} = i). \tag{7}$$

[8] W.l.o.g. we assume that there are no ties (ignoring the rounds where no post-click rewards have yet been observed). In fact, when there is a possibility of a tie, it can be seen that the arms have an even larger incentive to choose $s_i \geq s^*(\mu_i)$, since they are not guaranteed to be pulled again in the ensuing round when not clicked.

[9] However, note that the value of A_t for general t is not independent of s_i conditional on $i \in A_t$, since, e.g., for small s_i other arms will be played fewer times before round t, thereby reducing the probability of them being eliminated by round t.

We also see that as s_i decreases the number of utility-yielding rounds decreases in expectation, i.e., for $s_i < s^*(\mu_i)$:

$$\mathbb{E}_{(s_i,\sigma_{-i})}[N] < \mathbb{E}_{(s^*(\mu_i),\sigma_{-i})}[N] \tag{8}$$

since $\eta_k - \eta_{k-1} \sim \text{Geom}(s_{i_{\eta_k}})$. Finally, it follows from equations (7) and (8) and a technical lemma about the comparison of expectation under two measures (Lemma E.7.1 in Appendix E.7) that

$$\begin{aligned}\mathbb{E}_{(s_i,\sigma_{-i})}[m_T(i)] &= \mathbb{E}_{(s_i,\sigma_{-i})}\left[\sum_{k=1}^{N} \mathbb{1}_{\{i_{\eta_k}=i\}}\right] \\ &< \mathbb{E}_{(s^*(\mu_i),\sigma_{-i})}\left[\sum_{k=1}^{N} \mathbb{1}_{\{i_{\eta_k}=i\}}\right] = \mathbb{E}_{(s^*(\mu_i),\sigma_{-i})}[m_T(i)].\end{aligned}$$

Since a post-click reward is observed with probability s_i every time an arm is pulled by the learner, we have $\mathbb{E}_{(s_i,\sigma_{-i})}[m_t(i)] = \mathbb{E}_{(s_i,\sigma_{-i})}[n_t(i)]s_i$ so that $v_i(s_i,\sigma_{-i}) = \mathbb{E}_{(s_i,\sigma_{-i})}[m_t(i)]$. Now, from the above we see that for any $s_i < s^*(\mu_i)$, the strategy $s^*(\mu_i)$ is a strictly better response to σ_{-i} than s_i, i.e., $v_i(s^*(\mu_i),\sigma_{-i}) > v_i(s_i,\sigma_{-i})$. This shows that $s_i \geq s^*(\mu_i)$ for any $s_i \in \text{supp}(\sigma_i)$ with $\boldsymbol{\sigma} \in \text{NE(UCB-S)}$.

■

We continue the proof of Theorem V.5.2 by decomposing the number of times each arm is selected by UCB-S. Given (s_i,σ_{-i}) we can split $\mathbb{E}_{(s_i,\sigma_{-i})}[n_T(i)]$ into the time steps before τ_i and after τ, since arm i is never played in the rounds between τ_i and τ. Recall that UCB-S plays arms uniformly at random after round τ so that

$$\begin{aligned}\mathbb{E}_{(s_i,\sigma_{-i})}[n_T(i)] &= \mathbb{E}_{(s_i,\sigma_{-i})}\left[\sum_{t=1}^{\tau_i} \mathbb{1}_{\{i_t=i\}} + \sum_{t=\tau+1}^{T} \mathbb{1}_{\{i_t=i\}}\right] \\ &= \mathbb{E}_{(s_i,\sigma_{-i})}[n_{\tau_i}(i)] + \mathbb{E}_{(s_i,\sigma_{-i})}\Big[\frac{T-\tau}{K}\Big].\end{aligned} \tag{9}$$

The proof of Theorem V.5.2 proceeds by upper and lower bounding the quantities in (9), which will eventually lead to an approximate characterization of the best response s_i. More precisely, we establish the following bounds for $\boldsymbol{\sigma} \in \text{NE(UCB-S)}$ and $s_i \in \text{supp}(\sigma_i)$:

Lemma E.3.2: $\mathbb{E}_{(s_i,\sigma_{-i})}[n_{\tau_i}(i)] \leq \mathcal{O}\left(\frac{H^2 \log(T)}{s_i(s_i - s^*(\mu_i))^2}\right)$.

Lemma E.3.4: $T - \mathbb{E}_{(s_i,\sigma_{-i})}[\tau] \leq \mathcal{O}(1)$.

Lemma E.3.5: $\mathbb{E}_{(s_i,\sigma_{-i})}[n_T(i)] = \Omega\Big(\min\{\frac{\log(T)}{s_i\Delta_i^2}, s^*(\mu_i)\frac{T}{K}\}\Big)$.

E.3.1 Bounds on $n_T(i)$, $n_{\tau_i}(i)$, τ_i, and τ under UCB-S

We begin by bounding the number of allocations arm i receives before elimination. As one expects, UCB-S is able to detect that $s_i \neq s^*(\mu_i)$ with high probability after at most $\mathcal{O}(1/(s_i - s^*(\mu_i))^2)$ selections.

Lemma E.3.2. *Let $\boldsymbol{\sigma} \in \mathrm{NE}(\text{UCB-S})$ and $s_i \in \mathrm{supp}(\sigma_i)$ with $s_i \neq s^*(\mu_i)$. Then, the number of times that i is being selected before elimination, $n_{\tau_i)}(i)$, satisfies the following. For some constant $c_1 > 0$, it holds that*

$$\mathbb{P}_{(s_i,\sigma_{-i})}\left(n_{\tau_i}(i) \leq c_1 \frac{H^2 \log(T)}{s_i(s_i - s^*(\mu_i))^2}\right) \geq 1 - \frac{3}{T^2},$$

and as an immediate consequence for some $c_2 > 0$:

$$\mathbb{E}_{(s_i,\sigma_{-i})}[n_{\tau_i}(i)] \leq c_2 \frac{H^2 \log(T)}{s_i(s_i - s^*(\mu_i))^2}.$$

Proof. For simplicity, we consider w.l.o.g. the one-sided elimination rule checking whether the arm i's strategy s_i exceeds the desired strategy $s^*(\mu_i)$:

$$\underline{s}_i^t > \max_{\mu \in [\underline{\mu}_i^t, \overline{\mu}_i^t]} s^*(\mu). \tag{10}$$

Let $\alpha_t(i) = \sqrt{\frac{2\log(T)}{n_t(i)}}$ and $\beta_t(i) = \sqrt{\frac{2\log(T)}{m_t(i)}}$. Recall that $s^*(\mu)$ is H-Lipschitz. Then,

$$\max_{\mu \in [\underline{\mu}_i^t, \overline{\mu}_i^t]} s^*(\mu) \leq s^*(\hat{\mu}_i^t) + H\beta_t(i).$$

