Learning when Objectives are Hard to Specify

Kush Bhatia

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2022-172
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2022/EECS-2022-172.html

June 22, 2022

Learning when Objectives are Hard to Specify

by

Kush Bhatia

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Peter L. Bartlett, Co-chair
Professor Anca D. Dragan, Co-chair
Professor William Fithian
Professor Jacob Steinhardt

Summer 2022

Learning when Objectives are Hard to Specify

Abstract

Learning when Objectives are Hard to Specify

by

Kush Bhatia

Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor Peter L. Bartlett, Co-chair

Professor Anca D. Dragan, Co-chair

Deploying learning systems in the real-world requires aligning their objectives with those of the humans they interact with. Existing algorithmic approaches for this alignment try to infer these objectives through human feedback. The correctness of these algorithms crucially depends on several simplifying assumptions on 1) how humans represent these objectives, 2) how humans respond to queries given these objectives, and 3) how well the hypothesis space represents these objectives. In this thesis, we question the robustness of existing approaches to misspecifications in these assumptions and develop principled approaches to overcome such misspecifications.

We begin by studying misspecifications in the hypothesis class assumed by the learner and propose an agnostic learning setup where we demonstrate that all existing approaches based on learning from comparisons would incur constant regret. We further show that it is necessary for humans to provide more detailed feedback in the form of higher-order comparisons and obtain sharp bounds on the regret as a function of the order of comparisons. Next, we focus on misspecifications in human behavioral models and establish, through both theoretical and empirical analyses, that inverse RL methods can be extremely brittle in worst case. However, under reasonable assumptions, we exhibit that these methods do exhibit robustness and are able to recover underlying reward functions up to a small error term. We then proceed to study misspecifications in assumptions on how humans represent objective functions. We begin by showing that taking a uni-criterion approach to modeling human preferences fails to capture real-world human objectives and propose a new multi-criteria comparison based framework which overcome these limitations. In the next part, we shift our focus to hand-specified reward functions in reinforcement learning, an alternative to learning rewards from humans. We empirically study the effects of such misspecifications showing that over-optimizing such proxy rewards can hurt performance in the long run.

To Mummy, Sahil, and Neha.

Contents

Contents ii

List of Figures v

List of Tables ix

1 Introduction 1

I Reward learning with misspecified models 4

2 Agnostic learning with unknown utilities 5
2.1 Introduction . . . 5
2.2 Problem formulation . . . 10
2.3 Main results . . . 11
2.4 Binary decision-making with k-comparisons . . . 13
2.5 Instance-optimal guarantees for binary prediction . . . 24

3 Learning with misspecified human models 28
3.1 Introduction . . . 28
3.2 Problem formulation . . . 30
3.3 Worst-case instability of inference . . . 32
3.4 Stability under log-concavity . . . 33
3.5 Empirical Analysis . . . 35
3.6 Discussion . . . 40

4 Learning with multi-criteria preferences 42
4.1 Introduction . . . 42
4.2 Framework for preference learning along multiple criteria . . . 45
4.3 Statistical guarantees and computational approaches . . . 49
4.4 Autonomous driving user study . . . 54
4.5 Discussion and future work . . . 56

5 Reinforcement learning with misspecified rewards **57**
5.1 Introduction . . . 57
5.2 Experimental setup: Environments and reward functions . . . 59
5.3 How Agent Optimization Power Drives Misalignment . . . 62
5.4 Polynomaly: Mitigating reward misspecification . . . 67
5.5 Discussion . . . 68

6 Reward learning as doubly nonparametric bandits **70**
6.1 Introduction . . . 70
6.2 Framework: Doubly nonparametric Bandits . . . 72
6.3 Algorithm: Policy Learning via Reward Learning . . . 73
6.4 Query selection and statistical guarantees . . . 75
6.5 Bounds for kernel multi-armed bandits . . . 79

Bibliography **82**

II Appendices **98**

A Deferred content from Chapter 2 **99**
A.1 Deferred proofs from Section 2.4 . . . 99
A.2 Deferred proofs from Section 2.5 . . . 102

B Deferred content from Chapter 3 **106**
B.1 Proofs . . . 106
B.2 Experiment Details . . . 111

C Deferred content from Chapter 4 **114**
C.1 Blackwell's approachability . . . 114
C.2 Proof of main results . . . 116
C.3 Local asymptotic analysis for plug-in estimator . . . 126
C.4 Additional results and their proofs . . . 135
C.5 Details of user study . . . 140

D Deferred content from Chapter 5 **147**
D.1 Mapping The Effects of Reward Misspecification . . . 147
D.2 Polynomaly . . . 149

E Deferred content from Chapter 6 **152**
E.1 Technical details for proposed framework . . . 152
E.2 Proof of main results . . . 153
E.3 Gaussian process bandit optimization . . . 159
E.4 Adaptive sampling via GP-UCB . . . 167

E.5 Further details on experimental evaluation 169

List of Figures

1.1 Thesis overview. We look at different forms of misspecifications that arise when learning systems optimize unknown human objectives. Such misspecifications arise in the way the objectives are assumed to be represented, in assumptions on how humans provide feedback given these objectives, and in the representation ability of these learning systems. This work focuses on understanding the side-effects of such misspecifications and presents algorithmic approaches to overcome them. 2

2.1 Consider a binary decision-task with decisions G(reen) and B(lue). The instance space comprises of three equiprobable clusters of datapoints x_1, x_2 and x_3, and have associated utilities u^* for decisions B and G. The colour of the datapoints represents the decision with higher utility. The function class $\mathcal{F}$ consists of linear predictors. In the traditional learning setups where the dataset consists of pairs (x, y), no learner will have enough information to select between f_1 and f_2 since the $0-1$ error for both is $1/3$. In contrast, using a 2-comparison oracle, a learner can ask a query of the form "Which of $u^*(x_1, \mathrm{G}) + u^*(x_3, \mathrm{B})$ or $u^*(x_1, \mathrm{B}) + u^*(x_3, \mathrm{G})$ is bigger?". This allows them to infer that correctly predicting x_3 gives a higher overall utility and output the optimal decision function f_2. 6

3.1 Simple navigation environment where a near optimal policy violates Assumption 3.2. 33

3.2 Effect of transition error (measured as the degree of underestimation of unintended transitions) on (a) weighted policy divergence and (b) reward inference error on Gridworld environments. In (c), we show a scatter plot of the policy and reward errors for each P^*. 36

3.3 Effect of underestimating the discount factor on (a) weighted policy divergence and (b) reward inference error on Gridworld environments. In (c), we show a scatter plot of the policy and reward errors for each γ^*. 36

3.4 Gridworld environments. 36

3.5 Effect of transition error (measured as error in p) on (a) weighted policy divergence and (b) reward inference error on continuous Lunar Lander environments. In (c), we show a scatter plot of the policy and reward errors for fixed α. 37

3.6 Effect of amount of training on (a) weighted policy divergence and (b) reward inference error on continuous Lunar Lander environments. In (c), we show a scatter plot of the policy and reward errors for fixed number of training iterations. 38

3.7 Lunar Lander environments. 38
3.8 Effect of modeling human bias (measured by probability of acting according to human policy) on (a) weighted policy divergence and (b) reward inference error on discrete Lunar Lander environments. In (c), we show a scatter plot of the policy and reward errors for fixed probabilities. We see that more accurate human models correspond to lower reward inference error. 39
3.9 Visualization of trajectories under the human policy. 40

4.1 (a) Policy A focuses on optimizing comfort and policy B on speed, and these are compared pairwise in different environments. (b) Preference matrices, where entry (i, j) of the matrix contains the proportion of comparisons between the pair (i, j) that are won by object i. (The diagonals are set to half by convention). The overall pairwise comparisons are given by the matrix $\mathbf{P}_{\mathsf{ex}}^{\mathsf{Overall}}$, and preferences along each of the criteria by matrices $\mathbf{P}_{\mathsf{ex}}^{\mathsf{Comfort}}$ and $\mathbf{P}_{\mathsf{ex}}^{\mathsf{Speed}}$. Policy R is a randomized policy $1/2$ A $+1/2$ B. While the preference matrices satisfy the linearity assumption individually along speed and comfort, the assumption is violated overall, wherein R is preferred over both A and B. 43
4.2 Two target sets S_1 and S_2 for our example from Figure 4.1 that capture trade-offs between comfort and speed. Set S_1 requires feasible score vectors to satisfy 40% of the population along both comfort and speed. Set S_2 requires both scores to be greater than 0.3 but with a linear trade-off: the combined score must be at least 0.9. 47

5.1 An example of reward hacking when cars merge onto a highway. A human-driver model controls the grey cars and an RL policy controls the red car. The RL agent observes positions and velocities of nearby cars (including itself) and adjusts its acceleration to maximize the proxy reward. At first glance, both the proxy reward and true reward appear to incentivize fast traffic flow. However, smaller policy models allow the red car to merge, whereas larger policy models exploit the misspecification by stopping the red car. When the red car stops merging, the mean velocity increases (merging slows down the more numerous grey cars). However, the mean commute time also increases (the red car is stuck). This exemplifies a *phase transition*: the qualitative behavior of the agent shifts as the model size increases. 58
5.2 Increasing the RL policy's model size decreases true reward on three selected environments. The red line indicates a phase transition. 62
5.3 In addition to parameter count, we consider three other agent capabilities: training steps, action space resolution, and observation noise. In Figure 5.3a, an increase in the proxy reward comes at the cost of the true reward. In Figure 5.3b, increasing the granularity (from right to left) causes the agent to achieve similar proxy reward but lower true reward. In Figure 5.3c, increasing the fidelity of observations (by increasing the random testing rate in the population) tends to decrease the true reward with no clear impact on proxy reward. 63

5.4 The larger model prevents the AVs (in red) from moving to increase the velocity of the human cars (unobserved cars in white and observed cars in blue). However, this greatly increases the average commute per person. 64
5.5 For COVID, ICU usage is a proxy for public health and regulation stage is a proxy for economic health. The blue line indicates the maximum stage (right) enforced by the larger policy and the corresponding ICU level (left) at that stage. The red line is the equivalent for the smaller policy. Because the larger policy enforces regulations much sooner than the smaller policy, it maintains both low ICU usage and low regulation stage. However, the larger policy is politically unfavorable: regulations are high even though public signs of infection, such as ICU usage, are low. 65
5.6 Correlations between the proxy and true rewards, along with the reward hacking induced. In Figure 5.6a, we plot the proxy reward with "•" and the true reward with "×". In Figure 5.6b, we plot the trained checkpoint correlation and the early checkpoint correlation. 66

6.1 (a) Corroborating upper bound from Corollary 6.1. Our theoretical bounds predict a rate of $n^{-0.27}$ and the experiment shows an almost matching rate of $n^{-0.28}$. (b) As the dimension d is increased, the excess risk curves asymptote at different levels for different n. This shows that our algorithm achieves non-vacuous error for the doubly-nonparametric set in the regime $d \to \infty$. 81

B.1 Visualization of gridworld policies (as state-visitation distributions) with (a) different transition biases (probability p of unintended transitions), and (b) different discount factors. 111
B.2 Visualization of Lunar Lander trajectories for policies with (a) biased internal dynamics that underestimate left-right acceleration and (b) correct internal dynamics. 112
B.3 Effect of human bias (measured by probability of acting according to human policy) on (a) weighted policy divergence and (b) reward inference error on discrete Lunar Lander environments. In (c), we show a scatter plot of the policy and reward errors for fixed probabilities. 113

C.1 Instructions provided to the users before the experiment began. The users were asked to compare behavior of policies and were told to expect some policies to exhibit a randomized behavior. 144
C.2 Layout of the experiment where each panel shows a GIF exhibiting a Policy controlling the autonomous vehicle in one of the worlds of the environment. The users were instructed to compare behaviors across each of the columns before proceeding to answer the questions. 145
C.3 Layout of the questions panel comprising the 6 comparison questions and the form for reporting the relevance of each criterion in the overall evaluation. . . . 146

D.1 Additional model size scatter plots. Observe that not all misspecifications cause misalignment. We plot the proxy reward with "•" and the true reward with "×". The proxy reward is measured on the left-hand side of each figure and the true reward is measured on the right hand side of each figure. 147
D.2 Correlations between the proxy and true rewards, along with the reward hacking induced. In the left column, we plot the proxy reward with "•" and the true reward with "×". In the right column, we plot the trained checkpoint correlation and the randomly initialized checkpoint correlation. 148
D.3 ROC curves for Traffic-Mer - misweighting. 150
D.4 ROC curves for Traffic-Mer - scope. 150
D.5 ROC curves for Traffic-Mer - ontological. 150
D.6 ROC curves for Traffic-Bot - misweighting. 151
D.7 ROC curves for COVID - ontological. 151

List of Tables

5.1 Reward misspecifications across our four environments. 'Misalign' indicates whether the true reward drops and 'Transition' indicates whether this corresponds to a phase transition (sharp qualitative change). We observe 5 instances of misalignment and 4 instances of phase transitions. 'Mis.' is a misweighting and 'Ont.' is an ontological misspecification. 61

5.2 Performance of detectors on different subtasks. Each detector has at least one subtask with AUROC under 60%, indicating poor performance. 68

6.1 Our algorithm specializes to the case of kernel multi-armed bandits and yields strong bounds. For a d-dimensional Matérn kernel with smoothness ν, we outperform both GP-UCB and GP-TS unless $\nu \gtrsim d^2$. The only works to achieve better bounds for small ν are π-GP UCB, which was designed specifically for the Matérn kernel and a recent analysis of the SupKernelUCB which achieves near minimax rates. 79

C.1 Pairwise comparison between policies from user study 143

D.1 Benchmark statistics. We average over 5 rollouts in traffic and 32 rollouts in COVID. 149

Acknowledgments

These past six years of my Ph.D. have been some of the most enjoyable and enriching times of my life that have been only made possible by the amazing set of people around me. I will fondly remember my experiences at Berkeley which have not only made me a better researcher but have also allowed me to grow as an individual.

I must begin by thanking my advisors, Peter Bartlett and Anca Dragan, without whom this journey would have been impossible. I am grateful for teaching me about research, for their constant support and motivation, and for believing in me.

Throughout my time at Berkeley, Peter has been a calming influence for me, and has always kept me on the right path whenever I have deviated from it. He has always been around to push me towards a positive mindset and After every conversation with him, I would feel excited and motivated to think about research (and life in general) in the right way. I have been amazed by his technical insights and clarity during our research discussions – during most of our meetings, he would end up clarifying my thoughts and explaining what I had wanted to tell him. He created a very open and friendly environment during the group meetings which resulted in numerous friendships and collaborations.

I consider myself extremely fortunate to be one of Anca's first students. I was pleasantly surprised when I received an email from her when I had almost no exposure to human-robot interaction. I am glad that she took a bet on me and that I jumped on this opportunity; what followed was six years of a wonderful collaboration wherein she taught me all about interactions, humans and reward functions. Being (one of) her first student came with its own set of perks - she spent so much time and effort on making a better researcher out of me and pushing me to think out of my comfort zone. I am very thankful for this! I have been amazed by her enthusiasm and eagerness to learn, and her ability to always ask the right question. I will always cherish the wonderful conversations on fashion (how not to wear sandals), coffee, bread making, pandemic stuff, and more recently on human babies. I really appreciate her being available to listen to me throughout the PhD and hope to always have her on my speed dial!

In addition to my advisors, I have been fortunate to have found amazing mentors who have very generously guided me and helped me grow. I thank Jacob Steinhardt for welcoming me to his group and working with me on several projects which ended up being a crucial part of this thesis. I really enjoyed our intellectual conversations and learned how to transform vague ideas into concrete research problems. I really admire his openness to feedback and incorporating that to change himself, and I will always strive to incorporate in my own life.

Thanks to Mike Jordan and Will Fithian for serving on my various committees and providing valuable feedback which helped shape my research interests.

I am thankful to Karthik Sridharan for hosting me for an internship at Cornell in the wonderful summer of 2019. I am always amazed at his ability to explain the most complex mathematical concepts in very simple terms. I fondly remember the fun times from that summer, especially the amazing home-cooked food, the stories you shared, my laugh riot session about the TRON algorithms, and his independence day speech. Thanks a lot for

always being available to hear out my crazy ideas and being supportive of them; I will always keep pinging you on google chat!

From my undergraduate days, I would like to thank my mentor Manik Varma who was responsible for introducing me to the world of machine learning research. He has provided me valuable guidance and advice at every stage of my career and I consider myself very lucky to still be able to fallback on him whenever I am in doubt. I was overjoyed when he decided to come to Berkeley for his sabbatical and the time spent with him, Sachi, Mihir and Meha greatly added to my experience at Berkeley. I am grateful for him for providing me an opportunity to pursue research at MSR Bangalore as a research fellow.

At MSR Bangalore, I was fortunate to work with Prateek Jain who introduced me to the world of theoretical ML research. His principled way of looking at ML problems stuck with me and played a big part in encouraging me to pursue a PhD. I am thankful to Purushottam Kar for being an amazing collaborator and spending numerous hours debugging and explaining things to me. While at MSR, I found a great friend and mentor in Deeparnab Chakrabarty who pushed me to think big and showed me how to appreciate the elegance in simple things.

I have been fortunate to have collaborated with and learned a lot from phenomenal people during the course of my PhD: Yasin Abassi-Yadkori, Pieter Abbeel, Pranjal Awasthi, Nicolas Flammarion, Joey Hong, Sandy Huang, Michael Jordan, Sreenivas Gollapudi, Wenshuo Guo, Koulik Khamaru, Kostas Kollias, Ashish Kumar, Aditya Kusupati, Nevena Lazic, Yi-An Ma, Dhruv Malik, Aldo Pacchiano, Alexander Pan, Ashwin Pananjady, Pradeep Ravikumar, Manish Singh, Arun Suggala, Csaba Szepesvari, Martin Wainwright, and Gellert Weisz. Amongst these amazing people, I would like specially highlight the postdocs Nicolas and Yi-An for mentoring me at every stage during my first projects, and Ashwin Pananjady who introduced me to the field of statistical preference learning and helped me pivot my research. Also, thanks to Alex Pan and Joey Hong for trusting in me and allowing me to mentor them in some aspect.

I really enjoyed being a part of the vibrant research community at Berkeley, especially the Stat-learning group, InterACT lab, Steinhardt group, BAIR, BLISS and SAIL labs. I would like to thank Niladri Chatterji, Xiang Cheng, Yeshwanth Cherapanamjeri, Alan Malek, Wenlong Mou, Aldo Pacchiano, Juanky Perdomo, Alex Tsigler, Dong Yin, Andrea Bajcsy, Andreea Bobu, Daniel Brown, Lawrence Chan, Micah Caroll, Dylan Hadfield-Menell, Jerry He, Jaime Fisac, Smitha Milli, Aditi Raghunathan, Ellis Ratner, Sid Reddy, Dorsa Sadigh, Rohin Shah, Frances Ding, Dan Hendrycks, Meena Jagadeesan, Erik Jones, Adam Sealfon, Alex Wei, Ruiqi Zhong, Kabir Chandrasekher, Raaz Dwivedi, Avishek Ghosh, Vipul Gupta, Vidya Muthukumar, Ashwin Pananjady, Soham Phade, Nihar Shah, Melih Elibol, Chi Jin, Koulik Khamaru, Horia Mania, Eric Mazumdar, Robert Nishihara, Lydia Liu, Aaditya Ramdas, Esther Rolf, Ludwig Schmidt, Yan-Shuo Tan, Nilesh Tripuraneni, and Tijana Zrnic. Special thanks to Aldo, Niladri and Xiang for welcoming me into the Stat-learning group and making sure there was never a dearth of entertainment, to Andrea for sharing our pessimistic world-views and driving it towards optimism, and to Juanky and Nilesh for always being up for intriguing lunch conversations. I would also like to thank the

super helpful EECS and BAIR staff: Angie Abbatecola, Roxana Infante, Ami Katagiri, Jean Nguyen and Shirley Salanio, for always being ready to help with any concern.

This journey would be incomplete without the amazing set of friends, old and new, who always provided a fun environment and supported me. To Raaz, Sri Krishna and Soham, thanks for all the fun times travelling, cooking together, practising for the half-marathon, late-night chat sessions and more importantly, for always being available to listen me out. To Aboli and Prateek, thanks for welcoming us over and for providing us an escape from our stresses; the world appears a better place each time I meet you both. To Aldo and Niladri, thanks for always being ready to be a part of crazy schemes. To Aditya Grover, Kirankumar and Shivam, thanks for the endless conversations around research and philosophy that we had at Stanford. To Saurabh and Radhika, thanks for making me comfortable at Berkeley and taking me around to all the good restaurants around in the initial years; I still order all the same things at Bowl'd. To Adarsh Prasad, thanks for all the fun conversations and for being my go-to conference buddy. To Neha Yadav, thanks for all the encouragement and support and for continuously providing me with gossip from India. To Abhishek and Tanuja, thanks for bringing in joy and cheer to each of our gathering together. To Ayush Sekhari, thanks for listening in to all my rants and for introducing me to healthy eating; I have had to give up a lot of tasty food because of this.

Finally, and most importantly, I thank my family for all the support and encouragement throughout these years. I am really grateful to my nanaji and naniji for teaching me to always stay grounded and to live an honest life. I am thankful to my cousins, uncles and aunts, for setting up a support system which has allowed me to come this far. Special thanks to Pulleh and Pushti, for all the fun-filled conversations and to Sajal bhaiya for the guidance during the PhD. Thanks to Varun and Rupa for always welcoming me home, and to the girls, Anvesha and Adhya, for having me be a part of your fun-filled playful days. I owe everything in my life to my mom Poonam, *mummy*, who has been my pillar of strength throughout. She has been always there for me and has nurtured me with unconditional love. She has always supported my dreams and aspirations knowing how hard it would be for her to stay away from me. I respect all the sacrifices she has had to made to keep me going and am indebted to her for life. Next, thanks to my amazing brother, Sahil, for caring for me, for tolerating all my elder brother tantrums, and for all the memories, fun and not so fun, that we shared since childhood. I know I can always depend on you and am grateful for all that you have done for me. And finally, thanks to my wonderful girlfriend, Neha, for her unwavering support and understanding over the past nine years. I cannot thank you enough for the time and effort you put in to make sure my spirits are always up. I really admire your persistence and hard-working nature and you have always motivated me to become a better individual by being my most loving critic. Thanks for living up with all my craziness and I hope to share many more adventures together in the journey ahead!

Chapter 1

Introduction

The large-scale deployment of learning systems which interact with humans requires aligning what these systems optimize for with underlying human objectives and values. A major hurdle towards accomplishing this has been that it is hard for humans to precisely specify what it means to do the desired task well. Such situations arise, for instance, in autonomous driving, where one prefers a driving policy which is safe and comfortable [126, 119], in recommender systems, where one would like them to recommend content which promotes user's subjective well being [189], and also in classical natural language processing tasks like text summarization [159], where the notion of what makes a good summary is difficult to operationalize. Even in vanilla supervised learning, different mistakes may have different costs – for instance, misclassifying a stop sign is worse than mis-classifying a road-side postbox – and it is quite challenging for humans to correctly specify the relative mistakes of these costs [23].

Prior work on inferring these objectives from human feedback focuses on either learning these underlying preferences (reward learning) [152, 31, 187, 164] or relying on humans to specify these objectives by hand (reward engineering) [64]. Under the hood, these methods make a lot of assumptions: that people can consistently respond to complex queries, that we can represent the underlying objectives within our hypothesis space, and that humans are good at reward engineering. The proposed algorithmic approaches crucially rely on these assumptions to provide guarantees on the models they output.

This thesis is motivated by the concern that all these assumptions are not realistic: humans resort to simple heuristics when dealing with complex questions and biases their responses (e.g. representative heuristic) [199], it is often hard to represent everything that a person might care about (e.g. how courteous their car is to other drivers), and hand-written rewards by humans have quite often lead to unintended consequences (e.g. watch time as a proxy for to promote well-being by recommender systems lead to unhappy users [167]). Indeed, in the face of such misspecifications, the proposed learning algorithms can end up with incorrect models which could potentially make things worse instead of better. This work focuses on the design of value aligned models that are robust to such misspecifications by conceptually understanding how existing feedback modes fail when the underlying assump-

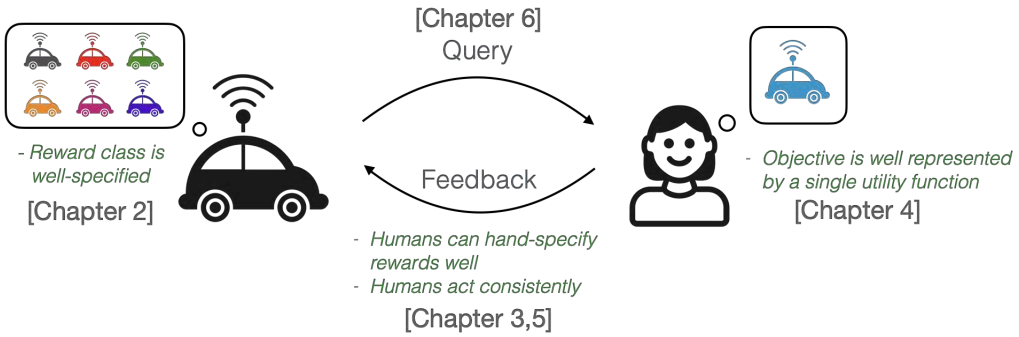


Figure 1.1. Thesis overview. We look at different forms of misspecifications that arise when learning systems optimize unknown human objectives. Such misspecifications arise in the way the objectives are assumed to be represented, in assumptions on how humans provide feedback given these objectives, and in the representation ability of these learning systems. This work focuses on understanding the side-effects of such misspecifications and presents algorithmic approaches to overcome them.

tions are violated and rethinking the right human feedbacks we need to develop provably robust models. Each chapter in the thesis addresses a different types of misspecifications that may arise in learning these underlying objectives and provides a systematic way to address it.

In Chapter 2, we look at the effects of misspecified reward classes, another form of a representational bias [23]. Algorithms for inferring reward functions from observed human decisions often posit that the true reward function belongs to some class of reward functions; for instance, it is a linear function of some pre-defined features. Such an assumption requires a learner to correctly identify everything a human might care about in their reward function, and is easily violated. Specifically, in this agnostic setup, we show that commonly used feedbacks like expert demonstrations as well as vanilla pairwise comparisons are information-theoretically insufficient to allow any learner to output a good model. To overcome this, we introduce a family of elicitation mechanisms by generalizing such comparisons, called the *k-comparison*, which enables the learner to ask for comparisons across k different inputs at once. This work brings out an interesting accuracy-elicitation trade-off – as the order k of the comparison increases, these queries become harder to elicit from humans but allow for more accurate learning.

In Chapter 3, we examine the effects of misspecified human behavioral models on reward inference [108]. Existing learning algorithms make very strong assumptions about how humans behave given their objectives. This is in stark contrast to decades of research in cognitive science, neuroscience, and behavioral economics, wherein obtaining such accurate human models remains a widely open question. This work asks the question: *how accurate do these models need to be in order for the reward inference to be accurate?* and studies it both theoretically and empirically. In a worst-case scenario, we show that it is unfortunately possible to construct small adversarial biases in behavior that lead to arbitrarily large errors in the inferred reward. However, on the positive front, we identify reasonable assumptions

under which the reward inference error can be bounded linearly in the error in the human model. From an empirical perspective, we verify our insights in discrete and continuous control tasks with both simulated biases, as well as real human data.

In Chapter 4, we study the effects of misspecification in the representation of the human's objective function [25]. Specifically, we study comparison-based preference learning models and exhibit how existing uni-criterion methods fail to infer the underlying preferences when the true utility function is represented via preferences along multiple criteria. To overcome these limitations, we propose a multi-criteria preference learning model and propose a new solution concept, Blackwell winner, by taking inspiration from Blackwell's approachability. Our proposed framework allows for non-linear aggregation of preferences across criteria, and generalizes the linearization-based approach from multi-objective optimization. From a theoretical standpoint, we show that the Blackwell winner of a multi-criteria problem instance can be computed as the solution to a convex optimization problem. Furthermore, given random samples of pairwise comparisons, we show that a simple "plug-in" estimator achieves near-optimal minimax sample complexity. We then showcase the practical utility of our framework in a user study on autonomous driving.

In Chapter 5, we consider the consequences of misspecified hand engineered reward functions in a reinforcement learning (RL) setup [157]. In several RL applications, the existing paradigm requires an engineer to specify a reward function which is then optimized to produce a policy. Misspecifications in these engineered rewards can lead to policies which are severely misaligned with the true objective, a phenomenon also termed as *reward hacking*. In this work, we systematically study this phenomenon from an empirical perspective. As opposed to supervised learning, where larger or more capable models have been seen to perform better than their shallower counterparts, we observe that more capable agents are able to better exploit reward misspecifications, causing them to attain higher proxy reward and lower true reward. Moreover, we find instances of *phase transitions*: capability thresholds at which the policy's behavior qualitatively shifts, leading to a sharp decrease in the true reward.

In Chapter 6, we propose a theoretical framework called *Doubly Nonparametric Bandits* for studying reward learning and the associated optimal experiment design problem [24]. Our proposed framework models rewards and policies as nonparametric functions belonging to subsets of Reproducing Kernel Hilbert Spaces (RKHSs). The learner receives (noisy) oracle access to a true reward and must output a policy that performs well under the true reward. For this setting, we first derive non-asymptotic excess risk bounds for a simple plug-in estimator based on ridge regression. We then solve the query design problem by optimizing these risk bounds with respect to the choice of query set and obtain a finite sample statistical rate, which depends primarily on the eigenvalue spectrum of a certain linear operator on the RKHSs. This framework, and the associated notion of policy regret, serves as a step towards studying the choice of adaptive queries for policy learning when both reward and policy classes maybe simultaneously misspecified.

Part I

Reward learning with misspecified models

Chapter 2

Agnostic learning with unknown utilities

2.1 Introduction

Our focus is on learning predictive models for decision-making tasks. Current paradigms for classification tasks use datasets consisting of scenarios[1] x along with the decisions y taken by human experts to learn a decision function[2] $f : \mathcal{X} \mapsto \mathcal{Y}$. For instance, in economics such decisions correspond to whether buyers bought an item at a suggested price [3, 22], in robotics such feedback comprises expert demonstrations in imitation learning [1, 12], and in machine learning literature such supervision consists of labels selected by human annotators [26, 70].

When we optimize models to predict correctly on these datasets, we often implicitly assume that all mistakes are equally costly, and that each scenario x in the data is just as important. In reality though, this is rarely the case. For instance, the standard $0-1$ loss for classification tasks assigns a unit of loss for each mistake, but misclassifying a stop sign is significantly more dangerous than misclassifying a road-side postbox. In Figure 2.1, we expand on this insight and illustrate how learning from such revealed decisions can often lead to suboptimal decision functions.

What is missing from this classical framework is that for most decision-making tasks there exists an underlying function $u^* : \mathcal{X} \times \mathcal{Y} \mapsto [0, 1]$ which evaluates the utility of a decision y depending on the surrounding context x. Depending on the decision task, such utility functions can encode buyer preferences in economics, rewards for robotic skills, or misprediction costs for classification. However, these utility functions are a priori unknown to the learner since the dataset consists only of context-decision pairs (x, y). Furthermore, asking human experts to write down these complex utility functions can be quite challenging and prone to serious errors [9].

One commonly studied approach, referred to as learning from revealed preferences in

[1] We use the term scenario/context/feature for the vector x interchangeably.

[2] We consider finite decision spaces $\mathcal{Y}$.

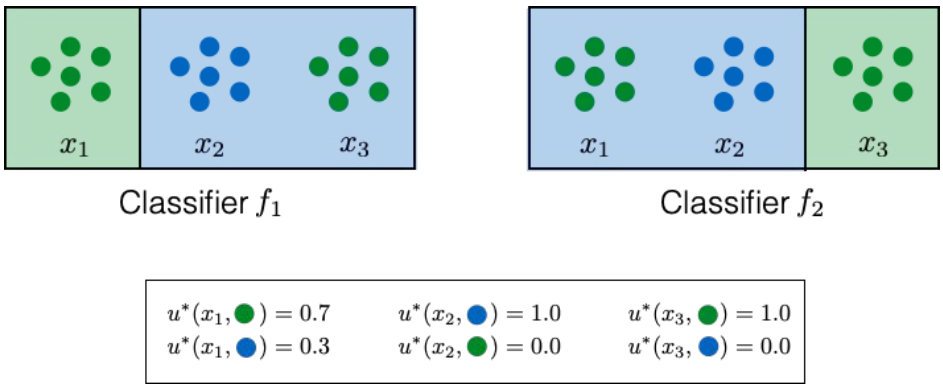


Figure 2.1. Consider a binary decision-task with decisions G(reen) and B(lue). The instance space comprises of three equiprobable clusters of datapoints x_1, x_2 and x_3, and have associated utilities u^* for decisions B and G. The colour of the datapoints represents the decision with higher utility. The function class $\mathcal{F}$ consists of linear predictors. In the traditional learning setups where the dataset consists of pairs (x, y), no learner will have enough information to select between f_1 and f_2 since the $0-1$ error for both is $1/3$. In contrast, using a 2-comparison oracle, a learner can ask a query of the form "Which of $u^*(x_1, \text{G}) + u^*(x_3, \text{B})$ or $u^*(x_1, \text{B}) + u^*(x_3, \text{G})$ is bigger?". This allows them to infer that correctly predicting x_3 gives a higher overall utility and output the optimal decision function f_2.

economics [22, 18] and inverse reinforcement learning (IRL) in the machine learning literature [150, 226], assumes that the utility function u^* belongs to some pre-specified class and uses the fact that decision y was the optimal decision for scenario x to learn estimates of these utilities. This setup is called the well-specified or realizable setup. However, this posited utility class can be misspecified in that the underlying utility u^* might not belong to this class. The correctness of such learning approaches crucially relies on the well specified assumption and offers no guarantees on how their performance degrades in the presence of class misspecifications.

We overcome this uncertainty in specifying the utility function u^* by proposing an *agnostic learning* framework which places no assumptions on the class of utility functions. Instead, we consider decision functions belonging to some class $\mathcal{F} = \{f \mid f : \mathcal{X} \mapsto \mathcal{Y}\}$ and study the objective of obtaining the "best" decision rule in $\mathcal{F}$ with respect to the unknown utility u^*. Formally, given the decision class $\mathcal{F}$ and samples from a distribution $\mathcal{D}_x$ over the feature space $\mathcal{X}$, the objective of the learner is to output a model $\hat{f} \in \mathcal{F}$ with small excess risk or regret

$$\mathsf{err}(\hat{f}, \mathcal{F}) := \sup_{f \in \mathcal{F}} \mathbb{E}_{x \sim \mathcal{D}_x}[u^*(x, f(x))] - \mathbb{E}_{x \sim \mathcal{D}_x}[u^*(x, \hat{f}(x))] \,. \tag{2.1}$$

Our proposed notion of excess risk measures the performance of an estimator $\hat{f}$ by comparing its decisions with those of the best predictive model in the class $\mathcal{F}$ under the utility u^*. Contrast this with the classical agnostic learning framework [100] where the evaluation metric

for classification measures what proportion of datapoints $\hat{f}$ predicts correctly

$$\mathsf{err}_{\mathsf{cl}}(\hat{f}, \mathcal{F}) := \sup_{f \in \mathcal{F}} \mathbb{E}_{x \sim \mathcal{D}_x}[\mathbb{I}[f(x) \neq y_x]] - \mathbb{E}_{x \sim \mathcal{D}_x}[\mathbb{I}[\hat{f}(x) \neq y_x]] \,, \tag{2.2}$$

where $y_x = \operatorname{argmax}_{y \in \mathcal{Y}} u^*(x, y)$ represents the expert decision (revealed decision) for scenario x. Our above framework generalizes the proper agnostic learning framework – we restrict our attention to proper learners which output models $\hat{f} \in \mathcal{F}$ and the decision class $\mathcal{F}$ is agnostic towards the unknown underlying utility u^*. Indeed, our agnostic framework allows for misspecification in the decision class $\mathcal{F}$ and allows for situations where no predictive model $f \in \mathcal{F}$ matches the expert predictions y_x for all instances x.

As highlighted by Figure 2.1, such a misspecification in the function class $\mathcal{F}$ implies that no decision function $f \in \mathcal{F}$ will be able to perfectly fit these optimal decisions y_x for all points $x \in S$. In order to solve the agnostic learning problem, it is necessary for the learner to understand the how costly these different mistakes are relative to each other. From the learners perspective, observing only the optimal decisions y_x for each instance x, such as revealed preferences or expert demonstrations, are clearly insufficient to obtain any information about these costs. One way to overcome this information-theoretic limit of revealed decisions is to directly elicit the utilities from humans – for scenarios x and decision y, ask an expert "What is the utility $u^*(x, y)$ for taking the decision y given situation x?". However, since the underlying utility u^* can be quite complex, humans are inept at answering them reliably [144, 186]. For instance, it can be challenging for humans to correctly specify the costs of mispredicting, say, a stop sign as a red signal relative to that of predicting it as a post-box.

On the other hand, it is often easier for humans to provide comparative evaluations based on these utilities [194, 87] and allow the learner to obtain relative feedback. Using these, the learner can query an expert with comparison or preference queries asking "For instance x, which of the two utilities $u^*(x, y_1)$ or $u^*(x, y_2)$ is larger?". Such vanilla comparisons can allow the learner to infer relative utilities for decisions y_1 and y_2 for a given context x; the learner can conclude that mispredicting stop sign as post-box is worse than mispredicting it as a red signal. However, such feedback still does not provide any information about the mistake costs across different examples – given a choice, should the learner correctly predict a stop-sign or correctly predict a post-box?

While vanilla comparisons are insufficient for the agnostic setup, let us consider the other extreme: suppose that we have access to an oracle which can provide us with comparisons of overall utilities for functions $f_1, f_2 \in \mathcal{F}$. That is, the oracle can answer question of the form "Which of the two overall utilities $\mathbb{E}_x[u^*(x, f_1(x))]$ or $\mathbb{E}_x[u^*(x, f_2(x))]$ is larger?". Given access to such an oracle, we will be able to find the optimal classifier in the class $\mathcal{F}$. We call this the ∞-comparison oracle since such preferences requires a human to reason about the utilities over the entire feature space $\mathcal{X}$ at once. Even for a small image classification task with a million images, this would require a human to compare the utility of a million simultaneous predictions! While this approach does allows for optimal estimation, the trade-off is that it puts the complete burden of learning on the human's side. It is worth highlighting that

the comparisons between lotteries used to establish the von Neumann-Morgenstern utility theorem [147] can be shown to be a special case of such an ∞-comparison oracle.

While comparison queries only allow comparison within a single instance, the ∞-comparison oracle takes the other extreme and requires a comparison along all instances. However, we need not restrict our self to either of these extremes; our key insight is that there is a natural spectrum of such comparisons, which we call k-comparisons which interpolate between the single or 1-comparison and the ∞-comparison oracle. Such comparison queries allow a learner to pick k instances $\{x_1, \ldots, x_k\}$ and two sets of corresponding decision, $\{y_1, \ldots, y_k\}$ and $\{y'_1, \ldots, y'_k\}$, and ask "Which of the cumulative utilities $\sum_i u^*(x_i, y_i)$ or $\sum_i u^*(x_i, y'_i)$ is bigger?". For instance, for the example in Figure 2.1, giving the learner access to a 2-comparison oracle allows the algorithm to output the optimal decision function.

These higher-order comparison oracles form a natural hierarchy of elicitation mechanisms for the learner with a k'-oracle being strictly more informative than a k-oracle for $k' > k$. They allow for a natural trade-off between accuracy and elicitation in the learning with unknown utilities framework. As we increase the order k of the oracle, the learner can obtain finer information about the utilities u^* and output functions with lower excess risk. However, this increase in information comes at the expense of asking for a harder elicitation from the human expert.

Our Contributions. We propose a novel framework, which we call *agnostic learning with unknown utilities*, for studying decision problems wherein the learner is evaluated with respect to an unknown utility function. Within this framework, we show that standard approaches which work well in the realizable setup, such as revealed preferences as well as vanilla comparisons, can perform quite poorly in the face of misspecification and can have excess risk $\Omega(1)$. To overcome this, we propose a family of elicitation mechanisms, the k-comparisons, which allows the learner access to finer information from an human expert with increasing values of the order k. Our main results, detailed in Section 2.3, provide a tight characterization of the excess risk as a function of the order k of the comparison oracle available to the learner. These result brings out an interesting accuracy-elicitation trade-off – as the order k of the oracle increases, the comparative queries allow for more accurate learning in our setup but become harder to elicit from humans.