As a consequence, we see that a sufficient condition for the elimination rule (10) to trigger is given by

$$\hat{s}_i^t - \alpha_t(i) > s^*(\hat{\mu}_i^t) + H\beta_t(i), \tag{11}$$

where by definition $\underline{s}_i^t = \hat{s}_i^t - \alpha_t(i)$. The following statements are always w.r.t. (s_i, σ_{-i}), i.e., w.r.t. the probability measure $\mathbb{P}_{(s_i,\sigma_{-i})}$. From Hoeffding's inequality, we know that with probability at least $1 - 1/T^2$:

$$|\hat{s}_i^t - s_i| \leq \alpha_t(i).$$

Similarly, using the Lipschitzness of $s^*(\mu)$, Hoeffding's inequality implies that with probability at least $1 - 1/T^2$:

$$|s^*(\hat{\mu}_i^t) - s^*(\mu_i)| \leq H\,|\hat{\mu}_i^t - \mu_i| \leq H\beta_t(i).$$

It then follows that with probability at least $1 - 2/T^2$

$$\hat{s}_i^t - s^*(\hat{\mu}_i^t) \geq \big(s_i - s^*(\mu_i)\big) - \big(\alpha_t(i) + \beta_t(i)\big) \geq \big(s_i - s^*(\mu_i)\big) - (H+1)\beta_t(i),$$

where we used that $\alpha_t(i) = \sqrt{\frac{2\log(T)}{n_t(i)}} \leq \sqrt{\frac{2\log(T)}{m_t(i)}} = \beta_t(i)$, since $n_t(i) > m_t(i)$ by definition. Therefore, the sufficient condition in equation (11) is satisfied with probability $1 - 2/T^2$ for

$$s_i - s^*(\mu_i) > 2(H+1)\beta_t(i) = 2(H+1)\sqrt{\frac{2\log(T)}{m_t(i)}}.$$

In other words, arm i has been eliminated by round t with probability at least $1 - 2/T^2$ if

$$m_t(i) > \frac{16H^2\log(T)}{(s_i - s^*(\mu_i))^2}. \tag{12}$$

Lastly, we translate this to a statement about $n_t(i)$. Recall that conditional on $n_t(i)$, we have $\mathbb{E}[m_t(i) \mid n_t(i)] = n_t(i)s_i$, since arm i is clicked with probability s_i. From Hoeffding's inequality, we then again have with probability $1 - 1/T^2$

$$|m_t(i) - n_t(i)s_i| \leq \sqrt{2n_t(i)\log(T)}$$

and thus $m_t(i) \geq n_t(i)s_i - \sqrt{2n_t(i)\log(T)}$. Then, in view of equation (12), if

$$n_t(i) > c_1 \frac{H^2\log(T)}{s_i(s_i - s^*(\mu_i))^2}$$

for some sufficiently large $c_2 > 0$, then with probability at least $1 - 3/T^2$ arm i has been eliminated before round t. Since τ_i denotes the round in which i is eliminated from A_t, this means that with probability $1 - 3/T^2$:

$$n_{\tau_i}(i) \leq c_1 \frac{H^2\log(T)}{s_i(s_i - s^*(\mu_i))^2}.$$

Since by definition $\tau_i \leq T$, this implies that for some $c_2 > 0$:

$$\mathbb{E}_{(s_i,\sigma_{-i})}[n_{\tau_i}(i)] \leq c_2 \frac{H^2\log(T)}{s_i(s_i - s^*(\mu_i))^2}.$$

■

We briefly recall a standard result often used in the context of MABs, which states that any probably correct decision rule needs $\Omega(\frac{1}{\varepsilon^2})$ samples to distinguish between two hypotheses for which the Bernoulli means lie ε apart. We only give a short outline of the proof and refer to the many expositions of such bounds for more detail (see, e.g., Theorem 1 in [MT04], Section 2 in [Sli+19], Section 14 in [LS20]).

Lemma E.3.3. *In order for us to reuse our current notation, suppose that $K = 1$. In this case, $n_{\tau_1}(1)$ simply denotes the number of samples from arm 1 before it gets eliminated, i.e., UCB-S asserts that $s_1 \neq s^*(\mu_1)$. For $s_1 \neq s^*(\mu_1)$, it holds that*

$$\mathbb{E}_{s_1}[n_{\tau_1}(1)] \geq \Omega\left(\frac{\log(T)}{(s_1 - s^*(\mu_1))^2}\right).$$

Proof. W.l.o.g. we can assume that $r_{t,1} = \mu_1$ for all t so that we are only concerned with the estimation of the Bernoulli mean s_1 (this clearly only reduces the number of samples the elimination rule would need). Note that the elimination rule is correct with probability $1 - 1/T^2$ by construction of the confidence sets around s_1, i.e., only eliminates arm 1 if it in fact deviated from $s^*(\mu_1)$. We can then consider the hypotheses

$$H_0 : s_1 = s^*(\mu_1) \quad \text{and} \quad H_1 : s_1 = s^*(\mu_1) + \varepsilon.$$

Then, since the elimination rule is correct with probability $1-1/T^2$, the standard hypothesis testing argument (see, e.g., Theorem 1 in [MT04]) yields for some constant $c > 0$ that $\mathbb{E}_{s_1}[n_{\tau_1}(1)] \geq \frac{c\log(T)}{\varepsilon^2} = \frac{c\log(T)}{(s_1 - s^*(\mu_1))^2}$.

■

The next lemma states that $\mathbb{E}_{(s_i,\sigma_{-i})}[\tau]$ is close to T. The intuition of this is quickly explained. If the set A_t becomes empty, UCB-S plays arms uniformly at random. However, if one arm would happen to remain in A_t this arm would always be played (as it has no competition). To do so, an arm simply has to ensure that it does not get eliminated too early. Now, in view of Lemma E.3.3, an arm can be sampled order x more times without getting eliminated for moving its strategy order $\sqrt{x}$ closer to $s^*(\mu_i)$. Writing the arms' utility as $v_i(s_i, \sigma_{-i}) = \mathbb{E}_{(s_i,\sigma_{-i})}[n_T(i)]s_i = \mathbb{E}_{(s_i,\sigma_{-i})}[n_T(i)]\big(s^*(\mu_i) + (s_i - s^*(\mu_i))\big)$ we see that a quadratic increase in $\mathbb{E}_{(s_i,\sigma_{-i})}[n_T(i)]$ will dominate a linear decrease in $s_i - s^*(\mu_i)$.

Lemma E.3.4. *Let $\boldsymbol{\sigma} \in$ NE(UCB-S) and $s_i \in \operatorname{supp}(\sigma_i)$. Then,*

$$\mathbb{E}_{(s_i,\sigma_{-i})}[\tau] \geq T - \mathcal{O}(1).$$

Proof. Let $s^*(\mu_i) \leq s_i' < s_i$. Due to delays for smaller click-rates (see proof of Lemma E.3.1) and the fact that under s_i' the probability of arm i being eliminated at any given round is smaller than under s_i, it holds for all $j \neq i$ that

$$\mathbb{E}_{(s_i',\sigma_{-i})}[n_{\tau_j}(j)] \leq \mathbb{E}_{(s_i,\sigma_{-i})}[n_{\tau_j}(j)].$$

By definition of τ, we have $\mathbb{E}_{(s_i,\sigma_{-i})}[\tau] = \sum_{j\in[K]} \mathbb{E}_{(s_i,\sigma_{-i})}[n_{\tau_j}(j)]$ so that the above implies

$$\sum_{j\neq i} \mathbb{E}_{(s_i',\sigma_{-i})}[n_{\tau_j}(j)] \leq \mathbb{E}_{(s_i,\sigma_{-i})}[n_{\tau_j}(j)] = \mathbb{E}_{(s_i,\sigma_{-i})}[\tau] - \mathbb{E}_{(s_i,\sigma_{-i})}[n_{\tau_i}(i)].$$

In other words, under any strategy $s^*(\mu_i) \leq s_i' < s_i$, all arms $j \neq i$ will be eliminated after a total of $\mathbb{E}_{(s_i,\sigma_{-i})}[\tau] - \mathbb{E}_{(s_i,\sigma_{-i})}[n_{\tau_i}(i)]$ rounds so that there are at least $\mathbb{E}_{(s_i,\sigma_{-i})}[n_{\tau_i}(i)] + T - \mathbb{E}_{(s_i,\sigma_{-i})}[\tau]$ many "uncontested" rounds.

For convenience, let $N(s_i, \sigma_{-i}) = T - \mathbb{E}_{(s_i,\sigma_{-i})}[\tau]$, i.e., the expected number of rounds that A_t is empty and arms are being selected uniformly at random. Now, in view of Lemma E.3.3, there exists s_i' with

$$s_i' - s^*(\mu_i) \geq \Omega\left(\sqrt{\frac{\log(T)}{\mathbb{E}_{(s_i,\sigma_{-i})}[n_{\tau_i}(i)] + N(s_i, \sigma_{-i})}}\right)$$

such that $\mathbb{E}_{(s_i',\sigma_{-i})}[n_{\tau_i}(i)] \geq \mathbb{E}_{(s_i,\sigma_{-i})}[n_{\tau_i}(i)] + N(s_i,\sigma_{-i})$.

The proof proceeds by contradiction. To this end, suppose the contrary is true, namely, that $N(s_i,\sigma_{-i})$ is not constant, but in fact increasing in T, i.e., $N(s_i,\sigma_{-i}) = w(1)$. We then show that $v_i(s_i',\sigma_{-i}) > v_i(s_i,\sigma_{-i})$, which is a contradiction to s_i being a best response to σ_{-i}. From Lemma E.3.2 we know that

$$s_i - s^*(\mu_i) \leq \mathcal{O}\left(\sqrt{\frac{\log(T)}{s^*(\mu_i)\mathbb{E}_{(s_i,\sigma_{-i})}[n_{\tau_i}(i)]}}\right),$$

where we used that $s_i \geq s^*(\mu_i)$ by Lemma E.3.1. Using that $\mathbb{E}_{(s_i',\sigma_{-i})}[n_T(i)] \geq \mathbb{E}_{(s_i',\sigma_{-i})}[n_{\tau_i}(i)]$, we then obtain

$$\begin{aligned}
&v_i(s_i',\sigma_{-i}) \\
&= \mathbb{E}_{(s_i',\sigma_{-i})}[n_T(i)]s_i' \\
&\geq \mathbb{E}_{(s_i',\sigma_{-i})}[n_{\tau_i}(i)]\big(s^*(\mu_i) + (s_i' - s^*(\mu_i))\big) \\
&\geq \big(\mathbb{E}_{(s_i,\sigma_{-i})}[n_{\tau_i}(i)] + N(s_i,\sigma_{-i})\big)\left(s^*(\mu_i) + \Omega\left(\sqrt{\frac{\log(T)}{\mathbb{E}_{(s_i,\sigma_{-i})}[n_{\tau_i}(i)] + N(s_i,\sigma_{-i})}}\right)\right) \\
&\geq \big(\mathbb{E}_{(s_i,\sigma_{-i})}[n_{\tau_i}(i)] + N(s_i,\sigma_{-i})\big)s^*(\mu_i) \\
&\qquad\qquad + \Omega\left(\sqrt{\log(T)\big(\mathbb{E}_{(s_i,\sigma_{-i})}[n_{\tau_i}(i)] + N(s_i,\sigma_{-i})\big)}\right) \\
&> \mathbb{E}_{(s_i,\sigma_{-i})}[n_{\tau_i}(i)]s^*(\mu_i) + \frac{2N(s_i,\sigma_{-i})}{K}s^*(\mu_i) + \mathcal{O}\left(\sqrt{\log(T)\mathbb{E}_{(s_i,\sigma_{-i})}[n_{\tau_i}(i)]}\right) \\
&\geq \big(\mathbb{E}_{(s_i,\sigma_{-i})}[n_{\tau_i}(i)] + \frac{N(s_i,\sigma_{-i})}{K}\big)\big(s_i + (s_i - s^*(\mu_i))\big) \\
&\geq \mathbb{E}_{(s_i,\sigma_{-i})}[n_T(i)]s_i \\
&= v_i(s_i,\sigma_{-i}).
\end{aligned}$$

Hence, s_i' is a better response to σ_{-i} than s_i, which is a contradiction to $s_i \in \mathrm{supp}(\sigma_i)$.

■

The next lemma *lower bounds* $\mathbb{E}_{(s_i,\sigma_{-i})}[n_T(i)]$ for which we distinguish between optimal and suboptimal arms in terms of post-click rewards μ.

Lemma E.3.5. *Let* $\boldsymbol{\sigma} \in \mathrm{NE}(\text{UCB-S})$.

(i) *For all* $i^* \in [K]$ *with* $\Delta_{i^*} = 0$ *and* $s_{i^*} \in \mathrm{supp}(\sigma_{i^*})$*:*

$$\mathbb{E}_{(s_{i^*},\sigma_{-i^*})}[n_T(i^*)] \geq s^*(\mu_{i^*})\,\Omega\left(\frac{T}{K}\right).$$

(ii) *For all* $i \in [K]$ *with* $\Delta_i > 0$ *and* $s_i \in \mathrm{supp}(\sigma_i)$*:*

$$\mathbb{E}_{(s_i,\sigma_{-i})}[n_T(i)] \geq \Omega\left(\min\left\{\frac{\log(T)}{s_i\Delta_i^2}, s^*(\mu_i)\frac{T}{K}\right\}\right).$$

Proof. ***(i)***: Let $\boldsymbol{\sigma} \in \mathrm{NE}(\text{UCB-S})$ and let $i^* \in [K]$ such that $\Delta_{i^*} = 0$. Recall that when playing strategy $s^*(\mu_{i^*})$ arm i^* is eliminated with low probability so that

$$\mathbb{P}_{(s^*(\mu_{i^*}),\sigma_{-i^*})}(i^* \in A_T) \geq 1 - 1/T^2.$$

Now, given that i^* is not going to be eliminated, the UCB-type selection rule of UCB-S selects any arm i^* with maximal post-click reward $\mu_{i^*} = \mu^*$ at least $\Omega(T/K)$ times so that $\mathbb{E}_{(s^*(\mu_{i^*}),\sigma_{-i^*})}[n_T(i^*)] \geq \Omega(T/K)$. Then, since s_{i^*} has to be a best response to σ_{-i^*}, we obtain

$$\begin{aligned}
\mathbb{E}_{(s_{i^*},\sigma_{-i^*})}[n_T(i^*)] &\geq s_{i^*}\mathbb{E}_{(s_{i^*},\sigma_{-i^*})}[n_T(i^*)] \\
&= v_{i^*}(s_{i^*},\sigma_{-i^*}) \\
&\geq v_{i^*}(s^*(\mu_{i^*}),\sigma_{-i^*}) \\
&\geq s^*(\mu_{i^*})\mathbb{E}_{(s^*(\mu_{i^*}),\sigma_{-i^*})}[n_T(i^*)] \geq s^*(\mu_{i^*})\,\Omega\left(\frac{T}{K}\right).
\end{aligned}$$

(ii): Once again, we use the desired strategy $s^*(\mu_i)$ to infer properties of s_i. Let us be reminded that under $s^*(\mu_i)$ arm i is eliminated with low probability, i.e.,

$$\mathbb{P}_{(s^*(\mu_i),\sigma_{-i})}(i \in A_T) \geq 1 - 1/T^2$$

so that when studying $(s^*(\mu_i),\sigma_{-i})$ the potential elimination of arm i is negligible.