We would like to highlight that increasing the order k of the comparisons could lead to potentially biased and noisy responses from the human expert. As a consequence, there might be an additional trade-off involving the *quality of the information* obtained by increasing the order. While we do not focus on this aspect of elicitation, it is an interesting direction for future work.

Related work

This paper sits at the intersection of multiple fields of study: agnostic learning , learning with nuisance parameters, and utility learning from preferences . Here, we review the papers that are most relevant to our contributions.

Agnostic learning. The framework of probably approximately correct (PAC) learning was introduced in their seminal work by Valiant [202]. This framework formalized the problem of learning from sampled data in a realizable setup. This was formally extended to the agnostic setup, with no assumptions on the data generating distribution, by Haussler [100]. Connections of learnability with uniform convergence were first established by Vapnik [205], and more recently it was established in [178] that for the general learning problem, such a uniform convergence is not necessary to establish learnability. Similar to the classical agnostic supervised learning, the learner does not know the distribution $\mathcal{D}_x$ but only has access to it via samples. The key difference is that the classical setup assumes that the utility function u^* is known to the learner while our framework does not.

Learning with nuisance parameters. Closely related to our setup is the problem of learning with a nuisance component [77] which comprises as special case the problems of heterogeneous treatment effect estimation [52], offline policy learning [14], and learning with missing data [94] amongst others. In this setup, objective is to learn a predictor with small excess risk and this risk depends on a underlying nuisance parameter which is unknown to the learner a priori. The unknown utility u^* of our setup can be seen as a nuisance component in their framework. However, the two problems differ in the form of information available to the learner – they allow the learner to directly elicit (possibly noisy) values of utility u^*. They additionally require that utility u^* belongs to some pre-specified function class and their bounds depend on the rate at which this utility function is learnable over this class.

Another line of work, called double/debiased machine learning in the statistics and econometrics literature [51, 53, 54], addresses semiparametric inference [168, 123] where the function class $\mathcal{F}$ is assumed to be a parametric family along with a non-parametric nuisance component. In addition to the differences mentioned above, this class of methods focuses on exact parameter recovery and conditions under which $\sqrt{n}$-consistent and asymptotically normal estimators can be obtained.

Utility estimation with preferences. The seminal work of von Neumann and Morgenstern [147] established that any rational agent whose preferences satisfy certain axioms will have a utility function. Furthermore, the proof of this expected utility theorem showed these utilities could be elicited from the agent using preferences over randomized lotteries. As discussed in Section 2.1, such preferences over lotteries can be seen as a special case of the ∞-comparison oracle. There have been several recent works studying the consequences of incomplete preferences [153, 88] which show the existence of a class of utility functions which are consistent with these incomplete preferences. Our k-comparison oracles can be seen as a quantitative approach to studying such incomplete preferences; for each value of $k \geq 1$, the human expert can only compare lotteries up to a granularity of $\frac{1}{k}$. Our work goes a step forwards and studies the consequences of such incomplete preferences for decision-making tasks.

2.2 Problem formulation

In this section, we formally state our learning with unknown utilities problem and introduce the k-comparison oracle. Let $\mathcal{X} \subseteq \mathbb{R}^d$ represent the space of feature vectors, $\mathcal{Y}$ denote the corresponding decision space and $\mathcal{F}$ denote a class of decision making functions, given as $\mathcal{F} = \{f \mid f : \mathcal{X} \mapsto \mathcal{Y}\}$. Our framework considers an underlying utility function $u^* : \mathcal{X} \times \mathcal{Y} \mapsto [0, 1]$ which assigns a non-negative real value for making a decision $y \in \mathcal{Y}$ given a situation $x \in \mathcal{X}$. Further, let us denote the set

$$\mathcal{U} = \{u \mid u : \mathcal{X} \times \mathcal{Y} \mapsto [0, 1]\} \tag{2.3}$$

of all possible such utility functions. For any distribution $\mathcal{D}_x$ over the feature space $\mathcal{X}$, we define the expected utility of a decision function $f \in \mathcal{F}$ as $U(f; u^*) := \mathbb{E}_{x\sim\mathcal{D}_x}[u^*(x, f(x))]$. Observe that such an expected utility model assumes that the utilities are additive across the different instances x and is a commonly studied model both in the machine learning, statistics and economics literature. We denote the excess risk of a function f with respect to the function class $\mathcal{F}$ by

$$\mathsf{err}(f, \mathcal{F}; u^*) := \max_{f' \in \mathcal{F}} U(f'; u^*) - U(f; u^*). \tag{2.4}$$

Further, we denote the optimal decision for any instance x with respect to the underlying utility u^* by $y_x := \operatorname{argmax}_{y\in\mathcal{Y}} u^*(x, y)$.

Similar to the classical agnostic learning setup [100], we assume that the learner does not know the underlying distribution $\mathcal{D}_x$ of the instances. However, our setup differs from it in that we do not assume that the underlying utility function u^* is known to the learner. Instead, we provide the learner access to an oracle which allows the learner to elicit responses to higher-order preferences queries.

Comparison Oracle Since the utility function u^* is unknown to the learner, our framework allows the learner access to an oracle which provides comparative feedback based on the utilities u^*. We consider a family of such oracles $\mathcal{O}_k$, each indexed by its order k which determines the number of different instances the learner is allowed to specify in the comparison query. For an oracle $\mathcal{O}_k$, a learner is allowed to select a set of k situations $\mathbf{x} \in \mathcal{X}^k$ and two pairs of corresponding decisions $\mathbf{y}_1, \mathbf{y}_2 \in \mathcal{Y}^k$. The oracle then compares, in a possibly noisy manner, the cumulative utilities of the pair $(\mathbf{x}, \mathbf{y}_1)$ and $(\mathbf{x}, \mathbf{y}_2)$ and responds with the feedback on which one is larger. As the order k of the oracle increases, the queries become more complex – an expert is required to evaluate a larger number of instances at once. This family of comparison oracles captures a natural hierarchy of elicitation mechanisms where with each increasing value of k, a learner has access to more information about the utility function u^*.

Formally, we represent a k-query by a tuple $(\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2)$ where the input $\mathbf{x} = (x_1, \ldots, x_k)$ comprises k feature vectors and the corresponding decision vectors $\mathbf{y}_1 = (y_1, \ldots, y_k)$ and

$\mathbf{y}_2 = (y'_1, \ldots, y'_k)$.[3] Given such a query q, the oracle $\mathcal{O}_k$ provides the learner a binary response

$$\mathcal{O}_k(q = (\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2)) = \begin{cases} \mathbb{I}\left[u^*(\mathbf{x}, \mathbf{y}_1) \geq u^*(\mathbf{x}, \mathbf{y}_2)\right] & \text{with prob. } 1 - \eta_q \\ 1 - \mathbb{I}\left[u^*(\mathbf{x}, \mathbf{y}_1) \geq u^*(\mathbf{x}, \mathbf{y}_2)\right] & \text{otherwise} \end{cases}, \tag{2.5}$$

where the parameter $0 \leq \eta_q < \frac{1}{2}$ represents the noise level corresponding to query q. Thus, the oracle[4] $\mathcal{O}_k$ provides noisy comparisons of the cumulative utilities $u^*(\mathbf{x}, \mathbf{y}_1)$ and $u^*(\mathbf{x}, \mathbf{y}_2)$ with varying noise level η_q. Observe that we allow the noise levels η_q to be different for each query q.

Problem Statement We are interested in the *agnostic learning with unknown utilities* problem where a learner is provided n samples $S = \{x_1, \ldots, x_n\}$ with each $x_i \sim \mathcal{D}_x$ and access to the k-comparison oracle described above, and is required to output a decision function $\hat{f} \in \mathcal{F}$ such that error $\mathsf{err}(\hat{f}, \mathcal{F})$ is small. The caveat is to do so with a minimum number of calls, which we term the query complexity n_q of learning, to the comparison oracle $\mathcal{O}_k$. Quantitatively, we would like to characterize the excess risk from equation (2.4) in terms of the number of sampled instances n, the order k of the comparison oracle and properties of the decision function class $\mathcal{F}$, and the associated oracle query complexity n_q to obtain this bound.

Obtaining such bounds on the excess risk $\mathsf{err}(f, \mathcal{F}; u^*)$ in terms of the order k allow us to quantify the trade-offs in learning better decision functions at the expense of requiring more complex information from the human expert. Going forward, we focus on the binary decision making problem where the label space $\mathcal{Y} = \{0, 1\}$ for clarity of exposition. Whenever our results can be extended to arbitrary decision sets, we provide a small remark about this extension.

2.3 Main results

With the formal problem setup in place, we discuss our main results for learning in this framework of unknown utilities. At a high level, our objective is to understand how the excess risk $\mathsf{err}(f, \mathcal{F}; u^*)$ defined in equation (2.4) behaves as a function of the oracle order k – specifically, at what rates does learning in our proposed framework get easier as we allow learner to elicit more complex information from the oracle?

For our main results, on the upper bound side, we design estimators for learning from the k-comparison oracle, and on the lower bound side, we study information-theoretic limits of learning with such higher-order comparisons. While we state our results for the binary

[3] We overload our notation and represent the cumulative utilities of the k inputs $(\mathbf{x}, \mathbf{y})$ by $u^*(\mathbf{x}, \mathbf{y}) = \sum_i u^*(x_i, y_i)$.

[4] Note that while the oracle depends on the underlying utility function u^*, our notation suppresses this dependence for clarity. We use the notation $\mathcal{O}_k(q; u^*)$ whenever we want to make this dependence explicit.

decision problem where the label space $\mathcal{Y} = \{0, 1\}$ for clarity, most of our results can be generalized to arbitrary outcome space $\mathcal{Y}$.

Excess risk with k-comparison oracle (Section 2.4)

We study a class of *plug-in* estimators which are based on the following two-step procedure:

i. Obtain estimate $\hat{u}$ of the true utility u^* on the sampled datapoints.

ii. Output utility maximizing function $\hat{f}_{k,n}$ with respect to the estimated utility $\hat{u}$.

For learning the parameters $\hat{u}$, we introduce the Comptron (Algorithm 1) and Rob-Comptron (Algorithm 2) algorithms for the noiseless and noisy comparison oracles respectively. We show that when these estimates $\hat{u}$ are combined with the two-step plug-in estimator, the excess risk of the function $\hat{f}_{k,n}$ scales as $O(\frac{1}{k})$ and an additive complexity term capturing uniform convergence of the decision class $\mathcal{F}$ with respect to the true utility u^*.

Theorem 2.1 (Informal, noiseless comparisons)**.** *Given n samples, the excess risk for the function $\hat{f}_{k,n} \in \mathcal{F}$ output by the plug-in estimator using estimates $\hat{u}$ from Comptron satisfies*

$$\mathsf{err}(\hat{f}_{k,n}, \mathcal{F}; u^*) \leq \mathsf{Complexity}_n(\mathcal{F}; u^*) + O\left(\frac{1}{k}\right) \cdot \left(\frac{1}{n}\sum_{i=1}^{n} \mathbb{I}[f_{\mathsf{ERM}}(x_i) \neq y_i]\right) ,$$

where the ERM function $f_{\mathsf{ERM}} \in \operatorname{argmax}_{f \in \mathcal{F}} \sum_{i=1}^{n} u^*(x_i, f(x_i))$. Furthermore, Comptron makes only $O(n \log k)$ queries to the oracle $\mathcal{O}_k$.

We make a few remarks on this result. First, observe that the complexity term depends on the true utility function u^* and not on the estimates $\hat{u}$. This ensures that the complexity term does not depend on the utility class $\mathcal{U}$ but rather only on the specific utility u^* – indeed, the class $\mathcal{U}$ consists of all bounded function and uniform convergence might not even be possible with finite sample for a large class of distributions $\mathcal{D}_x$. Second, the additional error of $O(\frac{1}{k})$ accounts for the fact that the utilities u^* are unknown. One can learn better decision functions by increasing the order k of the comparison oracle but this comes at the cost of the human expert answering a more complex set of queries. Furthermore, this error is multiplied by the $0-1$ prediction error of the optimal on-sample classifier $f_{\mathsf{ERM}} = \operatorname{argmax}_{f \in \mathcal{F}} \sum_i u^*(x_i, f(x_i))$. This implies that in the well-specified setup, where there exists an $f \in \mathcal{F}$ such that $f(x_i) = y_i$ on the sampled datapoints, the second term becomes 0 and the learner pays no additional error for not knowing the utilities u^*. Third, observe that our proposed algorithms, Comptron and Rob-Comptron, are query efficient; both require only $O(n \log k)$ calls to the k-comparison oracle to produce "good" estimates $\hat{u}$.

The proof of the above theorem proceeds in two steps. First, we adapt the classical proof for upper bounding the risk of ERM procedures to show that the gap $\mathsf{err}(\hat{f}_{k,n}, \mathcal{F})$ decomposes into the complexity term and estimation error $\|\hat{u} - u^*\|_{S,\infty}$, evaluated on the

dataset S. Next, we show that this estimation error scales as $O\left(\frac{1}{k}\right)$ for the Comptron and Rob-Comptron procedures.

Next, we address the optimality of the above plug-in procedure by studying the information-theoretic limits of learning with a k-comparison oracle. Specifically, in Theorem 2.3 we establish that the rate of $\frac{1}{k}$ is indeed minimax optimal – for any $k > 1$ and any predictor $\hat{f}$ in some class $\mathcal{F}$, we can construct utility functions u^* such that excess risk $\mathsf{err}(\hat{f}, \mathcal{F}; u^*) = \Omega\left(\frac{1}{k}\right)$. These lower bounds imply that traditional comparison based learning, corresponding to $k = 1$, is insufficient for learning good decision rules in our framework.

Instance-optimal learning (Section 2.5).

While the previous results show that the error rate of $O(\frac{1}{k})$ is optimal on worst-case instances, some instances of our learning with unknown utilities problem might be easier than these worst-case ones and one would expect the excess risk to be smaller for them. In this section, we study estimators whose error adapts to hardness of the specific problem instance.

To begin with, in Proposition 2.2 we establish that the plug-in estimator with Comptron estimates $\hat{u}$ is not optimal for all instances – it does not adapt to these easier instances. Inspired from the robust optimization literature, we introduce a randomized estimator p_{rob} and show that it is instance-optimal. Informally, we establish in Theorem 2.5 that for any instance $(\mathcal{D}_x, u^*, \mathcal{F})$ of the problem, the excess risk for p_{rob} is characterized by a local modulus of continuity; this modulus captures how quickly the optimal decision function in class $\mathcal{F}$ can change in a small neighborhood around u^* for the distribution $\mathcal{D}_x$. In Theorem 2.4, we derive a lower bound on the *local minimax excess risk* and show that the local modulus is indeed the correct instance-dependent complexity measure for this problem.

However, note that such adaptivity to the hardness of the instance comes at the cost of query efficiency. Our estimator p_{rob} makes an exponential number $O(n^k)$ of calls to the oracle $\mathcal{O}_k$.

2.4 Binary decision-making with k-comparisons

In this section, we obtain upper and lower bounds on the excess risk for the binary prediction problem with unknown utilities where the learner can elicit utility information using a k-comparison oracle. In Section 2.4, we introduce algorithms which learn decision-making rules from higher-order preference queries and obtain upper bounds on the excess risk for such estimators. Then, in Section 2.4, we turn to the information-theoretic limits of learning from k-queries and obtain lower bounds on the minimax risk of any estimator.

Recall from Section 2.2, our setup gives the learner access to a dataset $S = \{x_1, \ldots, x_n\}$ comprising n points, each sampled i.i.d. from an underlying distribution $\mathcal{D}_x$ and to a comparison oracle $\mathcal{O}_k$. Before proceeding to define the estimator, we introduce some notation. For any function $f \in \mathcal{F}$, let us denote the empirical cumulative utility with respect to utility

function u^* and the corresponding empirical utility maximizer as

$$\hat{U}_n(f;u^*) = \frac{1}{n}\sum_i u^*(x_i, f(x_i)) \quad \text{and} \quad f_{\mathsf{ERM}} \in \operatorname*{argmax}_{f\in\mathcal{F}} \hat{U}_n(f;u^*)\ , \tag{2.6}$$

where the subscript n encodes the dependence on the number of samples. If the underlying utility u^* were in fact known to the learner, it would have output the classifier f_{ERM}, which, from the classical learning theory literature, is known to have favorable generalization properties [179]. For the case of unknown utilities, we extend this ERM procedure to a natural two-stage plug-in estimator which outputs the minimizer with respect to an estimate $\hat{u}_k$ of these utilities.

Excess-risk upper bounds for plug-in estimator

Building on the ERM estimator f_{ERM} described in equation (2.6), we design a two stage *plug-in* estimator $\hat{f}_{k,n}$, where the subscript k represents the order of the comparison oracle used to obtain the estimate.

In the first stage, we form estimates $\hat{u}_k$ of the true utility function u^* on the sampled datapoints S using the k-comparison oracle. The predictor $\hat{f}_{k,n} \in \mathcal{F}$ is then given by the empirical utility maximizer with respect to $\hat{u}_k$, that is,

$$\hat{f}_{k,n} \in \operatorname*{argmax}_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} \hat{u}_k(x_i, f(x_i)). \tag{2.7}$$

Before detailing out the procedures for producing utility estimates $\hat{u}_k$, we present our first main result which shows that the excess risk $\mathsf{err}(\hat{f}_{k,n}, \mathcal{F}; u^*)$ can be upper bounded as a sum of two terms: (i) a complexity term corresponding to the rate of uniform convergence of the cumulative utility $U(f;u^*)$ over the decision class $\mathcal{F}$ and (ii) an estimation error term which denotes how well the estimates $\hat{u}_k$ approximate u^* on the sampled datapoints. Our result measures this estimation error in terms of a data-dependent norm

$$\|u\|_{S,\infty} := \sup_{i\in[n]} \sup_{y\in\mathcal{Y}} |u(x_i, y)|. \tag{2.8}$$

Recall from equation (2.6) that the function f_{ERM} is the minimizer of the empirical utility $\hat{U}_n(f;u^*)$. While the following results hold for general decision spaces $\mathcal{Y}$, we later specialize this in Proposition 2.1 for the binary prediction setup.

Theorem 2.2 (Excess-risk upper bound)**.** *Given datapoints $S = \{x_1, \ldots, x_n\}$ such that each $x_i \sim \mathcal{D}_x$, and an estimate $\hat{u}_k$ of the true utility function u^*, the plug-in estimate $\hat{f}_{k,n}$ from equation (2.7) satisfies*

$$\mathsf{err}(\hat{f}_{k,n}, \mathcal{F}; u^*) \le 2\cdot \sup_{f\in\mathcal{F}} \Big(|U(f;u^*) - \hat{U}_n(f;u^*)|\Big) + 2\|u^* - \hat{u}_k\|_{S,\infty} \cdot \left(\frac{1}{n}\sum_{i=1}^{n} \mathbb{I}[f_{\mathsf{ERM}}(x_i) \neq \hat{f}_{k,n}(x_i)]\right). \tag{2.9}$$

A few comments on Theorem 2.2 are in order. First, notice that the upper bound on the risk $\mathsf{err}(\hat{f}_{k,n}, \mathcal{F}; u^*)$ is a deterministic bound comprising two terms. The uniform convergence term captures how fast the empirical utility $\hat{U}_n(f; u^*)$ converge to the population utility $U(f; u^*)$ uniformly over the decision class $\mathcal{F}$. Using standard bounds [21], one can show that this term is upper bounded by the empirical Rademacher complexity of the class $\mathcal{F}$ on the datapoints S, that is,

$$\sup_{f \in \mathcal{F}} \Big(|U(f; u^*) - \hat{U}_n(f; u^*)| \Big) \leq \mathbb{E}_\varepsilon \left[\sup_{f \in \mathcal{F}} \left| \frac{1}{n} \sum_{i=1}^{n} \varepsilon_i u^*(x_i, f(x_i)) \right| \right] := \widehat{\mathfrak{R}}_n(\mathcal{F} \circ u^*) \tag{2.10}$$

where each ε_i is an i.i.d. Rademacher random variable taking values $\{-1, +1\}$ equiprobably. Such complexity measures are commonly studied in the learning theory literature and one can obtain sample complexity rates for a wide range of decision classes including parametric decision classes and non-parametric kernel classes amongst others.

The second term in equation (2.9) is given by a product of two terms. The first part $\|u^* - \hat{u}_k\|_{S,\infty}$ captures the *on-sample approximation error* of the estimates $\hat{u}_k$. Notice that, in general, the problem of estimating u^* uniformly over the space $\mathcal{X}$ is infeasible since the class $\mathcal{U}$ contains the set of all bounded functions on $\mathcal{X} \times \mathcal{Y}$. However, the fact that we are required to estimate the utilities u^* only on the sampled datapoints S makes learning feasible in our framework. The second part, $\frac{1}{n}\sum_{i=1}^{n} \mathbb{I}[f_{\mathsf{ERM}}(x_i) \neq \hat{f}_{k,n}(x_i)] \leq 1$ the mismatch between the predictions of f_{ERM}, obtained with complete knowledge of u^*, and of $\hat{f}_{k,n}$, obtained from estimates $\hat{u}_k$. Notice that whenever the function class $\mathcal{F}$ is correctly specified on S, that is, there exists a function $f \in \mathcal{F}$ such that $f(x_i) = y_i$), then the predictions of $\hat{f}_{k,n}$ and f_{ERM} will coincide. This follows since the labels y_i can be inferred using a 1-comparison. In such a well-specified setup, this second term vanishes and we recover the upper bound in terms of the uniform convergence term. Surprisingly, this exhibits that not knowing the utility u^* affects learnability only when the function class $\mathcal{F}$ is misspecified.

Proof. We begin by decomposing the excess error $\mathsf{err}(\hat{f}_{k,n}, \mathcal{F}; u^*)$ and then handle each term in the decomposition separately. Recall that the function f_{ERM} is the maximizer of the empirical utility $\hat{U}_n(f; u^*)$. Then, for any decision function $f \in \mathcal{F}$, consider the error

$$\begin{aligned}
\mathsf{err}(\hat{f}_{k,n}, f; u^*) &= U(f; u^*) - \hat{U}_n(f; u^*) + \hat{U}_n(f; u^*) - \hat{U}_n(f_{\mathsf{ERM}}; u^*) + \hat{U}_n(f_{\mathsf{ERM}}; u^*) - \hat{U}_n(\hat{f}_{k,n}; u^*) \\
&\quad + \hat{U}_n(\hat{f}_{k,n}; u^*) - U(\hat{f}_{k,n}; u^*) \\
&\overset{\text{(i)}}{\leq} 2 \sup_{f \in \mathcal{F}} \Big(|U(f; u^*) - \hat{U}_n(f; u^*)| \Big) + \underbrace{\hat{U}_n(f_{\mathsf{ERM}}; u^*) - \hat{U}_n(\hat{f}_{k,n}; u^*)}_{\text{Term (I)}},
\end{aligned} \tag{2.11}$$

where the inequality (i) follows by noting that f_{ERM} is the maximizer of $\hat{U}_n(f; u^*)$. We now

focus our attention on Term (I) in the above expression.

$$\begin{aligned}\hat{U}_n(f_{\mathsf{ERM}}; u^*) - \hat{U}_n(\hat{f}_{k,n}; u^*) &= \hat{U}_n(f_{\mathsf{ERM}}; u^*) - \hat{U}_n(f_{\mathsf{ERM}}; \hat{u}) + \hat{U}_n(f_{\mathsf{ERM}}; \hat{u}) - \hat{U}_n(\hat{f}_{k,n}; \hat{u}) \\ &\quad + \hat{U}_n(\hat{f}_{k,n}; \hat{u}) - \hat{U}_n(\hat{f}_{k,n}; u^*) \\ &\overset{\text{(i)}}{\leq} \frac{2}{n}\sum_{i=1}^{n} \mathbb{I}[f_{\mathsf{ERM}}(x_i) \neq \hat{f}_{k,n}(x_i)] \cdot \sup_{y \in \mathcal{Y}} |u^*(x_i, y) - \hat{u}(x_i, y)| \\ &\leq 2\|u^* - \hat{u}\|_{S,\infty} \cdot \left(\frac{1}{n}\sum_{i=1}^{n} \mathbb{I}[f_{\mathsf{ERM}}(x_i) \neq \hat{f}_{k,n}(x_i)]\right) ,\end{aligned}$$

where (i) follows by noting that $\hat{f}_{k,n}$ maximizes the utility $\hat{U}_n(f; \hat{u})$. Plugging the bound above in equation (2.11) completes the proof. □

We now specialize the result of Theorem 2.2 to the binary prediction setup where the label space $\mathcal{Y} = \{0, 1\}$. Recall that for each datapoint x_i, we denote the true label by $y_i = \operatorname{argmax}_y u^*(x_i, y)$. We now introduce the notion of utility gaps $u_{\mathsf{gap}}(x_i)$ which measures the excess utility a learner gains by predicting a datapoint x_i correctly relative to an incorrect prediction. Formally, the gap $u_{\mathsf{gap}}(x_i)$ for datapoint x_i with respect to some utility function $u \in \mathcal{U}$ is given as

$$u_{\mathsf{gap}}(x_i) := u(x_i, y_i) - u(x_i, \bar{y}_i) , \tag{2.12}$$

where we denote the incorrect label by $\bar{y} = 1 - y$. With this notation, the following proposition obtains an upper bound on the excess error of plug-in estimator $\hat{f}_{k,n}$ for the binary prediction problem in terms of the estimation error in these gaps $u_{\mathsf{gap}}(x_i)$.

Proposition 2.1 (Upper bounds for binary prediction)**.** *Consider the binary decision making problem with label space $\mathcal{Y} = \{0, 1\}$. Given n datapoints $\{x_1, \ldots, x_n\}$ such that each datapoint $x_i \sim \mathcal{D}_x$, and an estimate $\hat{u}_k$ of the utility function u^*, the plug-in estimator $\hat{f}_{k,n}$ from equation (2.7) satisfies*

$$\mathsf{err}(\hat{f}_{k,n}, \mathcal{F}; u^*) \leq 2 \cdot \sup_{f \in \mathcal{F}} \left(|U(f; u^*) - \hat{U}(f; u^*)|\right) + 2 \max_i [u^*_{\mathsf{gap}}(x_i) - \hat{u}_{\mathsf{gap}}(x_i)] \cdot \left(\frac{1}{n}\sum_{i=1}^{n} \mathbb{I}[f_{\mathsf{ERM}}(x_i) \neq y_i]\right) . \tag{2.13}$$

The proof of the above proposition follows similar to Theorem 2.2 and is deferred to Appendix A.1. This specializes the result of Theorem 2.2 and shows that for the binary prediction problem, estimating the utility gaps u_{gap} well for each datapoint suffices

The upper bound on excess risk given by Proposition 2.1 shows that the function $\hat{f}_{k,n}$ derived from estimates $\hat{u}_k$ will have small error as long as the estimates $\hat{u}_{\mathsf{gap}}(x_i)$ approximate the true utility gaps $u^*_{\mathsf{gap}}(x_i)$ for each datapoint x_i. Therefore, in the following sections, we focus on procedures for obtaining the utility estimates $\hat{u}_{\mathsf{gap}}$ using the k-comparison oracle. we separate the presentation based on whether the oracle $\mathcal{O}_k$ provides noiseless comparisons ($\eta_q = 0$ for all q) or whether the oracle evaluations are noisy.

Estimating u^*_{gap} with noiseless oracle

In this section, we propose our algorithm for estimating the gaps u^*_{gap} when the k-comparison oracle is noiseless. Recall from equation (2.5), for a query $q = (\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2)$ comprising k feature vectors $\mathbf{x} = (x_1, \ldots, x_k)$, and two decision vectors $\mathbf{y}_1 = (y_1, \ldots, y_k)$ and $\mathbf{y}_2 = (y'_1, \ldots, y'_k)$, such a noiseless oracle deterministically outputs

$$\mathcal{O}_k(q = (\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2)) = \mathbb{I}\left[u^*(\mathbf{x}, \mathbf{y}_1) \geq u^*(\mathbf{x}, \mathbf{y}_2)\right] \ ,$$

where recall that $u^*(\mathbf{x}, \mathbf{y}) = \sum_{i \in [k]} u^*(x_i, y_i)$ is the sum of the utilities under u^* for the tuple $(\mathbf{x}, \mathbf{y})$. In the binary prediction setup, such queries allow a learner to specify a set of k instances $\mathbf{x}$ and a subset $S_q \subset \mathbf{x}$ and ask the oracle "whether correctly predicting instances in S_q has higher utility or the instances in the complement $\mathbf{x} \setminus S_q$?".

Recall that Proposition 2.1 shows that excess risk for the plug-in estimator can be bounded by the worst-error $|u^*_{\mathsf{gap}}(x_i) - \hat{u}_{\mathsf{gap}}(x_i)|$ over the set of sampled datapoints S. To obtain such estimates, we introduce *Comptron* in Algorithm 1 which is a coordinate-wise variant of the classical perceptron algorithm [169]. At a high level, Comptron is an iterative procedure which estimates the utility gaps $u^*_{\mathsf{gap}}(x_i)$ for each x_i relative to the largest gap

$$u^*_{\mathsf{max}} := \max_{i \in [n]} u^*_{\mathsf{gap}}(x_i) \leq 1. \tag{2.14}$$

At each iteration t, the queries $q_{i,t}$ are selcted such that $\hat{u}^{t-1}_{\mathsf{gap}}(\mathbf{x}, \mathbf{y}_1) > \hat{u}^{t-1}_{\mathsf{gap}}(\mathbf{x}, \mathbf{y}_2)$ under the current estimates $\hat{u}^{t-1}_{\mathsf{gap}}$. If the oracle's response is $r_{i,t} = 1$, the estimates are consistent with the response and it keeps the current estimate. On the other hand, if the response $r_{i,t} = 0$, the algorithm decreases its current estimate of the i^{th} datapoint in order to be consistent with this query. Comptron repeats the above procedure for $T = \log_2 k - 1$ timesteps and finally outputs the estimates $\hat{u}^T_{\mathsf{gap}}$.

It is worth highlighting here that Comptron initializes all the estimates as the largest gap, that is, $\hat{u}^0_{\mathsf{gap}}(x_i) = u^*_{\mathsf{max}}$. Such an initialization is purely symbolic in nature and the algorithm *does not* require knowledge of this value. This is because the comparison queries $q_{i,t}$ allows the algorithm to compare the estimates $\hat{u}_{\mathsf{gap}}$ with u^*_{max} and the algorithm maintains its estimates $\hat{u}^t_{\mathsf{gap}}$ as a multiplicative factor of u^*_{max} for iterations t. Further, we can use symbolic estimates to output the plug-in estimator since it is invariant to scaling the utility gaps by a positive constant,

$$\begin{aligned}
\operatorname*{argmax}_{f \in \mathcal{F}} \sum_{i=1}^{n} \hat{u}(x_i, f(x_i)) \ &\equiv \ \operatorname*{argmax}_{f \in \mathcal{F}} \sum_{i=1}^{n} \hat{u}_{\mathsf{gap}}(x_i) \cdot \mathbb{I}[f(x_i) = y_i] \\
&\equiv \ \operatorname*{argmax}_{f \in \mathcal{F}} \sum_{i=1}^{n} \frac{\hat{u}_{\mathsf{gap}}(x_i)}{u^*_{\mathsf{max}}} \cdot \mathbb{I}[f(x_i) = y_i] \ .
\end{aligned}$$

The following lemma provides an upper bound on the estimation error of Comptron and shows that the output estimates $\hat{u}_{\mathsf{gap}}(x_i)$ are within a factor $O(\frac{u^*_{\mathsf{max}}}{k})$ of the true gaps $u^*_{\mathsf{gap}}(x_i)$.

Algorithm 1: Comptron: Comparison based Coordinate-Perceptron for estimating u^*_{gap}

Input: Datapoints $S = \{x_1, \ldots, x_n\}$, k-comparison oracle $\mathcal{O}_k$
Initialize: Set $T = \log_2 k - 1$
Obtain $y_i = \operatorname{argmax}_y u^*(x_i, y)$ for each i using 1-comparison.
Obtain index i_{max} using 2-comparisons such that $i_{\mathsf{max}} = \operatorname{argmax}_i u^*_{\mathsf{gap}}(x_i)$.
Set initial estimates $\hat{u}^0_{\mathsf{gap}} = [\hat{u}^0_{\mathsf{gap}}(x_1), \ldots, \hat{u}^0_{\mathsf{gap}}(x_n)] = u^*_{\mathsf{max}} := u^*_{\mathsf{gap}}(x_{i_{\mathsf{max}}})$.
(Note that exact value of u^*_{max} is not required since comparison queries are relative)
for $t = 1, \ldots, T$ **do**
  for $i = 1, \ldots, n$ **do**
    Denote by $\lambda = \frac{k}{2u^*_{\mathsf{max}}}\left(\hat{u}^{t-1}_{\mathsf{gap}}(x_i) - \frac{u^*_{\mathsf{max}}}{2^t}\right)$ and query $q_{i,t} = (\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2)$ where

$$\mathbf{x} = (\underbrace{x_i, \ldots, x_i}_{\frac{k}{2} \text{ times}}, \underbrace{x_{i_{\mathsf{max}}}, \ldots, x_{i_{\mathsf{max}}}}_{\lambda \text{ times}}), \quad \mathbf{y}_1 = (\underbrace{y_i, \ldots, y_i}_{\frac{k}{2} \text{ times}}, \underbrace{1 - y_{i_{\mathsf{max}}}, \ldots, 1 - y_{i_{\mathsf{max}}}}_{\lambda \text{ times}}), \quad \mathbf{y}_2 = 1 - \mathbf{y}_1.$$

    Query oracle $\mathcal{O}_k$ with $q_{i,t}$ and receive response $r_{i,t}$.
    Update $\hat{u}^t_{\mathsf{gap}}(x_i) = \hat{u}^{t-1}_{\mathsf{gap}}(x_i) - \mathbb{I}[r_{i,t} = 0] \cdot \frac{u^*_{\mathsf{max}}}{2^t}$.
Output: Gap estimates $\hat{u}^T_{\mathsf{gap}}$

Lemma 2.1 (Estimation error of Algorithm 1)**.** *Given access to datapoints $S = \{x_1, \ldots, x_n\}$ and k-comparison oracle $\mathcal{O}_k$, Comptron (Algorithm 1) uses $O(n \log k)$ queries to the oracle and produces estimates $\hat{u}_{\mathsf{gap}}$ such that*

$$\max_{i \in [n]} \left| \hat{u}_{\mathsf{gap}}(x_i) - u^*_{\mathsf{gap}}(x_i) \right| \leq \frac{2u^*_{\mathsf{max}}}{k} \,. \tag{2.15}$$

We defer the proof of the lemma to Appendix A.1. The proof proceed via an inductive argument where we show that the confidence interval around $u^*_{\mathsf{gap}}(x_i)$ shrinks by a factor of $\frac{1}{2}$ in each iteration for every datapoint x_i. Given the above estimation error guarantee for Comptron, the following corollary combines these with the excess risk bounds of Proposition 2.1 to obtain an upper bound on the excess risk of $\hat{f}_{k,n}$.

Corollary 2.1. *Consider the binary decision making problem with label space $\mathcal{Y} = \{0, 1\}$. Given n datapoints $\{x_1, \ldots, x_n\}$ such that each $x_i \sim \mathcal{D}_x$, the plug-in estimate $\hat{f}_{k,n}$ from equation (2.7), when instantiated with the output of Comptron (Algorithm 1), satisfies*

$$\mathsf{err}(\hat{f}_{k,n}, \mathcal{F}; u^*) \leq 2 \cdot \sup_{f \in \mathcal{F}} \left(|U(f; u^*) - \hat{U}(f; u^*)| \right) + \frac{2u^*_{\mathsf{max}}}{k} \cdot \left(\frac{1}{n} \sum_{i=1}^{n} \mathbb{I}[f_{\mathsf{ERM}}(x_i) \neq y_i] \right) \,.$$

We defer the proof of the corollary to Appendix A.1. Corollary 2.1 exhibits the advantage of using higher-order comparisons for the learning with unknown utilities problem – as the

order k increases, the error of the plug-in estimate decreases additively as $O\left(\frac{1}{k}\right)$. It is worth noting here that while the higher-order comparisons allow the learner to better estimate the underlying utilities, the problem gets harder from the side of the human expert. Indeed, with higher values of k, the expert is required to compare utilities across k different possible situations which can make the elicitation a harder task.

While the results in this section exhibit how the excess risk $\mathsf{err}(\hat{f}_{k,n}; \mathcal{F})$ varies as a function of k, they rely on the oracle responses being noiseless. In the next section, we consider the setup where the oracle responses can be noisy and propose a robust version of the Comptron algorithm for learning in this scenario.

Estimating u^*_{gap} with noisy oracle

In contrast to the deterministic noiseless oracle of the previous section, here, we consider learning with unkown utilities when the oracle $\mathcal{O}_k$ can output noisy responses to each query. Recall from equation (2.5), for any query q, the noisy k-comparison oracle the correct response with probability $1 - \eta_q$ and flips the response with probability η_q for some value of $\eta_q < \frac{1}{2}$. While we allow this error probability to vary across different queries, we assume that this error is bounded uniformly across all queries by some constant $\eta < \frac{1}{2}$.

Assumption 2.1. *For the noisy k-comparison oracle described in equation (2.5), we have that $\eta_q \leq \eta < \frac{1}{2}$ for all queries q.*

From an algorithmic perspective, it is well known that the perceptron algorithm itself is not noise-stable and can oscillate if there are datapoints x which have noisy labels. In order to overcome this limitation, several noise-robust perceptron variants have been proposed in the literature; see [117] for an extensive review.

We build on this line of work and present Rob-Comptron (Algorithm 2), a noise-robust variant of the deterministic Comptron algorithm. The main difference is the presence of an additional inner-loop with index j which repeatedly queries $q_{i,t}$ for $J = \tilde{O}\left(\frac{1}{(1-2\eta)^2}\right)$ times. In each iteration, the update is again a coordinate-wise perceptron update which matches the prediction of the current estimate with the average of the oracle responses. Such an averaging has been previously used in the context of learning halfspaces from noisy data both in a passive [43] and active [216] framework.

The following lemma, whose proof we defer to Appendix A.1, provides an upper bound on the estimation error of the gap estimates produced by Rob-Comptron.

Lemma 2.2 (Estimation error of Algorithm 2)**.** *Given access to datapoints $S = \{x_1, \ldots, x_n\}$ and noisy k-comparison oracle $\mathcal{O}_k$ satisfying Assumption 2.1 with parameter η, Rob-Comptron (Algorithm 2) uses $O\left(\frac{n}{(1-2\eta)^2} \cdot \log k \cdot \log \frac{n \log k}{\delta}\right)$ queries and produces estimates $\hat{u}_{\mathsf{gap}}$ such that*

$$\max_{i \in [n]} \left| \hat{u}_{\mathsf{gap}}(x_i) - u^*_{\mathsf{gap}}(x_i) \right| \leq \frac{2u^*_{\mathsf{max}}}{k} \,, \tag{2.16}$$

with probability at least $1 - \delta$.

Algorithm 2: Rob-Comptron: Robust Comptron for estimating u^*_{gap} with noisy oracle

Input: Datapoints $S = \{x_1, \ldots, x_n\}$, k-comparison oracle $\mathcal{O}_k$, noise level η, confidence δ

Initialize: $T = \log_2 k - 1$, $J = \frac{8}{(1-2\eta)^2} \log\left(\frac{nT}{\delta}\right)$

Obtain $y_i = \operatorname{argmax}_y u^*(x_i, y)$ for each i using 1-comparison.

Obtain index i_{max} using 2-comparisons such that $i_{\mathsf{max}} = \operatorname{argmax}_i u^*_{\mathsf{gap}}(x_i)$.

Set initial estimates $\hat{u}^0_{\mathsf{gap}} = [\hat{u}^0_{\mathsf{gap}}(x_1), \ldots, \hat{u}^0_{\mathsf{gap}}(x_n)] = u^*_{\mathsf{max}}$ symbolically

for $t = 1, \ldots, T$ **do**

  for $i = 1, \ldots, n$ **do**

    Denote by $\lambda = \frac{k}{2u^*_{\mathsf{max}}} \left(\hat{u}^{t-1}_{\mathsf{gap}}(x_i) - \frac{u^*_{\mathsf{max}}}{2^t}\right)$

    Set query $q_{i,t} = (\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2)$ where

$$\mathbf{x} = (\underbrace{x_i, \ldots, x_i}_{\frac{k}{2} \text{ times}}, \underbrace{x_{i_{\mathsf{max}}}, \ldots, x_{i_{\mathsf{max}}}}_{\lambda \text{ times}}), \quad \mathbf{y}_1 = (\underbrace{y_i, \ldots, y_i}_{\frac{k}{2} \text{ times}}, \underbrace{1 - y_{i_{\mathsf{max}}}, \ldots, 1 - y_{i_{\mathsf{max}}}}_{\lambda \text{ times}}), \quad \mathbf{y}_2 = 1 - \mathbf{y}_1.$$

    for $j = 1, \ldots, J$ **do**

      Query oracle $\mathcal{O}_k$ with $q_{i,t}$ and receive response $r_{i,j,t}$.

    Update $\hat{u}^t_{\mathsf{gap}}(x_i) = \hat{u}^{t-1}_{\mathsf{gap}}(x_i) - \mathbb{I}[\frac{1}{J}\sum_j r_{i,j,t} < \frac{1}{2}] \cdot \frac{u^*_{\mathsf{max}}}{2^t}$.

Output: Gap estimates $\hat{u}^T_{\mathsf{gap}}$

In comparison to Comptron which requires $O(n \log k)$ queries to the comparison oracle, the robust variant Rob-Comptron requires a fraction $\frac{1}{(1-2\eta)^2}$ more queries to achieve a similar estimation error. Such an increase in query complexity is typical of learning with such noisy oracles in the binary classification setup [17, 19, 61, 216].

Similar to Corollary 2.1 in the previous section, we can combine the above high-probability bound on the estimation error to obtain a bound on the excess risk which scales as $\frac{1}{k}$ with the order k of the comparison oracle.

Corollary 2.2. *Consider the binary decision making problem with label space $\mathcal{Y} = \{0, 1\}$. Given n datapoints $\{x_1, \ldots, x_n\}$ such that each $x_i \sim \mathcal{D}_x$, the plug-in estimate $\hat{f}_{k,n}$ from equation (2.7), when instantiated with the output of Comptron (Algorithm 1), satisfies*

$$\mathsf{err}(\hat{f}_{k,n}, \mathcal{F}; u^*) \leq 2 \cdot \sup_{f \in \mathcal{F}} \left(|U(f; u^*) - \hat{U}(f; u^*)|\right) + \frac{2u^*_{\mathsf{max}}}{k} \cdot \left(\frac{1}{n}\sum_{i=1}^{n} \mathbb{I}[f_{\mathsf{ERM}}(x_i) \neq y_i]\right) .$$

with probability at least $1 - \delta$.

We omit the proof of this corollary since it essentially follows the same steps as that for Corollary 2.1. This corollary establishes that by increasing the query complexity by a factor

of $O\left(1/(1-2\eta)^2\right)$, one can recover the same additive $\frac{1}{k}$ excess risk bound of the deterministic setup. Combined, Corollaries 2.1 and 2.2 establish the trade-offs in the reduction of the excess risk while eliciting more complex information about the underlying utility u^* through the k-comparison oracle.

Information-theoretic lower bounds

In the previous section, we studied the learning with unknown utility problem from an algorithmic perspective and showed that the plug-in estimator with Comptron estimates $\hat{u}$ achieve an excess risk bound which scales as $O(\frac{1}{k})$ with the order k of the comparison. In this section, we ask whether such a scaling of the error term is optimal and study this lower bound question from an information-theoretic perspective.

Recall from Theorem 2.2 that the excess risk decomposes into two terms: (i) a uniform convergence term for the decision class $\mathcal{F}$ with respect to utility function u^* and (ii) an estimation error term corresponding to how well $\hat{u}_k$ approximates u^* on the sampled data-points. When the underlying utility function u^* is known, classical results from the learning theory literature the uniform convergence complexity term is in general unavoidable [see 178, Theorem 6.8]. With this, we take the infinite-data limit, where the learner is assumed to have access to the distribution $\mathcal{D}_x$, and study whether the excess error of $O(\frac{1}{k})$ is necessary.

Our notion of minimax risk is based on the subset of utility functions which cannot be distinguished by any learner with access to a k-comparison oracle. Formally, given any oracle $\mathcal{O}_k(\cdot\,; u^*)$, where we have made the dependence on the utility u^* explicit, we denote by $\mathcal{U}_{k,u^*}$ the subset of utility functions in the class $\mathcal{U}$ which are consistent with the responses of $\mathcal{O}_k(\cdot\,; u^*)$. With this, we define the information-theoretic minimax risk $\mathfrak{M}_k(\mathcal{F}, \mathcal{D}_x)$ with respect to the function class $\mathcal{F}$ and distribution $\mathcal{D}_x$ as

$$\mathfrak{M}_k(\mathcal{F}, \mathcal{D}_x) := \sup_{\mathcal{O}_k(\cdot\,;u^*)} \inf_{p \in \Delta_{\mathcal{F}}} \sup_{u \in \mathcal{U}_{k,u^*}} \mathbb{E}_{f \sim p}\left[\mathsf{err}(f, \mathcal{F}; u)\right] \,, \tag{2.17}$$

where the infimum is taken over all procedures which take as input the distribution $\mathcal{D}_x$ over the instances and access to a k-comparison oracle, and output a possibly randomized estimate $p \in \Delta_{\mathcal{F}}$. The above notion of minimax risk can be viewed as a three-stage game between the learner and the environment. The sequence of supremum and infimum depicts the order in which information is revealed in this game. The environment first selects a k-query oracle $\mathcal{O}(\cdot\,; u^*)$ with underlying utility u^*. The learner is then provided access to the underlying distribution $\mathcal{D}_x$, function class $\mathcal{F}$ and the oracle $\mathcal{O}(\cdot\,; u^*)$ based on which it outputs a possibly randomized decision function given by $p \in \Delta_{\mathcal{F}}$. The environment is then allowed to select the worst-case utility u such that it is consistent with the k-oracle $\mathcal{O}(\cdot\,; u^*)$ and the learner is evaluated in expectation over this chosen utility. We call this the minimax risk of learning with respect to class $\mathcal{F}$ and distribution $\mathcal{D}_x$.

Our next main result shows that there exist instances of the binary prediction problem $(\mathcal{F}, \mathcal{D}_x)$ such that the minimax risk $\mathfrak{M}_k(\mathcal{F}, \mathcal{D}_x)$ is lower bounded by $\frac{1}{k}$ for any $k \geq 2$ up to some universal constants. Observe that this matches the corresponding upper bounds

obtained in Corollaries 2.1 and 2.2 exhibiting that the proposed plug-in estimator in equation (2.7) with Comptron (Rob-Comptron for noisy oracle) utilities is indeed minimax optimal for the binary prediction setup.

Theorem 2.3. *There exists a universal constant $c > 0$ such that for any $k \geq 2$, there exist a binary prediction problem instance $(\mathcal{F}, \mathcal{D}_x)$ such that*

$$\mathfrak{M}_k(\mathcal{F}, \mathcal{D}_x) \geq \frac{c}{k} \, .$$

A few comments on Theorem 2.3 are in order. First, the above result shows a family of lower bounds for our learning with unknown utilities framework – one for each value of the order k. Specifically, it shows that for every $k \geq 2$, there exists a worst-case instance such that any algorithm will incur an error of $\Omega(\frac{1}{k})$. Compare this with the upper bounds on excess risk from the previous section. In the limit of infinite data, Corollaries 2.1 and 2.2 exhibit that the excess risk $\mathsf{err}(\hat{f}_{k,n}, \mathcal{F}; u^*) = O(\frac{1}{k})$ for the plug-in estimator $\hat{f}_{k,n}$. This establishes that the plug-in estimator with Comptron and Rob-Comptron utility estimates is indeed minimax optimal.

Proof. In order to establish a lower bound on the minimax risk $\mathfrak{M}_k$, we will construct two utility functions $u_1, u_2 \in \mathcal{U}$ such that the k-comparison oracle has identical responses for both these utility functions. For the purpose of our construction, we will consider noiseless oracle; the problem only becomes harder for the learner if the oracle responses are noisy. Given these two utility functions, we next show that their maximizers f_1 and f_2 are different for some function class $\mathcal{F}$. We then combine these two insights to obtain the final minimax bound.

For our lower bound construction, we will focus on a setup where the features are one dimensional with $\mathcal{X} = \mathbb{R}$ and the linear decision function class

$$\mathcal{F}_{\mathsf{lin}} = \{f_a \mid f_a(x) = \mathrm{sign}(ax), \; a \in [-1, 1]\} \, .$$

Recall that for any point x, we represent by $u_{\mathsf{gap}}(x) = u(x, y_x) - u(x, \bar{y}_x)$ the utility gain corresponding to the function u. Before constructing the explicit example, we present a technical lemma which highlights a limitation of a k-comparison oracle – it establishes that a k-oracle will not be able to distinguish utility functions for which the utility gaps are in the range $(1 - \frac{1}{k}, 1)$.

Lemma 2.3. *Consider any utility functions $u_1, u_2 \in \mathcal{U}$. Let datapoints x have utility gain $u^i_{\mathsf{gap}}(x)$ for $i = \{1, 2\}$. For any two points x_1, x_2 such that*

$$u^1_{\mathsf{gap}}(x_1) = u^2_{\mathsf{gap}}(x_1) = u_{\mathsf{gap}}(x_1) \quad \textit{and} \quad \left(1 - \frac{1}{k}\right) \cdot u_{\mathsf{gap}}(x_1) \leq u^i_{\mathsf{gap}}(x_2) \leq u_{\mathsf{gap}}(x_1) \, ,$$

the oracle responses for any query $q = (\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2)$ comprising points x_1 and x_2 are identical for $u^* = u_1$ or $u^* = u_2$.

We defer the proof of the above lemma to Appendix A.1. Taking this as given, we proceed with our lower bound construction.

Utility functions u_1 and u_2. Our construction considers two datapoints $x_+ = +1$ and $x_- = -1$ and two utility functions u and $\tilde{u}$ satisfying

$$u_1(x_+, 1) > u_1(x_+, 0) \quad \text{and} \quad u_1(x_-, 1) > u_1(x_-, 0)\ ,$$
$$u_2(x_+, 1) > u_2(x_+, 0) \quad \text{and} \quad u_2(x_-, 1) > u_2(x_-, 0).$$

Observe that under these utilities, any function $f_a \in \mathcal{F}_{\mathsf{lin}}$ can make a correct decision for either point x_+ or point x_- but not for both simultaneously. Given these datapoints, the two utility functions are given by

$$u_1(x_+, 1) = 1, \quad u_1(x_-, 1) = 1 - \gamma_1 \ \text{ where } \gamma_1 = \frac{1}{2(3k+1)}$$
$$u_2(x_+, 1) = 1, \quad u_2(x_-, 1) = 1 - \gamma_2 \ \text{ where } \gamma_2 = \frac{2}{(3k+1)},$$

and $u_i(x, 0) = 0$ for both $i = \{1, 2\}$. Observe that both γ_1, γ_2 have been set to satisfy the conditions of Lemma 2.3, that is,

$$\left(1 - \frac{1}{k}\right) \cdot u_{\mathsf{gap}}(x_+) \le u^i_{\mathsf{gap}}(x_-) \le u_{\mathsf{gap}}(x_+) \text{ for } i = \{1, 2\}.$$

Distribution $\mathcal{D}_x$. For any $k > 2$, consider the distribution $\mathcal{D}_x$ over the points $\{x_+, x_-\}$ such that

$$\Pr(x = x_+) = \frac{3k}{6k+1} \quad \text{and} \quad \Pr(x = x_-) = \frac{3k+1}{6k+1}.$$

By Lemma 2.3, we have that using the k-comparison oracle, no learner can distinguish between the utility functions u_1 and u_2 on the distribution $\mathcal{D}_x$. Further, recall that any classifier $f_a \in \mathcal{F}_{\mathsf{lin}}$ can either predict x_+ or x_- correctly. We now obtain a bound on the excess risk $\mathsf{err}(f_a, \mathcal{F}; u)$ for both these cases separately.

Case 1: $f_a(x_+) = 1$. In this case, the utility gap is maximized by setting the utility $u = u_1$ in the minimax risk. The corresponding excess risk is given by

$$\mathsf{err}(f_a, \mathcal{F}; u_1) = \frac{(3k+1)(1-\gamma_1)}{6k+1} - \frac{3k}{6k+1} = \frac{1}{2(6k+1)}. \tag{2.18}$$

Case 2: $f_a(x_-) = 1$. In this case, the utility gap is maximized by setting the utility $u = u_2$ and the excess risk is given by

$$\mathsf{err}(f_a, \mathcal{F}; u_2) = \frac{3k}{6k+1} - \frac{(3k+1)(1-\gamma_2)}{6k+1} = \frac{1}{(6k+1)}. \tag{2.19}$$

Noting that any predictor $\hat{f}$ will output a function corresponding to one of the two cases above and combining equations (2.18) and (2.19) establishes the desired claim. □

While the information theoretic results of this section showed that the plug-in estimator is minimax optimal, the next section focuses on whether this estimator is able to *adapt* to easier problem instances – specifically, whether our estimation procedures Comptron and Rob-Comptron are optimal for every problem instance? We answer this in the negative and introduce a new estimator which is instance optimal. However, such an adaptivity to easier instances comes at the cost of an exponential query complexity.

2.5 Instance-optimal guarantees for binary prediction

In the previous section, we proposed query-efficient algorithms, Comptron and Rob-Comptron, for learning a function $\hat{f}_{k,n}$ with small excess risk using only $\tilde{O}(n \log k)$ queries to the k-comparison oracle. Further, the upper bounds in Corollaries 2.1 and 2.2 along with the lower bound of Theorem 2.3 establish that our proposed algorithms are indeed minimax optimal over the class of utility functions $\mathcal{U}$. Given this, it is natural to ask whether our proposed algorithms are instance wise-optimal, that is, do they achieve the best possible excess-risk bounds for *all* $u^* \in \mathcal{U}$?

To simplify our presentation, we study this question at the population level,[5] where we assume that the learner has access to the underlying distribution $\mathcal{D}_x$. This allows us to focus on the excess risk as a function of the order k of the comparison oracle and ignore the uniform convergence term. We also restrict our attention to the deterministic noiseless oracle since one can reduce the noisy oracle to the noiseless oracle by using the averaging technique presented in Section 2.4.

The following proposition shows that the plug-in estimator with Comptron utilities are *not instance-optimal*, that is, it does not adapt to the hardness of the learning with unknown utilities problem instance. Specifically, it constructs a problem instance $(\mathcal{F}, \mathcal{D}_x)$ with a noiseless oracle and shows that the estimate[6] $\hat{f}_k$ from equation (2.7) with Comptron utility estimates has an excess risk of $\frac{1}{k}$ while there exists an estimator, which uses all k-queries and is able to achieve zero excess risk.

Recall that for any utility $u^* \in \mathcal{U}$, we denote by $\mathcal{U}_{k,u^*}$ the subset of utility functions in the class $\mathcal{U}$ which are indistinguishable from u^* under the k-comparison oracle $\mathcal{O}(\cdot\,; u^*)$.

Proposition 2.2 (Plug-in with Comptron estimates is not instance-optimal)**.** *For every $k > 2$, there exists an binary prediction instance $(\mathcal{F}, \mathcal{D}_x)$ along with an oracle $\mathcal{O}_k$ such that*

[5] Our analysis could be extended to the finite sample setup using the bound obtained in Theorem 2.2.

[6] Since we are working at the population level, we have dropped the subscript n from $\hat{f}_{k,n}$

a) The error of the plug-in estimate $\hat{f}_k$ from equation (2.7) with estimated utilities $\hat{u}_k$ from Comptron (Algorithm 1) is non-zero, that is,

$$\mathsf{err}(\hat{f}_k, \mathcal{F}; u^*) = \frac{1}{k}.$$

b) There exists an optimal predictor $\tilde{f}$ with zero excess-risk, that is,

$$\sup_{u \in \mathcal{U}_{k,u^*}} \mathsf{err}(\tilde{f}, \mathcal{F}; u) = 0.$$

We make a few remarks about the proposition. While the first part of the proposition shows that the excess risk $\mathsf{err}(\hat{f}_k, \mathcal{F}; u^*) = \frac{1}{k}$, the second part makes a stronger claim about the performance of $\tilde{f}$ on all utilities $u \in \mathcal{U}_{k,u^*}$. This shows that the predictor $\tilde{f}$ performs well when evaluated on an entire neighborhood around the true utility u^*. We defer the proof of the proposition to Appendix A.2.

Having established that our estimators from the previous section are not adaptive, we introduce a notion of *local minimax risk* and study estimators which are instance-optimal. We begin by precisely defining this notion of instance-wise minimax optimality. Recall from Section 2.4, our notion of minimax risk $\mathfrak{M}_k(\mathcal{F}, \mathcal{D}_x)$ was a worst-case notion – the minimax risk was defined as a supremum over all oracles $\mathcal{O}_k(\cdot\,; u^*)$. We extend this global minimax notion to a local minimax one. In particular, for any $u^* \in \mathcal{U}$, we define the local minimax risk around u^* as

$$\mathfrak{M}_k(\mathcal{F}, \mathcal{D}_x; u^*) := \inf_{\hat{f}} \sup_{u \in \mathcal{U}_{|u^*}} \left[\mathsf{err}(\hat{f}, \mathcal{F}; u) \right], \tag{2.20}$$

where the infimum is again over the set of all estimators which output a function $\hat{f} \in \mathcal{F}$ given access to distribution $\mathcal{D}_x$ and k-comparison oracle $\mathcal{O}_k$. Observe that this local notion of minimax risk concerns the performance of an algorithm $\hat{f}$ around a specific instance u^* as compared to the worst-case instance.

For any utility function $u \in \mathcal{U}$, we define its population maximizer $f_u \in \operatorname{argmax}_{f \in \mathcal{F}} U(f; u)$. With this notation, our next theorem provides a lower bound on this local minimax risk in terms of a local modulus of continuity with respect to the set $\mathcal{U}_{k,u^*}$.

Theorem 2.4 (Local minimax lower bound)**.** *For any distribution $\mathcal{D}_x$ over feature space $\mathcal{X}$, utility function $u^* \in \mathcal{U}$, function class $\mathcal{F}$ and order k of the comparison oracle, the local minimax risk*

$$\mathfrak{M}_k(\mathcal{F}, \mathcal{D}_x; u^*) \geq \frac{1}{2} \cdot \sup_{u_1, u_2 \in \mathcal{U}_{k,u^*}} \left(U(f_{u_1}; u_1) - U(f_{\frac{u_1+u_2}{2}}; u_1) \right). \tag{2.21}$$

Proof. Consider any two utility functions $u_1, u_2 \in \mathcal{U}_{k,u^*}$ and let $\bar{u} = \frac{u_1+u_2}{2}$. We can then lower bound the minimax risk as

$$\begin{aligned}\mathfrak{M}_k(\mathcal{F}, \mathcal{D}_x; u^*) &\geq \inf_{f\in\mathcal{F}} \left(\frac{1}{2}\,\mathsf{err}(f, \mathcal{F}; u_1) + \frac{1}{2}\,\mathsf{err}(f, \mathcal{F}; u_2)\right) \\ &= \frac{1}{2}\,\mathsf{err}(f_{\bar{u}}, \mathcal{F}; u_1) + \frac{1}{2}\,\mathsf{err}(f_{\bar{u}}, \mathcal{F}; u_2) \\ &\geq \frac{1}{2}\left(U(f_{u_1}; u_1) - U(f_{\bar{u}}; u_1)\right),\end{aligned}$$

where the last equality follows by noting that $\mathsf{err}(f_{\bar{u}}, \mathcal{F}; u_2) \geq 0$. Since the above holds for any choice of u_1, u_2, the desired bound follows by taking a supremum over these values. □

A few comments on Theorem 2.4 are in order. The theorem establishes that the local minimax risk $\mathfrak{M}_k(\mathcal{F}, \mathcal{D}_x)$ is lower bounded by a local modulus of continuity,

$$\sup_{u_1,u_2\in\,\mathcal{U}_{k,u^*}} \left(U(f_{u_1}; u_1) - U(f_{\frac{u_1+u_2}{2}}; u_1)\right), \tag{2.22}$$

which captures the worst-case variation in the performance of utility maximizers of utility in a neighborhood of u^*. For any two utilities $u_1, u_2 \in \mathcal{U}_{k,u^*}$, it measures the performance drop in the utility of a learner uses the maximizer $f_{\frac{u_1+u_2}{2}}$ in place of f_{u_1} when the underlying utility is u_1.

Given this lower bound on the local minimax risk $\mathfrak{M}_k(\mathcal{F}, \mathcal{D}_x)$, it is natural to ask whether this local modulus of continuity exactly captures the instance-specific hardness of the problem. To this end, our next result answers this in the affirmative. In particular, it shows that for any u^*, the randomized minimax robust estimator $p_{\mathsf{rob}} \in \Delta_{\mathcal{F}}$, given by

$$p_{\mathsf{rob}} \in \operatorname*{argmin}_{p\in\Delta_{\mathcal{F}}} \sup_{u\in\mathcal{U}_{k,u^*}} \mathbb{E}_{f\sim p}[\mathsf{err}(f, \mathcal{F}; u)], \tag{2.23}$$

(nearly-)obtains the same excess-risk bound as that given by the lower bound in Theorem 2.4.

Theorem 2.5 (Upper bounds for p_{rob})**.** *For any distribution $\mathcal{D}_x$ over feature space $\mathcal{X}$, utility function $u^* \in \mathcal{U}$ and function class $\mathcal{F}$, the expected excess risk of the randomized estimator given by the distribution $p_{\mathsf{rob}} \in \Delta_{\mathcal{F}}$ is*

$$\begin{aligned}\mathbb{E}[\mathsf{err}(p_{\mathsf{rob}}, \mathcal{F}; u^*)] &= \sup_{p_u} \left(\mathbb{E}_{u'\sim p_u}\left[U(f_{u'}; u') - U(f_{p_u}; u')\right]\right) \\ &\leq \sup_{u_1,u_2\in\,\mathcal{U}_{k,u^*}} \left(U(f_{u_1}; u_1) - U(f_{u_2}; u_1)\right),\end{aligned} \tag{2.24}$$

where the distribution $p_u \in \Delta_{\mathcal{U}_{k,u^*}}$ is over the space of utility functions consistent with u^*.

We defer the proof of Theorem 2.5 to Appendix A.2. Compared with the lower bound of Theorem 2.4, the bound in (2.24) shows that the local minimax risk can indeed be upper

bounded by a similar local modulus of continuity. Observe that the while the lower bound evaluates the performance loss of the maximizer $f_{\frac{u_1+u_2}{2}}$, the upper bound is evaluated on f_{u_2}. While the minimax estimator p_{rob} in equation (2.23) is defined at the population level, we can naturally extend it to the finite sample regime as

$$\hat{p}_{\mathsf{rob},n} \in \operatorname*{argmin}_{p\in\Delta_{\mathcal{F}}} \sup_{u\in\hat{\mathcal{U}}_{k,u^*}} \mathbb{E}_{f\sim p}[\hat{U}(f_u;u) - \hat{U}(f;u)] \tag{2.25}$$

where the class of utilities $\hat{\mathcal{U}}_{k,u^*}$ represents the set of all n-dimensional vectors in $[0,1]^n$ which are consistent with responses to all k-queries on the set of sampled datapoints S. Using a similar analysis as in Theorem 2.2, one can then upper bound the excess risk of this estimator in terms of the local modulus on the dataset S and an additional uniform convergence term.

In comparison to the Comptron procedure which uses $O(n \log k)$ queries to the comparison oracle for estimating utilities, the estimator $\hat{p}_{\mathsf{rob},n}$ uses $O(n^k)$ queries to construct the set $\hat{\mathcal{U}}_{k,u^*}$. Thus, while this estimator adapts to the problem hardness, such an adaptation comes at the cost of an exponential increase in query complexity. Achieving instance-optimality by using fewer queries is an interesting question for future research.

Chapter 3

Learning with misspecified human models

3.1 Introduction

The expanding interest in the area of reward learning stems from the concern that it is difficult or even impossible to specify what we actually want AI agents to optimize when it comes to increasingly complex, real-world tasks [227, 148]. At the core of reward learning is the idea that human behavior serves as evidence about the underlying objective. Therefore, the fundamental assumption we are making when pursuing research in this area is that by modeling the link between human behavior and the desired objective, we can draw useful inferences about the latter from the former.

Research on inferring rewards typically uses noisy-rationality as a model for human behavior: the human will take higher value actions with higher probability. This has its roots in mathematical psychology and economics with Luce's Axiom of Choice [137], which later became the Luce-Shephard choice rule [138]. It has enjoyed great success in a variety of reward inference applications [225, 206, 215], but researchers have also started to come up against its limitations [165][cite here work on inferring beta?]. This is not surprising, given decades of research in *behavioral* economics that has identified a deluge of systematic biases people have when making decisions on how to act, like myopia/hyperbolic discounting [95], optimism bias [182], prospect theory [114], and many more [193, 65]. These and others end up creeping into the reward learning tasks AI researchers are interested in. For instance, in shared autonomy, a human operating a robotic arm to grasp objects may behave suboptimally due to being unfamiliar with the control interface or the robot's dynamics.

Recent work in reward learning attempts to go beyond noisy rationality and consider more accurate models of human behavior, by for instance looking at biases as variations on the Bellman update [48], modeling the human's false beliefs [165] or learning their suboptimal perception process [166]. And while we might be getting closer, we will realistically never have a *perfect* model of human behavior.

This raises an obvious question: *Does the human model need to be perfect in order for reward inference to be successful?* On the one hand, if small errors in the model can lead to catastrophic error in inference, the entire framework of reward learning seems ill-fated, especially as it applies to value alignment: we will never have perfect models, and we will therefore never have guarantees that the agent does not do something catastrophically bad with respect to what people actually value. On the other hand, if we can show that as our models improve, we have a guarantee that reward accuracy also improves, then there is hope: yes, modeling human behavior is difficult, but at least we know that as we get closer, our AI agents will be more and more aligned with us.

The main goal of this work is to study whether we can bound the reward inference error by some function of the distance between the assumed and true human model. We study this question both theoretically and empirically. Our first result is a negative answer: we show that given a finite dataset of demonstrations, it is possible to hypothesize a true human model that generated the dataset and is "close" to the assumed model, but results in arbitrarily large error in the reward we would infer via maximum likelihood estimation (MLE). This unfortunately holds for a very strong notion of closeness, where the assumed model needs to be close to the true one across every possible reward and every possible state. However, we also find reason for hope. We identify mild assumptions on the true human behavior, under which we can actually bound the reward inference error linearly by the error of the human model. Thus, if these assumptions hold, refining the human model will monotonically improve the accuracy of the learned reward. We also show how this bound simplifies for particular biases like false internal dynamics or myopia.

Empirically, we validate our theoretical conclusions on both diagnostic gridworld domains [85], as well as the Lunar Lander game, which involves continuous control over a continuous state space. First, we verify that under various simulated biases, when the conditions on the human model are likely to be satisfied, small divergences in human models do not lead to large reward errors. We also demonstrate the same finding when the bias is grounded by real human demonstration data. Overall, our results suggest an optimistic perspective on the framework of reward learning, and that efforts in improving human models will further enhance the quality of the inferred rewards.

Related Work

Inverse reinforcement learning (IRL) aims to use expert demonstrations, often from a human, to infer a reward function [151, 225]. Maximum-entropy (MaxEnt) IRL is a popular IRL framework that models the demonstrator as noisily optimal, maximizing reward while also randomising actions as much as possible [225, 224]. This is equivalent to modeling humans as Boltzmann rational. MaxEnt IRL is preferred in practice over Bayesian IRL [164], which learns a posterior over reward functions rather than a point estimate, due to better scaling in high-dimensional environments [215]. More recently, Guided Cost Learning [75] and Adversarial IRL [83] learn reward functions more robust to environment changes, but build off similar modeling assumptions as MaxEnt IRL. Gleave and Toyer [91] connected MaxEnt

IRL to maximum likelihood estimation (MLE), which is the framework that we consider in this work. One of the challenges with IRL is that rewards are not always uniquely identified from expert demonstrations [45, 118]. Since identifiability is orthogonal to the main message of our work–sensitivity to misspecified human models–we assume that the dataset avoids this ambiguity.

Recent IRL algorithms attempt to account for possible irrationalities in the expert [73, 165, 177]. Reddy, Dragan, and Levine [165] consider when experts behave according to an internal dynamics, and show that explicitly learning these dynamics improves accuracy of the learned reward. Shah et al. [177] propose learning general biases using demonstrations across similar tasks, but conclude that doing so without prior knowledge is difficult. Finally, Chan, Critch, and Dragan [48] show that knowing the type of irrationality the expert exhibits can improve reward inference over even an optimal expert. In this work, we do not assume the bias can be uncovered, but rather analyze how sensitive reward inference is to such biases.

More generally, reward learning is a specific instantiation of an inverse problem, which is well-studied in existing literature. In the framework of Bayesian inverse problems, prior work has analyzed how misspecified likelihood models affect the accuracy of the inferred quantity when performing Bayesian inference. Owhadi, Scovel, and Sullivan [155] showed that two similar models can lead to completely opposite inference of the desired quantity. Meanwhile, Sprungk [184] showed inference is stable under a different measure of distance between models. In this work, we also derive both instability and stability results, but consider a different problem of reward learning using MLE.

3.2 Problem formulation

Reward parameters. We consider Markov decision processes (MDP), which are defined by a tuple $(\mathcal{S}, \mathcal{A}, P, r, \gamma)$. Here, $\mathcal{S}, \mathcal{A}$ represent state and action spaces, $P(s'|s,a)$ and $r(s,a)$ represent the dynamics and reward function, and $\gamma \in (0,1)$ represents the discount factor. In this work, we are interested in the setting where the reward function r is unknown and needs to be inferred by a learner. We assume rewards are bounded $|r(s,a)| \leq R_{\max}$. We assume that the reward can be parameterized by *reward function parameters* $\theta \in \Theta$. We denote by $r(\cdot;\theta)$ the reward function with θ as parameters.

True vs. assumed human policy. Instead of having access to the reward, we observe the behavior of an "expert" demonstrator. Let $\pi^* : \Theta \times \mathcal{S} \to \Delta(\mathcal{A})$ be the reward-conditioned *demonstrator policy*, and $\mathcal{D} = \{(s_t, a_t)\}_{t=1}^n$ be a *dataset* of demonstrations provided to the learner, sampled from π^*. We use $(s,a) \sim w^\pi$ to denote that observations generated by policy π, where w^π denotes the discounted stationary distribution. We shorthand w^{π^*} as w^*. Finally, let $\tilde{\pi} : \Theta \times \mathcal{S} \to \Delta(\mathcal{A})$ be the *model* that the learner assumes generated the dataset; in practice, this is often the Boltzmann rational policy [225], but we can expect that to change as research in human models evolves.

Reward inference using the assumed policy. Many popular algorithms in inverse reinforcement learning (IRL) [225, 224] infer the reward function parameters via maximum-

likelihood estimation (MLE). This is because unlike Bayesian IRL methods that learn a posterior over rewards, such MLE methods are shown to scale to high-dimensional environments [215]. Using a dataset $\mathcal{D}$, the learner would estimate parameters

$$\tilde{\theta} = \arg\min_{\theta} \frac{1}{n} \sum_{t=1}^{n} -\log \tilde{\pi}(a_t \mid s_t; \theta) := \arg\min_{\theta} L(\theta; \tilde{\pi}, \mathcal{D}) \,. \tag{3.1}$$

Let θ^* be the true reward function parameters. Though θ^* cannot always uniquely determined in general [45, 118], for simplicity, we make the following assumption that θ^* is identifiable.

Assumption 3.1. *There exists a unique θ^* satisfying $\theta^* = \arg\min_\theta L(\theta; \pi^*, \mathcal{D})$.*

Though Assumption 3.1 is rather strong, we make it only because we view identifiability as orthogonal to the subject of our work–sensitivity to misspecified models.

Goal: effect of error in the model on the error in the inferred reward. The goal of our paper is to answer whether we can bound the distance between the inferred reward and the true reward, $d_\theta(\theta^*, \tilde{\theta})$, as a function of the distance between the assumed human model and the true human policy, $d_\pi(\pi^*, \tilde{\pi})$, for some useful notions of distance. If so, then we know that more accurate policies will monotonically improve the fidelity of the learned rewards. We discuss our choice of distances below.

Reward inference error. A majority of existing work in reward learning ultimately measures the performance of the policy optimized over the inferred reward [151, 225]. However, there are two issues with using this policy based distance metric: (1) we do not necessarily know what environments to evaluate on, as they could have different start distributions or dynamics than the training environment, and, (2) it is much difficult to disentangle errors in the inferred reward from suboptimality in the training of the policy. For these reasons, we investigate a more straight-forward distance metric, specifically the distance between the inferred and true parameters $d_\theta(\theta^*, \tilde{\theta}) = \|\tilde{\theta} - \theta^*\|_2^2$

Human model error. Since policies are probability distributions, we can measure error in the human model as the KL-divergence between the model and demonstrator policies. We consider two different instatiations of policy divergence. The first is a *worst-case policy divergence* that takes the supremum over all reward parameters and states:

$$d_\pi^{\mathsf{wc}}(\pi^*, \tilde{\pi}) = \sup_{\theta \in \Theta} \sup_{s \in \mathcal{S}} D_{\mathrm{KL}}(\pi^*(\cdot \mid s; \theta) || \tilde{\pi}(\cdot \mid s; \theta)) \,. \tag{3.2}$$

Alternatively, we consider a weaker–potentially more practical–divergence that is only over the true reward parameters θ^* and in expectation over states, which we dub the *weighted policy divergence*:

$$d_\pi^{\mathsf{w}}(\pi^*, \tilde{\pi}) = \mathbb{E}_{s \sim w^*} \left[D_{\mathrm{KL}}(\pi^*(\cdot \mid s; \theta^*) \;||\; \tilde{\pi}(\cdot \mid s; \theta^*)) \right] . \tag{3.3}$$

The weighted policy divergence only looks at the states visited under the true human behavioral policy π^* as compared to the worst case metric in eq. (3.2) which compares the policies on all states and rewards.

3.3 Worst-case instability of inference

We begin our theoretical analysis by exhibiting a negative result. We prove that even under the worst-case policy divergence from eq. (3.2), a small difference in the assumed policies $d_\pi^{\mathsf{wc}}(\pi^*, \pi) < \varepsilon$ can lead to a large inference error $d_\theta(\theta^*, \tilde{\theta})$. Note that for our lower bound construction, we use the much stronger notion of worst-case policy divergence and our theorem shows that despite the two policies being close on each state and reward pair, the inference procedure can be extremely unstable and lead to large errors.

Theorem 3.1. *There exists an MDP $\mathcal{M}$ such that for any policy error $\varepsilon > 0$, assumed model $\tilde{\pi}$, and dataset $\mathcal{D}$, there exists a demonstrator policy π^* that generates $\mathcal{D}$ such that the worst-case policy divergence satisfies $d_\pi^{\mathsf{wc}}(\pi^*, \tilde{\pi}) < \varepsilon$, and the reward inference error*

$$\|\tilde{\theta} - \theta^*\|_2^2 > \frac{1}{2} \sup_{\theta, \theta' \in \Theta} \|\theta - \theta'\|_2^2 .$$

A few comments on the theorem are in order. First, the theorem shows that there exists worst case MDPs such that even a small perturbation in the assumed human model can lead to large inference error – the inference error will be lower bounded by half the range of possible reward parameters. Second, note that the theorem works for an arbitrarily small perturbation $\varepsilon > 0$ for which we construct the MDP $\mathcal{M}$ with a continuous action space. Additionally, observe that the theorem holds true for any observed dataset $\mathcal{D}$. We are able to prove such a strong result by perturbing the policy π^* on only the *observed* state-action pairs in the dataset. We defer the proof of the theorem to Appendix B.1 but provide an illustrative example below that captures the key essence of the proof.

Illustrative example: Let us consider a stochastic bandit with continuous actions $\mathcal{A} \in (0, 1)$. Since bandits consist of a single, stationary state, we drop dependence on state in all quantities. The reward for choosing action $a \in \mathcal{A}$ is $r(a; \theta) = a^\theta (1 - a)^{1-\theta}$, for some parameter $\theta \in (0, 1)$. When θ is close to 0, the reward is higher for actions close to 0, and vice-versa when θ is close to 1.

For simplicity, let us only consider a dataset of a single action a_1. Let us consider a Boltzmann rational policy as the assumed model, namely $\tilde{\pi}(a; \theta) \propto \exp(r(a; \theta))$ and have the demonstrator policy π^* be an adversarial perturbation of $\tilde{\pi}$ that overestimates the reward of a_1:

$$\pi^*(a; \theta) \propto \begin{cases} \exp(r(a;\theta)) \left(\mathbb{1}\{a \notin (a_1 - \frac{\delta}{2}, a_1 + \frac{\delta}{2})\} + 10^9 \, \mathbb{1}\{a \in (a_1 - \frac{\delta}{2}, a_1 + \frac{\delta}{2})\} \right) & \text{if } \theta < 0.001 \\ \exp(r(a;\theta)) & \text{otherwise}, \end{cases}$$

for some $\delta \in (0, 1)$. The interpretation of this is that the human is believed to be noisily optimal; however, the human actually overestimates the value of an infinitesimal region centered at action a_1 only if θ is close to 0. Note that $d_\pi^{\mathsf{wc}}(\pi^*, \tilde{\pi}) < c\delta$ for some constant c, so we can choose δ such that the two policies are "close" to each other. When $a_1 = 1$, we will infer $\tilde{\theta} \approx 1$; however, $\theta^* \approx 0$, leading to reward inference error equal to the range of reward parameters.

3.4 Stability under log-concavity

Theorem 3.1 paints a pessimistic picture on the feasibility of reward inference from human demonstrations. In this section, we show that under reasonable assumptions on the true policy we can indeed obtain a positive stability result wherein we can upper bound the reward inference error by a linear function of the weighted policy error. Our analysis requires the following log-concavity assumption on the the true policy π^*.

Assumption 3.2. *The true and model policies $\pi^*, \tilde{\pi}$ are strongly log-concave with respect to reward parameters $\theta \in \Theta$. Formally, there exists constant $c > 0$ such that for any $s \in \mathcal{S}, a \in \mathcal{A}$, π^* satisfies*

$$\log \pi^*(a \mid s; \theta') \leq \log \pi^*(a \mid s; \theta) + \nabla_\theta \log \pi^*(a \mid s; \theta)^\top (\theta - \theta') - \frac{c}{2}\|\theta - \theta'\|_2^2 ,$$

and analogously for $\tilde{\pi}$.

The adversarial construction of demonstrator policy π^* in deriving Theorem 3.1 violate the above log-concavity assumption as they involve drastic perturbations of the actions that we happened to observe, and reward parameters that deviate from the inferred parameters.

Intuition. We know that log-concavity is violated by unnatural, adversarial constructions, but, *when does log-concavity always hold outside of such contrived examples?* Intuitively, we notice that log-concavity holds only if, as the reward parameter increases, an action that has become less preferred cannot become more preferred in the future. This appears to be a natural property of many policies; however, we show that there are simple problems where this is violated. In Figure 3.1 (left), we present a simple navigation example where Assumption 3.2 is violated. The environment is a 3×3 gridworld with deterministic transitions and discount $\gamma = 1$. Let s be the center cell, and a be going up. In Figure 3.1 (right), we show that a natural policy that chooses a according to $\pi(a \mid s; \theta) \propto \exp(\max(\theta, 10 - \theta))$ violates log-concavity. The reason is that a is optimal for $\theta \in [0, 4] \cup [6, 10]$ but not in between.

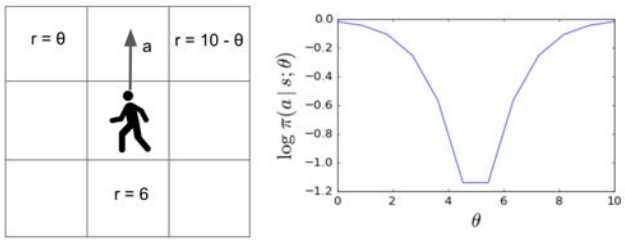


Figure 3.1. Simple navigation environment where a near optimal policy violates Assumption 3.2.

Under Assumption 3.2, we can show that the reward inference error can be bounded linearly by weighted policy divergence. We state the formal result below, and defer its proof to Appendix B.1.