We will argue about $\mathbb{E}_{(s^*(\mu_i),\sigma_{-i})}[n_T(i)]$ via $\mathbb{E}_{(s^*(\mu_i),\sigma_{-i})}[m_T(i)]$. To isolate the rounds in which arms are clicked, i.e., post click-rewards are observed, we will re-use the rounds $\eta_1, \eta_2, \ldots$, which determine the phases of the UCB selection rule (introduced in Lemma E.3.1). On the rounds $\eta_1, \eta_2, \ldots$, the UCB-selection rule of line 4 is analogous to standard UCB in a MAB. We can then use well-known results from the instance-dependent lower bound analysis of the MAB problem. From Lemma 16.3 in [LS20] it then follows that for some constant $c_1 > 0$:[10]

$$\mathbb{E}_{(s^*(\mu_i),\sigma_{-i})}[m_T(i)] \geq \frac{\frac{1}{2}\log(T) + \log\left(\frac{c_1\Delta_i}{\sqrt{K}}\right)}{2\Delta_i^2}.$$

We see that this lower bound is only meaningful for sufficiently large Δ_i, as the numerator may become negative for $\Delta_i = \mathcal{O}\big(\sqrt{K/T}\big)$. For now let us assume that Δ_i is sufficiently large. Recall that $\mathbb{E}_{(s^*(\mu_i),\sigma_{-i})}[m_T(i)] =$

[10] We here used that the standard minimax bandit regret of UCB in MABs is bounded by $\tilde{\mathcal{O}}(\sqrt{KT})$.

$\mathbb{E}_{(s^*(\mu_i),\sigma_{-i})}[n_T(i)]s^*(\mu_i)$ as arm i is clicked with probability $s^*(\mu_i)$. Since s_i must be a best response to σ_{-i}, it must then hold that

$$\begin{aligned}\mathbb{E}_{(s^*(\mu_i),\sigma_{-i})}[n_T(i)]s_i &= v_i(s_i,\sigma_{-i})\\ &\geq v_i(s^*(\mu_i),\sigma_{-i})\\ &= \mathbb{E}_{(s^*(\mu_i),\sigma_{-i})}[m_T(i)] \geq c_2\frac{\log(T)}{\Delta_i^2}\end{aligned}$$

for some $c_2 > 0$. Solving for $\mathbb{E}_{(s^*(\mu_i),\sigma_{-i})}[n_T(i)]$ then yields

$$\mathbb{E}_{(s^*(\mu_i),\sigma_{-i})}[n_T(i)] \geq c_2\frac{\log(T)}{s_i\Delta_i^2}.$$

Next, for $\Delta_i \leq \mathcal{O}(\sqrt{K/T})$ it is well-known that the number of times UCB plays arm i is order at least $\Omega(T/K)$. We then have $\mathbb{E}_{s^*(\mu_i,\sigma_{-i})}[n_T(i)] = \Omega\big(\frac{T}{K}\big)$, so that

$$\begin{aligned}\mathbb{E}_{(s_i,\sigma_{-i})}[n_T(i)] &\geq s_i\mathbb{E}_{(s_i,\sigma_{-i})}[n_T(i)]\\ &= v_{i^*}(s_i,\sigma_{-i})\\ &\geq v_i(s^*(\mu_i),\sigma_{-i})\\ &\geq s^*(\mu_i)\mathbb{E}_{(s^*(\mu_i),\sigma_{-i})}[n_T(i)] \geq s^*(\mu_i)\,\Omega\left(\frac{T}{K}\right).\end{aligned}$$

■

E.3.2 Connecting the Bounds

Finally, using the lower and upper bound on $\mathbb{E}_{(s_i,\sigma_{-i})}[n_T(i)]$, we obtain the following approximate characterization of the strategies in the Nash equilibrium $\boldsymbol{\sigma} \in \mathrm{NE}(\text{UCB-S})$. For $i^* \in [K]$ with $\Delta_{i^*} = 0$, it follows from equation (9) and Lemma E.3.2, Lemma E.3.4, Lemma E.3.5 that

$$s^*(\mu_{i^*})\,\Omega\left(\frac{T}{K}\right) \leq \mathbb{E}_{(s_{i^*},\sigma_{-i^*})}[n_T(i^*)] \leq \mathcal{O}\left(\frac{H^2\log(T)}{s_{i^*}(s_{i^*}-s^*(\mu_{i^*}))^2}\right) + \mathcal{O}\left(\frac{1}{K}\right).$$

Solving for $s_{i^*} - s^*(\mu_{i^*})$, we obtain

$$s_{i^*}\big(s_{i^*} - s^*(\mu_{i^*})\big)^2 \leq \mathcal{O}\left(\frac{H^2K\log(T)}{T\,s^*(\mu_{i^*})}\right),$$

Finally, using that $s^*(\mu_{i^*}) \leq s_{i^*}$ by Lemma E.3.1 yields the claimed bound (note that $s^*(\mu)$ is bounded away from zero by assumption (A3))

$$s_{i^*} - s^*(\mu_{i^*}) \leq \mathcal{O}\left(H\sqrt{\frac{K\log(T)}{T\,s^*(\mu_{i^*})^2}}\right).$$

For $i \in [K]$ with $\Delta_i > 0$ suppose that $\frac{\log(T)}{s_i \Delta_i^2} \leq s^*(\mu_i)\frac{T}{K}$. Then, we have

$$\Omega\left(\frac{\log(T)}{s_i\Delta_i^2}\right) \leq \mathbb{E}_{(s_i,\sigma_{-i})}[n_T(i^*)] \leq \mathcal{O}\left(\frac{H^2\log(T)}{s_i(s_i - s^*(\mu_i))^2}\right) + \mathcal{O}\left(\frac{1}{K}\right),$$

which after solving for $s_i - s^*(\mu_i)$ yields

$$s_i - s^*(\mu_i) \leq \mathcal{O}\left(H\Delta_i\right).$$

For $i \in [K]$ with $\frac{\log(T)}{s_i\Delta_i^2} > s^*(\mu_i)\frac{T}{K}$, it follows, analogously to the case of $\Delta_i = 0$, from Lemma E.3.2, Lemma E.3.4, and Lemma E.3.5 that

$$s_i - s^*(\mu_i) \leq \mathcal{O}\left(H\sqrt{\frac{K\log(T)}{T\,s^*(\mu_i)^2}}\right).$$