Theorem 3.2. *Under Assumption 3.2 with parameter $c > 0$, for any policies $\pi^*, \tilde{\pi}$ with corresponding MLE reward parameters $\tilde{\theta}, \theta^*$, the reward inference error $d_\theta(\theta^*, \tilde{\theta})$ is bounded as*

$$\mathbb{E}_{\mathcal{D}\sim\pi^*}\left[\|\tilde{\theta} - \theta^*\|_2^2\right] \leq \frac{2}{c}\mathbb{E}_{s\sim d^*}\left[D_{KL}(\pi^*(\cdot \mid s; \theta^*) \;||\; \tilde{\pi}(\cdot \mid s; \theta^*))\right] .$$

Theorem 3.2 differs from Theorem 3.1 in two important ways: (1) the reward inference error is in expectation over sampled datasets, and (2) the policy divergence is the weighted policy divergence. Both these properties are desirable, as we are agnostic to tail events due to randomness in dataset sampling, and we use the weaker of the two notions of divergence in the upper bound.

Instantiating the upper bound for specific biases

Theorem 3.2 shows that the reward inference error can be bounded by the expected KL-divergence between the assumed and true policies. To understand the result in more detail, we now consider different systematic biases that could appear in human behavior, and show how they affect the weighted policy divergence.

We parameterize both the true and assumed policies as acting noisily optimal with respect to their own "Q-functions", *i.e.*, $\pi^*(a \mid s;\theta) \propto \exp(Q^*(s,a;\theta))$ and $\tilde{\pi}(a \mid s;\theta) \propto \exp(\tilde{Q}(s,a;\theta))$. Importantly, note that even though this parameterization is used in MaxEnt IRL with the soft Q-values [225, 224], neither Q^* nor $\tilde{Q}$ need necessarily be optimal – in this analysis, we will use $\tilde{Q}$, the human model, as the soft Q-value function, and show what happens when the true model coming from Q^* suffers from certain biases. Following prior work [165, 48], we examine biases that can be modelled as deviations from the Bellman update. For a tabular MDP M with $|\mathcal{S}|, |\mathcal{A}| < \infty$, the soft Bellman update satisfies:

$$Q(s,a;\theta) := r(s,a;\theta) + \gamma \sum_{s'} P(s' \mid s,a) V(s;\theta)\,, \;\; V(s;\theta) := \log \left(\sum_{a \in \mathcal{A}} \exp(Q(s,a;\theta)) \right). \tag{3.4}$$

Formally, we assume that the human demonstrator's Q-values $Q^*(s,a;\theta)$ satisfy (3.4) but under a biased MDP M^*. We consider two specific sources of bias in the MDP: (1) the transition model P and (2) the discounting factor γ. By parameterizing the biases in this way, we now have an intuitive notion of the degree of bias, and can study how the magnitude of the bias affects the policy divergence in (3.3). For brevity, we simply state the results as corollaries and defer proofs to Appendix B.1.

Internal dynamics. We first consider irrationalities that result from human demonstrators having an *internal dynamics model* P^* that is misspecified. For example, studies in cognitive science have shown that humans tend to underestimate the effects of inertia in projectile motion [46]. Similar studies have also shown that humans overestimate their control over randomness in the environment [193], dubbed *illusion of control*. The latter irrationality can be formalized in our parameterization by assuming that $P^*(\cdot \mid s,a) \propto (P(\cdot \mid s,a))^n$, where as $n \to \infty$, the human will believe the dynamics of the MDP are increasingly more deterministic. In Corollary 3.1, we show that the policy distance can be bounded linearly by the bias in transition dynamics:

Corollary 3.1. *Let* $\Delta_P = \sup_{s,a} \|P^*(\cdot \mid s,a) - \tilde{P}(\cdot \mid s,a)\|_1$. *Also, let* $\pi^*, \tilde{\pi}$ *be the policies that result from from value iteration using* (3.4) *with dynamics models* $P^*, \tilde{P}$, *respectively. Then, their weighted policy divergence is bounded as*

$$\mathbb{E}_{s\sim d^*}\left[D_{KL}(\pi^*(\cdot \mid s;\theta^*) \;||\; \tilde{\pi}(\cdot \mid s;\theta^*))\right] \leq \frac{2|\mathcal{A}|R_{\max}}{(1-\gamma)^2}\,\Delta_P\,.$$

Myopia Bias. The other irrationality we study is when humans overvalue near-term rewards, dubbed *myopia* [95]. Such bias can be captured in our parameterization through a biased discount factor γ^*, where as $\gamma^* \to 0$, the human will act more greedily and prioritize immediate reward. In Corollary 3.2, we bound the distance between policies by the absolute difference in their internal discount factor.

Corollary 3.2. *Let* $\pi^*, \tilde{\pi}$ *be the policies that result from value iteration using* (3.4) *with discount factors* $\gamma^*, \tilde{\gamma}$, *respectively. Then, their weighted policy divergence is bounded as*

$$\mathbb{E}_{s\sim d^*}\left[D_{KL}(\pi^*(\cdot \mid s;\theta^*) \;||\; \tilde{\pi}(\cdot \mid s;\theta^*))\right] \leq \frac{2|\mathcal{A}|R_{\max}}{(1-\tilde{\gamma})(1-\gamma^*)}\,|\tilde{\gamma}-\gamma^*|\,.$$

The above result shows that the degree of bias linearly upper-bounds the weighted policy divergence and hence, from Theorem 3.2, the expected reward inference error.

3.5 Empirical Analysis

Since our theory points to both reasons to be concerned as well as reasons to be optimistic, we also conduct an empirical analysis to check how different biases affect reward inference. Namely, in practice, do we find a stable relationship between policy divergence and reward error.

We tackle this in three ways: (1) simulating the specific biases we analyzed in Section 3.4, (2) simulating a non-Bellman-update structured kind of bias (a demonstrator that is still learning about the environment), and (3) collecting real human policies. We consider both tabular navigation tasks on gridworld, as well as more challenging continuous control tasks on the Lunar Lander game [36].

Experiment design. Each experiment has a bias we study and an environment (gridworld or LunarLander). When considering simulated biases, we manipulate π^* by manipulating the *magnitude of the bias* starting at $\pi^* = \tilde{\pi}$ the Boltzmann optimal policy. This helps us simulate different hypothetical humans, and see what degree of deviation from optimality ends up negatively impacting reward inference. When modeling bias with real human data, we instead fix π^* as the real human policy, and manipulate $\tilde{\pi}$ by interpolating between the Boltzmann optimal policy and the real human policy – this emulates a practical process where human models get increasingly more accurate.

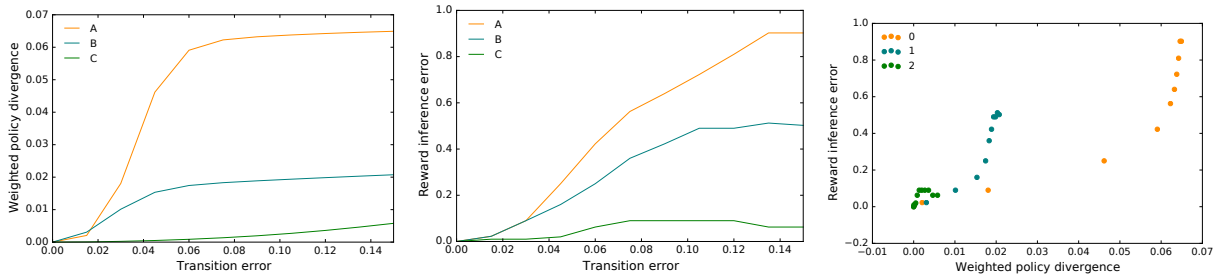


Figure 3.2. Effect of transition error (measured as the degree of underestimation of unintended transitions) on (a) weighted policy divergence and (b) reward inference error on Gridworld environments. In (c), we show a scatter plot of the policy and reward errors for each P^*.

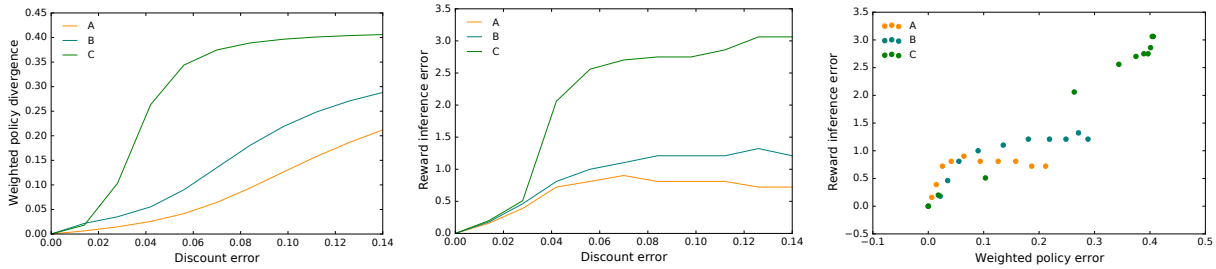


Figure 3.3. Effect of underestimating the discount factor on (a) weighted policy divergence and (b) reward inference error on Gridworld environments. In (c), we show a scatter plot of the policy and reward errors for each γ^*.

Tabular experiments with structured biases

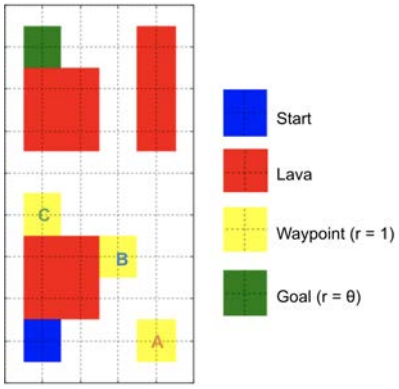


Figure 3.4. Gridworld environments.

First, we consider tabular navigation in gridworld domains [84], where the task is the reach the goal state and earn a reward of $\theta > 1$, which is not known to the agent, while avoiding getting trapped at lava states. To further complicate the task, the agent can also get stuck at "waypoint" states that yield a reward of 1. Depending on the environment, it can be better for the agent to stop at the waypoint state, to circumvent taking the longer, more treacherous path to the goal state. The agent is able to move in either of the four directions, or choose to stay still. To introduce stochasticity in the transition dynamics, there is a 30% chance that the agent travels in a different direction than commanded. We consider three different gridworlds (which we simply call environments A, B, and C) where we vary in the location of the waypoint state (shown in Figure 3.4).

In each environment, we want to the learn the underlying reward parameter θ from demonstrations; however, the model $\tilde{\pi}$ is noisily optimal, whereas the demonstrator policy π^* is irrational by suffering from false internal dynamics, or myopia. We model these irrationalities by either modifying the transition matrix or discount factor, respectively, in the

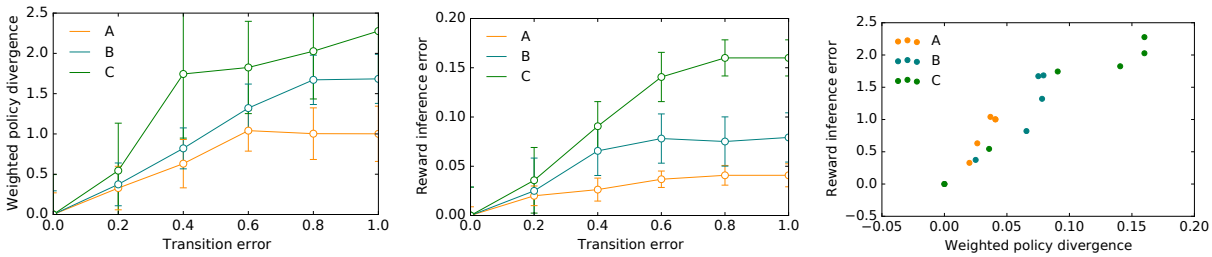


Figure 3.5. Effect of transition error (measured as error in p) on (a) weighted policy divergence and (b) reward inference error on continuous Lunar Lander environments. In (c), we show a scatter plot of the policy and reward errors for fixed α.

soft Bellman update in Equation (3.4). Note that the stationary distribution w^π for a policy π can be exactly computed; hence, instead of sampling data from π^*, we use w^* to compute exact quantities (see Appendix B.2 for technical details).

Internal dynamics. The first irrationality we consider is illusion of control, where the demonstrator policy significantly underestimates the stochasticity in the environment. Such biased policies π^* are obtained via value iteration on a biased transition matrix P^*, where the human wrongly believes the probability p of unintended transitions is smaller than the true value. As $p \to 0$, the demonstrator becomes more confident that they can reach the goal state, and will prefer reaching the goal over the waypoint state, even when the latter is much closer and safely reachable (see Appendix B.2 for visualizations of the biased policies). In Figure 3.2, we show the effect of the transition bias (error in p) on both the weighted policy divergence, and the reward inference error. The sub-linear trend in Figure 3.2a agrees with Corollary 3.1. Figure 3.2b and c show a sub-linear dependence of the reward inference error on the policy divergence, as predicted by Theorem 3.2. For environment A, the the reward error goes up most quickly with the dynamics error, but so does the weighted policy divergence, making this divergence a better indicator of reward error than simply the dynamics error.

Myopia. The next irrationality we look at is myopia, where the demonstrator policy assumes a biased discounting factor γ^* that underestimates the true one. As $\gamma^* \to 0$, the biased agent will much more strongly prefer the closer waypoint state over the goal state. In Figure 3.3, we see analogous results to the internal dynamics bias. Namely, Figure 3.3a agrees with Corollary 3.2, and Figure 3.3b and c shows a sub-linear correlation between policy and reward error, as predicted by Theorem 3.2.

Continuous control experiments

Next, we consider a more challenging domain of navigation with continuous states and actions. The exact navigation environment is a modification the Lunar Lander game with continuous actions [38], where the agent receives a reward for landing safely on the landing pad. The agent is able to take a continuous action in $[-1, 1]^2$ that encodes the directions it wants to move (left, right, up) via its sign, as well as how much power it wants to use in each direction via its cardinality. In constrast to the classic version of the game where the

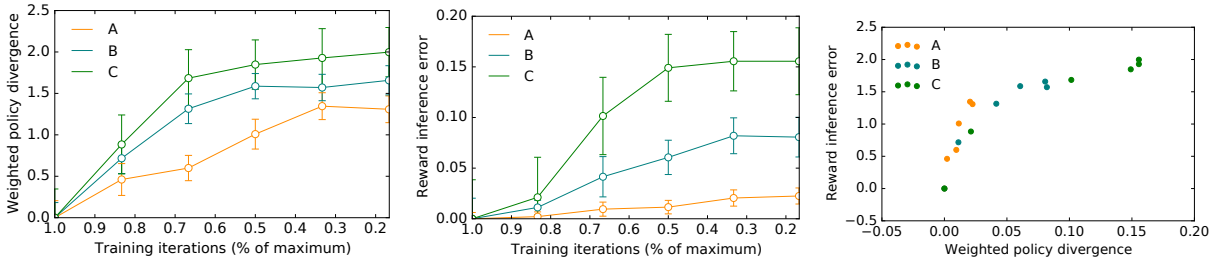


Figure 3.6. Effect of amount of training on (a) weighted policy divergence and (b) reward inference error on continuous Lunar Lander environments. In (c), we show a scatter plot of the policy and reward errors for fixed number of training iterations.

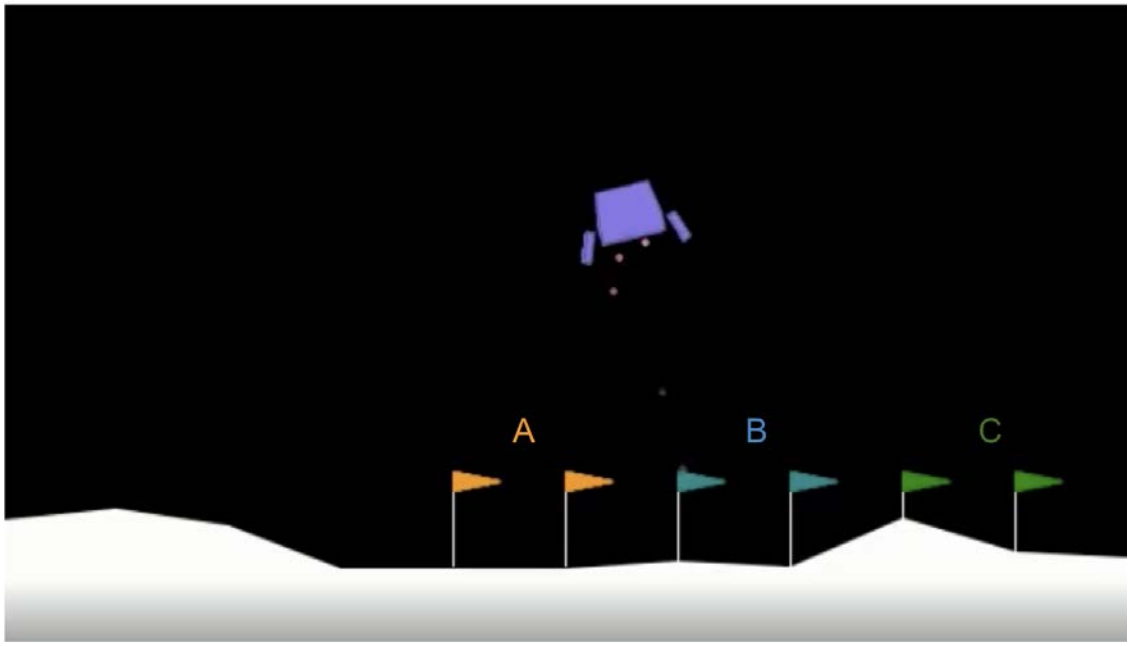


Figure 3.7: Lunar Lander environments.

landing pad is always in the middle, we vary its location. The unknown reward parameter $\theta \in (0, 1)$ is the location of the landing pad (as a horizontal displacement normalized by the total width of the environment). We consider three different environments that differ in the location of the landing pad (see Figure 3.7). In each environment, the human model $\tilde{\pi}$ is the near-optimal one obtained by soft actor-critic [96]. We provide details on the training procedure in Appendix B.2. In these experiments, we simulate the internal dynamics bias as well as a new one based on the notion of a demonstrator that is still themselves learning.

Internal dynamics. We first study of the effect of demonstrator policies with biased dynamics models. We bias the dynamics model by varying a parameter p that describes how much one unit of power will increase acceleration in the corresponding direction. This is a plausibly natural human bias as people tend to underestimate the effects of inertia in

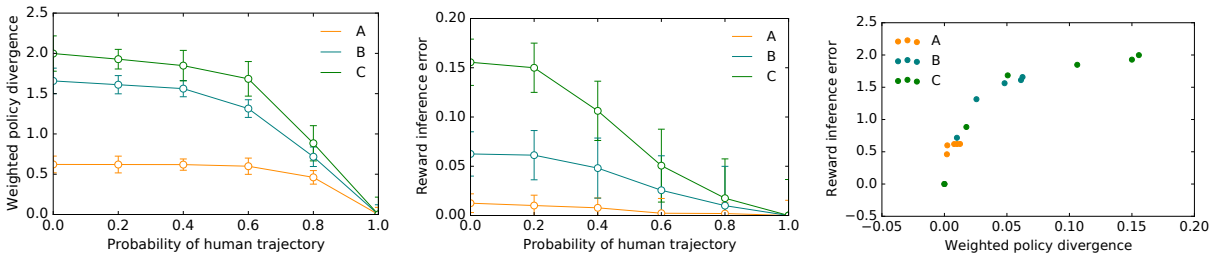


Figure 3.8. Effect of modeling human bias (measured by probability of acting according to human policy) on (a) weighted policy divergence and (b) reward inference error on discrete Lunar Lander environments. In (c), we show a scatter plot of the policy and reward errors for fixed probabilities. We see that more accurate human models correspond to lower reward inference error.

projectile motion [46]. For each false setting of p, we learn a biased policy that is near-optimal for that p. As p increase, the biased policy tends to underestimate the amount of power required to move the lander enough to the right to reach the landing pad (see Appendix B.2 for visualizations). In Figure 3.5, we show the effect of the transition bias (error in p) on both the weighted policy divergence and the reward inference error. We see that even in a challenging continuous control domain, Theorem 3.2 still holds.

Demonstrators that are learning. We next simulate a bias that might arise from humans that are learning how to do the task (as would be the case, for instance, in our Lunar Lander task). We do so by varying the amount of training iterations in learning the policy. The degree of such bias is captured in a parameter $\rho \in (0, 1)$, that denotes the number of training iterations, normalized by the amount used to learn the near-optimal believed policy. In Figure 3.6, we show the effect of ρ on both the weighted policy divergence and the reward inference error. Reassuringly, we again notice a sub-linear correlation in line with Theorem 3.2.

Analysis of real human policies

The previous experiments have considered natural but simulated biases to construct biased demonstrator policies. However, it remains to be seen whether the findings for simulated biases hold for biases grounded in real human demonstrations. We consider the same Lunar Lander game in Section 3.5 but with a discrete action space; this action space consists of only the 3 directions (left, right, up), where the power in each direction are now fixed constants. We discretize the action space to create a more intuitive environment that humans can easily interact in. Using this environment, we create a demonstrator policy grounded in real human demonstrations. We do this by collecting trajectories from 10 human demonstrators, then learning a policy that imitates human behavior by running behavior cloning (BC) on the aggregated trajectories. We visualize trajectories from this policy in Figure 3.9. We observe that in general, humans tend to be unable to properly account for the effect of gravity, causing them to crash the lander before it has moved enough horizontally, particularly in

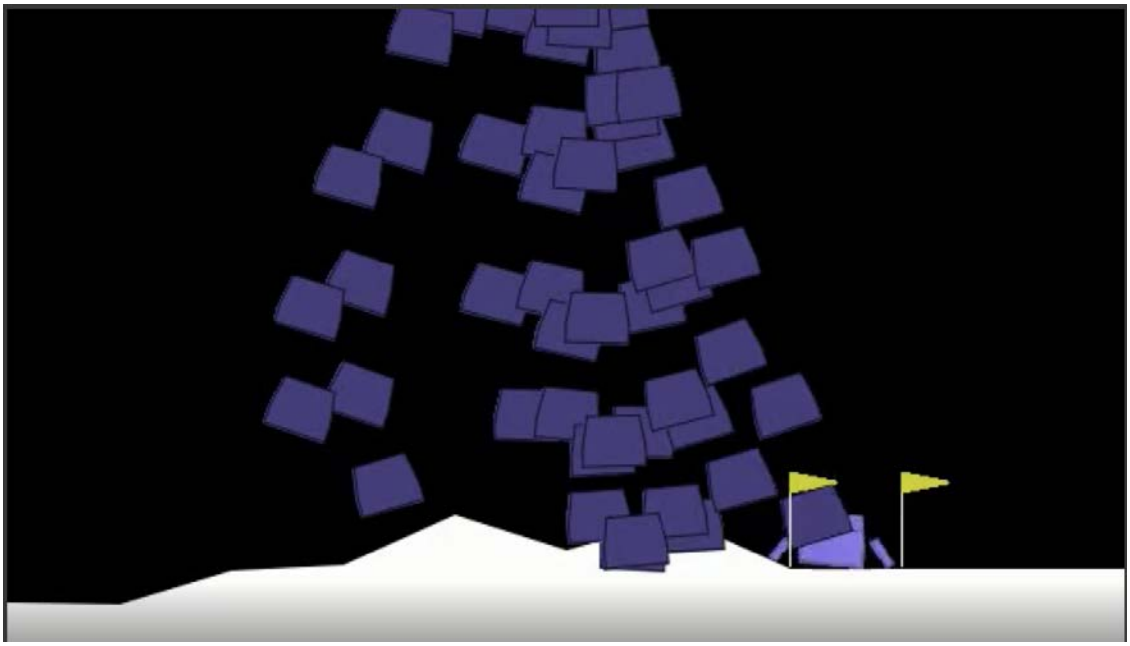


Figure 3.9: Visualization of trajectories under the human policy.

environments B and C where the landing pad is horizontally displaced from the middle.

Then, we emulate a process through which the model $\tilde{\pi}$ would evolve to align more and more with the true human policy. Specifically, we vary $\tilde{\pi}$ to interpolate between near-optimal and the true human policy, while keeping π^* fixed to the latter. To do so, we vary a parameter α that controls the probability of sampling from the human policy (vs. the near-optimal one). In Figure 3.8, we show the affect of α on the weighted policy divergence and reward inference error, and conclude that larger α result in smaller policy divergence as well as reward error. In addition, we also match the simulation experiments by keeping $\tilde{\pi}$ fixed as the optimal policy, and interpolate between the optimal policy and the real human policy for π^* (see Appendix B.2). This suggests that Theorem 3.2 might broadly apply, giving us hope that better human models $\tilde{\pi}$ can translate to better reward inference.

3.6 Discussion

Summary. In this paper, we conduct a theoretical and empirical study of how sensitive reward learning from human demonstrations is to misspecification of the human model. First, we provide an ominous result that arbitrarily small divergences in the assumed human model can result in large reward inference error. However, we also show if the true human policy satisfies a relatively mild assumption, then reward error is upper-bounded linearly by the policy divergence. Experiments with multiple biases in different environments, as well as an analysis of the true human policy, reassuringly show remarkably consistent results: over and over again, we see that as the human model and the true human behavior are more and more aligned, the reward error decreases. Overall, our results convey the optimistic message

that reward learning improves as we obtain better human models.

Limitations and future work. Our upper-bound relies on Assumption 3.2 of log-concavity of the human policy. However, we hypothesize that weaker assumptions exist from which we can derive similar bounds as Theorem 3.2. In addition, via Assumption 3.1, we ignore ambiguity in reward identification. It may be important in the future to consider reward error and identifiability jointly, potentially through equivalence classes of reward functions. Interesting directions of further investigation include: (1) how to better model human biases, or orthogonally, (2) how to modify existing reward inference algorithms to be more robust to misspecification. A negative side-effect of our work could occur when we mistakenly rely on the upper-bound in Theorem 3.2 when its conditions are not met, *i.e.*, Assumption 3.2 does not hold, resulting in catastrophically bad inference without knowing it. More broadly, reward learning in general has the issue that it does not specify *whose* reward to learn, and how to combine different people's values.

Chapter 4

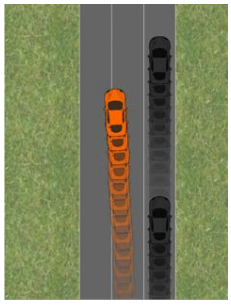
Learning with multi-criteria preferences

4.1 Introduction

Economists, social scientists, engineers, and computer scientists have long studied models for human preferences, under the broad umbrella of social choice theory [28, 13]. Learning from human preferences has found applications in interactive robotics for learning reward functions [172, 156], in medical domains for personalizing assistive devices [221, 27], and in recommender systems for optimizing search engines [49, 106]. The recent focus on safety in AI has popularized human-in-the-loop learning methods that use human preferences in order to promote value alignment [57, 173, 8].

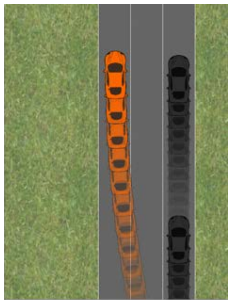
The most popular form of preference elicitation is to make pairwise comparisons [194, 34, 136]. Eliciting such feedback involves showing users a pair of objects and asking them a query: Do you prefer object A or object B? Depending on the application, an object could correspond to a product in a search query, or a policy or reward function in reinforcement learning. A vast body of classical work dating back to Condorcet and Borda [58, 33] has focused on defining and producing a "winning" object from the result of a set of pairwise comparisons.

In relatively recent work, Dudik et al. [71] proposed the concept of a von Neumann winner, corresponding to a distribution over objects that beats or ties every other object in the collection. They showed that under an expected utility assumption, such a randomized winner always exists and overcomes limitations of existing winning concepts—the Condorcet winner does not always exist, while the Borda winner fails an independence of clones test [175]. However, the assumption of expected utility relies on a strong hypothesis about how humans evaluate distributions over objects: it posits that the probability with which any distribution over objects π beats an object is linear in π.



Policy A

Vs.



Policy B

(a)

	A	B
A	0.50	0.60
B	0.40	0.50
R	0.55	0.63

$\mathbf{P}_{\text{ex}}^{\text{Overall}}$

	A	B
A	0.50	0.80
B	0.20	0.50
R	0.35	0.65

$\mathbf{P}_{\text{ex}}^{\text{Comfort}}$

	A	B
A	0.50	0.30
B	0.70	0.50
R	0.60	0.40

$\mathbf{P}_{\text{ex}}^{\text{Speed}}$

(b)

Figure 4.1. (a) Policy A focuses on optimizing comfort and policy B on speed, and these are compared pairwise in different environments. (b) Preference matrices, where entry (i, j) of the matrix contains the proportion of comparisons between the pair (i, j) that are won by object i. (The diagonals are set to half by convention). The overall pairwise comparisons are given by the matrix $\mathbf{P}_{\text{ex}}^{\text{Overall}}$, and preferences along each of the criteria by matrices $\mathbf{P}_{\text{ex}}^{\text{Comfort}}$ and $\mathbf{P}_{\text{ex}}^{\text{Speed}}$. Policy R is a randomized policy $1/2$ A $+1/2$ B. While the preference matrices satisfy the linearity assumption individually along speed and comfort, the assumption is violated overall, wherein R is preferred over both A and B.

Consequences of assuming linearity: In order to better appreciate these consequences, consider as an example the task of deciding between two policies (say A and B) to deploy in an autonomous vehicle. Suppose that these policies have been obtained by optimizing two different objectives, with policy A optimized for comfort and policy B optimized for speed. Figure 4.1(a) shows a snapshot of these two policies. When compared overall, 60% of the people preferred Policy A over B – making it the von Neumann winner. The linearity assumption then posits that a randomized policy that mixes between A and B can *never* be better than both A and B; but we see that the Policy R = $1/2$ A + $1/2$ B is actually preferred by a majority over both A and B! Why is the linearity assumption violated here?

One possible explanation for such a violation is that the comparison problem is actually *multi-criteria* in nature. If we look at the preferences for the criterion speed and comfort individually in Figure 4.1(b), we see that Policy A does quite poorly on the speed axis while B lags behind in comfort. In contrast, Policy R does acceptably well along both the criteria and hence is preferred overall to both Policies A and B. It is indeed impossible to come to this conclusion by only observing the overall comparisons. This observation forms the basis of our main proposal: decompose the single overall comparison and ask humans to provide preferences along *simpler* criteria. This decomposition of the comparison task allows us to place structural assumptions on comparisons along each criterion. For instance, we may now posit the linearity assumption along each criterion separately rather than on the overall comparison task. In addition to allowing for simplified assumptions, breaking up the task into such simpler comparisons allows us to obtain richer and more accurate feedback as compared to the single overall comparison. Indeed, such a motivation for eliciting simpler feedback

from humans finds its roots in the the study of cognitive biases in decision making, which suggests that the human mind resorts to simple heuristics when faced with a complicated questions [199].

Contributions: In this paper, we formalize these insights and propose a new framework for preference learning when pairwise comparisons are available along multiple, possibly conflicting, criteria. As shown by our example in Figure 4.1, a single distribution which is the von Neumann winner along every criteria might not exist. To counter this, we formulate the problem of finding the "best" randomized policy by drawing on tools from the literature on vector valued pay-offs in game theory. Specifically, we take inspiration from Blackwell's approachability [29] and introduce the notion of a Blackwell winner. This solution concept strictly generalizes the concept of a von Neumann winner, and recovers the latter when there is only a single criterion present. Section 4.2 describes this framework in detail, and Section 4.3 collects our statistical and computational guarantees for learning the Blackwell winner from data. Section 4.4 describes a user study with an autonomous driving environment, in which we ask human subjects to compare self-driving policies along multiple criteria such as safety, aggressiveness, and conservativeness. Our experiment demonstrates that the Blackwell winner is able to better trade off utility along these criteria and produces randomized policies that outperform the von Neumann winner for the overall preferences.

Related work

This paper sits at the intersection of multiple fields of study: learning from pairwise comparisons, multi-objective optimization, preference aggregation, and equilibrium concepts in games. Here, we review the papers that are most relevant to our contributions.

Winners from pairwise comparisons. Most closely related to our work is the field of computational social choice, which has focused on defining notions of winners from overall pairwise comparisons (see the survey [35] for a review). Amongst them, three deterministic notions of a winner—the Condorcet [58], Borda [33], and Copeland [59] winners—have been widely studied. In addition, Dudik et al. [71] recently introduced the notion of a (randomized) von Neumann winner.

Starting with the work of Yue et al. [219], there have been several research papers studying an online version of preference learning, called the Dueling Bandits problem. This is a partial information version of the classic K-armed bandit problem where the feedback comprises comparisons between a pair of arms. Algorithms have been proposed to compete with Condorcet [229, 230, 6], Copeland [228, 214], Borda [111] and von Neumann [71] winners.

Multi-criteria decision making. The theoretical foundations of decision making based on multiple criteria have been widely studied within the operations research community . This sub-field—called multiple-criteria decision analysis— has focused largely on scoring,

classification, and sorting based on multiple-criteria feedback. See the surveys [163, 231] for thorough overviews of existing methods and their associated guarantees. The problem of eliciting the user's relative weighting of the various criteria has also been considered [67]. However, relatively less attention has been paid to the study of randomized decisions and statistical inference, both of which form the focus of our work. From an applied perspective, the combination of multi-criteria assessments has received attention in disparate fields such as psychometrics [158, 141], healthcare [192], and recidivism prediction [210]. In many of these cases, a variety of approaches—both linear and non-linear—have been empirically evaluated [66]. Justification for non-linear aggregation of scores along the criteria has a long history in psychology and the behavioral sciences [92, 82, 200].

Blackwell's approachability. In the game theory literature, Blackwell [29] introduced the notion of approachability as a generalization of a zero-sum game with vector-valued payoffs (for a detailed discussion see Appendix C.1). Blackwell's approachability and its connections with no-regret learning and calibrated forecasting have been extensively studied [2, 162, 140]. These connections have enabled applications of Blackwell's results to problems ranging from constrained reinforcement learning [145] to uncertainty estimation for question-answering tasks [127]. In contrast with such applications of the repeated vector-valued game, our framework for preference learning along multiple criteria deals with a single shot game and uses the idea of the target set to define the concept of a Blackwell winner.

Stability of Nash equilibria. Another body of literature related to our work studies Nash equilibria in games with perturbed payoffs, under both robust [5, 132] and uncertain (or Bayesian) [86] formulations (see the recent survey by Perchet [161]). Perturbation theory for Nash equilibria has been derived in these contexts, and it is well-known that the Nash equilibrium is not (at least in general) stable to perturbations of the payoff matrix. On the other hand, the results of [71] consider Nash equilibria of perturbed, symmetric, zero-sum games, but show that the *payoff* of the perturbed Nash equilibrium is indeed stable. That is, even if the equilibrium itself can change substantially with a small perturbation of the payoff matrix, the payoff that this perturbation obtains is still close to the payoff of the original equilibrium. Our work provides a similar characterization for the multi-criteria setting.

4.2 Framework for preference learning along multiple criteria

We now set up our framework for preference learning along multiple criteria. We consider a collection of d objects over which comparisons can be elicited along k different criteria. We index the objects by the set $[d] := \{1, \dots, d\}$ and the criteria by the set $[k]$.

Probabilistic model for comparisons

Since human responses to comparison queries are typically noisy, we model the pairwise preferences as random variables drawn from an underlying population distribution. In particular, the result of a comparison between a pair of objects (i_1, i_2) along criterion j is modeled as a draw from a Bernoulli distribution, with $p(i_1, i_2; j) = \mathbb{P}(i_1 \succeq i_2 \text{ along criterion } j)$. By symmetry, we must have

$$p(i_2, i_1; j) = 1 - p(i_1, i_2; j) \text{ for each triple } i_1 \in [d],\ i_2 \in [d], \text{ and } j \in [k]. \tag{4.1}$$

We let $\pi_1, \pi_2 \in \Delta_d$ represent[1] two distributions over the d objects. With a slight abuse of notation, let $p(\pi_1, \pi_2; j)$ denote the probability with which an object drawn from distribution π_1 beats an object drawn from distribution π_2 along criterion j. We assume for each individual criterion j that the probability $p(\pi_1, \pi_2; j)$ is linear in the distributions π_1 and π_2, i.e. that it satisfies the relation

$$p(\pi_1, \pi_2; j) := \mathbb{E}_{i_1 \sim \pi_1, i_2 \sim \pi_2} \left[p(i_1, i_2; j) \right]. \tag{4.2}$$

Equation (4.2) encodes the per-criterion linearity assumption highlighted in Section 4.1. We collect the probabilities $\{p(i_1, i_2; j)\}$ into a *preference tensor* $\mathbf{P} \in [0,1]^{d \times d \times k}$ and denote by $\mathcal{P}_{d,k}$ the set of all preference tensors that satisfy the symmetry condition (4.1). Specifically, we have

$$\mathcal{P}_{d,k} = \{\mathbf{P} \in [0,1]^{d \times d \times k} \mid \mathbf{P}(i_1, i_2; j) = 1 - \mathbf{P}(i_2, i_1; j) \text{ for all } (i_1, i_2, j)\}\ . \tag{4.3}$$

Let $\mathbf{P}^j$ denote the $d \times d$ matrix corresponding to the comparisons along criterion j, so that $p(\pi_1, \pi_2; j) = \pi_1^\top \mathbf{P}^j \pi_2$. Also note that a comparison between a pair of objects (i_1, i_2) induces a *score vector* containing k such probabilities. Denote this vector by $\mathbf{P}(i_1, i_2) \in [0,1]^k$, whose j-th entry is given by $p(i_1, i_2; j)$. Denote by $\mathbf{P}(\pi_1, \pi_2)$ the score vector for a pair of distribution (π_1, π_2).

In the single criterion case when $k = 1$, each comparison between a pair of objects is along an *overall* criterion. We let $\mathbf{P}_{\mathsf{ov}} \in [0,1]^{d \times d}$ represent such an overall comparison matrix. As mentioned in Section 4.1, most preference learning problems are multi-objective in nature, and the overall preference matrix $\mathbf{P}_{\mathsf{ov}}$ is derived as a non-linear combination of per-criterion preference matrices $\{\mathbf{P}^j\}_{j=1}^k$. Therefore, even when the linearity assumption (4.2) holds across each criterion, it might not hold for the *overall* preference $\mathbf{P}_{\mathsf{ov}}$. In contrast, when the matrices $\mathbf{P}^j$ are aggregated linearly to obtain the overall matrix $\mathbf{P}_{\mathsf{ov}}$, we recover the assumptions of Dudik et al. [71].

Blackwell winner

Given our probabilistic model for pairwise comparisons, we now describe our notion of a Blackwell winner. When defining a winning distribution for the multi-criteria case, it would

[1] We let Δ_d denote the d-dimensional simplex.

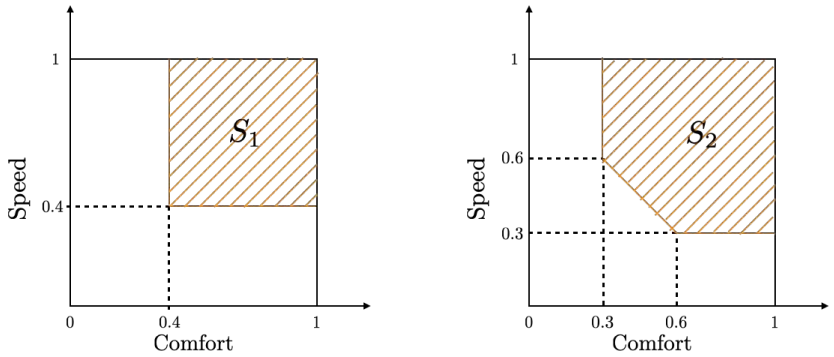


Figure 4.2. Two target sets S_1 and S_2 for our example from Figure 4.1 that capture trade-offs between comfort and speed. Set S_1 requires feasible score vectors to satisfy 40% of the population along both comfort and speed. Set S_2 requires both scores to be greater than 0.3 but with a linear trade-off: the combined score must be at least 0.9.

be ideal to find a distribution π^* that is a von Neumann winner along *each* of the criteria separately. However, as shown in our example from Figure 4.1, such a distribution need not exist: policy A is preferred along the comfort axis, while policy B along speed. We thus need a generalization of the von Neumann winner that explicitly accounts for conflicts between the criteria.