■

E.4 Proof of Theorem V.5.3

Proof of Theorem V.5.3. Let $\boldsymbol{\sigma} \in \mathrm{NE}(\text{UCB-S})$ and let $i^* \in [K]$ be any arm with $\Delta_{i^*} = 0$. We begin with a standard regret decomposition into the number of times each arm is played and the rounds before i^* is eliminated. It holds that

$$\begin{aligned}
&R_T(\text{UCB-S}, \boldsymbol{\sigma}) \\
&= \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\sigma}}\left[\sum_{t=1}^{T} u(s^*,\mu^*) - u(s_{i_t},\mu_{i_t})\right] \\
&= \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\sigma}}\left[\sum_{t=1}^{\tau_{i^*}} u(s^*,\mu^*) - u(s_{i_t},\mu_{i_t})\right] + \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\sigma}}\left[\sum_{t=\tau_{i^*}+1}^{T} u(s^*,\mu^*) - u(s_{i_t},\mu_{i_t})\right] \\
&\leq \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\sigma}}\left[\sum_{i\in[K]} \mathbb{E}_{\boldsymbol{s}}[n_{\tau_{i^*}}(i)]\big(u(s^*,\mu^*) - u(s_i,\mu_i)\big)\right] + (T - \mathbb{E}_{\boldsymbol{\sigma}}[\tau_{i^*}]). \qquad (13)
\end{aligned}$$

From Lemma E.4.1 we know that $T - \mathbb{E}_{\boldsymbol{\sigma}}[\tau_{i^*}] \leq \sqrt{KT}$. We continue to split the arms into two cases. To this end, let $\Delta_i' := \sqrt{\frac{K\log(T)}{T\,s^*(\mu_i)^2}}$ and let $\Delta_*' = \sqrt{\frac{K\log(T)}{T\,s^*(\mu^*)^2}}$. For $i \in [K]$, we then distinguish between two cases: (a) $\Delta_i \leq \Delta_i'$ and (b) $\Delta_i > \Delta_i'$.

We begin with (a). Recall that $s^* := s^*(\mu^*)$. For the proof we will need one last technicality, namely, that $\Delta_i' \leq 2\Delta_*'$. We here assume that $s^*(\mu^*) > 2H\Delta_i'$.[11]

[11] Otherwise there is nothing to prove since the regret bound of Theorem V.5.3 is of order T.

Then, since $|s^*(\mu^*) - s^*(\mu_i)| \le H\Delta_i \le H\Delta'_i$, we get

$$\begin{aligned}\Delta'_i &= \frac{1}{s^*(\mu_i)}\sqrt{\frac{K\log(T)}{T}} \\ &\le \frac{1}{s^*(\mu^*) - H\Delta'_i}\sqrt{\frac{K\log(T)}{T}} \\ &\le \frac{2}{s^*(\mu^*)}\sqrt{\frac{K\log(T)}{T}} = 2\Delta'_*.\end{aligned}$$

We can now apply Theorem V.5.2 to obtain for any $\boldsymbol{s} \in \operatorname{supp}(\boldsymbol{\sigma})$ that

$$\begin{aligned}&\sum_{i:\Delta_i \le \Delta'_i} \mathbb{E}_{\boldsymbol{s}}[n_{\tau_{i^*}}(i)]\big(u(s^*,\mu^*) - u(s_i,\mu_i)\big) \\ &\le L \sum_{i:\Delta_i \le \Delta'_i} \mathbb{E}_{\boldsymbol{s}}[n_{\tau_{i^*}}(i)]\big(|s^*(\mu^*) - s_i| + |\mu^* - \mu_i|\big) \\ &\le L \sum_{i:\Delta_i \le \Delta'_i} \mathbb{E}_{\boldsymbol{s}}[n_{\tau_{i^*}}(i)]\left(|s^*(\mu^*) - s^*(\mu_i)| + \mathcal{O}\left(H\sqrt{\frac{K\log(T)}{T\,s^*(\mu_i)^2}}\right) + \Delta_i\right) \\ &\le L \sum_{i:\Delta_i \le \Delta'_i} \mathbb{E}_{\boldsymbol{s}}[n_{\tau_{i^*}}(i)]\left((H+1)\Delta_i + \mathcal{O}\left(H\sqrt{\frac{K\log(T)}{T\,s^*(\mu_i)^2}}\right) + \Delta_i\right) \\ &\le L(H+2) \sum_{i:\Delta_i \le \Delta'_i} \mathbb{E}_{\boldsymbol{s}}[n_{\tau_{i^*}}(i)]\,\Delta'_i \qquad (14) \\ &\le 2L(H+2)\Delta'_* \sum_{i:\Delta_i \le \Delta'_i} \mathbb{E}_{\boldsymbol{s}}[n_{\tau_{i^*}}(i)] \\ &\le LH \cdot \mathcal{O}\left(H\sqrt{\frac{K\log(T)}{T\,s^*(\mu^*)^2}}\right) \sum_{i:\Delta_i \le \Delta'_i} \mathbb{E}_{\boldsymbol{s}}[n_{\tau_{i^*}}(i)] \\ &\le \frac{LH}{s^*(\mu^*)}\,\mathcal{O}\left(\sqrt{KT\log(T)}\right),\end{aligned}$$

where we used that $\sum_{i:\Delta_i \le \Delta'_i} \mathbb{E}_{\boldsymbol{s}}[n_{\tau_{i^*}}(i)] \le T$ in the last line.

For taking care of the sum over arms satisfying (b), define the "good event" $\mathcal{E} = \{\underline{\mu}_i^t \le \mu_i \le \overline{\mu}_i^t \ \forall i \in [K] \ \forall t \in [T]\}$. We know that $\mathcal{E}$ occurs with probability at least $1 - 1/T^2$ for any $\boldsymbol{s} \in [0,1]^K$ by merit of Hoeffding's inequality. Under $\mathcal{E}$, we obtain from the standard UCB argument for all $t \le \tau_{i^*}$ that

$$\mu_{i_t} + 2\sqrt{\frac{2\log(T)}{m_t(i_t)}} \ge \overline{\mu}_{i_t}^t \ge \overline{\mu}_{i^*}^t \ge \mu_{i^*}.$$