Blackwell [29] asked a related question for the theory of zero-sum games: how can one generalize von Neumann's minimax theorem to vector-valued games? He proposed the notion of a *target set*: a set of acceptable payoff vectors that the first player in a zero-sum game seeks to attain. Within this context, Blackwell proposed the notion of approachability, i.e. how the player might obtain payoffs in a repeated game that are close to the target set on average. We take inspiration from these ideas to define a solution concept for the multi-criteria preference problem.

Our notion of a winner also relies on a target set, which we denote by $S \subset [0,1]^k$, and which in our setting contains *score vectors*. This set provides a way to combine different criteria by specifying combinations of preference scores that are acceptable. Figure 4.2 provides an example of two such sets. Observe that for our preference learning problem, the target set S is by definition monotonic with respect to the orthant ordering, that is, if $z_1 \geq z_2$ coordinate-wise, then $z_2 \in S$ implies $z_1 \in S$. Our goal is to then produce a distribution π^* that can achieve a target score vector for any distribution with which it is compared—that is $\mathbf{P}(\pi^*, \pi) \in S$ for all $\pi \in \Delta_d$. When such a distribution π^* exists, we say that the problem instance $(\mathbf{P}, S)$ is *achievable*.

On the other hand, it is clear that there are problem instances $(\mathbf{P}, S)$ that are not achievable. While Blackwell's workaround was to move to the setting of repeated games, preference aggregation is usually a one-shot problem. Consequently, our relaxation instead introduces the notion of a *worst-case distance* to the target set. In particular, we measure the distance between any pair of score vectors $u, v \in [0,1]^k$ as $\rho(u,v) = \|u - v\|$ for some

norm $\|\cdot\|$. Using the shorthand $\rho(u, S) := \inf_{v\in S} \|u - v\|$, the *Blackwell winner* π^* for an instance $(\mathbf{P}, S, \|\cdot\|)$ is now defined as the one which minimizes the maximum distance to the set S, i.e.,

$$\pi(\mathbf{P}, S, \|\cdot\|) \in \operatorname*{argmin}_{\pi\in\Delta_d}[v(\pi; \mathbf{P}, S, \|\cdot\|)], \quad \text{where} \quad v(\pi; \mathbf{P}, S, \|\cdot\|) := \max_{\pi'\in\Delta_d} \rho(\mathbf{P}(\pi, \pi'), S)\ . \tag{4.4}$$

Observe that equation (4.4) has an interpretation as a zero-sum game, where the objective of the minimizing player is to make the score vector $\mathbf{P}(\pi, \pi')$ as close as possible to the target set S.

We now look at commonly studied frameworks for single criterion preference aggregation and multi-objective optimization and show how these can be naturally derived from our framework.

Example: Preference learning along a single criterion. A particular special case of our framework is when we have a single criterion ($k = 1$) and the preferences are given by a matrix $\mathbf{P}_{\mathsf{ov}}$. The score $\mathbf{P}_{\mathsf{ov}}(i_1, i_2)$ is a scalar representing the probability with which object i_1 beats object i_2 in an overall comparison. As a consequence of the von Neumann minimax theorem, we have

$$\max_{\pi_1\in\Delta_d} \min_{\pi_2\in\Delta_d} \mathbf{P}_{\mathsf{ov}}(\pi_1, \pi_2) = \min_{\pi_2\in\Delta_d} \max_{\pi_1\in\Delta_d} \mathbf{P}_{\mathsf{ov}}(\pi_1, \pi_2) = \frac{1}{2}, \tag{4.5}$$

with any maximizer above called the von Neumann winner [71]. Thus, for *any* preference matrix $\mathbf{P}_{\mathsf{ov}}$, a von Neumann winner is preferred to any other object with probability at least $\frac{1}{2}$.

Let us show how this uni-criterion formulation can be derived as a special case of our framework. Consider the target set $S = [\frac{1}{2}, 1]$ and choose the distance function $\rho(a, b) = |a - b|$. By equation (4.5), the target set $S = [\frac{1}{2}, 1]$ is achievable *for all* preference matrices $\mathbf{P}_{\mathsf{ov}}$, and so the von Neumann winner and the Blackwell winner $\pi(\mathbf{P}_{\mathsf{ov}}, [\frac{1}{2}, 1], |\cdot|)$ coincide. ♣

Example: Weighted combinations of a multi-criterion problem. We saw in the previous example that the single criterion preference learning problem is quite special: achievability can be guaranteed by the von Neumann winner for set $S = [\frac{1}{2}, 1]$ for any preference matrix $\mathbf{P}_{\mathsf{ov}}$. One of the common approaches used in multi-objective optimization is to reduce a multi-dimensional problem to a uni-dimensional counterpart is by introducing a weighted combinations of objectives.

Formally, consider a weight vector $w \in \Delta_k$ and the corresponding preference matrix

$$\mathbf{P}(w) := \sum_{j\in[k]} w_j \mathbf{P}^j$$

obtained by combining the preference matrices along the different criteria. A winning distribution can then be obtained by solving for the von Neumann winner of $\mathbf{P}(w)$ given by

$\pi(\mathbf{P}(w), [\frac{1}{2}, 1], |\cdot|)$. The following proposition establishes that such an approach is a particular special case of our framework, and conversely, that there are problem instances in our general framework which cannot be solved by a simple linear weighing of the criteria.

Proposition 4.1. *(a) For every weight vector $w \in \Delta_k$, there exists a target set $S_w \in [0,1]^k$ such that for any norm $\|\cdot\|$, we have*

$$\pi(\mathbf{P}, S_w, \|\cdot\|) = \pi(\mathbf{P}(w), [1/2, 1], |\cdot|) \quad \textit{for all} \quad \mathbf{P} \in \mathcal{P}_{d,k}.$$

(b) Conversely, there exists a set S and a preference tensor $\mathbf{P}$ with a unique *Blackwell winner π^* such that for all $w \in \Delta_k$, exactly one of the following is true:*

$$\pi(\mathbf{P}(w), [1/2, 1], |\cdot|) \neq \pi^* \quad \textit{or} \quad \operatorname*{argmax}_{\pi \in \Delta_d} \min_{i \in [d]} \mathbf{P}(\pi, i) = \Delta_d \, .$$

Thus, while the Blackwell winner is always able to recover any linear combination of criteria, the converse is not true. Specifically, part (b) of the proposition shows that for a choice of preference tensor $\mathbf{P}$ and target set S, either the von Neumann winner for $\mathbf{P}(w)$ is not equal to the Blackwell winner, or it degenerates to the entire simplex Δ_d and is thus uninformative. Consequently, our framework is strictly more general that weighting the individual criteria. ♣

4.3 Statistical guarantees and computational approaches

In this section, we provide theoretical results on computing the Blackwell winner from samples of pairwise comparisons along the various criteria.

Observation model and evaluation metrics.

We operate in the natural passive observation model, where a sample consists of a comparison between two randomly chosen objects along a randomly chosen criterion. Specifically, we assume access to an oracle that when queried with a tuple $\eta = (i_1, i_2, j)$ comprising a pair of objects (i_1, i_2) and a criterion j, returns a comparison $y(\eta) \sim \mathsf{Ber}(p(i_1, i_2; j))$. Each query to the oracle constitutes one sample. In the passive sampling model, the tuple of objects and criterion is sampled uniformly, with replacement, that is $(i_1, i_2) \sim \mathsf{Unif}\{\binom{[d]}{2}\}$ and $j \sim \mathsf{Unif}\{[k]\}$ where $\mathsf{Unif}\{A\}$ denotes the uniform distribution over the elements of a set A.

Given access to samples $\{y_1(\eta_1), \ldots, y_n(\eta_n)\}$ from this observation model, we define the empirical preference tensor (specifically the upper triangular part)

$$\widehat{\mathbf{P}}_n(i_1, i_2, j) := \frac{\sum_{\ell=1}^n y_\ell(\eta_\ell) \mathbb{I}[\eta_\ell = (i_1, i_2, j)]}{1 \vee \sum_\ell \mathbb{I}[\eta_\ell = (i_1, i_2, j)]} \quad \text{for } i_1 < i_2 \, , \tag{4.6}$$

where each entry of the upper-triangular tensor is estimated using a sample average and the remaining entries are calculated to ensure the symmetry relations implied by the inclusion $\widehat{\mathbf{P}}_n \in \mathcal{P}_{d,k}$.

As mentioned before, we are interested in computing the solution $\pi^* := \pi(\mathbf{P}, S, \|\cdot\|)$ to the optimization problem (4.4), but with access only to samples from the passive observation model. For any estimator $\widehat{\pi} \in \Delta_d$ obtained from these samples, we evaluate its error based on its value with respect to the tensor $\mathbf{P}$, i.e.,

$$\Delta_{\mathbf{P}}(\widehat{\pi}, \pi) := v(\widehat{\pi}; S, \mathbf{P}, \|\cdot\|) - v(\pi^*; S, \mathbf{P}, \|\cdot\|). \tag{4.7}$$

Note that the error $\Delta_{\mathbf{P}}$ implicitly also depends on the set S and the norm $\|\cdot\|$, but we have chosen our notation to be explicit only in the preference tensor $\mathbf{P}$. For the rest of this section, we restrict our attention to convex target sets S and refer them to as *valid sets*. Having established the background, we are now ready to provide sample complexity bounds on the estimation error $\Delta_{\mathbf{P}}(\widehat{\pi}, \pi^*)$.

Upper bounds on the error of the plug-in estimator

Recall the definition of the function v from equation (4.4), and define, for each preference tensor $\widetilde{\mathbf{P}}$, the optimizer

$$\pi(\widetilde{\mathbf{P}}) \in \operatorname*{argmin}_{\pi \in \Delta_d} v(\pi; S, \widetilde{\mathbf{P}}, \|\cdot\|)\ . \tag{4.8}$$

Also recall the empirical preference tensor $\widehat{\mathbf{P}}_n$ from equation (4.6). With this notation, the plug-in estimator is given by $\widehat{\pi}_{\mathsf{plug}} = \pi(\widehat{\mathbf{P}}_n)$ and the target (or true) distribution by $\pi^* = \pi(\mathbf{P})$.

While, our focus in this section is to provide upper bounds on the error of the plug-in estimator $\widehat{\pi}_{\mathsf{plug}}$, we first state a general perturbation bound which relates the error of the optimizer $\pi(\widetilde{\mathbf{P}})$ to the deviation of the tensor $\widetilde{\mathbf{P}}$ from the true tensor $\mathbf{P}$. We use $\mathbf{P}(\cdot, i) \in [0,1]^{d \times k}$ to denote a matrix formed by viewing the i-th slice of $\mathbf{P}$ along its second dimension. Finally, recall our definition of the error $\Delta_{\mathbf{P}}(\widehat{\pi}, \pi^*)$ from equation (4.7).

Theorem 4.1. *Suppose the distance ρ is induced by the norm $\|\cdot\|_q$ for some $q \geq 1$. Then for each valid target set S and preference tensor $\widetilde{\mathbf{P}}$, we have*

$$\Delta_{\mathbf{P}}(\pi(\widetilde{\mathbf{P}}), \pi^*) \leq 2 \max_{i \in [d]} \|\widetilde{\mathbf{P}}(\cdot, i) - \mathbf{P}(\cdot, i))\|_{\infty, q}. \tag{4.9}$$

Note that this theorem is entirely deterministic: it bounds the deviation in the optimal solution to the problem (4.4) as a function of perturbations to the tensor $\mathbf{P}$. It also applies *uniformly* to all valid target sets S. In particular, this result generalizes the perturbation result of Dudik et al. [71, Lemma 3] which obtained such a deviation bound for the single criterion problem with π^* as the von Neumann winner. Indeed, one can observe that by setting the distance $\rho(u, v) = |u - v|$ in Theorem 4.1 for the uni-criterion setup, we have the error $\Delta_{\mathbf{P}}(\pi(\widetilde{\mathbf{P}}), \pi^*) \leq 2\|\widetilde{\mathbf{P}} - \mathbf{P}\|_{\infty,\infty}$, matching the bound of [71].

Let us now illustrate a consequence of this theorem by specializing it to the plug-in estimator, and with the distances given by the ℓ_∞ norm.

Corollary 4.1. *Suppose that the distance ρ is induced by the ℓ_∞-norm $\|\cdot\|_\infty$. Then there exists a universal constant $c > 0$ such that given a sample size $n > cd^2k\log(\frac{cdk}{\delta})$, we have for each valid target set S*

$$\Delta_{\mathbf{P}}(\widehat{\pi}_{\mathsf{plug}}, \pi^*) \leq c\sqrt{\frac{d^2k}{n}\log\left(\frac{cdk}{\delta}\right)}, \tag{4.10}$$

with probability greater than $1-\delta$.

The bound (4.10) implies that the plug-in estimator $\widehat{\pi}_{\mathsf{plug}}$ is an ϵ-approximate solution whenever the number of samples scales as $n = \widetilde{O}(\frac{d^2k}{\epsilon^2})$. Observe that this sample complexity scales quadratically in the number of objects d and linearly in the number of criteria k. This scaling represents the effective dimensionality of the problem instance, since the underlying preference tensor $\mathbf{P}$ has $O(d^2k)$ unknown parameters. Notice that the corollary holds for sample size $n = \widetilde{O}(d^2k)$; this should not be thought of as restrictive, since otherwise, the bound (4.10) is vacuous.

Information-theoretic lower bounds

While Corollary 4.1 provides an upper bound on the error of the plug-in estimator that holds *for all* valid target sets S, it is natural to ask if this bounds is sharp, i.e., whether there is indeed a target set S for which one can do no better than the plug-in estimator. In this section, we address this question by providing lower bounds on the minimax risk

$$\mathfrak{M}_{n,d,k}(S, \|\cdot\|_\infty) := \inf_{\widehat{\pi}} \sup_{\mathbf{P}\in\mathcal{P}} \mathbb{E}\left[\Delta_{\mathbf{P}}(\widehat{\pi}, \pi^*)\right], \tag{4.11}$$

where the infimum is taken over *all* estimators that can be computed from n samples from our observation model. It is important to note that the error $\Delta_{\mathbf{P}}$ is computed using the ℓ_∞ norm and for the set S. Our lower bound will apply to the particular choice of target set $S_0 = [1/2, 1]^k$.

Theorem 4.2. *There is a universal constant c such that for all $d \geq 4$, $k \geq 2$, and $n \geq cd^4k$, we have*

$$\mathfrak{M}_{n,d,k}(S_0, \|\cdot\|_\infty) \geq c\sqrt{\frac{d^2k}{n}}. \tag{4.12}$$

Comparing equations and (4.10) and (4.12), we see that for the ℓ_∞-norm and the set S_0, we have provided upper and lower bounds that match up to a logarithmic factor in the dimension. Thus, the plug-in estimator is indeed optimal for this pair $(\|\cdot\|_\infty, S_0)$. Further, observe that the above lower bound is non-asymptotic, and holds for all values of $n \gtrsim d^4k$.

This condition on the sample size arises as a consequence of the specific packing set used for establishing the lower bound, and improving it is an interesting open problem.

However, this raises the question of whether the set S_0 is special, or alternatively, whether one can obtain an S-dependent lower bound. The following proposition shows that at least *asymptotically*, the sample complexity for *any* polyhedral set S obeys a similar lower bound.

Proposition 4.2 (Informal)**.** *Suppose that we have a valid polyhedral target set S, and that $d \geq 4$. There exists a positive integer $n_0(d, k, S)$ such that for all $n \geq n_0(d, k, S)$ we have*

$$\mathfrak{M}_{n,d,k}(S, \|\cdot\|_\infty) \gtrsim \sqrt{\frac{d^2 k}{n}}\,. \tag{4.13}$$

We defer the formal statement and proof of this proposition to Appendix C.2. This proposition establishes that the plugin estimator $\widehat{\pi}_{\mathsf{plug}}$ is indeed optimal in the ℓ_∞ norm for broad class of sets S. Note that the result is asymptotic in nature: in order for the proposition to hold, we require that the number of samples are greater than the value n_0. This number n_0 depends on problem dependent parameters and we provide an exact expression for n_0 in the appendix.

Instance-specific analysis for plug-in estimator

In the previous section we established that the error $\Delta_{\mathbf{P}}(\widehat{\pi}_{\mathsf{plug}}, \pi^*)$ of the plug-in estimator scales as $\widetilde{O}\left(\sqrt{\frac{d^2 k}{n}}\right)$ for any choice of preference tensor $\mathbf{P}$ and target set S when the distance function $\rho = \|\cdot\|_\infty$. In this section, we study the adaptivity properties of the plug-in estimator $\widehat{\pi}_{\mathsf{plug}}$ and obtain upper bounds on the error $\Delta_{\mathbf{P}}(\widehat{\pi}_{\mathsf{plug}}, \pi^*)$ which depend on the properties of the underlying problem instance.

In the main text, we will restrict our focus to the uni-criterion setup with $k = 1$ with the target set $S = [\frac{1}{2}, 1]$ in which case the Blackwell winner coincides with the von Neumann winner. Furthermore, we will consider the case where the preference matrix $\mathbf{P}$ has a unique von Neumann winner π^*. This is formalized in the following assumption.

Assumption 4.1 (Unique Nash equilibrium)**.** *The matrix $\mathbf{P}$ belongs to the set of preference matrices $\mathcal{P}_{d,1}$ and has a unique mixed Nash equilibrium π^*, that is, $\pi^*_i > 0$ for all $i \in [d]$.*

For the more general analysis, we refer the reader to Appendix C.3. For any preference matrix $\mathbf{P} \in \mathcal{P}_{d,1}$ and the Bernoulli passive sampling model discussed in Section 4.3 let us represent by Σ_i the diagonal matrix corresponding to the variances along the i^{th} column of the matrix $\mathbf{P}$ with

$$\Sigma_i = \mathrm{diag}(\mathbf{P}(1,i) \cdot (1 - \mathbf{P}(1,i)), \ldots, \mathbf{P}(d,i) \cdot (1 - \mathbf{P}(d,i)).$$

Given this notation, we now state an informal corollary (of Theorem C.1) which shows that the error $\Delta_{\mathbf{P}}(\widehat{\pi}_{\mathsf{plug}}, \pi^*)$ depends on the worst-case alignment of the Nash equilibrium π^* with the underlying covariance matrices Σ_i.

Corollary 4.2 (Informal). *For any preference matrix* $\mathbf{P}$ *satisfying Assumption 4.1, confidence* $\delta > 0$*, and number of samples* $n > n_0(\mathbf{P}, \delta)$*, we have that the error* $\Delta_{\mathbf{P}}$ *of the plug-in estimate* $\widehat{\pi}_{\mathsf{plug}}$ *satisfies*

$$\Delta_{\mathbf{P}}(\widehat{\pi}_{\mathsf{plug}}, \pi^*) \leq c \cdot \sqrt{\frac{\sigma_{\mathbf{P}}^2 d^2}{n} \log\left(\frac{d}{\delta}\right)}, \tag{4.14}$$

with probability at least $1 - \delta$ *and the variance* $\sigma_{\mathbf{P}}^2 := \max_{i \in [d]} (\pi^*)^\top \Sigma_i \pi^*$.

We defer the proof of the above to Appendix C.3. A few comments on the above corollary are in order. Observe that the bound above is a high probability bound on the error $\Delta_{\mathbf{P}}$ of the plug-in estimator $\widehat{\pi}_{\mathsf{plug}}$. Compared with the upper bounds of Corollaries 4.1 and C.2, the asymptotic bound on the error above is instance dependent – the effective variance $\sigma_{\mathbf{P}}^2$ depends on the underlying preference matrix $\mathbf{P}$. In particular, this variance measures how well does the underlying von Neumann winner π^* align with the variance associated with each column of the matrix $\mathbf{P}$. In the worst case, since each entry of $\mathbf{P}$ is bounded above by 1, the variance $\sigma_{\mathbf{P}}^2 = 1$ and we recover back the upper bounds from Corollaries 4.1 and C.2 for the uni-criterion case.

Computing the plug-in estimator

In the last few sections, we discussed the statistical properties of the plug-in estimator, and showed that its sample complexity was optimal in a minimax sense. We now turn to the algorithmic question: how can the plug-in estimator $\widehat{\pi}_{\mathsf{plug}}$ be computed? Our main result in this direction is the following theorem that characterizes properties of the objective function $v(\pi; \mathbf{P}, S, \|\cdot\|)$.

Theorem 4.3. *Suppose that the distance function is given by an* ℓ_q *norm* $\|\cdot\|_q$ *for some* $q \geq 1$*. Then for each valid target set* S*, the objective function* $v(\pi; \mathbf{P}, S, \|\cdot\|_q)$ *is convex in* π*, and Lipschitz in the* ℓ_1 *norm, i.e.,*

$$|v(\pi_1; \mathbf{P}, S, \|\cdot\|_q) - v(\pi_2; \mathbf{P}, S, \|\cdot\|_q)| \leq k^{\frac{1}{q}} \cdot \|\pi_1 - \pi_2\|_1 \text{ for each } \pi_1, \pi_2 \in \Delta_d.$$

Theorem 4.3 establishes that the plug-in estimator can indeed be computed as the solution to a (constrained) convex optimization problem. In Appendix C.4, we discuss a few specific algorithms based on zeroth-order and first-order methods for obtaining such a solution and an analysis of the corresponding iteration complexity for these methods; see Propositions C.3 and C.4 in the appendix. These methods differ in the way they access the target set S: while zeroth-order methods require a *distance oracle* to the target set, the first-order methods require a stronger *projection oracle* to this constraint set.

4.4 Autonomous driving user study

In order to evaluate the proposed framework, we applied it to an autonomous driving environment. The objective is to study properties of the randomized policies obtained by our multi-criteria framework—the Blackwell winner for specific choices of the target set—and compare them with the alternative approaches of linear combinations of criteria and the single-criterion (overall) von Neumann winner. We briefly describe the components of the experiment here; see Appendix C.5 for more details.

Self-driving Environment. Figure 4.1(a) shows a snapshot of one of the worlds in this environment with the autonomous car shown in orange. We construct three different worlds in this environment:

W1: The first world comprises an empty stretch of road with no obstacles (20 steps).

W2: The second world consists of a sequence of cones placed in certain sequences (80 steps).

W3: The third world has additional cars driving at varying speeds in their fixed lanes (80 steps).

Policies. For our *base policies*, we design five different reward functions encoding different self-driving behaviors. These polices, named Policy A-E, are then set to be the model predictive control based policies based on these reward functions wherein we fix the planning horizon to 6. We defer the details of these reward functions to Appendix C.5. A *randomized policy* $\pi \in \Delta_5$ is given by a distribution over the base policies A-E. Such a randomized policy is implemented in our environment by randomly sampling a base policy from the mixture distribution after every $H = 18$ time steps and executing this selected policy for that duration. To account for the randomization, we execute each such policy for 5 independent runs in each of the worlds and record these behaviors.

Subjective Criteria. We selected five subjective criteria to compare the policies, with questions asking which of the two policies was C1: Less aggressive, C2: More predictable, C3: More quick, C4: More conservative, and had C5: Less collision risk. Such a framing of question ensures that higher score value along any of C1-C5 is preferred; thus a higher score along C1 would imply less aggressive while along C2 would mean more predictable.

In addition to the these base criteria, we also consider an *Overall Preference* which compares any pair of policies in an aggregate manner. For this criterion, the users were asked to select the policy they would prefer when riding to their destination. Additionally, we also asked the users to rate the importance of each criterion in their overall preference.

Main Hypotheses. The central focus of the main hypotheses is on comparing the randomized policies given by the Blackwell winner, the overall von Neumann winner, and those given by weighing the criteria linearly.

MH1 There exists a set S such that the Blackwell winner with respect to S and ℓ_∞-norm produced by our framework outperforms the overall von Neumann winner.

MH2 The Blackwell winner for oblivious score sets S outperforms both oblivious[2] and data-driven weights for linear combination of criteria.

Independent Variables. The independent variable of our experiment is the choice of algorithms for producing the different randomized winners. These comprise the von Neumann winner based on overall comparisons, Blackwell winners based on two oblivious target sets, and 9 different linear combinations weights (3 data-driven and 6 oblivious).

We begin with the two target sets S_1 and S_2 for our evaluation of the Blackwell winner which were selected in a data-oblivious manner. Set S_1 is an axis-aligned set promoting the use of safer policies with score vector constrained to have a larger value along the collision risk axis. Similar to Figure 4.2(b), the set S_2 adds a linear constraint along aggressiveness and collision risk. This target set thus favors policies which are less aggressive and have lower collision risk. For evaluating hypothesis MH2, we considered several weight vectors, both oblivious and data-dependent, comprising average of the users' self-reported weights, that obtained by regressing the overall criterion on C1-C5, and a set of oblivious weights. See Appendix C.5 for details of the sets S_1 and S_2, and the weights $w_{1:9}$.

Data collection. The experiment was conducted in two phases, both of which involved human subjects on Amazon Mechanical Turk (Mturk) (see Appendix C.5 for an illustration of the questionnaire).

The first phase of the experiment involved preference elicitation for the five base policies A-E. Each user was asked to provide comparison data for all ten combinations of policies. The cumulative comparison data is given in Appendix C.5, and the average weight vector elicited from the users was found to be $w_1 = [0.21, 0.19, 0.20, 0.18, 0.22]$. We ran this study with 50 subjects.

In the overall preference elicitation, we saw an approximate ordering amongst the base policies: C $\succ$ E $\succsim$ D $\succsim$ B $\succ$ A. Thus, Policy C was the von Neumann winner along the overall criterion. For each of the linear combination weights w_1 through w_9, Policy C was the weighted winner. The Blackwell winners R1 and R2 for the sets S_1 and S_2 with the ℓ_∞ distance were found to be R1 $= [0.09, 0.15, 0.30, 0.15, 0.31]$ and R2 $= [0.01, 0.01, 0.31, 0.02, 0.65]$.

In the second phase, we obtained preferences from a set of 41 subjects comparing the randomized polices R1 and R2 with the baseline policies A-E. The results are aggregated in Table C.1 in Appendix C.5.

Analysis for main hypotheses. Given that the overall von Neumann winner and those corresponding to weights $w_{1:9}$ were all Policy C, hypotheses MH1 and MH2 reduced whether

[2]We use the term oblivious to denote variables that were *fixed* before the data collection phase and data-driven to denote those which are based on collected data.

users prefer at least one of {R1, R2} to the deterministic policy C, that is whether $\mathbf{P}_{\mathsf{ov}}(\mathrm{C}, \mathrm{R1}) < 0.5$ or $\mathbf{P}_{\mathsf{ov}}(\mathrm{C}, \mathrm{R2}) < 0.5$.

Policies C and E were preferred to R1 by 0.71 and 0.61 fraction of the respondents, respectively. On the other hand, R2 was preferred to the von Neumann winner C by 0.66 fraction of the subjects. Using the data, we conducted a hypothesis test with the null and alternative hypotheses given by

$$H_0 : \mathbf{P}_{\mathsf{ov}}(\mathrm{C}, \mathrm{R2}) \geq 0.5, \quad \text{and} \quad H_1 : \mathbf{P}_{\mathsf{ov}}(\mathrm{C}, \mathrm{R2}) < 0.5.$$

Among the hypotheses that make up the (composite) null, our samples have the highest likelihood for the distribution $\mathsf{Ber}(0.5)$. We therefore perform a one-sided hypothesis test with the Binomial distribution with number of samples $n = 41$, success probability $p = 0.5$ and number of successes $x = 14$ (indicating number of subjects which preferred Policy C to R1). The p-value for this test was obtained to be 0.0298. This supports both our claimed hypotheses MH1 and MH2.

4.5 Discussion and future work

In this paper, we considered the problem of eliciting and learning from preferences along multiple criteria, as a way to obtain rich feedback under weaker assumptions. We introduced the notion of a Blackwell winner, which generalizes many known winning solution concepts. We showed that the Blackwell winner was efficiently computable from samples with a simple and optimal procedure, and also that it outperformed the von Neumann winner in a user study on autonomous driving. Our work raises many interesting follow-up questions: How does the sample complexity vary as a function of the preference tensor $\mathbf{P}$? Can the process of choosing a good target set be automated? What are the analogs of our results in the setting where pairwise comparisons can be elicited *actively*?

Chapter 5

Reinforcement learning with misspecified rewards

5.1 Introduction

As reinforcement learning agents are trained with better algorithms, more data, and larger policy models, they are at increased risk of overfitting their objectives [171]. *Reward hacking*, or the gaming of misspecified reward functions by RL agents, has appeared in a variety of contexts, such as game playing [110], text summarization [159], and autonomous driving [119]. These examples show that better algorithms and models are not enough; for human-centered applications such as healthcare [218], economics [197] and robotics [120], RL algorithms must be safe and aligned with human objectives [32, 109].

Reward misspecifications occur because real-world tasks have numerous, often conflicting desiderata. In practice, reward designers resort to optimizing a proxy reward that is either more readily measured or more easily optimized than the true reward. For example, consider a recommender system optimizing for users' subjective well-being (SWB). Because SWB is difficult to measure, engineers rely on more tangible metrics such as click-through rates or watch-time. Optimizing for misspecified proxies led YouTube to overemphasize watch-time and harm user satisfaction [189], as well as to recommended extreme political content to users [167].

Addressing reward hacking is a first step towards developing human-aligned RL agents and one goal of ML safety [101]. However, there has been little systematic work investigating when or how it tends to occur, or how to detect it before it runs awry. To remedy this, we study the problem of reward hacking across four diverse environments: traffic control [213], COVID response [122], blood glucose monitoring [78], and the Atari game Riverraid [37]. Within these environments, we construct nine misspecified proxy reward functions (Section 5.2).

Using our environments, we study how increasing optimization power affects reward hacking, by training RL agents with varying resources such as model size, training time, action

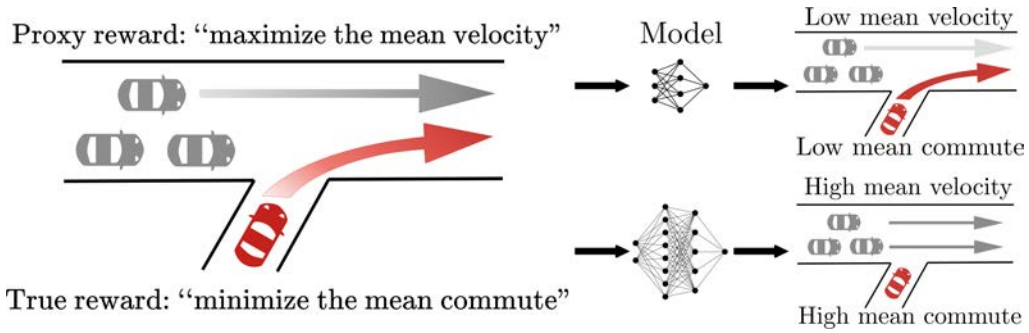


Figure 5.1. An example of reward hacking when cars merge onto a highway. A human-driver model controls the grey cars and an RL policy controls the red car. The RL agent observes positions and velocities of nearby cars (including itself) and adjusts its acceleration to maximize the proxy reward. At first glance, both the proxy reward and true reward appear to incentivize fast traffic flow. However, smaller policy models allow the red car to merge, whereas larger policy models exploit the misspecification by stopping the red car. When the red car stops merging, the mean velocity increases (merging slows down the more numerous grey cars). However, the mean commute time also increases (the red car is stuck). This exemplifies a *phase transition*: the qualitative behavior of the agent shifts as the model size increases.

space resolution, and observation space noise (Section 5.3). We find that more powerful agents often attain higher proxy reward but lower true reward, as illustrated in Figure 5.1. Since the trend in ML is to increase resources exponentially each year [134], this suggests that reward hacking will become more pronounced in the future in the absence of countermeasures.

More worryingly, we observe several instances of *phase transitions*. In a phase transition, the more capable model pursues a qualitatively different policy that sharply decreases the true reward. Figure 5.1 illustrates one example: An RL agent regulating traffic learns to stop any cars from merging onto the highway in order to maintain a high average velocity of the cars on the straightaway.

Since there is little prior warning of phase transitions, they pose a challenge to monitoring the safety of ML systems. Spurred by this challenge, we propose an anomaly detection task [102, 190]: Can we detect when the true reward starts to drop, while maintaining a low false positive rate in benign cases? We instantiate our proposed task, POLYNOMALY, for the traffic and COVID environments (Section 5.4). Given a trusted policy with moderate performance, one must detect whether a given policy is aberrant.

Related work

Previous works have focused on classifying different types of reward hacking and sometimes mitigating its effects. One popular setting is an agent on a grid-world with an erroneous sensor. Hadfield-Menell et al. [99] show and mitigate the reward hacking that arises due to an incorrect sensor reading at test time in a 10x10 navigation grid world. Leike et al. [133]

show examples of reward hacking in a 3x3 boat race and a 5x7 tomato watering grid world. Everitt et al. [74] theoretically study and mitigate reward hacking caused by a faulty sensor.

Game-playing agents have also been found to hack their reward. Baker et al. [16] exhibit reward hacking in a hide-and-seek environment comprising 3-6 agents, 3-9 movable boxes and a few ramps: without a penalty for leaving the play area, the hiding agents learn to endlessly run from the seeking agents. Toromanoff, Wirbel, and Moutarde [195] briefly mention reward hacking in several Atari games (Elevator Action, Kangaroo, Bank Heist) where the agent loops in a sub-optimal trajectory that provides a repeated small reward.

Agents optimizing a learned reward can also demonstrate reward hacking. Ibarz et al. [110] show an agent hacking a learned reward in Atari (Hero, Montezuma's Revenge, and Private Eye), where optimizing a frozen reward predictor eventually achieves high predicted score and low actual score. Christiano et al. [56] show an example of reward hacking in the Pong game where the agent learns to hit the ball back and forth instead of winning the point. Stiennon et al. [188] show that a policy which over-optimizes the learnt reward model for text summarization produces lower quality summarizations when judged by humans.

5.2 Experimental setup: Environments and reward functions

In this section, we describe our four environments (Section 5.2) and taxonomize our nine corresponding misspecified reward functions (Section 5.2).

Environments

We chose a diverse set of environments and prioritized complexity of action space, observation space, and dynamics model. Our aim was to reflect real-world constraints in our environments, selecting ones with several desiderata that must be simultaneously balanced. Table 5.1 provides a summary.

Traffic Control. The traffic environment is an autonomous vehicle (AV) simulation that models vehicles driving on different highway networks. The vehicles are either controlled by a RL algorithm or pre-programmed via a human behavioral model. Our misspecifications are listed in Table 5.1.

We use the Flow traffic simulator, implemented by Wu et al. [213] and Vinitsky et al. [207], which extends the popular SUMO traffic simulator [135]. The simulator uses cars that drive like humans, following the Intelligent Driver Model (IDM) [196], a widely-accepted approximation of human driving behavior. Simulated drivers attempt to travel as fast as possible while tending to decelerate whenever they are too close to the car immediately in front.

The RL policy has access to observations only from the AVs it controls. For each AV, the observation space consists of the car's position, its velocity, and the position and velocity

of the cars immediately in front of and behind it. The continuous control action is the acceleration applied to each AV. Figure 5.4 depicts the Traffic-Mer network, where cars from an on-ramp attempt to merge onto the straightaway. We also use the Traffic-Bot network, where cars (1-4 RL, 10-20 human) drive through a highway bottleneck where lanes decrease from four to two to one.

COVID Response. The COVID environment, developed by Kompella et al. [122], simulates a population using the SEIR model of individual infection dynamics. The RL policymaker adjusts the severity of social distancing regulations while balancing economic health (better with lower regulations) and public health (better with higher regulations), similar in spirit to Trott et al. [197]. The population attributes (proportion of adults, number of hospitals) and infection dynamics (random testing rate, infection rate) are based on data from Austin, Texas.

Every day, the environment simulates the infection dynamics and reports testing results to the agent, but not the true infection numbers. The policy chooses one of three discrete actions: INCREASE, DECREASE, or MAINTAIN the current regulation stage, which directly affects the behavior of the population and indirectly affects the infection dynamics. There are five stages in total.

Atari Riverraid. The Atari Riverraid environment is run on OpenAI Gym [37]. The agent operates a plane which flies over a river and is rewarded by destroying enemies. The agent observes the raw pixel input of the environment. The agent can take one of eighteen discrete actions, corresponding to either movement or shooting within the environment.

Glucose Monitoring. The glucose environment, implemented in Fox et al. [78], is a continuous control problem. It extends a FDA-approved simulator [139] for blood glucose levels of a patient with Type 1 diabetes. The patient partakes in meals and wears a continuous glucose monitor (CGM), which gives noisy observations of the patient's glucose levels. The RL agent administers insulin to maintain a healthy glucose level.

Every five minutes, the agent observes the patient's glucose levels and decides how much insulin to administer. The observation space is the previous four hours of glucose levels and insulin dosages.

Misspecifications

Using the above environments, we constructed nine instances of misspecified proxy rewards. To help interpret these proxies, we taxonomize them as instances of misweighting, incorrect ontology, or incorrect scope. We elaborate further on this taxonimization using the traffic example from Figure 5.1.

- **Misweighting.** Suppose that the true reward is a linear combination of commute time and acceleration (for reducing carbon emissions). Downweighting the acceleration term

Env.	Type	Objective	Proxy	Misalign?	Transition?
Traffic	Mis.	minimize commute and accelerations	underpenalize acceleration	No	No
	Mis.		underpenalize lane changes	Yes	Yes
	Ont.		velocity replaces commute	Yes	Yes
	Scope		monitor velocity near merge	Yes	Yes
COVID	Mis.	balance economic, health, political cost	underpenalize health cost	No	No
	Ont.		ignore political cost	Yes	Yes
Atari	Mis.	score points under smooth movement	downweight movement	No	No
	Ont.		include shooting penalty	No	No
Glucose	Ont.	minimize health risk	risk in place of cost	Yes	No

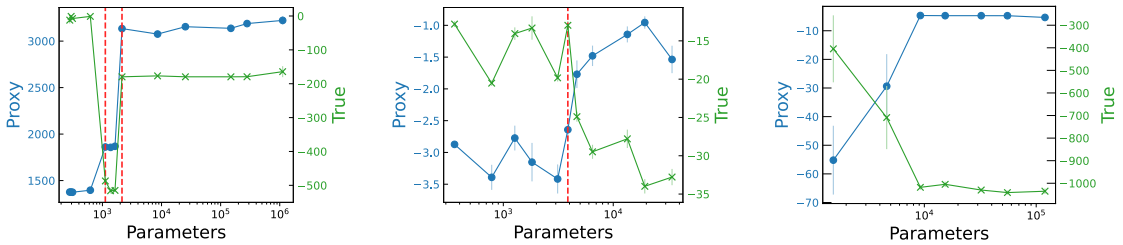
Table 5.1. Reward misspecifications across our four environments. 'Misalign' indicates whether the true reward drops and 'Transition' indicates whether this corresponds to a phase transition (sharp qualitative change). We observe 5 instances of misalignment and 4 instances of phase transitions. 'Mis.' is a misweighting and 'Ont.' is an ontological misspecification.

thus underpenalizes carbon emissions. In general, misweighting occurs when the proxy and true reward capture the same desiderata, but differ on their relative importance.

- **Ontological.** Congestion could be operationalized as either high average commute time or low average vehicle velocity. In general, ontological misspecification occurs when the proxy and true reward use different desiderata to capture the same concept.
- **Scope.** If monitoring velocity over all roads is too costly, a city might instead monitor them only over highways, thus pushing congestion to local streets. In general, scope misspecification occurs when the proxy measures desiderata over a restricted domain (e.g. time, space).

We include a summary of all nine tasks in Table 5.1 and provide full details in Appendix D.1. Table 5.1 also indicates whether each proxy leads to misalignment (i.e. to a policy with low true reward) and whether it leads to a phase transition (a sudden qualitative shift as model capacity increases). We investigate both of these in Section 5.3.

Evaluation protocol. For each environment and proxy-true reward pair, we train an agent using the proxy reward and evaluate performance according to the true reward. We use PPO [174] to optimize policies for the traffic and COVID environments, SAC [97] to optimize the policies for the glucose environment, and torchbeast [128], a PyTorch implementation of IMPALA [72], to optimize the policies for the Atari environment. When available, we adopt the hyperparameters (except the learning rate and network size) given by the original codebase.



(a) Traffic - Ontological (b) COVID - Ontological (c) Glucose - Ontological

Figure 5.2. Increasing the RL policy's model size decreases true reward on three selected environments. The red line indicates a phase transition.

5.3 How Agent Optimization Power Drives Misalignment

To better understand reward hacking, we study how it emerges as agent optimization power increases. We define optimization power as the effective search space of policies the agent has access to, as implicitly determined by model size, training steps, action space, and observation space.