This implies that $\Delta_i \le 2\sqrt{\frac{2\log(T)}{m_{\tau_{i^*}}(i)}}$. Hence, $i \in [K]$ with $\Delta_i > 0$ we get that $m_{\tau_{i^*}}(i) \le \frac{c\log(T)}{\Delta_i^2}$. Now, post-click rewards are observed for arm i with probability

s_i every time i is played by UCB-S, which tells us that $\mathbb{E}_{\boldsymbol{s}}[m_{\tau_{i^*}}(i)] = \mathbb{E}_{\boldsymbol{s}}[n_{\tau_{i^*}}(i)]s_i$. It follows from Theorem V.5.2 that

$$\begin{aligned}
&\sum_{i:\Delta_i>\Delta_i'} \mathbb{E}_{\boldsymbol{s}}[n_{\tau_{i^*}}(i)]\big(u(s^*,\mu^*) - u(s_i,\mu_i)\big) \\
&\leq \sum_{i:\Delta_i>\Delta_i'} \mathbb{E}_{\boldsymbol{s}}[n_{\tau_{i^*}}(i)]\,\frac{u(s^*,\mu^*) - u(s_i,\mu_i)}{s_i} \\
&\leq L \sum_{i:\Delta_i>\Delta_i'} \mathbb{E}_{\boldsymbol{s}}[n_{\tau_{i^*}}(i)]\,\frac{|s^*(\mu^*) - s_i| + |\mu^* - \mu_i|}{s_i} \\
&\leq L \sum_{i:\Delta_i>\Delta_i'} \mathbb{E}_{\boldsymbol{s}}[n_{\tau_{i^*}}(i)]\,\frac{|s^*(\mu^*) - s^*(\mu_i)| + \mathcal{O}(H\Delta_i) + \Delta_i}{s_i} \\
&\leq L \sum_{i:\Delta_i>\Delta_i'} c\log(T)\,\frac{H\Delta_i + \mathcal{O}(H\Delta_i) + \Delta_i}{s_i\Delta_i^2} \\
&\leq LH \sum_{i:\Delta_i>\Delta_i'} \mathcal{O}\left(\frac{\log(T)}{s_i\Delta_i}\right) \\
&\leq LH \sum_{i:\Delta_i>\Delta_i'} \mathcal{O}\left(\frac{\log(T)}{s^*(\mu_i)\Delta_i}\right),
\end{aligned}$$

where the last line used that $s_i \geq s^*(\mu_i)$ for all $s_i \in \operatorname{supp}(\sigma_i)$ shown in Lemma E.3.1. This completes the proof of Theorem V.5.3.

■

Lemma E.4.1. *Let $i^* \in [K]$ with $\Delta_{i^*} = 0$. For all $\delta > 0$:*

$$\mathbb{P}_{\boldsymbol{\sigma}}\Big(\tau_{i^*} > T - \frac{\sqrt{KT}}{1-\delta}\Big) > 1 - \delta.$$

Proof. Suppose the contrary is true, i.e., $\mathbb{P}_{\boldsymbol{\sigma}}\Big(\tau_{i^*} > T - \frac{\sqrt{KT}}{1-\delta}\Big) \leq \delta$. Since $\tau_{i^*} \leq T$ by definition, this implies that

$$\mathbb{E}_{\boldsymbol{\sigma}}[\tau_{i^*}] \leq \delta \cdot T + (1-\delta)\Big(T - \frac{\sqrt{KT}}{1-\delta}\Big) = T - \sqrt{KT}. \tag{15}$$

Now, let $s_{i^*} \in \operatorname{supp}(\sigma_{i^*})$ with $\mathbb{E}_{(s_{i^*},\sigma_{-i^*})}[\tau_{i^*}] \leq T - \sqrt{KT}$. Note that such s_{i^*} must exist for (15) to hold. We now show that there exists a strategy s'_{i^*} which is a better response to σ_{-i^*} than s_{i^*}. To this end, similarly to the proof of Lemma E.3.4, Lemma E.3.3 tells us that there exists $s'_{i^*} \in [0,1]$ with

$$s'_{i^*} - s^*(\mu_{i^*}) = \Omega\left(\sqrt{\frac{\log(T)}{\mathbb{E}_{(s_i,\sigma_{-i})}[n_{\tau_i}(i)] + \sqrt{KT}}}\right)$$

such that $\mathbb{E}_{(s'_{i^*},\sigma_{-i^*})}[\tau_{i^*}] = T - \mathcal{O}(1)$. Moreover, recall from Lemma E.3.4 that $T - \mathbb{E}_{(s_{i^*},\sigma_{-i^*})}[\tau] < \mathcal{O}(1)$. Then, using $\frac{x+\frac{y}{K}}{\sqrt{x+y}} \geq \sqrt{x+y} - \frac{y}{\sqrt{x+y}}$, equation (9), and Lemma E.3.2, we obtain

$$
\begin{aligned}
&v_{i^*}(s_{i^*}, \sigma_{-i^*}) \\
&\leq \mathbb{E}_{(s_{i^*},\sigma_{-i^*})}[n_{\tau_{i^*}}(i^*)] s_{i^*} + \mathcal{O}(1/K) \\
&\leq \mathbb{E}_{(s_{i^*},\sigma_{-i^*})}[n_{\tau_{i^*}}(i^*)] \big(s^*(\mu_{i^*}) + (s_{i^*} - s^*(\mu_{i^*}))\big) + \mathcal{O}(1/K) \\
&\leq \mathbb{E}_{(s_{i^*},\sigma_{-i^*})}[n_{\tau_{i^*}}(i^*)] \left(s^*(\mu_{i^*}) + \mathcal{O}\left(\sqrt{\frac{\log(T)}{\mathbb{E}_{(s_{i^*},\sigma_{-i^*})}[n_{\tau_{i^*}}(i^*)]}}\right)\right) \\
&\leq \mathbb{E}_{(s_{i^*},\sigma_{-i^*})}[n_{\tau_{i^*}}(i^*)] s^*(\mu_{i^*}) + \mathcal{O}\left(\sqrt{\log(T)\mathbb{E}_{(s_{i^*},\sigma_{-i^*})}[n_{\tau_{i^*}}(i^*)]}\right) \\
&\leq \mathbb{E}_{(s_{i^*},\sigma_{-i^*})}[n_{\tau_{i^*}}(i^*)] s^*(\mu_{i^*}) + \mathcal{O}\left(\sqrt{\log(T)(\mathbb{E}_{(s_{i^*},\sigma_{-i^*})}[n_{\tau_{i^*}}(i^*)] + \sqrt{KT})}\right) \\
&< \left(\mathbb{E}_{(s_{i^*},\sigma_{-i^*})}[n_{\tau_{i^*}}(i^*)] + \frac{\sqrt{KT}}{K}\right)\left(s^*(\mu_{i^*}) + \Omega\left(\sqrt{\frac{\log(T)}{\mathbb{E}_{(s_i,\sigma_{-i})}[n_{\tau_i}(i)] + \sqrt{KT}}}\right)\right) \\
&\leq \left(\mathbb{E}_{(s_{i^*},\sigma_{-i^*})}[n_{\tau_{i^*}}(i^*)] + \frac{\sqrt{KT}}{K}\right)\big(s^*(\mu_{i^*}) + (s'_{i^*} - s^*(\mu_{i^*}))\big) \\
&\leq \mathbb{E}_{(s'_{i^*},\sigma_{-i^*})}[n_T(i^*)] s'_{i^*} = v_{i^*}(s'_{i^*}, \sigma_{i^*}).
\end{aligned}
$$

Hence, $v_{i^*}(s_{i^*}, \sigma_{-i^*}) < v_{i^*}(s'_{i^*}, \sigma_{-i^*})$, a contradiction. ■

E.5 Proof of Corollary V.5.4

Proof of Corollary V.5.4. The argument roughly follows the standard way to translate an instance-dependent regret bound in multi-armed bandits to a minimax bound (see, e.g., [LS20]). However, the difference lies in that we split the arms not according to some fixed gap Δ', but according to the arm-specific gap

$$\Delta'_i := \sqrt{\frac{K \log(T)}{T\, s^*(\mu_i)^2}},$$

which we already used in the proof of Theorem V.5.3. This is necessary due to the guarantees of Theorem V.5.2 being gap-dependent.