In Section 5.3, we consider the quantitative effect of optimization power for all nine environment-misspecification pairs; we primarily do this by varying model size, but also use training steps, action space, and observation space as robustness checks. Overall, more capable agents tend to overfit the proxy reward and achieve a lower true reward. We also find evidence of phase transitions on four of the environment-misspecification pairs. For these phase transitions, there is a critical threshold at which the proxy reward rapidly increases and the true reward rapidly drops.

In Section 5.3, we further investigate these phase transitions by qualitatively studying the resulting policies. At the transition, we find that the quantitative drop in true reward corresponds to a qualitative shift in policy behavior. Extrapolating visible trends is therefore insufficient to catch all instances of reward hacking, increasing the urgency of research in this area.

In Section 5.3, we assess the faithfulness of our proxies, showing that reward hacking occurs even though the true and proxy rewards are strongly positively correlated in most cases.

Quantitative Effects vs. Agent Capabilities

As a stand-in for increasing agent optimization power, we first vary the model capacity for a fixed environment and proxy reward. Specifically, we vary the width and depth of the actor and critic networks, changing the parameter count by two to four orders of magnitude depending on the environment. For a given policy, the actor and critic are always the same size.

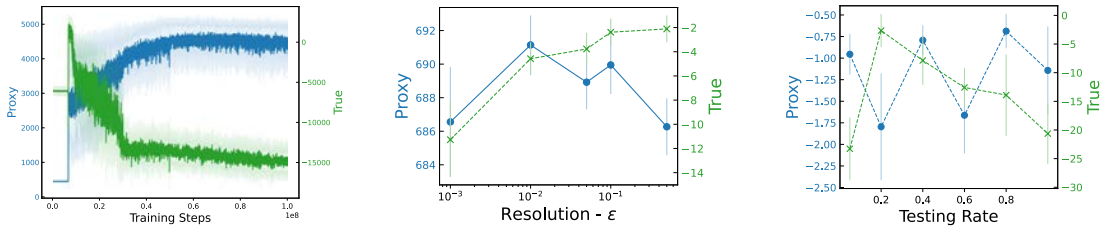
Model Capacity. Our results are shown in Figure 5.2, with additional plots included in Appendix D.1. We plot both the proxy (blue) and true (green) reward vs. the number of parameters. As model size increases, the proxy reward increases but the true reward decreases. This suggests that reward designers will likely need to take greater care to specify reward functions accurately and is especially salient given the recent trends towards larger and larger models [134].

The drop in true reward is sometimes quite sudden. We call these sudden shifts *phase transitions*, and mark them with dashed red lines in Figure 5.2. These quantitative trends are reflected in the qualitative behavior of the policies (Section 5.3), which typically also shift at the phase transition.

Model capacity is only one proxy for agent capabilities, and larger models do not always lead to more capable agents [11]. To check the robustness of our results, we consider several other measures of optimization: observation fidelity, number of training steps, and action space resolution.

Number of training steps. Assuming a reasonable RL algorithm and hyperparameters, agents which are trained for more steps have more optimization power. We vary training steps for an agent trained on the Atari environment. The true reward incentivizes staying alive for as many frames as possible while moving smoothly. The proxy reward misweights these considerations by underpenalizing the smoothness constraint. As shown in Figure 5.3a, optimizing the proxy reward for more steps harms the true reward, after an initial period where the rewards are positively correlated.

Action space resolution. Intuitively, an agent that can take more precise actions is more capable. For example, as technology improves, an RL car may make course corrections every millisecond instead of every second. We study action space resolution in the traffic



(a) Atari - Misweighting (b) Traffic - Ontological (c) COVID - Ontological

Figure 5.3. In addition to parameter count, we consider three other agent capabilities: training steps, action space resolution, and observation noise. In Figure 5.3a, an increase in the proxy reward comes at the cost of the true reward. In Figure 5.3b, increasing the granularity (from right to left) causes the agent to achieve similar proxy reward but lower true reward. In Figure 5.3c, increasing the fidelity of observations (by increasing the random testing rate in the population) tends to decrease the true reward with no clear impact on proxy reward.

environment by discretizing the output space of the RL agent. Specifically, under resolution level ε, we round the action $a \in \mathbb{R}$ output by the RL agent to the nearest multiple of ε and use that as our action. The larger the resolution level ε, the lower the action space resolution. Results are shown in Figure 5.3b for a fixed model size. Increasing the resolution causes the proxy reward to remain roughly constant while the true reward decreases.

Observation fidelity. Agents with access to better input sensors, like higher-resolution cameras, should make more informed decisions and thus have more optimization power. Concretely, we study this in the COVID environment, where we increase the random testing rate in the population. The proxy reward is a linear combination of the number of infections and severity of social distancing, while the true reward also factors in political cost. As shown in Figure 5.3c, as the testing rate increases, the model achieves similar proxy reward at the cost of a slightly lower true reward.

Qualitative Effects

In the previous section, quantitative trends showed that increasing a model's optimization power often hurts performance on the true reward. We shift our focus to understanding *how* this decrease happens. In particular, we typically observe a qualitative shift in behavior associated with each of the phase transitions, three of which we describe below.

Traffic Control. We focus on the Traffic-Mer environment from Figure 5.2a, where minimizing average commute time is replaced by maximizing average velocity. In this case, smaller policies learn to merge onto the straightaway by slightly slowing down the other vehicles (Figure 5.4a). On the other hand, larger policy models stop the AVs to prevent them from merging at all (Figure 5.4b). This increases the average velocity, because the vehicles on the straightaway (which greatly outnumber vehicles on the on-ramp) do not need to slow down for merging traffic. However, it significantly increases the average commute time, as the passengers in the AV remain stuck.

(a) Traffic policy of smaller network (b) Traffic policy of larger network

Figure 5.4. The larger model prevents the AVs (in red) from moving to increase the velocity of the human cars (unobserved cars in white and observed cars in blue). However, this greatly increases the average commute per person.

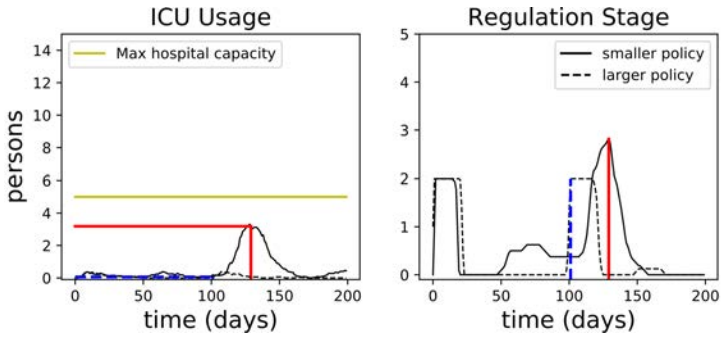


Figure 5.5. For COVID, ICU usage is a proxy for public health and regulation stage is a proxy for economic health. The blue line indicates the maximum stage (right) enforced by the larger policy and the corresponding ICU level (left) at that stage. The red line is the equivalent for the smaller policy. Because the larger policy enforces regulations much sooner than the smaller policy, it maintains both low ICU usage and low regulation stage. However, the larger policy is politically unfavorable: regulations are high even though public signs of infection, such as ICU usage, are low.

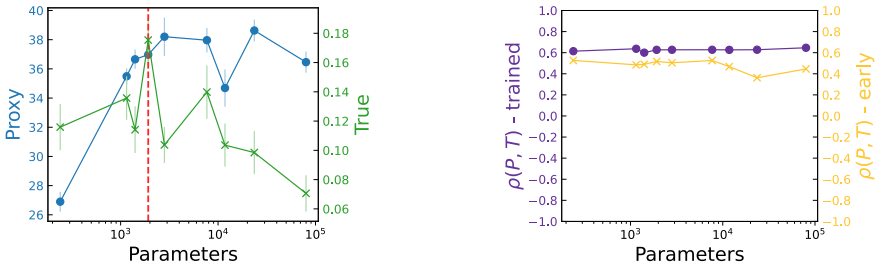
COVID Response. Suppose the RL agent optimizes solely for the public and economic health of a society, without factoring politics into its decision-making. This behavior is shown in Figure 5.5. The larger model chooses to increase the severity of social distancing restrictions earlier than the smaller model. As a result, larger models are able to maintain low average levels of both ICU usage (a proxy for public health) and social distancing restrictions (a proxy for economic health). These preemptive regulations may however be politically costly, as enforcing restrictions without clear signs of infection may foment public unrest [30].

Atari Riverraid. We create an ontological misspecification by rewarding the plane for staying alive as long as possible while shooting as little as possible: a "pacifist run". We then measure the game score as the true reward. We find that agents with more parameters typically maneuver more adeptly. Such agents shoot less frequently, but survive for much longer, acquiring points (true reward) due to passing checkpoints. In this case, therefore, the proxy and true rewards are well-aligned so that reward hacking does not emerge as capabilities increase.

We did, however, find that some of the agents exploited a bug in the simulator that halts the plane at the beginning of the level. The simulator advances but the plane itself does not move, thereby achieving high pacifist reward.

Glucose Monitoring. Consider an RL agent that optimizes solely for a patient's health, without considering the economic costs of its treatment plans. In this case, the proxy reward is based off of a glycemic risk measure, which reflects the likelihood that a patient will suffer an acute hypoglycemic episode, developed by the medical community [124].

However, a less economically-privileged patient may opt for the treatment plan with the least expected cost [104, 79], not the one with the least amount of risk. From this patient's



(a) *Traffic-Mer* - Space (b) Correlation for Figure 5.6a

Figure 5.6. Correlations between the proxy and true rewards, along with the reward hacking induced. In Figure 5.6a, we plot the proxy reward with "•" and the true reward with "×". In Figure 5.6b, we plot the trained checkpoint correlation and the early checkpoint correlation.

perspective, the true reward is the expected cost of the treatment plan, which includes the expected cost of hospital visits and the cost of administering the insulin.

Although larger model treatments reduce hypoglycemic risk more smaller model treatments, they administer more insulin. Based on the average cost of an ER visit for a hypogylcemic episode ($1350 from Bronstone and Graham [39]) and the average cost of a unit of insulin ($0.32 from Lee [131]), we find that it is actually more expensive to pursue the larger model's treatment.

Quantitative Effects vs Proxy-True Reward Correlation

We saw in Sections 5.3 and 5.3 that agents often pursue proxy rewards at the cost of the true reward. Perhaps this only occurs because the proxy is greatly misspecified, i.e., the proxy and true reward are weakly or negatively correlated. If this were the case, then reward hacking may pose less of a threat. To investigate this intuition, we plot the correlation between the proxy and true rewards.

The correlation is determined by the state distribution of a given policy, so we consider two types of state distributions. Specifically, for a given model size, we obtain two checkpoints: one that achieves the highest proxy reward during training and one from early in training (less than 1% of training complete). We call the former the "trained checkpoint" and the latter the "early checkpoint".

For a given model checkpoint, we calculate the Pearson correlation ρ between the proxy reward P and true reward T using 30 trajectory rollouts. Reward hacking occurs even though there is significant positive correlation between the true and proxy rewards (see Figure 5.6). The correlation is lower for the trained model than for the early model, but still high. Further figures are shown in Appendix D.1. Among the four environments tested, only the Traffic-Mer environment with ontological misspecification had negative Pearson correlation.

5.4 Polynomaly: Mitigating reward misspecification

In Section 5.3, we saw that reward hacking often leads to phase transitions in agent behaviour. Furthermore, in applications like traffic control or COVID response, the true reward may be observed only sporadically or not at all. Blindly optimizing the proxy in these cases can lead to catastrophic failure [223, 191].

This raises an important question: Without the true reward signal, how can we mitigate misalignment? We operationalize this as an anomaly detection task: the detector should flag instances of misalignment, thus preventing catastrophic rollouts. To aid the detector, we provide it with a *trusted policy*: one verified by humans to have acceptable (but not maximal) reward. Our resulting benchmark, POLYNOMALY, is described below.

Problem Setup

We train a collection of policies by varying model size on the traffic and COVID environments. For each policy, we estimate the policy's true reward by averaging over 5 to 32 rollouts. One author labeled each policy as acceptable, problematic, or ambiguous based on its true reward score relative to that of other policies. We include only policies that received a non-ambiguous label.

For both environments, we provide a small-to-medium sized model as the trusted policy model, as Section 5.3 empirically illustrates that smaller models achieve reasonable true reward without exhibiting reward hacking. Given the trusted model and a collection of policies, the anomaly detector's task is to predict the binary label of "acceptable" or "problematic" for each policy.

Table D.1 in Appendix D.2 summarizes our benchmark. The trusted policy size is a list of the hidden unit widths of the trusted policy network (not including feature mappings).

Evaluation

We propose two evaluation metrics for measuring the performance of our anomaly detectors.

- *Area Under the Receiver Operating Characteristic (AUROC).* The AUROC measures the probability that a detector will assign a random anomaly a higher score than a random non-anomalous policy [62]. Higher AUROCs indicate stronger detectors.

- *Max F-1 score.* The F-1 score is the harmonic mean of the precision and the recall, so detectors with a high F-1 score have both low false positives and high true negatives. We calculate the max F-1 score by taking the maximum F-1 score over all possible thresholds for the detector.

Baselines

In addition to the benchmark datasets described above, we provide baseline anomaly detectors based on estimating distances between policies. We estimate the distance between the trusted policy and the unknown policy based on either the Jensen-Shannon divergence (JSD) or the Hellinger distance. Specifically, we use rollouts to generate empirical action distributions. We compute the distance between these action distributions at each step of the rollout, then aggregate across steps by taking either the mean or the range. For full details, see Appendix D.2. Table 5.2 reports the AUROC and F-1 scores of several such detectors. We provide full ROC curves in Appendix D.2.

Baseline Detectors	Mean Jensen-Shannon		Mean Hellinger		Range Hellinger	
Env. - Misspecification	AUROC	Max F-1	AUROC	Max F-1	AUROC	Max F-1
Traffic-Mer - misweighting	81.0%	0.824	81.0%	0.824	76.2%	0.824
Traffic-Mer - scope	74.6%	0.818	74.6%	0.818	57.1%	0.720
Traffic-Mer - ontological	52.7%	0.583	55.4%	0.646	71.4%	0.842
Traffic-Bot - misweighting	88.9%	0.900	88.9%	0.900	74.1%	0.857
COVID - ontological	45.2%	0.706	59.5%	0.750	88.1%	0.923

Table 5.2. Performance of detectors on different subtasks. Each detector has at least one subtask with AUROC under 60%, indicating poor performance.

We observe that different detectors are better for different tasks, suggesting that future detectors could do better than any of our baselines. Our benchmark and baseline provides a starting point for further research on mitigating reward hacking.

5.5 Discussion

In this work, we designed a diverse set of environments and proxy rewards, uncovered several instances of phase transitions, and proposed an anomaly detection task to help mitigate these transitions. Our results raise two questions: How can we not only detect phase transitions, but prevent them in the first place? And how should phase transitions shape our approach to safe ML?

On preventing phase transitions, anomaly detection already offers one path forward. Once we can detect anomalies, we can potentially prevent them, by using the detector to purge the unwanted behavior (e.g. by including it in the training objective). Similar policy shaping has recently been used to make RL agents more ethical [103]. However, since the anomaly detectors will be optimized against by the RL policy, they need to be adversarially robust [93]. This motivates further work on adversarial robustness and adversarial anomaly detection. Another possible direction is optimizing policies against a distribution of rewards [40, 113], which may prevent over-fitting to a given set of metrics.

Regarding safe ML, several recent papers propose extrapolating empirical trends to forecast future ML capabilities [115, 105, 68], partly to avoid unforeseen consequences from ML. While we support this work, our results show that trend extrapolation alone is not enough to ensure the safety of ML systems. To complement trend extrapolation, we need better interpretability methods to identify emergent model behaviors early on, before they dominate performance [154]. ML researchers should also familiarize themselves with emergent behavior in self-organizing systems [217], which often exhibit similar phase transitions [10]. Indeed, the ubiquity of phase transitions throughout science suggests that ML researchers should continue to expect surprises–and should therefore prepare for them.

Chapter 6

Reward learning as doubly nonparametric bandits

6.1 Introduction

Specifying the reward function accurately for a desired objective, or *reward engineering*, is challenging to perform by hand, as the consequences of even small errors can be drastic [98]. To address this, reward learning seeks to learn a predictive model of the reward function from data, which is obtained from carefully selected queries to human annotators. The learned reward model is then used as the optimization objective for policy learning. Reward learning has achieved significant empirical success in domains such as text summarization [187, 31], robot locomotion [60], predicting driving styles [126], and Atari game playing [57].

Despite their success, reward learning methods still lack theoretical grounding. Moreover, their behavior can be brittle even on simple tasks, due to the difficulty of choosing appropriate queries and due to feedback loops from adaptive querying [80]. Indeed, an ablation study in Christiano et al. [57] suggests that random queries can outperform or be competitive with adaptive query procedures. To address these issues, we provide a theoretical framework for analyzing reward learning, framing it as a *doubly nonparametric experimental design* problem. This framework helps elucidate the role of query selection [47] and also enables us to derive scaling laws—how the sizes of the policy and reward models affect the query complexity—for reward learning [116].

Proposed framework. In our framework, we suppose we are given a reward class C_r and policy class C_π. Our goal is to find a policy $\hat{\pi} \in C_\pi$ that performs well according to an unknown true reward $r^* \in C_r$. To do this, we query policies $\pi \in C_\pi$, observing noisy estimates of their true reward, and use this information to choose the eventual policy $\hat{\pi}$.

To be compatible with modern nonparametric learning methods (i.e. neural nets), we view C_r and C_π as subsets of Reproducing Kernel Hilbert Spaces (RKHS). A salient feature of our proposed framework is that the learner therefore optimizes a nonparametric reward function over a nonparametric space of policies, making the task "doubly" nonparametric.

In contrast, previous work considers a nonparametric function class or reward class, but typically not both. For instance, nonparametric zeroth order or bandit optimization [185, 146, 211] considers a nonparametric function on a *finite-dimensional* input space. Conversely, nonparametric supervised learning [208, 107] minimizes a *known* loss function over a nonparametric input space.

The doubly nonparametric nature of our task poses new challenges. The (possibly) infinite-dimensional RKHS requires the learner to select which subspace to explore given a finite number of queries. Furthermore, the unknown reward function makes it challenging for the learner to reason about the information gained from the selected query policies. We address these challenges by deriving a risk upper bound for a family of plug-in estimators based on ridge regression, and then optimizing this bound to solve the optimal design task. Our results show that the quality of the output policy depends on how well the query set $\mathcal{Q}$ is aligned with the eigenfunctions of the policy space.

In addition to the optimal design problem, our framework allows us to study scaling laws with respect to the reward (or policy) class by varying the rate of decay of their corresponding eigenspectrum. This decay rate determines the effective dimensionality of a RKHS [222], and provides a natural proxy for varying the the size of the reward or policy class. Qualitatively, our main results show that the excess risk asymptotically vanishes as long as the policy class grows at a slower rate relative to the reward class.

Sharpness of analysis. Our risk bounds apply to reward and policy classes of arbitrary or even infinite dimensionality. Despite this generality, we show they provide stronger guarantees than previous bounds for the specialized settings of compact policy sets and kernel multi-armed bandits.

In Section 6.4, we look at a special case of our problem when the policy set C_π is a compact subspace and thus has finite rank. For these instances, we show that our learning algorithm obtains a better excess risk $O(n^{-\frac{\beta}{\beta+2}})$ versus a rate of $O(n^{-\frac{\beta-1}{2(\beta+1)}})$ obtained by the adaptive GP-UCB algorithm [185], where $\beta > 0$ is a power law decay rate.

In Section 6.5, we specialize our general results to the well-studied problem of Gaussian process bandit optimization [212], also known as kernel multi-armed bandit (MAB). Specifically, for the class of Matérn kernels with parameter ν in d dimensions, we show that our algorithm achieves a regret bound of $\tilde{O}(T^{\frac{4\nu+d(4d+6)}{6\nu+d(4d+7)}})$ which is strictly better than those achieved by the GP-UCB and GP-Thompson Sampling (GP-TS) [55] algorithms and comparable with π-GP UCB [112] and supKernelUCB [203, 201]; see Table 6.1 for details. GP-UCB and GP-TS are only yield sub-linear regret bounds when the smoothness of the kernel $\nu > d^2$—thus in high dimensions, these bounds essentially become vacuous. The π-GP UCB algorithm was designed specifically to overcome this issue. Our proposed algorithm achieves sublinear regret for all $\nu > 3/2$.

Our Contributions. We propose doubly-nonparametric bandits as a framework for theoretically studying the reward learning problem. Within this framework, we obtain finite sample risk bounds for a ridge regression based plug-in estimator and derive scaling laws for reward learning. From a technical standpoint, we study the optimal design problem for our

estimator to select informative query points by showing that the excess risk depends only on the spectral properties of a certain operator of the two RKHSs and the empirical covariance matrix. As a corollary of our risk bounds, we provide sharper regret bounds for a class of kernel MAB problems compared to several existing algorithms, showing that the doubly-nonparametric lens of reward learning is fruitful even for "singly-nonparametric" tasks. To obtain these bounds, our reduction carefully constructs two different RKHSs to embed the input space and reward function into a policy and reward class.

6.2 Framework: Doubly nonparametric Bandits

Our framework considers non-parametric policy learning with non-parametric reward models. We let $\pi \in \mathbb{H}_\pi$ denote an arbitrary policy and $r \in \mathbb{H}_r$ denote an arbitrary reward function, where $\mathbb{H}_\pi$ and $\mathbb{H}_r$ are Reproducing Kernel Hilbert Spaces. For technical reasons, we assume the corresponding kernel functions $\mathcal{K}_\pi$ and $\mathcal{K}_r$ both satisfy the Hilbert-Schmidt condition (see Appendix E.1 for details).

We let $F(\pi, r) \in \mathbb{R}$ denote the reward obtained by selecting policy π under reward function r and consider the case where the evaluation functional F is linear in both π and r. In other words, $F(\pi, r) = \langle r, M\pi \rangle_{\mathbb{H}_r}$ where $M : \mathbb{H}_\pi \mapsto \mathbb{H}_r$ is a known linear mapping from the policy space to the reward space. Since $\mathbb{H}_\pi$ and $\mathbb{H}_r$ may be infinite-dimensional, linearity is only a weak restriction–e.g. the map $f \mapsto f(x)$ is linear in f for any RKHS.

To incorporate problem structure, we let r^* denote the true reward function and assume that $r^* \in C_r$ for some known set $C_r \subseteq \mathbb{H}_r$ such that $\|r^*\|_{\mathbb{H}_r} = 1$. We further assume that policies π are restricted to lie in some C_π which is a subset of the unit ball in $\mathbb{H}_\pi$ (for instance, C_π might incorporate physical constraints on implementable policies). Thus, given the true reward r^*, the optimal policy (for a compact C_π) is $\pi^* \in \operatorname{argmax}_{\pi \in C_\pi} F(\pi, r^*)$. This proposed framework, which allows for infinite-dimensional policy as well as reward classes, allows us to study how both the policy and reward space affect the difficulty of learning.

Query access to reward r^*. The true reward function r^* is unknown to the learner but is accessible via queries to an oracle (e.g. a human expert), which provide noisy zeroth-order (or bandit) evaluations of the reward r^*. When queried with a policy $\pi \in C_\pi$, the oracle provides a response

$$\text{Oracle } \mathcal{O}r^* : \pi \mapsto F(\pi, r^*) + \epsilon \quad \text{with } \epsilon \sim \mathcal{N}(0, \tau^2) \,, \tag{6.1}$$

with τ^2 denoting the variance of the response. There are two possible query models: passive queries [15, 176], where the learner selects all queries at the same time, and active queries [42, 130], where the learner is allowed to select queries sequentially. Our focus in this work will be on the passive query model, but in many cases we will outperform existing active query algorithms.

Problem statement. Given passive access to the oracle $\mathcal{O}_{r^*}$, the objective of the learner is to output a policy $\hat{\pi} \in C_\pi$ that has small excess risk Δ, defined as

$$\Delta(\hat{\pi}; r^*) := F(\pi^*, r^*) - F(\hat{\pi}, r^*) \,. \tag{6.2}$$

We think of queries to the oracle as expensive, and are interested in achieving low excess risk with as few queries as possible. This notion of excess risk is also studied by the term *simple regret* in pure exploration bandit problems [130].

Representations in $\ell_2(\mathbb{N})$. By Mercer's theorem, we can represent any RKHS as a subset of $\ell_2(\mathbb{N})$. Formally, the policy and the reward spaces are isomorphic to the ellipsoids

$$\mathbb{H}_\pi := \left\{ \sum_{j=1}^{\infty} \kappa_{\pi,j}\phi_{\pi,j} \;\middle|\; (\kappa_{\pi,j})_{j=1}^{\infty} \in \ell^2(\mathbb{N}) \text{ with } \sum_{j=1}^{\infty} \frac{\kappa_{\pi,j}^2}{\mu_{\pi,j}^2} < \infty \right\}$$

$$\mathbb{H}_r := \left\{ \sum_{j=1}^{\infty} \kappa_{r,j}\phi_{r,j} \;\middle|\; (\kappa_{r,j})_{j=1}^{\infty} \in \ell^2(\mathbb{N}) \text{ with } \sum_{j=1}^{\infty} \frac{\kappa_{r,j}^2}{\mu_{r,j}^2} < \infty \right\}, F$$

for appropriately chosen eigenfunctions $\phi_{\pi,j}$ and $\phi_{r,j}$, and corresponding eigenvalues $\mu_{\pi,j}$ and $\mu_{r,j}$ [209]. These are defined with respect to a base measure $\mathbb{P}$ over the input domain; see Appendix E.1 for details. With a slight abuse of notation, going forward, we will use π and r to denote the corresponding coefficients $(\kappa_{\pi,j})$ and $(\kappa_{r,j})$ in the expansion above. [1] With this, the inner products associated with $\mathbb{H}_\pi$ and $\mathbb{H}_r$ simplify

$$\langle \pi_1, \pi_2 \rangle_{\mathbb{H}_\pi} := \sum_{j=1}^{\infty} \frac{\pi_{1,j}\pi_{2,j}}{\mu_{\pi,j}} \quad \text{and} \quad \langle r_1, r_2 \rangle_{\mathbb{H}_r} := \sum_{j=1}^{\infty} \frac{r_{1,j}r_{2,j}}{\mu_{r,j}} \,. \tag{6.3}$$

Also let $S_r := \mathrm{diag}(\mu_{r,j}^{-1})$ and $S_\pi := \mathrm{diag}(\mu_{\pi,j}^{-1})$ be diagonal matrices comprising the inverse of the eigenvalues of $\mathbb{H}_r$ and $\mathbb{H}_\pi$. With this notation, if we view the map M as a (infinite-dimensional) matrix, its Hermitian adjoint[2] is equal to $M^* = S_\pi^{-1} M^\top S_r$.

In order for the evaluation functional $g(\pi, r^*)$ to be finite for all $\pi \in \mathbb{H}_\pi$, the operator norm $\|S_r^{\frac{1}{2}} M S_\pi^{-\frac{1}{2}}\|_{\mathsf{op}}$ must be bounded (see Appendix E.1). We will see later that the decay of this operator's singular values is closely related to the difficulty of learning in our setting.

6.3 Algorithm: Policy Learning via Reward Learning

Given the setup above, we now describe a meta-algorithm, policy learning via reward learning (Algorithm 3), for the non-parametric policy learning problem. The algorithm is a three-stage procedure: it (i) selects a subset of policies $\mathcal{Q}$ to query for reward feedback, (ii) uses the responses to learn a reward estimate $\hat{r}$, and (iii) optimizes this learnt estimate to output the policy $\hat{\pi}_{\mathsf{plug}}$, that is, $\hat{\pi}_{\mathsf{plug}} \in \mathrm{argmin}_{\pi \in C_\pi} \langle \hat{r}, M\pi \rangle_{\mathbb{H}_r}$. Such general plug-in procedure have been studied in the statistics [204] and the machine learning [63] literature. We analyze the excess risk of this estimator for our doubly-nonparametric setup and use this risk bound to select our query set $\mathcal{Q}$. We now discuss the two key design choices in our algorithm: the choice of the reward estimation procedure as well as the choice of query set $\mathcal{Q}$.

[1] While the eigenfunctions ϕ_π and ϕ_r can be different, this representation can still be used by modifying the map M appropriately. This is detailed in Appendix E.1.

[2] Recall the Hermitian adjoint of M satisfies $\langle r, M\pi \rangle_{\mathbb{H}_r} = \langle M^* r, \pi \rangle_{\mathbb{H}_\pi}$

Algorithm 3: Policy Learning via Reward Learning

Input: Number of queries n, policy set C_π, oracle $\mathcal{O}_{r^*}$
Select n policies $\mathcal{Q} = \{\pi_1, \dots, \pi_n\}$ and receive noisy reward evaluations $y_i = \mathcal{O}_{r^*}(\pi_i)$.
Estimate $\hat{r}$ using observed responses $\{(\pi_1, y_1), \dots, (\pi_n, y_n)\}$ using ridge regression (6.4).
Obtain plug-in policy $\hat{\pi}_{\mathsf{plug}} \in \operatorname{argmax}_{\pi \in C_\pi} F(\pi, \hat{r})$.
Output: Policy $\hat{\pi}_{\mathsf{plug}}$

Reward learning via ridge regression. We estimate the reward $\hat{r}$ via ridge regression in the RKHS $\mathbb{H}_r$ [81, 183]. Suppose that in the first step of the algorithm, we have already queried the oracle on n policies and let $\{(\pi_i, y_i)\}_{i=1}^n$ represent the query-response pairs. For a regularization parameter $\lambda_{\mathsf{reg}} > 0$, the ridge regression estimate of the reward function is

$$\hat{r} \in \operatorname*{argmin}_{r \in \mathbb{H}_r} \frac{1}{n} \sum_{i=1}^{n} (y_i - \langle r, M\pi_i \rangle_{\mathbb{H}_r})^2 + \lambda_{\mathsf{reg}} \|r\|_{\mathbb{H}_r}^2 \,. \tag{6.4}$$

The parameter λ_{reg}, which is usually set as a function of n, controls the bias-variance trade-off in estimating r^*—smaller values of λ_{reg} reduce bias while larger values help reduce variance.

Excess risk bound for fixed query set. Observe that the plug-in estimator $\hat{\pi}_{\mathsf{plug}}(\mathcal{Q})$ is implicitly a function of the query set $\mathcal{Q}$. Ideally, we want to choose the set $\mathcal{Q}$ which minimizes the expected risk of the plugin estimator. This requires us to solve the optimization problem

$$\mathcal{Q} = \operatorname*{argmin}_{S : |S| \le n} \mathbb{E}[\Delta(\hat{\pi}_{\mathsf{plug}}(S); r^*)] \,. \tag{6.5}$$

However, solving the above precisely requires knowledge about the underlying reward function r^*, and the combinatorial nature of the optimization problem makes it hard to find an exact solution. To address this, we first upper bound the excess risk of the plug-in policy $\hat{\pi}_{\mathsf{plug}}$ in terms of the query set $\mathcal{Q} = \{\pi_1, \dots, \pi_n\}$. The following theorem[3] bounds the excess risk in terms of the spectrum of the spaces $\mathbb{H}_r$ and $\mathbb{H}_\pi$, as well as the covariance matrix of the queried policies $\Sigma_{\mathcal{Q}} := \frac{1}{n} \sum_{\pi \in \mathcal{Q}} \pi\pi^\top$.

Theorem 6.1 (Excess risk of plug-in)**.** *For any query set $\mathcal{Q}$ consisting of n policies and regularization parameter $\lambda_{\mathsf{reg}} > 0$, the excess risk of the plug-in estimator $\hat{\pi}_{\mathsf{plug}}$ is upper bounded as*

$$\mathbb{E}[\Delta(\hat{\pi}_{\mathsf{plug}}; r^*)] \le 2\mathbb{E}[\|M^*(r^* - \hat{r})\|_{\mathbb{H}_\pi}] \,. \tag{6.6}$$

In addition, letting $A = M\Sigma_{\mathcal{Q}} M^\top S_r + \lambda_{\mathsf{reg}} I$, the expected squared distance is equal to

$$\mathbb{E}[\|M^*(r^* - \hat{r})\|_{\mathbb{H}_\pi}^2] = \lambda_{\mathsf{reg}}^2 \cdot \|M^* A^{-1} r^*\|_{\mathbb{H}_\pi}^2 + \frac{\tau^2}{n} \cdot \operatorname{tr}\left[S_\pi (M^* A^{-1} M) \Sigma_{\mathcal{Q}} (M^* A^{-1} M)^\top\right] . \tag{6.7}$$

[3]Throughout the paper, for clarity purposes, we denote by c a universal constant whose value changes across lines. All our proofs in the appendices explicitly track this constant.

The proof follows a standard analysis of ridge regression and is deferred to Appendix E.2. Observe that in the above theorem, the query set $\pi \in \mathcal{Q}$ participates in the excess risk only via the covariance $\Sigma_{\mathcal{Q}}$. The risk bound is the sum of two term: the first corresponding to the bias and the second corresponding to the variance. In both these terms, $\Sigma_{\mathcal{Q}}$ appears as part of A^{-1}—thus query sets $\mathcal{Q}$ which induce a larger correlation with the map M will generally have lower excess risk. Choices of queries which are orthogonal to the right singular vectors of M will have a constant excess risk, since for those directions the matrix $A \approx \lambda_{\mathsf{reg}} I$.

As shown later in the appendix, in the special case when the policy set consists of the entire unit ball $C_\pi = \{\pi \in \mathbb{H}_\pi \mid \|\pi\|_{\mathbb{H}_\pi} \leq 1\}$, the excess risk bound can be improved by a quadratic factor $\mathbb{E}[\Delta(\hat{\pi}_{\mathsf{plug}}; r^*)] \leq O\left(\|M^*(r^* - \hat{r})\|^2_{\mathbb{H}_\pi}\right)$. Such a phenomenon was first observed in the finite-dimensional setup by [170].

6.4 Query selection and statistical guarantees

We now show how to select the query set $\mathcal{Q}$ effectively and study the excess risk of the corresponding plug-in estimator $\hat{\pi}_{\mathsf{plug}}$ obtained via this query set. We will start with the special case where the policy set C_π is the unit ball in $\mathbb{H}_\pi$ and the map M is diagonal, and then generalize to arbitrary policy sets. In both cases, low excess risk can be achieved by repeatedly querying (approximations of) the projections of top eigenvectors of M^*M onto the $\mathbb{H}_\pi$ space. For the special case when the map M is diagonal, this is reduces to querying the top eigenvectors of $\mathbb{H}_\pi$.

The excess risk will ultimately depend on the the eigenspectrum of the operator $S_\pi^{-\frac{1}{2}} M^\top S_r M S_\pi^{-\frac{1}{2}}$, which is similar to the operator M^*M. Additionally, to interpret our results, we instantiate them for a power law spectrum with exponent $\beta > 0$, that is, $\sigma_j(S_\pi^{-\frac{1}{2}} M^\top S_r M S_\pi^{-\frac{1}{2}}) \asymp j^{-\beta}$, where σ_j corresponds to the j^{th} singular value of the corresponding operator. Such power law spectra have been observed in a variety of practical settings, for instance, in the spectrum of Hessian of trained deep neural networks [90].

Warm-up: C_π = unit ball, M = diagonal

In order to get some intuition, we study the special case where the policy set C_π consists of the entire unit ball in the space $\mathbb{H}_\pi$ and the map M is diagonal with $M = \mathrm{diag}(\nu_j)$. Further, let us denote the operator $\tilde{M} = S_r^{1/2} M S_\pi^{-1/2}$.

For this special case, our sampling algorithm (Algorithm 4) simply selects the top J eigenvectors of the space $\mathbb{H}_\pi$ to query, for some value J which depends on the decay exponent β. To see why, observe that for a diagonal map M, the right singular vectors of the operator $\tilde{M}$ are the same as the eigenvectors of the policy space $\mathbb{H}_\pi$. Therefore, the choice of policy π_j in our algorithm is simply the scaled eigenfunction $\sqrt{\mu_{\pi,j}} \cdot \phi_{\pi,j}$. Having selected these J queries, the algorithm queries each one of the $\frac{n}{J}$ times and uses this as query set $\mathcal{Q}$.

The intuition for this choice of query set $\mathcal{Q}$ is that since we are in the passive setup with no knowledge of r^*, any policy $\pi \in C_\pi$ can be an optimal policy. By querying the top J ones

out of these, we can obtain a good enough approximation to the performance of any policy in the unit ball. The particular choice of the parameter J depends on the number of queries n available. Since the oracle responses are noisy, to reduce variance in the responses along those directions, our algorithm performs multiple queries along the same direction.

If we further consider the special case when the policies and rewards correspond to the unit balls in the finite dimensional spaces $\mathbb{R}^{d_\pi}$ and $\mathbb{R}^{d_r}$ respectively, our choice of query set queries the directions $\{e_i\}_{i=1}^{d_\pi}$, each for $J = \frac{n}{d_\pi}$ number of times. Intuitively, this strategy works well because without any prior over the unknown reward function, the optimal strategy in the passive setup is to explore all directions equally and this is precisely our set of chosen queries. This simple query strategy enjoys the following excess risk bound.

Proposition 6.1 (Risk bound for C_π = unit ball.)**.** *For any $J \le n$ and regularization parameter $\lambda_{\mathsf{reg}} > 0$, consider the plug-in estimator obtained via the passive sampling algorithm which explores the first J eigenfunctions of $\mathbb{H}_\pi$. The excess risk satisfies*

$$\mathbb{E}[\Delta(\hat{\pi}_{\mathsf{plug}}; r^*)] \le c \cdot \left(1 + \frac{\tau^2}{n\lambda_{\mathsf{reg}}^2}\right) \cdot \max\left\{\sup_{j \le J} \frac{\lambda_{\mathsf{reg}}^2 J^2 \zeta_j}{\zeta_j^2 + \lambda_{\mathsf{reg}}^2 J^2}, \sup_{j > J} \zeta_j\right\},$$

where the quantity $\zeta_j = \frac{\nu_j^2 \mu_{\pi,j}}{\mu_{r,j}}$ and $c > 0$ is some universal constant.

We defer the proof of the above proposition to Appendix E.2. The choice of the exploration parameter J allows us to trade-off between the two terms inside the maximum. Typically, the second term will be maximized at $j = J+1$. For the first term, the supremum depends on the choice of λ_{reg} — for small values of λ_{reg}, the sup is achieved at $j = 1$ while for larger values, it is achieved at $j = J$. In order to gain more intuition about this bound, we instantiate this for the power law decay.

Corollary 6.1 (Risk bound for power-law decay)**.** *Suppose that eigenvalues of the police space $\mathbb{H}_\pi$ decay as $j^{-\beta_\pi}$, reward space $\mathbb{H}_r$ as $j^{-\beta_r}$ and the singular values of map M as $j^{-\beta_M}$. This satisfies the power law assumption with exponent $\beta = \beta_\pi + \beta_M - \beta_r$. The plug-in estimator with exploration parameter $J = n^{\frac{1}{\beta+2}}$ and regularization $\lambda_{\mathsf{reg}} = n^{-\frac{\beta+1}{\beta+2}}$ satisfies $\mathbb{E}[\Delta(\hat{\pi}_{\mathsf{plug}}; r^*)] \le c n^{-\frac{\beta}{\beta+2}}$.*

The proof of the corollary upper bounds the risk bound with the specific choices of J and λ_{reg}. The above bound shows that our algorithm can learn in the framework as long as $\beta > 0$ or equivalently $\beta_\pi + \beta_M > \beta_r$, with better rates for larger values of β. Thus, for a fixed size of reward class β_r, the learning rate improves as the policy class grows smaller (β_π increases) – this is intuitive since we are required to search over a smaller policy space. On the other hand, for a fixed policy class β_π, our excess risk rate gets better as the reward class grows in size (β_r increases) – this is because a larger set of reward functions have similar optimal policies and hence learning gets easier.