We begin by recalling from equation (14) in the proof of Theorem V.5.3 that

$$
\begin{aligned}
&\sum_{i:\Delta_i \leq \Delta'_i} \mathbb{E}_{\boldsymbol{s}}[n_{\tau_{i^*}}(i)]\big(u(s^*, \mu^*) - u(s_i, \mu_i)\big) \\
&\qquad \leq L(H+2) \sum_{i:\Delta_i \leq \Delta'_i} \mathbb{E}_{\boldsymbol{s}}[n_{\tau_{i^*}}(i)]\, \Delta'_i \qquad (16)
\end{aligned}
$$

$$\leq \sum_{i:\Delta_i \leq \Delta_i'} \frac{LH}{s^*(\mu_i)} \mathcal{O}\left(\sqrt{KT\log(T)}\right),$$

where we coarsely upper bounded $\mathbb{E}_{\boldsymbol{s}}[n_{\tau_{i^*}}(i)] \leq T$.

For all arms i with $\Delta_i > \Delta_i'$, we also get similarly to the proof of Theorem V.5.3:

$$\begin{aligned}
&\sum_{i:\Delta_i > \Delta_i'} \mathbb{E}_{\boldsymbol{s}}[n_{\tau_{i^*}}(i)]\big(u(s^*,\mu^*) - u(s_i,\mu_i)\big) \\
&\qquad \leq LH \sum_{i:\Delta_i > \Delta_i'} \mathcal{O}\left(\frac{\log(T)}{s^*(\mu_i)\Delta_i'}\right) \\
&\qquad \leq LH \sum_{i:\Delta_i > \Delta_i'} \mathcal{O}\left(\sqrt{\frac{T\log(T)}{K}}\right) \qquad (17) \\
&\qquad \leq \sum_{i:\Delta_i > \Delta_i'} \frac{LH}{s^*(\mu_i)} \mathcal{O}\left(\sqrt{KT\log(T)}\right),
\end{aligned}$$

where we used a very coarse upper bound in the last line by simply adding a factor of $K/s^*(\mu_i)$. Note that the bound in the second last line is a much stronger bound than the one claimed in Corollary V.5.4. Combining these two bounds yields the first statement of the corollary.

Recall the definition of $s_{\min} := \min_{i\in[K]} s^*(\mu_i)$ and note that

$$\sqrt{\frac{K\log(T)}{T\, s_{\min}^2}} = \max_{i\in[K]} \Delta_i'. \qquad (18)$$

To get the more refined bound in Corollary V.5.4, we can continue from equation (16) and bound the right hand side via a maximum using (18) to get

$$L(H+2) \sum_{i:\Delta_i \leq \Delta_i'} \mathbb{E}_{\boldsymbol{s}}[n_{\tau_{i^*}}(i)]\Delta_i' \leq \frac{LH}{s_{\min}} \mathcal{O}\left(\sqrt{KT\log(T)}\right).$$

Lastly, note that in view of equation 17, we have

$$\begin{aligned}
&\sum_{i:\Delta_i > \Delta_i'} \mathbb{E}_{\boldsymbol{s}}[n_{\tau_{i^*}}(i)]\big(u(s^*,\mu^*) - u(s_i,\mu_i)\big) \\
&\leq LH\mathcal{O}\left(\sqrt{KT\log(T)}\right) \\
&\leq \frac{LH}{s_{\min}} \mathcal{O}\left(\sqrt{KT\log(T)}\right).
\end{aligned}$$

The corollary then follows from the regret decomposition in equation (13)

■

E.6 Proof of Theorem V.5.5

Proof of Theorem V.5.5. We work under the utility function $u(s,\mu) = s\mu$. In the strategic click-bandit model there are two distributions associated with each arm, the click distribution $P_{s_i} = \mathrm{Bern}(s_i)$ and the reward distribution P_{μ_i} with mean μ_i. We here assume that arm i's reward distribution is Bernoulli with mean $\mu_i \in [0,1]$. For convenience, w.l.o.g. we assume that the learner observes both, the click-event and the post-click reward every round. This clearly makes the learning problem easier for the learner. To summarise the distributions of arm i we let $P_{s_i,\mu_i} = P_{s_i} \times P_{\mu_i}$ denote the product distribution.

We consider problem instances

$$\boldsymbol{\mu} = \big(\frac{1}{2}, \dots, \frac{1}{2}, \frac{1}{2} + \Delta, \frac{1}{2}, \dots, \frac{1}{2}\big)$$

with $\mu_{i^*} = \frac{1}{2} + \Delta$. For convenience, we assume that M is index-independent, i.e., if arm i and arm j have identical distributions $P_{s_i,\mu_i} = P_{s_j,\mu_j}$, then $(n_T(i), n_T(j))$ and $(n_T(j), n_T(i))$ have the same distribution. If M is not index-independent, we can consider different indices i^* for the maximal element in $\boldsymbol{\mu}$. Let us choose $\Delta = c\sqrt{K/T}$ for some constant $c > 0$ to be chosen sufficiently small later.

Let us suppose that M is better than the claimed lower bound so that $R_T(M, \boldsymbol{s}, \boldsymbol{\mu}) \le o(\sqrt{KT})$ for some $\boldsymbol{s} \in \mathrm{NE}(M, \boldsymbol{\mu})$.[12] By choice of Δ in $\boldsymbol{\mu}$, it then directly follows that $\mathbb{E}_{\boldsymbol{s},\boldsymbol{\mu}}[n_T(i)] \le o\big(\frac{T}{K}\big)$ for all $i \neq i^*$, otherwise $R_T(M, \boldsymbol{s}, \boldsymbol{\mu}) \ge \Omega\big(\sqrt{KT}\big)$. Since $\sum_{i\in[K]} \mathbb{E}_{\boldsymbol{s},\boldsymbol{\mu}}[n_T(i)] = T$, this entails $\mathbb{E}_{\boldsymbol{s},\boldsymbol{\mu}}[n_T(i^*)] \ge \Omega\big(\frac{T}{K}\big)$.

We now show that $\mathbb{E}_{\boldsymbol{s},\boldsymbol{\mu}}[n_T(i)] = o\big(\frac{T}{K}\big)$ cannot hold when $\boldsymbol{s}$ is a Nash equilibrium. To this end, consider an alternative strategy s_i'. Now, let $s_i' = s_j$ with

$$j = \operatorname*{argmax}_{k\in[K]} \mathbb{E}_{(s_k, s_{-i})}[n_T(k)].$$

Generally, we would expect $j = i^*$, however, j could be any other index in $[K]$ (except for i as we see now). Since $\sum_{k\in[K]} \mathbb{E}_{\tilde{\boldsymbol{s}}}[n_T(k)] = T$ for any $\tilde{\boldsymbol{s}}$, we get that $\mathbb{E}_{(s_i', s_{-i}),\boldsymbol{\mu}}[n_T(j)] \ge \frac{T}{K}$. If $i = j$, this would be a contradiction to the statement that $\mathbb{E}_{(s_i, s_{-i}),\boldsymbol{\mu}}[n_T(i)] = o\big(\frac{T}{K}\big)$.