Algorithm 4: Passive querying strategy

Input: Number of queries n, map M, policy set C_π, exploration parameter J

Construct linear map $\tilde{M} = S_r^{\frac{1}{2}} M S_\pi^{-\frac{1}{2}}$ and compute eigenvectors $\{\phi_{\tilde{M},j}\}_j$ of $\tilde{M}^\top \tilde{M}$

Set policy $\pi_j = \Phi_\pi S_\pi^{-\frac{1}{2}} \Phi_\pi^\top \phi_{\tilde{M},j}$ for all $j \leq J$

Obtain policy $\tilde{\pi}_j \in C_\pi$ such that $\tilde{\pi}_j \tilde{\pi}_j^\top \succeq c_\pi \pi_j \pi_j^\top$

Form query set $\mathcal{Q} = \{\tilde{\pi}_1^{(n/J)}, \ldots, \tilde{\pi}_{n^\alpha}^{(n/J)}\}$ where $a^{(b)} = \{a, \ldots, a\}$ repeated b times

Output: Query set $\mathcal{Q}$

General policy sets

We now describe our choice of query sets $\mathcal{Q}$ for general policy sets C_π. Our strategy, described in Algorithm 4, differs from the above special case in that we need to take into account the interaction of the policy space $\mathbb{H}_\pi$ with the map M. Specifically, we show in Appendix E.2 that the upper bound in Theorem 6.1 can be diagonalized for this general case via a transformation.

Let us denote the operator $\tilde{M} = S_r^{1/2} M S_\pi^{-1/2}$. Our transformation reveals that the relevant directions to query for this general case corresponds to the columns of $\Phi_\pi S_\pi^{-1/2} \Phi_\pi^\top V_M$ where , then V_M are the eigenvectors of the self-adjoint operator $\tilde{M}^\top \tilde{M}$ – and it is precisely a subset of these directions that our algorithm queries.

In order to be able to query these policies, we require the set C_π to contain some policies which align well with them. We formally state this regularity assumption below.

Assumption 6.1 (Regularity assumption on C_π)**.** *For any eigenfunction $\phi_{\tilde{M},j}$ of the operator $\tilde{M}^\top \tilde{M}$, consider the policy $\pi_j = \Phi_\pi S_\pi^{-1/2} \Phi_\pi^\top \phi_{\tilde{M},j}$. There exists a policy $\tilde{\pi}_j$ in policy set C_π such that for some constant $c_\pi > 0$, we have $\tilde{\pi}_j \tilde{\pi}_j^\top \succeq c_\pi \pi_j \pi_j^\top$.*

The above assumption requires that for every choice of the policy π_j in Algorithm 4, the set C_π has the another policy $\tilde{\pi}_j$ which is collinear with it. This assumption can be relaxed in various ways (for instance via convexification) but we omit this as it is not needed for our results. Given this assumption, the following theorem, a generalization of Proposition 6.1, provides a bound on the excess risk for the plug-in estimate for general policy sets C_π.

Theorem 6.2 (Risk bound for general policy sets C_π.)**.** *For any $J \leq n$, regularization parameter $\lambda_{\mathsf{reg}} > 0$ and set C_π satisfying Assumption 6.1, let $\hat{\pi}_{\mathsf{plug}}$ be the estimator output by Algorithm 3. The squared excess risk satisfies*

$$(\mathbb{E}[\Delta(\hat{\pi}_{\mathsf{plug}}; r^*)])^2 \leq c \cdot \left(1 + \frac{\tau^2}{n\lambda_{\mathsf{reg}}^2}\right) \cdot \max\left\{\sup_{j \leq J} \frac{\lambda_{\mathsf{reg}}^2 J^2 \zeta_j}{\zeta_j^2 + \lambda_{\mathsf{reg}}^2 J^2}, \sup_{j > J} \zeta_j\right\},$$

where the values ζ_j correspond to the j^{th} eigen values of the operator $\tilde{M}^* \tilde{M}$ with $\tilde{M} = S_r^{\frac{1}{2}} M S_\pi^{\frac{1}{2}}$.

We defer the proof of this theorem to Appendix E.2. The proof of this theorem goes via a transformation which diagonalizes the excess risk bound and reduces the problem to a similar setup as that of Proposition 6.1. Additionally, Assumption 6.1 allows us to generalize the results to arbitrary policy sets C_π. Note that the above upper bounds the square of the excess risk. As discussed in Section 6.3, one can obtain a quadratic improvement in this rate if the set C_π is the entire unit ball in $\mathbb{H}_\pi$. We specialize the above bound for the power law decay assumption in the following corollary.

Corollary 6.2 (Risk bound for power-law decay)**.** *Suppose that eigenspectrum of the operator $S_\pi^{-\frac{1}{2}} M^\top S_r M S_\pi^{-\frac{1}{2}}$ satisfy the power law assumption with exponent $\beta > 0$, that is, $\sigma_j(S_\pi^{-\frac{1}{2}} M^\top S_r M S_\pi^{-\frac{1}{2}}) \asymp j^{-\beta}$. The plug-in estimator $\hat{\pi}_{\mathsf{plug}}$ with parameter $J = n^{\frac{1}{\beta+2}}$ and regularization $\lambda_{\mathsf{reg}} = n^{-\frac{\beta+1}{\beta+2}}$ satisfies $\mathbb{E}[\Delta(\hat{\pi}_{\mathsf{plug}}; r^*)] \leq c n^{-\frac{\beta}{2(\beta+2)}}$ for some universal constant $c > 0$.*

The above bound indicates that for the general case, learning is possible if the spectrum decay has parameter $\beta > 0$. To get such a spectrum decay with the operator defined in the above corollary, one sufficient condition is that the map M does not flip the larger eigenvectors of $\mathbb{H}_\pi$ towards the smaller eigenvectors of $\mathbb{H}_r$, that is, the map M preserves the ordering of the eigenvectors of $\mathbb{H}_\pi$ when transformed to the space $\mathbb{H}_r$. Such a misaligned scenario would require learning a very accurate representation of the reward to learn a good policy and will make learning harder. It is worth highlighting that while we discuss our bounds with such a power law assumption on the relevant eigenvalues, one can also obtain similar rates for singular values with exponential decay, by optimizing the value of J to trade off the bias and variance terms.

Comparison with UCB-style adaptive algorithms

We next turn to evaluating the sharpness of Theorem 6.2. Existing frameworks for studying "singly"-nonparametric setups require the input domain to be compact. In our doubly-nonparametric setup, the input space is the policy set C_π which is often non-compact (i.e. the unit ball is not compact in infinite dimensions). We address this for singly-nonparametrics algorithm by taking a finite-dimensional approximation.

Even though our proposed method is passive, it achieves better rates than well-known *adaptive* sampling algorithms. Specifically, in the power law setting of Section 6.4, the analysis of GP-UCB algorithm [185] provides a rate of $O(n^{-\frac{\beta-1}{2(\beta+1)}})$, which is strictly worse than the $O(n^{-\frac{\beta}{\beta+1}})$ obtained by our analysis in Corollary 6.1. We refer the reader to Proposition E.1 in Appendix E.4 for an exact statement. The proof adapts the analysis from [185], which hinges on a quantity called the information gain, which we bound for our setup. While we are comparing upper bounds for the two algorithms, we believe that our improved bound is due to a better algorithm and not an analysis gap. While we expect adaptive algorithms to perform better than passive ones in general [129], UCB style algorithms require the construction

Algorithm	Regret $\mathfrak{R}_T$	Non-vacuous regime
GP-UCB [185], GP-TS [55]	$\tilde{O}(T^{\frac{2\nu+d(3d+3)}{4\nu+d(2d+2)}})$	$\nu > \frac{d^2+d}{2}$
Our work	$\tilde{O}(T^{\frac{4\nu+d(4d+6)}{6\nu+d(4d+7)}})$	$\nu > \frac{3}{2}$
π-GP-UCB [112]	$\tilde{O}(T^{\frac{2\nu+d(2d+3)}{4\nu+d(2d+4)}})$	$\nu > 1$
SupKernelUCB [201]	$\tilde{O}(T^{\frac{\nu+d}{2\nu+d}})$	$\nu > 1$

Table 6.1. Our algorithm specializes to the case of kernel multi-armed bandits and yields strong bounds. For a d-dimensional Matérn kernel with smoothness ν, we outperform both GP-UCB and GP-TS unless $\nu \gtrsim d^2$. The only works to achieve better bounds for small ν are π-GP UCB, which was designed specifically for the Matérn kernel and a recent analysis of the SupKernelUCB which achieves near minimax rates.

of confidence intervals around input points, which crucially dictate the regret bounds of such algorithms. In the frequentist setup, the best known such bounds [201] are known to yield suboptimal regret rates and it is an open question as to whether these can be improved.

6.5 Bounds for kernel multi-armed bandits

In the previous subsection, we saw that our passive sampling algorithm actually outperforms existing adaptive sampling algorithms for the reward learning task we care about. Here we take this a step further—we specialize our algorithm to the case of kernel MABs, and show that it outperforms standard algorithms for that setting and is competitive with a specialized algorithm for Matérn kernels.

We consider the task of maximizing an unknown function $f^* : \mathcal{X} \mapsto \mathbb{R}$ over its domain $\mathcal{X} \subset \mathbb{R}^d$. In the kernel multi-armed bandit (MAB) setup, this unknown function f belongs to an RKHS $\mathbb{H}$, equipped with a positive-definite kernel[4] $\mathcal{K}$, such that $\|f^*\|_{\mathbb{H}} = 1$. Let us further restrict our attention to the space of input points $\mathcal{X} = \{x \in \mathbb{R}^d \mid \|x\|_2 \leq 1\}$. The learner is allowed to access this function via a noisy zeroth-order oracle

$$\mathcal{O}_{f^*} : x \mapsto f^*(x) + \eta \,\text{where}\, \eta \sim \mathcal{N}(0, \tau^2) \,. \tag{6.8}$$

Going forward we will assume that $\tau = 1$. The above oracle is similar to the reward oracle $\mathcal{O}_{r^*}$, except that the query points x belong to a finite dimensional space and f^* is a non-linear function of the query point x. The goal in MAB is to minimize the T-step regret $\mathfrak{R}_T := T \cdot \max_{x \in \mathcal{X}} f^*(x) - \sum_{t=1}^{T} f^*(x_t)$, where x_t is the datapoint queried in the t^{th} round. There have been several algorithms proposed to solve this problem including general purpose UCB

[4] We require that the kernel $\mathcal{K}$ be a Mercer's kernel satisfying $\mathcal{K}(x, x) = c$ for all $x \in \mathcal{X}$.

algorithms [185, 55], Thompson sampling approaches [55], and special-purpose algorithms for specific kernels [112].

We next show that kernel MAB can be cast as a special case of our non-parametric policy learning framework. The resulting regret bounds, derived from an application of Theorem 6.3, are better than several general purpose algorithms (GP-UCB, IGP-UCB, GP-TS) and comparable to those specialized for the Matérn kernel (π-GP-UCB) and SupKernelUCB.

In order to reduce kernel MAB to our framework, we need to introduce three elements – the policy space $\mathbb{H}_\pi$, the reward space $\mathbb{H}_r$ and the map M. We would like spaces $\mathbb{H}_r$ and $\mathbb{H}_\pi$ such that (1) the resulting objective $F(r, \pi)$ is linear in this space, (2) the resulting rewards and policies have unit norm in their respective space, and (3) we have a good understanding of the eigenvalues of the resulting operator. This last point ensures that we can employ our upper bounds from Section 6.4.

Before we define these, we let $\mathcal{C}_\epsilon$ denote an ϵ-net of the input space $\mathcal{X}$ under the ℓ_2 norm and denote its size by $N_{\mathsf{cov}}(\epsilon)$. We define the kernel matrix $K \in \mathbb{R}^{N_{\mathsf{cov}} \times N_{\mathsf{cov}}}$ on points selected in the cover as $K(i, j) = \mathcal{K}(x_i, x_j)$ for all $(x_i, x_j) \in \mathcal{C}_\epsilon \times \mathcal{C}_\epsilon$.

Reward space $\mathbb{H}_r$. Given the RKHS $\mathbb{H}$ as well as the elements of the cover $\mathcal{C}_\epsilon$, we view the reward function as a map from $\mathcal{C}_\epsilon$ to $\mathbb{R}$, or equivalently as a vector in $\mathbb{R}^{N_{\mathsf{cov}}(\epsilon)}$. More precisely, letting $\tilde{f} = [f(x_1), \ldots, f(x_{N_{\mathsf{cov}}})]$ denote the vector of evaluations of a function f, we define $\mathbb{H}_r := \mathrm{span}\{\tilde{f} \mid f \in \mathbb{H}\}$ with $\langle \tilde{f}_1, \tilde{f}_2 \rangle_{\mathbb{H}_r} := \tilde{f}_1^\top K^{-1} \tilde{f}_2$. With this notation, we define the true reward $r^* := \tilde{f}^* = [f^*(x_1), \ldots, f^*(x_{N_{\mathsf{cov}}})]$.

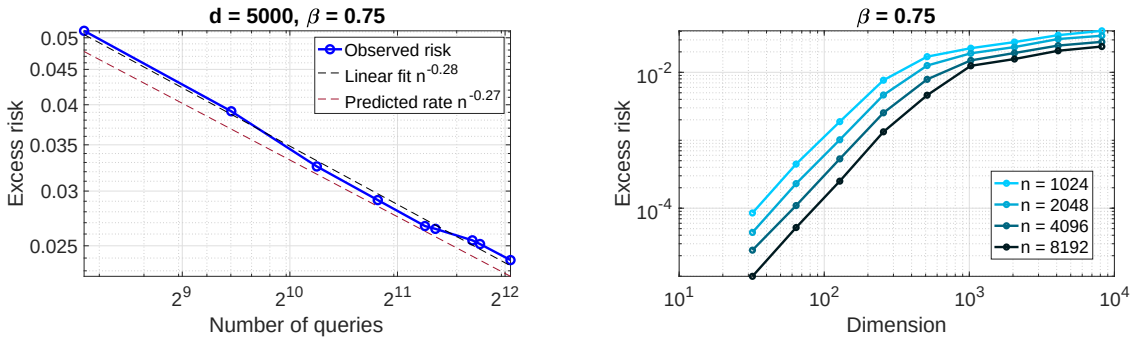
Policy Space $\mathbb{H}_\pi$. Similarly to rewards, we will embed policies in $\mathbb{R}^{N_{\mathsf{cov}}}$. For any point $x \in \mathcal{C}_\epsilon$, let $k_x = [\mathcal{K}(x, x_1), \ldots, \mathcal{K}(x, x_{N_{\mathsf{cov}}})]$ denote the corresponding vector in $\mathbb{R}^{N_{\mathsf{cov}}}$ obtained by evaluating the kernel $\mathcal{K}$ over the cover. Then, the space $\mathbb{H}_\pi := \mathrm{span}\{k_x \mid x \in \mathcal{C}_\epsilon\}$ with $\langle k_1, k_2 \rangle_{\mathbb{H}_\pi} := \langle k_1, K^{-2} k_2 \rangle$. The choice of the above norm ensures that $\langle k_i, k_j \rangle_{\mathbb{H}_\pi} = \langle K^{-1} k_i, K^{-1} k_j \rangle = \delta_{i,j}$ for all $(x_i, x_j) \in \mathcal{C}_\epsilon \times \mathcal{C}_\epsilon$. Thus in particular, $\mathbb{H}_\pi$ contains an orthonormal embedding of the set of vectors $\{k_x\}_{x \in \mathcal{C}_\epsilon}$.

Map M. Both the reward space $\mathbb{H}_r$ and policy space $\mathbb{H}_\pi$ can be associated with $\mathbb{R}^{N_{\mathsf{cov}}}$. Under this transformation, the evaluation $f^*(x)$ for any $x \in \mathcal{C}_\epsilon$ corresponds to the standard inner product with $F(r^*, \pi_x) = f^*(x) = (\tilde{f}^*)^\top K^{-1} k_x = \langle r^*, k_x \rangle_{\mathbb{H}_r}$. This indicates that we should take the map M to be the identity. Furthermore, as a simple application of Mercer's theorem it follows that this map M is a bounded linear operator.

We make an additional assumption on the kernel function $\mathcal{K}$, requiring it to be Lipschitz in its input arguments. This assumption is often satisfied, in particular for the Matérn kernel when $\nu > 3/2$.

Assumption 6.2 (Lipschitz Kernel $\mathcal{K}$)**.** *The Kernel $\mathcal{K}$ associated with the Hilbert space $\mathbb{H}$ is $L_\mathcal{K}$-Lipschitz with respect to the ℓ_2-norm for some $L_\mathcal{K} > 0$: $|\mathcal{K}(x, y) - \mathcal{K}(x, x)| \leq L_\mathcal{K} \|x - y\|_2$ for all $x \in \mathcal{X}, y \in \mathcal{X}$. Furthermore, the kernel satisfies $\mathcal{K}(x, x) = 1$ for all points $x \in \mathcal{X}$.*

Applying Theorem 6.2 under the above assumption, we obtain the following excess risk bound for the plug-in estimator evaluated on the unknown function f^*.



(a) (b)

Figure 6.1. (a) Corroborating upper bound from Corollary 6.1. Our theoretical bounds predict a rate of $n^{-0.27}$ and the experiment shows an almost matching rate of $n^{-0.28}$. (b) As the dimension d is increased, the excess risk curves asymptote at different levels for different n. This shows that our algorithm achieves non-vacuous error for the doubly-nonparametric set in the regime $d \to \infty$.

Theorem 6.3 (Excess risk for Kernel MAB)**.** *Suppose that the eigenvalues of a $L_{\mathcal{K}}$-Lipschitz kernel $\mathcal{K}$ satisfy the power-law decay $\mu_j \asymp j^{-\beta}$. Let $\hat{x}_{\mathsf{plug}}$ be the output of Algorithm 3 using n queries to the oracle $\mathcal{O}_{f^*}$. Then, for any value of $\beta > 1 + \frac{2}{d} + \log(\frac{1}{\delta})$ and $\epsilon \in (0,1)$, the excess risk satisfies*

$$\max_{x:\|x\|_2 \le 1} f^*(x) - f^*(\hat{x}_{\mathsf{plug}}) \lesssim N_{\mathsf{cov}}^{\frac{1}{\beta+2}}(\epsilon) \cdot n^{\frac{-\beta}{2(\beta+2)}} + N_{\mathsf{cov}}^{\frac{1-\beta}{2}}(\epsilon) + \sqrt{L_{\mathcal{K}}\epsilon}\,,$$

with probability at least $1-\delta$.

For Matérn kernels, it is known that the eigenvalues decay with parameter $\beta = 1 + \frac{2\nu}{d}$ [112]. Using this, we can obtain the following corollary.

Corollary 6.3 (Regret bound for Matérn Kernel)**.** *Consider the family of Matérn kernels with parameter $\nu > \frac{3}{2}$ defined with the Euclidean norm over $\mathbb{R}^d$. The T-step regret of our algorithm is $\mathfrak{R}_{\mathsf{mat},T} = \tilde{O}\left(T^{\frac{4\nu+d(6+4d)}{6\nu+d(7+4d)}}\right)$.*

The above bound is for regret, which is an online notion, while our previous results are offline notions. We get from one to the other using a standard batch-to-online conversion bound based on an explore-then-commit strategy. Table 6.1 compares the above bound to the existing literature.

Bibliography

[1] Pieter Abbeel and Andrew Y Ng. "Apprenticeship learning via inverse reinforcement learning". In: *Proceedings of the twenty-first international conference on Machine learning*. 2004.

[2] Jacob Abernethy, Peter L Bartlett, and Elad Hazan. "Blackwell approachability and no-regret learning are equivalent". In: *Proceedings of the Conference on Learning Theory*. 2011.

[3] Sydney N Afriat. "The construction of utility functions from expenditure data". In: *International economic review* 8.1 (1967).

[4] Alekh Agarwal, Ofer Dekel, and Lin Xiao. "Optimal algorithms for online convex optimization with multi-point bandit feedback." In: *Proceedings of the Conference on Learning Theory*. 2010.

[5] Michele Aghassi and Dimitris Bertsimas. "Robust game theory". In: *Mathematical Programming* 107 (2006), pp. 231–273.

[6] Nir Ailon, Zohar Karnin, and Thorsten Joachims. "Reducing dueling bandits to cardinal bandits". In: *Proceedings of the International Conference on Machine Learning*. 2014.

[7] Alexey Rostislavovich Alimov and Igor'Germanovich Tsar'kov. "Connectedness and other geometric properties of suns and Chebyshev sets". In: *Fundamentalnaya i Prikladnaya Matematika* 19.4 (2014), pp. 21–91.

[8] Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. "Power to the people: The role of humans in interactive machine learning". In: *AI Magazine* 35.4 (2014), pp. 105–120.

[9] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. "Concrete problems in AI safety". In: *arXiv preprint arXiv:1606.06565* (2016).

[10] Philip W Anderson. "More is different". In: *Science* 177.4047 (1972), pp. 393–396.

[11] Marcin Andrychowicz, Anton Raichuk, Piotr Stańczyk, Manu Orsini, Sertan Girgin, Raphael Marinier, Léonard Hussenot, Matthieu Geist, Olivier Pietquin, and Marcin Michalski. "What matters in on-policy reinforcement learning? A large-scale empirical study". In: *arXiv preprint arXiv:2006.05990* (2020).

[12] Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. "A survey of robot learning from demonstration". In: *Robotics and autonomous systems* 57.5 (2009).

[13] Kenneth Joseph Arrow. *Social Choice and Individual Values*. Wiley, 1951.

[14] Susan Athey and Stefan Wager. "Efficient policy learning". In: *arXiv preprint arXiv:1702.02896* (2017).

[15] Anthony C Atkinson. "The usefulness of optimum experimental designs". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 58 (1996).

[16] Bowen Baker, Ingmar Kanitscheider, Todor Markov, Yi Wu, Glenn Powell, Bob McGrew, and Igor Mordatch. "Emergent Tool Use From Multi-Agent Autocurricula". In: *International Conference on Learning Representations*. 2020.

[17] Maria-Florina Balcan, Andrei Broder, and Tong Zhang. "Margin based active learning". In: *International Conference on Computational Learning Theory*. Springer. 2007, pp. 35–50.

[18] Maria-Florina Balcan, Amit Daniely, Ruta Mehta, Ruth Urner, and Vijay V Vazirani. "Learning economic parameters from revealed preferences". In: *International Conference on Web and Internet Economics*. 2014.

[19] Maria-Florina Balcan and Phil Long. "Active and passive learning of linear separators under log-concave distributions". In: *Conference on Learning Theory*. 2013, pp. 288–316.

[20] Vitor Balestro, Horst Martini, and Ralph Teixeira. "Convex analysis in normed spaces and metric projections onto convex bodies". In: *arXiv preprint arXiv:1908.08742* (2019).

[21] Peter L Bartlett and Shahar Mendelson. "Rademacher and Gaussian complexities: Risk bounds and structural results". In: *Journal of Machine Learning Research* 3.Nov (2002), pp. 463–482.

[22] Eyal Beigman and Rakesh Vohra. "Learning from revealed preference". In: *Proceedings of the 7th ACM Conference on Electronic Commerce*. 2006.

[23] Kush Bhatia, Peter L Bartlett, Anca D Dragan, and Jacob Steinhardt. "Agnostic learning with unknown utilities". In: *arXiv preprint arXiv:2104.08482* (2021).

[24] Kush Bhatia, Wenshuo Guo, and Jacob Steinhardt. "Reward Learning as Doubly Nonparametric Bandits: Optimal Design and Scaling Laws". In: (2022).

[25] Kush Bhatia, Ashwin Pananjady, Peter Bartlett, Anca Dragan, and Martin J Wainwright. "Preference learning along multiple criteria: A game-theoretic perspective". In: *Advances in neural information processing systems* 33 (2020), pp. 7413–7424.

[26] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[27] Erdem Bıyık, Nicolas Huynh, Mykel J Kochenderfer, and Dorsa Sadigh. "Active preference-based gaussian process regression for reward learning". In: *arXiv preprint arXiv:2005.02575* (2020).

[28] Duncan Black. "On the rationale of group decision-making". In: *Journal of Political Economy* 56.1 (1948), pp. 23–34.

[29] David Blackwell. "An analog of the minimax theorem for vector payoffs." In: *Pacific Journal of Mathematics* 6.1 (1956), pp. 1–8.

[30] Peter Boettke and Benjamin Powell. "The political economy of the COVID-19 pandemic". In: *Southern Economic Journal* 87.4 (2021), pp. 1090–1106.

[31] Florian Böhm, Yang Gao, Christian M Meyer, Ori Shapira, Ido Dagan, and Iryna Gurevych. "Better rewards yield better summaries: Learning to summarise without references". In: *arXiv preprint arXiv:1909.01214* (2019).

[32] Rishi Bommasani et al. "On the Opportunities and Risks of Foundation Models". In: *arXiv preprint arXiv:2108.07258* (2021). arXiv: 2108.07258.

[33] JC de Borda. "Mémoire sur les élections au scrutin". In: *Histoire de l'Academie Royale des Sciences pour 1781* (1784).

[34] Ralph Allan Bradley and Milton E Terry. "Rank analysis of incomplete block designs: I. The method of paired comparisons". In: *Biometrika* 39.3/4 (1952), pp. 324–345.

[35] Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D Procaccia. *Handbook of Computational Social Choice*. Cambridge University Press, 2016.

[36] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. *OpenAI Gym*. 2016. eprint: arXiv:1606.01540.

[37] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. *OpenAI Gym*. 2016. eprint: arXiv:1606.01540.

[38] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. "Openai gym". In: *arXiv preprint arXiv:1606.01540* (2016).

[39] Amy Bronstone and Claudia Graham. "The Potential Cost Implications of Averting Severe Hypoglycemic Events Requiring Hospitalization in High-Risk Adults With Type 1 Diabetes Using Real-Time Continuous Glucose Monitoring". In: *Journal of Diabetes Science and Technology* 10 (2016).

[40] Daniel Brown, Russell Coleman, Ravi Srinivasan, and Scott Niekum. "Safe Imitation Learning via Fast Bayesian Reward Inference from Preferences". In: *Proceedings of the 37th International Conference on Machine Learning*. 2020.

[41] Sébastien Bubeck. "Convex optimization: Algorithms and complexity". In: *Foundations and Trends® in Machine Learning* 8 (2015).

[42] Sébastien Bubeck, Rémi Munos, Gilles Stoltz, and Csaba Szepesvári. "X-Armed Bandits." In: *Journal of Machine Learning Research* 12.5 (2011).

[43] Tom Bylander. "Learning linear threshold functions in the presence of classification noise". In: *Proceedings of the seventh annual conference on Computational learning theory*. 1994.

[44] Xu Cai and Jonathan Scarlett. "On lower bounds for standard and robust Gaussian process bandit optimization". In: *International Conference on Machine Learning*. PMLR. 2021.

[45] Haoyang Cao, Samuel N. Cohen, and Lukasz Szpruch. "Identifiability in inverse reinforcement learning". In: *CoRR* abs/2106.03498 (2021).

[46] Alfonso Caramazza, Michael McCloskey, and Bert Green. "Naive beliefs in "sophisticated" subjects: Misconceptions about trajectories of objects". In: *Cognition* 9 (1981).

[47] Kathryn Chaloner and Isabella Verdinelli. "Bayesian experimental design: A review". In: *Statistical Science* (1995), pp. 273–304.

[48] Lawrence Chan, Andrew Critch, and Anca Dragan. "Human irrationality: both bad and good for reward inference". In: *preprint arXiv:2111.06956* (2021).

[49] Olivier Chapelle, Thorsten Joachims, Filip Radlinski, and Yisong Yue. "Large-scale validation and analysis of interleaved search evaluation". In: *ACM Transactions on Information Systems (TOIS)* 30.1 (2012), pp. 1–41.

[50] Xiaohui Chen and Yun Yang. "Hanson–Wright inequality in Hilbert spaces with application to K-means clustering for non-Euclidean data". In: *Bernoulli* 27.1 (2021), pp. 586–614.

[51] Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins. *Double/debiased machine learning for treatment and structural parameters*. 2018.

[52] Victor Chernozhukov, Matt Goldman, Vira Semenova, and Matt Taddy. "Orthogonal machine learning for demand estimation: High dimensional causal inference in dynamic panels". In: *arXiv preprint arXiv:1712.09988* (2017).

[53] Victor Chernozhukov, Denis Nekipelov, Vira Semenova, and Vasilis Syrgkanis. "Plug-in regularized estimation of high-dimensional parameters in nonlinear semiparametric models". In: *arXiv preprint arXiv:1806.04823* (2018).

[54] Victor Chernozhukov, Whitney K Newey, and James Robins. *Double/de-biased machine learning using regularized Riesz representers*. Tech. rep. cemmap working paper, 2018.

[55] Sayak Ray Chowdhury and Aditya Gopalan. "On kernelized multi-armed bandits". In: *International Conference on Machine Learning*. 2017.

[56] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. "Deep Reinforcement Learning from Human Preferences". In: *Advances in Neural Information Processing Systems*. 2017.

[57] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. "Deep reinforcement learning from human preferences". In: *Advances in Neural Information Processing Systems*. 2017.

[58] Marquis de Condorcet. "Essai sur l'application de l'analyse a la probabilité des decisions rendues a la pluralité des voix". In: (1785).

[59] Arthur H Copeland. *A reasonable social welfare function*. Tech. rep. mimeo, University of Michigan, 1951.

[60] Christian Daniel, Malte Viering, Jan Metz, Oliver Kroemer, and Jan Peters. "Active Reward Learning." In: *Robotics: Science and systems*. Vol. 98. 2014.

[61] Sanjoy Dasgupta, Adam Tauman Kalai, and Adam Tauman. "Analysis of Perceptron-Based Active Learning." In: *Journal of Machine Learning Research* 10.2 (2009).

[62] Jesse Davis and Mark Goadrich. "The Relationship between Precision-Recall and ROC Curves". In: *International Conference on Machine Learning*. 2006.

[63] Luc Devroye, László Györfi, and Gábor Lugosi. *A probabilistic theory of pattern recognition*. Vol. 31. Springer Science & Business Media, 2013.

[64] Daniel Dewey. "Reinforcement learning and the reward engineering principle". In: *2014 AAAI Spring Symposium Series*. 2014.

[65] Amy M Do, Alexander V Rupert, and George Wolford. "Evaluations of pleasurable experiences: The peak-end rule". In: *Psychonomic Bulletin & Review* (2008).

[66] Karen M Douglas and Robert J Mislevy. "Estimating classification accuracy for complex decision rules based on multiple scores". In: *Journal of Educational and Behavioral Statistics* 35.3 (2010), pp. 280–306.

[67] Michael Doumpos and Constantin Zopounidis. "Regularized estimation for preference disaggregation in multiple criteria decision making". In: *Computational Optimization and Applications* 38.1 (2007), pp. 61–80.

[68] Jasha Droppo and Oguz Elibol. "Scaling Laws for Acoustic Models". In: *arXiv preprint arXiv:2106.09488* (2021).

[69] John C Duchi, Michael I Jordan, Martin J Wainwright, and Andre Wibisono. "Optimal rates for zero-order convex optimization: The power of two function evaluations". In: *IEEE Transactions on Information Theory* 61.5 (2015).

[70] Richard O Duda, Peter E Hart, and David G Stork. *Pattern classification*. John Wiley & Sons, 2012.

[71] Miroslav Dudík, Katja Hofmann, Robert E Schapire, Aleksandrs Slivkins, and Masrour Zoghi. "Contextual Dueling Bandits". In: *Proceedings of the Conference on Learning Theory*. 2015.

[72] Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Volodymyr Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Robert Dunning, Shane Legg, and Koray Kavukcuoglu. "IMPALA: Scalable Distributed Deep-RL with Importance Weighted Actor-Learner Architectures". In: 2018.

[73] Owain Evans, Andreas Stuhlmueller, and Noah D. Goodman. "Learning the Preferences of Ignorant, Inconsistent Agents". In: *AAAI Conference on Artificial intelligence*. 2016.

[74] Tom Everitt, Victoria Krakovna, Laurent Orseau, and Shane Legg. "Reinforcement Learning with a Corrupted Reward Channel". In: *International Joint Conference on Artificial Intelligence*. 2017.

[75] Chelsea Finn, Sergey Levine, and Pieter Abbeel. "Guided Cost Learning: Deep Inverse Optimal Control via Policy Optimization". In: *International Conference on Machine Learning (ICML)*. 2016.

[76] Abraham D Flaxman, Adam Tauman Kalai, and H Brendan McMahan. "Online convex optimization in the bandit setting: gradient descent without a gradient". In: *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*. 2005.

[77] Dylan J Foster and Vasilis Syrgkanis. "Orthogonal statistical learning". In: *arXiv preprint arXiv:1901.09036* (2019).

[78] Ian Fox, Joyce Lee, Rodica Pop-Busui, and Jenna Wiens. "Deep Reinforcement Learning for Closed-Loop Blood Glucose Control". In: *Machine Learning for Healthcare Conference*. 2020.

[79] M. Fralick and A. S. Kesselheim. "The U.S. Insulin Crisis - Rationing a Lifesaving Medication Discovered in the 1920s". In: *New England Journal of Medicine* 381.19 (2019), pp. 1793–1795.

[80] Pedro Freire, Adam Gleave, Sam Toyer, and Stuart Russell. In: *Proceedings of the Workshop on Deep Reinforcement Learning at NeurIPS*. 2020.

[81] Jerome Friedman, Trevor Hastie, Robert Tibshirani, et al. *The elements of statistical learning*. Vol. 1. 10. Springer series in statistics New York, 2001.

[82] Deborah Frisch and Robert T Clemen. "Beyond expected utility: rethinking behavioral decision research." In: *Psychological Bulletin* 116.1 (1994), p. 46.

[83] J. Fu, K. Luo, and S. Levine. "Learning Robust Rewards with Adversarial Inverse Reinforcement Learning". In: *International Conference on Learning Representations (ICLR)*. 2018.

[84] Justin Fu, Aviral Kumar, Matthew Soh, and Sergey Levine. "Diagnosing Bottlenecks in Deep Q-learning Algorithms". In: *Proceedings of the 36th International Conference on Machine Learning*. PMLR, 2019.

[85] Justin Fu, Aviral Kumar, Matthew Soh, and Sergey Levine. "Diagnosing bottlenecks in deep Q-learning algorithms". In: *arXiv preprint arXiv:1902.10250* (2019).

[86] Drew Fudenberg and David K Levine. "Self-confirming equilibrium". In: *Econometrica: Journal of the Econometric Society* (1993), pp. 523–545.

[87] Johannes Fürnkranz and Eyke Hüllermeier. "Preference learning and ranking by pairwise comparison". In: *Preference learning*. 2010.

[88] Tsogbadral Galaabaatar and Edi Karni. "Subjective expected utility with incomplete preferences". In: *Econometrica* 81.1 (2013).

[89] Saeed Ghadimi and Guanghui Lan. "Stochastic first-and zeroth-order methods for nonconvex stochastic programming". In: *SIAM Journal on Optimization* 23.4 (2013).

[90] Behrooz Ghorbani, Shankar Krishnan, and Ying Xiao. "An investigation into neural net optimization via hessian eigenvalue density". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 2232–2241.

[91] Adam Gleave and Sam Toyer. "A Primer on Maximum Causal Entropy Inverse Reinforcement Learning". In: *preprint arXiv:2203.11409* (2022).

[92] William M Goldstein and Jane Beattie. "Judgments of relative importance in decision making: The importance of interpretation and the interpretation of importance". In: *Frontiers of Mathematical Psychology*. 1991, pp. 110–137.

[93] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples". In: *arXiv preprint arXiv:1412.6572* (2014).

[94] Bryan S Graham. "Efficiency bounds for missing data models with semiparametric restrictions". In: *Econometrica* 79.2 (2011).

[95] Till Grüne-Yanoff. "Models of temporal discounting 1937–2000: An interdisciplinary exchange between economics and psychology". In: *Science in context* (2015).

[96] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor". In: *arXiv*. 2018. URL: https://arxiv.org/pdf/1801.01290.pdf.

[97] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor". In: *International conference on machine learning*. 2018.

[98] Dylan Hadfield-Menell, Smitha Milli, Pieter Abbeel, Stuart Russell, and Anca Dragan. "Inverse reward design". In: *arXiv preprint arXiv:1711.02827* (2017).

[99] Dylan Hadfield-Menell, Smitha Milli, Pieter Abbeel, Stuart J Russell, and Anca Dragan. "Inverse Reward Design". In: *Advances in Neural Information Processing Systems*. 2017.

[100] David Haussler. "Decision theoretic generalizations of the PAC model for neural net and other learning applications". In: *Information and computation* 100.1 (1992).

[101] Dan Hendrycks, Nicholas Carlini, John Schulman, and Jacob Steinhardt. "Unsolved Problems in ML Safety". In: *arXiv preprint arXiv:2109.13916* (2021).

[102] Dan Hendrycks and Kevin Gimpel. "A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks". In: *International Conference on Learning Representations* (2017).

[103] Dan Hendrycks, Mantas Mazeika, Andy Zou, Sahil Patel, Christine Zhu, Jesus Navarro, Dawn Song, Bo Li, and Jacob Steinhardt. "What Would Jiminy Cricket Do? Towards Agents That Behave Morally". In: (2021).

[104] Darby Herkert, Pavithra Vijayakumar, Jing Luo, Jeremy I. Schwartz, Tracy L. Rabin, Eunice DeFilippo, and Kasia J. Lipska. "Cost-Related Insulin Underuse Among Patients With Diabetes". In: *JAMA Internal Medicine* 179.1 (2019), pp. 112–114.

[105] Danny Hernandez, Jared Kaplan, Tom Henighan, and Sam McCandlish. "Scaling laws for transfer". In: *arXiv preprint arXiv:2102.01293* (2021).

[106] Katja Hofmann, Shimon Whiteson, and Maarten De Rijke. "A probabilistic method for inferring preferences from clicks". In: *Proceedings of the International Conference on Information and Knowledge Management*. 2011.

[107] Thomas Hofmann, Bernhard Schölkopf, and Alexander J Smola. "Kernel methods in machine learning". In: *The annals of statistics* 36.3 (2008), pp. 1171–1220.

[108] Joey Hong, Kush Bhatia, and Anca Dragan. "On the Sensitivity of Reward Inference to Misspecified Human Models". In: (2022).

[109] Evan Hubinger, Chris van Merwijk, Vladimir Mikulik, Joar Skalse, and Scott Garrabrant. "Risks from learned optimization in advanced machine learning systems". In: *arXiv preprint arXiv:1906.01820* (2019).

[110] Borja Ibarz, J. Leike, Tobias Pohlen, Geoffrey Irving, S. Legg, and Dario Amodei. "Reward learning from human preferences and demonstrations in Atari". In: *Advances in Neural Information Processing Systems*. 2018.

[111] Kevin Jamieson, Sumeet Katariya, Atul Deshpande, and Robert Nowak. "Sparse dueling bandits". In: *Proceedings of the International Conference on Artificial Intelligence and Statistics*. 2015.

[112] David Janz, David Burt, and Javier González. "Bandit optimisation of functions in the Matérn kernel RKHS". In: *International Conference on Artificial Intelligence and Statistics*. 2020.

[113] Zaynah Javed, Daniel S Brown, Satvik Sharma, Jerry Zhu, Ashwin Balakrishna, Marek Petrik, Anca Dragan, and Ken Goldberg. "Policy Gradient Bayesian Robust Optimization for Imitation Learning". In: *Proceedings of the 38th International Conference on Machine Learning*. 2021.

[114] Daniel Kahneman and Amos Tversky. "Prospect theory: An analysis of decision under risk". In: *Handbook of the fundamentals of financial decision making: Part I* (2013).

[115] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. "Scaling laws for neural language models". In: *arXiv preprint arXiv:2001.08361* (2020).

[116] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. "Scaling laws for neural language models". In: *arXiv preprint arXiv:2001.08361* (2020).

[117] Roni Khardon and Gabriel Wachman. "Noise tolerant variants of the perceptron algorithm". In: *Journal of Machine Learning Research* 8 (2007).

[118] Kuno Kim, Shivam Garg, Kirankumar Shiragur, and Stefano Ermon. "Reward Identification in Inverse Reinforcement Learning". In: *Proceedings of the 38th International Conference on Machine Learning*. 2021.

[119] W. Bradley Knox, Alessandro Allievi, Holger Banzhaf, Felix Schmitt, and Peter Stone. "Reward (Mis)design for Autonomous Driving". In: *arXiv e-prints arXiv:2104.13906* (2021).

[120] Jens Kober, J Andrew Bagnell, and Jan Peters. "Reinforcement learning in robotics: A survey". In: *The International Journal of Robotics Research* 32.11 (2013), pp. 1238–1274.

[121] Vladimir Koltchinskii and Karim Lounici. "Concentration inequalities and moment bounds for sample covariance operators". In: *Bernoulli* 23 (2017).