If $j \neq i^*$, we find that $\mathrm{KL}(P_{s_j,\mu_j}, P_{s_i',\mu_i}) = 0$, since i and j have identical click and reward distribution. More generally, we obtain from the chain rule that

$$\mathrm{KL}(P_{s_j,\mu_j}, P_{s_i',\mu_i}) = \mathrm{KL}(P_{s_j}, P_{s_i'}) + \mathrm{KL}(P_{\mu_j}, P_{\mu_i}) = \mathrm{KL}(P_{\mu_j}, P_{\mu_i}) \le 8\Delta^2,$$

where we used that $\mathrm{KL}(P_{\mu_j}, P_{\mu_i}) \le \mathrm{KL}(P_{\mu_{i^*}}, P_{\mu_i}) = \mathrm{KL}\big(\mathrm{Bern}(\frac{1}{2}), \mathrm{Bern}(\frac{1}{2} + \Delta)\big) \le 8\Delta^2$ (see, e.g., Theorem 2.4 in [Sli+19]). Recall that $\Delta = c\sqrt{K/T}$. For sufficiently small constant $c > 0$, Theorem 3 in [GMS19] then yields that either

$$\mathbb{E}_{(s_i', s_{-i}),\boldsymbol{\mu}}[n_T(i)] \ge \frac{T}{K} \quad \text{or} \quad \mathbb{E}_{(s_i', s_{-i}),\boldsymbol{\mu}}\left[\frac{n_T(i)}{n_T(j)}\right] \ge \frac{1}{2}. \tag{19}$$

[12] We consider pure strategy NE here, though, mixed strategies can be handled analogously.

Assuming $n_T(j) \geq 1$, using some algebra (Lemma E.7.2), the latter can be seen to imply that

$$\mathbb{E}_{(s_i', s_{-i}), \boldsymbol{\mu}}[n_T(i)] \geq \frac{1}{2}\mathbb{E}_{(s_i', s_{-i}), \boldsymbol{\mu}}[n_T(j)] \geq \frac{T}{2K},$$

where the last inequality holds due to the choice of j. Hence, from equation 19 we obtain that

$$\mathbb{E}_{(s_i', s_{-i}), \boldsymbol{\mu}}[n_T(i)] \geq \frac{T}{2K}.$$

This leads to a contradiction, as s_i' is a better response to s_{-i} than s_i. We have thus shown that $R_T(M, s, \boldsymbol{\mu}) = \Omega\big(\sqrt{KT}\big)$ for any $\boldsymbol{s} \in \mathrm{NE}(M, \boldsymbol{\mu})$.

■

E.7 Technical Lemmas

Lemma E.7.1. *Let $\mathbb{P}$ and $\tilde{\mathbb{P}}$ be two probability measure (and let $\mathbb{E}$ and $\tilde{\mathbb{E}}$ denote the respective expectations). Suppose that for integer-valued random variables $N, X_1, X_2, \ldots$, it holds for all $k \in \mathbb{N}$ and some $i \in \mathbb{N}$:*

$$\mathbb{E}[N] < \tilde{\mathbb{E}}[N] \quad \textit{and} \quad 0 < \mathbb{P}(X_k = i \mid N) \leq \tilde{\mathbb{P}}(X_k = i \mid N) \;\; a.s. \tag{20}$$

Then,

$$\mathbb{E}\left[\sum_{k=1}^{N} \mathbb{1}_{\{X_k = i\}}\right] < \tilde{\mathbb{E}}\left[\sum_{k=1}^{N} \mathbb{1}_{\{X_k = i\}}\right]. \tag{21}$$

Proof. Note that if N and $X_1, X_2, \ldots$ were independent and $X_1, X_2, \ldots$ i.i.d. this would immediately follow from Wald's lemma.

We prove the lemma via factorization. It holds that

$$\begin{aligned}
\mathbb{E}\left[\sum_{k=1}^{N} \mathbb{1}_{\{X_k = i\}}\right] &= \sum_{n=1}^{\infty} \mathbb{E}\left[\sum_{k=1}^{n} \mathbb{1}_{\{X_k = i\}} \mid N = n\right] \mathbb{P}(N = n) \\
&= \sum_{n=1}^{\infty} \sum_{k=1}^{n} \mathbb{P}(X_k = i \mid N = n)\, \mathbb{P}(N = n) \\
&\leq \sum_{n=1}^{\infty} \sum_{k=1}^{n} \tilde{\mathbb{P}}(X_k = i \mid N = n)\, \mathbb{P}(N = n) \\
&= \mathbb{E}\left[\sum_{k=1}^{N} \tilde{\mathbb{P}}(X_k = i \mid N)\right] \\
&< \tilde{\mathbb{E}}\left[\sum_{k=1}^{N} \tilde{\mathbb{P}}(X_k = i \mid N)\right] = \tilde{\mathbb{E}}\left[\sum_{k=1}^{N} \mathbb{1}_{\{X_k = i\}}\right],
\end{aligned}$$

where in the last line we used that $\tilde{\mathbb{P}}(X_k = i \mid N) > 0$ almost surely. ■

Lemma E.7.2. *Let X and Y be two random variables (which are not necessarily independent) and $Y \geq 1$. Suppose that*

$$\mathbb{E}\left[\frac{X}{Y}\right] \geq \frac{1}{2}.$$

Then,

$$\mathbb{E}[X] \geq \frac{\mathbb{E}[Y]}{2}.$$

Proof. Basic algebra yields that

$$\mathbb{E}\left[\frac{2X}{Y}\right] - 1 = \mathbb{E}\left[\frac{2X}{Y}\right] - \mathbb{E}\left[\frac{Y}{Y}\right] = \mathbb{E}\left[\frac{2X - Y}{Y}\right] \leq \mathbb{E}\left[2X - Y\right].$$

Hence, if $\mathbb{E}\left[\frac{2X}{Y}\right] \geq 1$, it follows that $\mathbb{E}[2X] \geq \mathbb{E}[Y]$.

■

E.8 More Related Work

In other related work, [GH13; Hro+22; Hu+23; LH18] study incentive design in online recommendation and are interested in incentivizing agents to contribute high-quality content. They differ to our work primarily in that either the strategies are directly observable, or no bandit learning together with incentive design is performed simultaneously. There is also a multitude of additional work on auction-based mechanism design with unknown agent values and bandit feedback [GLT12; Kan+23; Naz+16, e.g.]. Similar to the previously discussed auction design in MABs [BKS15; BSS09; DK09], [Gao+21] study an auction-based combinatorial multi-armed bandit with payments, where each arm can misreport the cost for its selection. Other related areas of research are dynamic mechanism design [BV19; PST14] as well as online mechanism design [Par07].

E.9 Future Work

A natural extension to the studied setting would be to assume that CTRs are user-dependent or more generally dependent on contextual information. Another direction would be to consider multi-slot recommendations in which the learner selects a subset of arms every round and the selected arms compete for the click (and our observations are therefore relative). In fact, the case where the learner selects a set of arms and each arm i is clicked with probability s_i independently of the other arms can be handled with exactly the same methods as presented in this paper. More generally, we believe that the idea of introducing a screening rule based on confidences of each arm's strategy can be extended to various settings and many of our techniques reused.