[122] Varun Kompella, Roberto Capobianco, Stacy Jong, Jonathan Browne, Spencer Fox, Lauren Meyers, Peter Wurman, and Peter Stone. *Reinforcement Learning for Optimization of COVID-19 Mitigation policies*. 2020. arXiv: 2010.10560 [cs.LG].

[123] Michael R Kosorok. *Introduction to empirical processes and semiparametric inference*. 2007.

[124] BorIs. P. Kovatchev, Martin Straume, Daniel J. Cox, and Leon.S Farhy. "Risk analysis of blood glucose data:A quantitative approach to optimizing the control of insulin dependent diabetes". In: *Journal of Theoretical Medicine* 3.1 (2000), pp. 1–10.

[125] Erwin Kreyszig. *Introductory functional analysis with applications*. Vol. 1. wiley New York, 1978.

[126] Markus Kuderer, Shilpa Gulati, and Wolfram Burgard. "Learning driving styles for autonomous vehicles from demonstration". In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2015, pp. 2641–2646.

[127] Volodymyr Kuleshov and Stefano Ermon. "Estimating uncertainty online against an adversary". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 2017.

[128] Heinrich Küttler, Nantas Nardelli, Thibaut Lavril, Marco Selvatici, Viswanath Sivakumar, Tim Rocktäschel, and Edward Grefenstette. "TorchBeast: A PyTorch Platform for Distributed RL". In: *arXiv preprint arXiv:1910.03552* (2019).

[129] Tor Lattimore and Botao Hao. "Bandit Phase Retrieval". In: *arXiv preprint arXiv:2106.01660* (2021).

[130] Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

[131] Benita Lee. *How Much Does Insulin Cost? Here's How 23 Brands Compare*. 2020.

[132] Ehud Lehrer. "Partially specified probabilities: decisions and games". In: *American Economic Journal: Microeconomics* 4.1 (2012), pp. 70–100.

[133] Jan Leike, Miljan Martic, Victoria Krakovna, Pedro A. Ortega, Tom Everitt, Andrew Lefrancq, Laurent Orseau, and Shane Legg. *AI Safety Gridworlds*. 2017. arXiv: 1711.09883 [cs.LG].

[134] Michael L. Littman, Ifeoma Ajunwa, Guy Berger, Craig Boutilier, Morgan Currie, Finale Doshi-Velez, Gillian Hadfield, Michael C. Horowitz, Charles Isbell, Hiroaki Kitano, Karen Levy, Terah Lyons, Melanie Mitchell, Julie Shah, Steven Sloman, Shannon Vallor, and Toby Walsh. *Gathering Strength, Gathering Storms: The One Hundred Year Study on Artificial Intelligence (AI100) 2021 Study Panel Report*. Tech. rep. Stanford University, Stanford, CA, 2021.

[135] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. "Microscopic Traffic Simulation using SUMO". In: *International Conference on Intelligent Transportation Systems*. 2018.

[136] R Duncan Luce. *Individual choice behavior: A theoretical analysis*. John Wiley, 1959.

[137] R. Duncan Luce. *Individual choice behavior*. Oxford, England: John Wiley, 1959.

[138] R.Duncan Luce. "The choice axiom after twenty years". In: *Journal of Mathematical Psychology* 15 (1977).

[139] Chiara Dalla Man, Francesco Micheletto, Dayu Lv, Marc Breton, Boris Kovatchev, and Claudio Cobelli. "The UVA/PADOVA Type 1 Diabetes Simulator: New Features". In: *Journal of Diabetes Science and Technology* 8.1 (2014), pp. 26–34.

[140] Shie Mannor, Vianney Perchet, and Gilles Stoltz. "Approachability in unknown games: Online learning meets multi-objective optimization". In: *Proceedings of the Conference on Learning Theory*. 2014.

[141] Matthew T McBee, Scott J Peters, and Craig Waterman. "Combining scores in multiple-criteria assessment systems: The impact of combination rule". In: *Gifted Child Quarterly* 58.1 (2014), pp. 69–89.

[142] Peter McMullen. "The maximum numbers of faces of a convex polytope". In: *Mathematika* 17.2 (1970), pp. 179–184.

[143] J. Mercer. "Functions of Positive and Negative Type, and their Connection with the Theory of Integral Equations". In: *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* 209 (1909).

[144] George A Miller. "The magical number seven, plus or minus two: Some limits on our capacity for processing information." In: *Psychological review* 63.2 (1956).

[145] Sobhan Miryoosefi, Kianté Brantley, Hal Daume III, Miro Dudik, and Robert E Schapire. "Reinforcement Learning with Convex Constraints". In: *Advances in Neural Information Processing Systems*. 2019.

[146] Jonas Mockus. *Bayesian approach to global optimization: theory and applications*. Vol. 37. Springer Science & Business Media, 2012.

[147] Oskar Morgenstern and John Von Neumann. *Theory of games and economic behavior*. Princeton university press, 1953.

[148] Katharina Muelling, Arun Venkatraman, Jean-Sebastien Valois, John E Downey, Jeffrey Weiss, Shervin Javdani, Martial Hebert, Andrew B Schwartz, Jennifer L Collinger, and J Andrew Bagnell. "Autonomy infused teleoperation with application to brain computer interface controlled manipulation". In: *Autonomous Robots* (2017).

[149] Yurii Nesterov and Vladimir Spokoiny. "Random gradient-free minimization of convex functions". In: *Foundations of Computational Mathematics* 17.2 (2017).

[150] Andrew Y Ng and Stuart J Russell. "Algorithms for inverse reinforcement learning." In: *International Conference on Machine Learning*. Vol. 1. 2000, p. 2.

[151] Andrew Y. Ng and Stuart Russel. "Algorithms for Inverse Reinforcement Learning". In: *International Conference on Machine Learning*. 2000.

[152] Andrew Y. Ng and Stuart Russell. "Algorithms for Inverse Reinforcement Learning". In: *in Proc. 17th International Conf. on Machine Learning*. Morgan Kaufmann, 2000, pp. 663–670.

[153] Efe A Ok. "Utility representation of an incomplete preference relation". In: *Journal of Economic Theory* 104.2 (2002).

[154] Chris Olah, Arvind Satyanarayan, Ian Johnson, Shan Carter, Ludwig Schubert, Katherine Ye, and Alexander Mordvintsev. "The building blocks of interpretability". In: *Distill* 3.3 (2018), e10.

[155] Houman Owhadi, Clint Scovel, and Tim Sullivan. "On the Brittleness of Bayesian Inference". In: *SIAM Review* (2015).

[156] Malayandi Palan, Nicholas C Landolfi, Gleb Shevchuk, and Dorsa Sadigh. "Learning reward functions by integrating human demonstrations and preferences". In: *arXiv preprint arXiv:1906.08928* (2019).

[157] Alexander Pan, Kush Bhatia, and Jacob Steinhardt. "The effects of reward misspecification: Mapping and mitigating misaligned models". In: *arXiv preprint arXiv:2201.03544* (2022).

[158] John P Papay. "Different tests, different answers: The stability of teacher value-added estimates across outcome measures". In: *American Educational Research Journal* 48.1 (2011), pp. 163–193.

[159] Romain Paulus, Caiming Xiong, and Richard Socher. "A Deep Reinforced Model for Abstractive Summarization". In: *International Conference on Learning Representations*. 2018.

[160] Jean-Paul Penot and Robert Ratsimahalo. "Characterizations of metric projections in Banach spaces and applications". In: *Abstract and Applied Analysis*. Vol. 3. 1970.

[161] Vianney Perchet. "A note on robust Nash equilibria with uncertainties". In: *RAIRO-Operations Research* 48.3 (2014), pp. 365–371.

[162] Vianney Perchet. "Approachability, regret and calibration; implications and equivalences". In: *arXiv preprint arXiv:1301.2663* (2013).

[163] Jean-Charles Pomerol and Sergio Barba-Romero. *Multicriterion decision in management: principles and practice*. Vol. 25. Springer Science & Business Media, 2012.

[164] Deepak Ramachandran and Eyal Amir. "Bayesian Inverse Reinforcement Learning". In: *IJCAI*. 2007.

[165] Siddharth Reddy, Anca D. Dragan, and Sergey Levine. "Where Do You Think You're Going?: Inferring Beliefs about Dynamics from Behavior". In: *Advances in Neural Information Processing Systems*. 2018.

[166] Siddharth Reddy, Sergey Levine, and Anca D. Dragan. "Assisted Perception: Optimizing Observations to Communicate State". In: *CoRR* abs/2008.02840 (2020).

[167] Manoel Horta Ribeiro, Raphael Ottoni, Robert West, Virgıélio A. F. Almeida, and Wagner Meira. "Auditing Radicalization Pathways on YouTube". In: *Conference on Fairness, Accountability, and Transparency*. New York, NY, USA, 2020.

[168] Peter M Robinson. "Root-N-consistent semiparametric regression". In: *Econometrica: Journal of the Econometric Society* (1988).

[169] Frank Rosenblatt. "The Perceptron: a probabilistic model for information storage and organization in the brain." In: *Psychological review* 65.6 (1958).

[170] Paat Rusmevichientong and John N Tsitsiklis. "Linearly parameterized bandits". In: *Mathematics of Operations Research* 35 (2010).

[171] Stuart Russell. *Human Compatible: Artificial Intelligence and the Problem of Control*. Penguin, 2019.

[172] Dorsa Sadigh, Anca D Dragan, Shankar Sastry, and Sanjit A Seshia. "Active preference-based learning of reward functions." In: *Robotics: Science and Systems*. 2017.

[173] William Saunders, Girish Sastry, Andreas Stuhlmueller, and Owain Evans. "Trial without error: towards safe reinforcement learning via human intervention". In: *Proceedings of the International Conference on Autonomous Agents and MultiAgent Systems*. 2018.

[174] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. "Proximal policy optimization algorithms". In: *arXiv preprint arXiv:1707.06347* (2017).

[175] Markus Schulze. "A new monotonic, clone-independent, reversal symmetric, and condorcet-consistent single-winner election method". In: *Social Choice and Welfare* 36.2 (2011), pp. 267–303.

[176] Paola Sebastiani and Henry P Wynn. "Maximum entropy sampling and optimal Bayesian experimental design". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 62 (2000).

[177] Rohin Shah, Noah Gundotra, Pieter Abbeel, and Anca D. Dragan. "On the Feasibility of Learning, Rather than Assuming, Human Biases for Reward Inference". In: *International Conference on Machine Learning*. 2019.

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

[179] Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. "Learnability, stability and uniform convergence". In: *The Journal of Machine Learning Research* 11 (2010).

[180] Ohad Shamir. "An optimal algorithm for bandit and zero-order convex optimization with two-point feedback". In: *The Journal of Machine Learning Research* 18.1 (2017).

[181] Ohad Shamir. "On the complexity of bandit and derivative-free stochastic convex optimization". In: *Proceedings of the Conference on Learning Theory*. 2013.

[182] Tali Sharot, Alison M Riccardi, Candace M Raio, and Elizabeth A Phelps. "Neural mechanisms mediating optimism bias". In: *Nature* (2007).

[183] John Shawe-Taylor, Nello Cristianini, et al. *Kernel methods for pattern analysis*. Cambridge university press, 2004.

[184] Björn Sprungk. "On the local Lipschitz stability of Bayesian inverse problems". In: *arxiv preprint arxiv:1906.07120* (2020).

[185] Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias W Seeger. "Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design". In: *International Conference on Machine Learning*. 2010.

[186] Neil Stewart, Gordon DA Brown, and Nick Chater. "Absolute identification by relative judgment." In: *Psychological review* 112.4 (2005).

[187] Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. "Learning to summarize from human feedback". In: *arXiv preprint arXiv:2009.01325* (2020).

[188] Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. "Learning to summarize from human feedback". In: *arXiv preprint arXiv:2009.01325* (2020).

[189] Jonathan Stray. "Aligning AI Optimization to Community Well-Being". In: *International Journal of Community Well-Being* 3.4 (2020), pp. 443–463.

[190] Jihoon Tack, Sangwoo Mo, Jongheon Jeong, and Jinwoo Shin. "CSI: Novelty Detection via Contrastive Learning on Distributionally Shifted Instances". In: *Advances in Neural Information Processing Systems* (2020).

[191] Jessica Taylor. "Quantilizers: A Safer Alternative to Maximizers for Limited Optimization". In: *AAAI Workshop: AI, Ethics, and Society*. 2016.

[192] Armando Teixeira-Pinto and Sharon-Lise T Normand. "Statistical methodology for classifying units on the basis of multiple-related measures". In: *Statistics in Medicine* 27.9 (2008), pp. 1329–1350.

[193] Suzanne C. Thompson. "Illusions of Control: How We Overestimate Our Personal Influence". In: *Current Directions in Psychological Science* (1999).

[194] Louis L Thurstone. "A law of comparative judgment." In: *Psychological review* 34.4 (1927), p. 273.

[195] Marin Toromanoff, Emilie Wirbel, and Fabien Moutarde. *Is Deep Reinforcement Learning Really Superhuman on Atari? Leveling the playing field*. 2019. arXiv: 1908.04683.

[196] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. "Congested traffic states in empirical observations and microscopic simulations". In: *Physical review E* 62.2 (2000), p. 1805.

[197] Alexander Trott, Sunil Srinivasa, Douwe van der Wal, Sebastien Haneuse, and Stephan Zheng. "Building a Foundation for Data-Driven, Interpretable, and Robust Policy Design using the AI Economist". In: *arXiv preprint arXiv:2108.02904* (2021).

[198] Alexandre B Tsybakov. *Introduction to Nonparametric Estimation*. Springer Science & Business Media, 2008.

[199] Amos Tversky and Daniel Kahneman. "Judgment under uncertainty: Heuristics and biases". In: *Science* 185.4157 (1974), pp. 1124–1131.

[200] Amos Tversky and Daniel Kahneman. "Prospect theory: An analysis of decision under risk". In: *Econometrica* 47.2 (1979), pp. 263–291.

[201] Sattar Vakili, Kia Khezeli, and Victor Picheny. "On information gain and regret bounds in Gaussian process bandits". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2021, pp. 82–90.

[202] Leslie G Valiant. "A theory of the learnable". In: *Communications of the ACM* 27.11 (1984).

[203] Michal Valko, Nathaniel Korda, Rémi Munos, Ilias Flaounas, and Nelo Cristianini. "Finite-time analysis of kernelised contextual bandits". In: *arXiv preprint arXiv:1309.6869* (2013).

[204] Aad W Van der Vaart. *Asymptotic statistics*. Vol. 3. Cambridge university press, 2000.

[205] Vladimir Vapnik. "Principles of risk minimization for learning theory". In: *Advances in neural information processing systems*. 1992, pp. 831–838.

[206] Dizan Vasquez, Billy Okal, and Kai O. Arras. "Inverse Reinforcement Learning algorithms and features for robot navigation in crowds: An experimental comparison". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2014.

[207] Eugene Vinitsky, Aboudy Kreidieh, Luc Le Flem, Nishant Kheterpal, Kathy Jang, Cathy Wu, Fangyu Wu, Richard Liaw, Eric Liang, and Alexandre M. Bayen. "Benchmarks for reinforcement learning in mixed-autonomy traffic". In: *Conference on Robot Learning*. 2018.

[208] Grace Wahba. *Spline models for observational data*. SIAM, 1990.

[209] Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*. Cambridge University Press, 2019.

[210] Glenn D Walters. "Taking the next step: Combining incrementally valid indicators to improve recidivism prediction". In: *Assessment* 18.2 (2011), pp. 227–233.

[211] Yining Wang, Sivaraman Balakrishnan, and Aarti Singh. "Optimization of smooth functions with noisy observations: Local minimax rates". In: *Advances in Neural Information Processing Systems*. 2018, pp. 4338–4349.

[212] Christopher K Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*. Vol. 2. 3. MIT press Cambridge, MA, 2006.

[213] Cathy Wu, Abdul Rahman Kreidieh, Kanaad Parvate, Eugene Vinitsky, and Alexandre M. Bayen. "Flow: A Modular Learning Framework for Mixed Autonomy Traffic". In: *IEEE Transactions on Robotics* (2021).

[214] Huasen Wu and Xin Liu. "Double Thompson sampling for dueling bandits". In: *Advances in Neural Information Processing Systems*. 2016.

[215] M. Wulfmeier, P. Ondruska, and I. Posner. "Maximum Entropy Deep Inverse Reinforcement Learning". In: *Neural Information Processing Systems Conference, Deep Reinforcement Learning Workshop*. 2015.

[216] Songbai Yan and Chicheng Zhang. "Revisiting perceptron: Efficient and label-optimal learning of halfspaces". In: *Advances in Neural Information Processing Systems*. 2017.

[217] F Eugene Yates. *Self-organizing systems: The emergence of order*. Springer Science & Business Media, 2012.

[218] Chao Yu, Jiming Liu, and Shamim Nemati. "Reinforcement learning in healthcare: A survey". In: *arXiv preprint arXiv:1908.08796* (2019).

[219] Yisong Yue, Josef Broder, Robert Kleinberg, and Thorsten Joachims. "The k-armed dueling bandits problem". In: *Journal of Computer and System Sciences* 78.5 (2012), pp. 1538–1556.

[220] Luděk Zajíček. "On the Fréchet differentiability of distance functions". In: *Proceedings of the Winter School on Abstract Analysis* (1984), pp. 161–165.

[221] Juanjuan Zhang, Pieter Fiers, Kirby A Witte, Rachel W Jackson, Katherine L Poggensee, Christopher G Atkeson, and Steven H Collins. "Human-in-the-loop optimization of exoskeleton assistance during walking". In: *Science* 356.6344 (2017), pp. 1280–1284.

[222] Tong Zhang. "Effective dimension and generalization of kernel learning". In: *NIPs*. Vol. 4. 1. Citeseer. 2002, pp. 454–461.

[223] Simon Zhuang and Dylan Hadfield-Menell. "Consequences of Misaligned AI". In: *Advances in Neural Information Processing Systems*. 2020.

[224] B. D. Ziebart, J. A. Bagnell, and A. K. Dey. "Modeling Interaction via the Principle of Maximum Causal Entropy". In: *International Conference on Machine Learning (ICML)*. 2010.

[225] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey. "Maximum Entropy Inverse Reinforcement Learning". In: *International Conference on Artificial Intelligence (AAAI)*. 2008.

[226] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. "Maximum entropy inverse reinforcement learning." In: *Aaai*. Vol. 8. 2008.

[227] Brian D Ziebart, Nathan Ratliff, Garratt Gallagher, Christoph Mertz, Kevin Peterson, J Andrew Bagnell, Martial Hebert, Anind K Dey, and Siddhartha Srinivasa. "Planning-based prediction for pedestrians". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2009.

[228] Masrour Zoghi, Zohar S Karnin, Shimon Whiteson, and Maarten De Rijke. "Copeland dueling bandits". In: *Advances in Neural Information Processing Systems*. 2015.

[229] Masrour Zoghi, Shimon Whiteson, Remi Munos, and Maarten De Rijke. "Relative upper confidence bound for the k-armed dueling bandit problem". In: *arXiv preprint arXiv:1312.3393* (2013).

[230] Masrour Zoghi, Shimon Whiteson, and Maarten de Rijke. "MergeRUCB: A method for large-scale online ranker evaluation". In: *Proceedings of the International Conference on Web Search and Data Mining*. 2015.

[231] Constantin Zopounidis and Michael Doumpos. "Multicriteria classification and sorting methods: A literature review". In: *European Journal of Operational Research* 138.2 (2002), pp. 229–246.

Part II

Appendices

Appendix A

Deferred content from Chapter 2

A.1 Deferred proofs from Section 2.4

Proof of Proposition 2.1

The first part of the proof essentially follows the same as that for Theorem 2.2. The proof differs in how we upper bound Term (I) from equation (2.11).

$$\begin{aligned}
\hat{U}(f_{\mathsf{ERM}}; u^*) - \hat{U}(\hat{f}_{k,n}; u^*) &\leq \frac{1}{n}\sum_{i=1}^{n}(\mathbb{I}[f_{\mathsf{ERM}}(x_i) = y_i] - \mathbb{I}[\hat{f}_{k,n}(x_i) = y_i])(u^*_{\mathsf{gap}}(x_i) - \hat{u}_{\mathsf{gap}}(x_i)) \\
&= \frac{1}{n}\sum_{i=1}^{n}(\mathbb{I}[\hat{f}_{k,n}(x_i) \neq y_i] - \mathbb{I}[f_{\mathsf{ERM}}(x_i) \neq y_1])(\hat{u}_{\mathsf{gap}}(x_i) - u^*_{\mathsf{gap}}(x_i)) \qquad \text{(A.1)} \\
&\overset{\text{(i)}}{\leq} \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}[f_{\mathsf{ERM}}(x_i) \neq y_i](u^*_{\mathsf{gap}}(x_i) - \hat{u}_{\mathsf{gap}}(x_i)) \\
&\leq \max_i[u^*_{\mathsf{gap}}(x_i) - \hat{u}_{\mathsf{gap}}(x_i)] \cdot \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}[f_{\mathsf{ERM}}(x_i) \neq y_i], \qquad \text{(A.2)}
\end{aligned}$$

where inequality (i) follows from the fact that $\hat{u}$ is a lower estimate of u^*. This establishes the desired claim. □

Proof of Lemma 2.1

We begin by noting that for any given datapoint x_i, the deterministic comparison oracle $\mathcal{O}_k$ when queried with $q_{i,t}$ outputs

$$\mathcal{O}_k(q_{i,j}) = \mathbb{I}\left[\frac{k}{2}u^*_{\mathsf{gap}}(x_i) \geq \lambda u^*_{\mathsf{max}}\right] ,$$

for values[1] of $\lambda \in [\frac{k}{2}]$. This effectively allows one to compare the utility gap u^*_i with u^*_{max} at a multiplicative granularity of $\frac{2}{k}$. With this observation, let us establish that for any time $t \in [T]$, for any datapoint $x_i \in S$, we have

$$\hat{u}^t_{\mathsf{gap}}(x_i) - \frac{u^*_{\mathsf{max}}}{2^t} \leq u^*_{\mathsf{gap}}(x_i) \leq \hat{u}^t_{\mathsf{gap}}(x_i). \tag{A.3}$$

The proof will proceed via an inductive argument.

Base Case. For initial time $t = 0$, by the boundedness of the utility functions, we have for all x_i,

$$\hat{u}^0_{\mathsf{gap}}(x_i) - u^*_{\mathsf{max}} = 0 \leq u^*_{\mathsf{gap}}(x_i) \leq u^*_{\mathsf{max}} = \hat{u}^0_{\mathsf{gap}}(x_i).$$

Induction Step. Assume that for some $t = s$, equation (A.3) holds for all $x_i \in S$. We will now show that it holds for $t = s + 1$. Note that by the induction hypothesis, the value of λ at time $s + 1$ can be equivalently written as

$$\lambda = \frac{k}{2u^*_{\mathsf{max}}} \cdot \left(\frac{\hat{u}^s_{\mathsf{gap}}(x_i) - \frac{u^*_{\mathsf{max}}}{2^s} + \hat{u}^s_{\mathsf{gap}}(x_i)}{2} \right),$$

that is, as a scaled mid-point of the confidence interval at time s. the query $q_{i,t}$ then compares the gap $u^*_{\mathsf{gap}}(x_i)$ with the mid-point of the confidence interval.

Case 1. If the response $r_{i,t} = 1$ which implies that $\hat{u}^s_{\mathsf{gap}}(x_i) \geq \frac{2\lambda}{k}$, the upper estimate remains the same and the lower estimate is (implicitly) moved to the mid-point $\frac{2\lambda}{k}$ since we know from the oracle's response that $u^*_{\mathsf{gap}}(x_i)$ is greater than the mid-point. Thus, after each update, the confidence interval shrinks by a factor of $\frac{1}{2}$ and reduces to $\frac{1}{2^{s+1}}$ at the end of time $t = s+1$.

Case 2. On the other hand if $r_{i,t} = 0$, the estimate $\hat{u}^{s+1}_{\mathsf{gap}}(x_i)$ is updated to be the midpoint $\frac{u^*_{\mathsf{max}}}{2^{s+1}}$ while the lower estimate remains the same because of the oracle's response.

Combining both the cases above, we see that at time $t = s + 1$, the confidence interval for $u^*_{\mathsf{gap}}(x_i)$ is exactly $\frac{1}{2^{s+1}}$ for both the cases. Thus, we must have that

$$\hat{u}^{s+1}_{\mathsf{gap}}(x_i) - \frac{u^*_{\mathsf{max}}}{2^{s+1}} \leq u^*_{\mathsf{gap}}(x_i) \leq \hat{u}^{s+1}_{\mathsf{gap}}(x_i).$$

This establishes the first part of the claim. The bound on the query complexity follows from the fact that for each datapoint x_i, we use $\log_2 k - 1$ queries to the oracle in the procedure. This establishes the desired claim. □

[1] We denote by $[d]$ the set of integers $\{1, \ldots, d\}$.

Proof of Corollary 2.1

The excess risk of the plug-in estimator can be upper-bounded from Proposition 2.1 as

$$\mathsf{err}(\hat{f}_{k,n}, \mathcal{F}; u^*) \leq 2 \cdot \sup_{f \in \mathcal{F}} \left(|U(f; u^*) - \hat{U}(f; u^*)| \right) + \max_i [u^*_{\mathsf{gap}}(x_i) - \hat{u}_{\mathsf{gap}}(x_i)] \cdot \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}[f_{\mathsf{ERM}}(x_i) \neq y_i]$$

$$\overset{\text{Lemma 2.1}}{\leq} 2 \cdot \sup_{f \in \mathcal{F}} \left(|U(f; u^*) - \hat{U}(f; u^*)| \right) + \frac{2u^*_{\mathsf{max}}}{k} \cdot \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}[f_{\mathsf{ERM}}(x_i) \neq y_i] \,, \tag{A.4}$$

where the last inequality follows by noting that Comptron produces an estimate $\hat{u}_{\mathsf{gap}}$ of the utility gap $u^*_{\mathsf{gap}}(x_i)$ with an additive error of $\frac{2u^*_{\mathsf{max}}}{k}$. □

Proof of Lemma 2.2

To establish the above claim, we show that the updates to the gap estimates $\hat{u}^t_{\mathsf{gap}}$ performed by Rob-Comptron mirror those performed by the deterministic Comptron with high probability. For any datapoint x_i and any time $t \in [T]$, denote by $r^*_{i,t} = \mathbb{E}[r_{i,j,t}]$ the expected value of the response for query $q_{i,j,t}$. By Assumption 2.1, we have that $\mathbb{I}[r^*_{i,t} < \frac{1}{2}]$ provides the true label for the query $q_{i,j,t}$. By an application of the Hoeffding's inequality, we have,

$$\Pr\left(\mathbb{I}\left[\frac{1}{J} \sum_j r_{i,j,t} < \frac{1}{2} \right] \neq \mathbb{I}\left[r^*_{i,t} < \frac{1}{2} \right] \right) \leq \exp\left(\frac{-J(1-2\eta)^2}{4} \right).$$

Taking a union bound over all datapoints x_i and time $t \in [T]$, and substituting the value of $J = \frac{8}{(1-2\eta)^2} \log(\frac{nT}{\delta})$, we have,

$$\Pr\left(\exists i, t \text{ s.t. } \mathbb{I}\left[\frac{1}{J} \sum_j r_{i,j,t} < \frac{1}{2} \right] \neq \mathbb{I}\left[r^*_{i,t} < \frac{1}{2} \right] \right) \leq \delta. \tag{A.5}$$

From the above equation, we have that with probability at least $1-\delta$, every update performed by Rob-Comptron uses the correct label. Combining the above with the proof of Lemma 2.1 establishes the required claim. □

Proof of Lemma 2.3

Observe that from the conditions of the lemma statement, we have

$$\left(1 - \frac{1}{k}\right) \cdot u_{\mathsf{gap}}(x_1) \leq u^1_{\mathsf{gap}}(x_2), u^2_{\mathsf{gap}}(x_2) \leq u_{\mathsf{gap}}(x_1).$$

Assume without loss of generality that $u^1_{\mathsf{gap}}(x_2) > u^2_{\mathsf{gap}}(x_2)$. Observe that any k-query comprising only points x_1 and x_2 must have the form

$$\mathbf{x} = (\underbrace{x_1, \ldots, x_1}_{j_1 \text{ times}}, \underbrace{x_2, \ldots, x_2}_{j_2 \text{ times}}), \quad \mathbf{y}_1 = (\underbrace{y_1, \ldots, y_1}_{j_1 \text{ times}}, \underbrace{\bar{y}_2, \ldots, \bar{y}_2}_{j_2 \text{ times}}), \quad \mathbf{y}_1 = (\underbrace{\bar{y}_1, \ldots, \bar{y}_1}_{j_1 \text{ times}}, \underbrace{y_2, \ldots, y_2}_{j_2 \text{ times}})$$

with $j_1 + j_2 = k$. For any query q to be different under the oracles $\mathcal{O}_k(\cdot\,; u_1)$ and $\mathcal{O}_k(\cdot\,; u_2)$, we should have

$$\mathbb{I}[j_1 u^1_{\mathsf{gap}}(x_1) > j_2 u^1_{\mathsf{gap}}(x_2)] = 1 \quad \text{and} \quad \mathbb{I}[j_1 u^2_{\mathsf{gap}}(x_1) > j_2 u^2_{\mathsf{gap}}(x_2)] = 0$$

since $u^1_{\mathsf{gap}}(x_2) > u^2_{\mathsf{gap}}(x_2)$. In order for the above equation to be satisfied, we requires that the ratio $\frac{j_1}{j_2} \geq 1 - \frac{1}{k}$. However, under the constraints $j_1 + j_2 = k$, this is not possible. Hence, it is not possible to distinguish between the utilities u_1 and u_2 using a k comparison oracle. □

A.2 Deferred proofs from Section 2.5

Proof of Proposition 2.2

Our example construction will focus on the real-valued feature space $\mathcal{X} = \mathbb{R}$, binary decision space $\mathcal{Y} = \{0, 1\}$, and the class of linear decision functions

$$\mathcal{F}_{\mathsf{lin}} = \{f_a \mid f_a(x) = \text{sign}(ax),\ a \in [-1, 1]\}\ .$$

Distribution $\mathcal{D}_x$. Our example will focus on three points $x_1 = 1, x_2 = 2, x_3 = -1$ with their population probabilities given by

$$\Pr(x = x_1) = p, \quad Pr(x = x_2) = p, \quad \text{and} \quad \Pr(x = x_3) = 1 - 2p\ ,$$

for some value $p > 0$ which we define later. Note that our final choice of p will depend on the order k of the comparison oracle.

Utility function u^*. Given the above three points, we set the utility $u^*(x_i, 0) = 0$ for all datapoints x_i. The utilities for label $y = 1$ are given by

$$u^*(x_1, 1) = 1, \quad u^*(x_2, 1) = \frac{4}{k}, \quad \text{and} \quad u^*(x_3, 1) = \frac{2}{k^2}.$$

With these utilities, observe that the true label $y_i = 1$ for all the datapoints. Further, any predictor $f \in \mathcal{F}_{\mathsf{lin}}$ can either correctly predict the points $\{x_1, x_2\}$ or the point x_3 but not all three simultaneously.

Performance of predictors. For this setup described above, we now proceed to describe the optimal function f^*, the plug-in estimate $\hat{f}_k$ and an alternate predictor $\tilde{f}$ which outperforms the plug-in estimate. Observe that any estimator will pick either f_{-1} or f_{+1} depending on the value of p.

Optimal Classifier. The difference in the expected utility between the classifiers f_{+1} and f_{-1} is given by

$$\begin{aligned} U(f_{+1};u^*) - U(f_{-1};u^*) &= p\cdot\left(1+\frac{4}{k}\right) - (1-2p)\cdot\left(\frac{2}{k^2}\right) \\ &= \frac{p}{k^2}\cdot(k+2)^2 - \frac{2}{k^2} \\ &= \frac{1}{k^2}\left(p(k+2)^2 - 2\right). \end{aligned}$$

Given the calculation above, the optimal classifier f^* is given by

$$f^* = \begin{cases} f_{+1} & \text{for } p \geq \frac{2}{(k+2)^2} \\ f_{-1} & \text{otherwise} \end{cases}. \tag{A.6}$$

Plug-in estimate $\hat{f}_k$. We now study the prediction $\hat{f}_k$ obtained by using the prediction $\hat{u}$ from Comptron (Algorithm 1). Recall that since Comptron produces upper estimates for u^*_{gap} (which is equivalent to u^* since $u^*(x,0)=0$) within an error of $\frac{2}{k}$, the output estimates will be

$$\hat{u}(x_1,1) = 1, \quad \hat{u}(x_2,1) = \frac{4}{k}, \quad \text{and} \quad \hat{u}(x_3,1) = \frac{2}{k}.$$

Observe that while Comptron is able to correctly learn the utilities for x_1 and x_2, it overestiamtes the utility for the point x_3. Let us look at the difference of estimated utilities

$$\begin{aligned} U(f_{+1};\hat{u}) - U(f_{-1};\hat{u}) &= p\cdot\left(1+\frac{4}{k}\right) - (1-2p)\cdot\left(\frac{2}{k}\right) \\ &= \frac{p}{k}\cdot(k+8) - \frac{2}{k} \\ &= \frac{1}{k}\cdot(p(k+8)-2). \end{aligned}$$

Given the above calculations, we see that the function $\hat{f}_k$ is given by

$$\hat{f}_k = \begin{cases} f_{+1} & \text{for } p \geq \frac{2}{k+8} \\ f_{-1} & \text{otherwise} \end{cases}. \tag{A.7}$$

Alternate estimator $\tilde{f}$. While Comptron compares the utilities of both x_2 and x_3 with respect to x_1 (equivalently $x_{i_{\mathsf{max}}}$), consider the alternate procedure which differs in the estimation of utility gap $u^*_{\mathsf{gap}}(x_3)$. Instead of using the proposed queries $q_{3,t}$ of Comptron, we modify those as $\tilde{q}_{3,t} = (\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2)$ where

$$\mathbf{x} = (\underbrace{x_3,\ldots,x_3}_{\frac{k}{2}\text{ times}}, \underbrace{x_2,\ldots,x_2}_{\lambda\text{ times}}), \quad \mathbf{y}_1 = (\underbrace{y_3,\ldots,y_3}_{\frac{k}{2}\text{ times}}, \underbrace{1-y_2,\ldots,1-y_2}_{\lambda\text{ times}}), \quad \mathbf{y}_2 = 1-\mathbf{y}_1.$$

Following the same proof as of Lemma 2.1, we can show that one can obtain an upper estimate $\tilde{u}(x_3, 1) = \frac{8}{k^2}$. This follows from the fact that we can deduce that $u^*(x_3) \in [0, \frac{2u^*(x_2)}{k}]$ from the above queries and combining this with the fact that $u^*(x_2) \leq \frac{4}{k}$. Evaluating the difference between the utilities with respect to $\tilde{u}$, we get

$$\begin{aligned} U(f_{+1}; \tilde{u}) - U(f_{-1}; \tilde{u}) &= p \cdot \left(1 + \frac{4}{k}\right) - (1 - 2p) \cdot \left(\frac{8}{k^2}\right) \\ &= \frac{p}{k^2} \cdot ((k+2)^2 + 12) - \frac{8}{k^2} \\ &= \frac{1}{k^2} \left(p((k+2)^2 + 12) - 8\right). \end{aligned}$$

Using such estimates $\tilde{u}$ with the plug-in estimator in equation (2.7), we have that the function

$$\tilde{f} = \begin{cases} f_{+1} & \text{for } p \geq \frac{8}{(k+2)^2+12} \\ f_{-1} & \text{otherwise} \end{cases}. \tag{A.8}$$

Thus, the three estimators f^*, $\hat{f}_k$ and $\tilde{f}$ differ in the threshold for p for switching between the functions f_{+1} and f_{-1}. Setting a value of $p = \frac{1}{k+8}$, we see that for $k > 10$

$$\frac{2}{(k+2)^2} < \frac{8}{(k+2)^2 + 12} < \underbrace{\frac{1}{k+8}}_{p} < \frac{2}{k+8}.$$

Thus, for this setting of p, while the predictor $f^* = \tilde{f} = f_{+1}$, the estimator $\hat{f}_k = f_{-1}$ and hence it incurs an excess risk $\mathsf{err}(\hat{f}_k, \mathcal{F}; u^*) = \frac{1}{k}$. This establishes the first part of the claim.

For the second part, observe that the estimator $\tilde{f}$ outputs f_+ for the particular setting of p for all $\tilde{u}_3 \in [0, \frac{8}{k^2}]$. This set precisely captures the set of all utilities which are consistent with the k oracle $\mathcal{O}(\cdot\,; k)$. Since the optimal decision function $f^* = f_+$, this establishes the second part of the claim. □

Proof of Theorem 2.5

Let us represent by $\Delta_{\mathcal{F}}$ the space of probability distributions over the function $\mathcal{F}$. The error of the estimator p_{rob} can then be upper bounded as

$$\begin{aligned} \mathbb{E}[\mathsf{err}(p_{\mathsf{rob}}, \mathcal{F}; u^*)] &= \inf_{p \in \Delta_{\mathcal{F}}} \sup_{u' \in \mathcal{U}_{|u^*}} \mathbb{E}_f[\mathsf{err}(f, \mathcal{F}; u')] \\ &= \inf_{p \in \Delta_{\mathcal{F}}} \sup_{u' \in \mathcal{U}_{|u^*}} \sup_{f' \in \mathcal{F}} \mathbb{E}_f[U(f'; u') - U(f; u')] \\ &\overset{\text{(i)}}{=} \sup_{p \in \Delta_{\mathcal{F} \times \mathcal{U}_{|u^*}}} \inf_{f \in \mathcal{F}} \mathbb{E}_{(f', u')} \left[U(f'; u') - U(f; u')\right], \end{aligned}$$

where the equality (i) follows from an application of Sion's minimax theorem and the space $\Delta_{\mathcal{F}\times\mathcal{U}_{|u^*}}$ denotes the space of all distributions over the joint space $\mathcal{F}\times\mathcal{U}_{|u^*}$. Let us decompose the distribution $p = q_u \cdot q_{f|u}$ where q_u represents the marginal distribution over the space $\mathcal{U}_{|u^*}$ and $q_{f|u}$ denotes the conditional distirbution of sampling a function $f \in \mathcal{F}$ given utility function u. Denote by

$$f_u := \operatorname*{argmax}_{f\in\mathcal{F}} U(f;u) \quad \text{and} \quad f_p := \operatorname*{argmax}_{f\in\mathcal{F}} \mathbb{E}_{u\sim p}[U(f;u)]$$

as the maximizers for the corresponding (expected) utility functions. Then, the excess risk

$$\begin{aligned}
\mathbb{E}[\mathsf{err}(p_{\mathsf{rob}},\mathcal{F};u^*)] &= \sup_{p\in\Delta_{\mathcal{F}\times\mathcal{U}_{|u^*}}} \inf_{f\in\mathcal{F}} \mathbb{E}_{(f',u')}\left[U(f';u') - U(f;u')\right] \\
&= \sup_{p_u}\sup_{p_{f|u}}\inf_{f\in\mathcal{F}} \left(\mathbb{E}_{u'\sim p_u}\mathbb{E}_{f'\sim p_{f|u'}}\left[U(f';u')\right] - \mathbb{E}_{u'\sim p_u}\left[U(f;u')\right]\right) \\
&\overset{\text{(i)}}{=} \sup_{p_u}\sup_{p_{f|u}} \left(\mathbb{E}_{u'\sim p_u}\mathbb{E}_{f'\sim p_{f|u'}}\left[U(f';u')\right] - \mathbb{E}_{u'\sim p_u}\left[U(f_{p_u};u')\right]\right) \\
&\overset{\text{(ii)}}{=} \sup_{p_u}\left(\mathbb{E}_{u'\sim p_u}\left[U(f_{u'};u') - U(f_{p_u};u')\right]\right),
\end{aligned}$$

where the inequality (i) follows from the fact that f_{p_u} maximizes the expected utility with respect to p_u and (ii) follows by noting that the maximizing distribution $p_{f|u'} = \mathbb{I}[f = f_{u'}]$. Noting that f_{p_u} is the maximizer corresponding to the distribution p_u, we have,

$$\begin{aligned}
\mathbb{E}[\mathsf{err}(p_{\mathsf{rob}},\mathcal{F};u^*)] &= \sup_{p_u}\left(\mathbb{E}_{u'\sim p_u}\left[U(f_{u'};u') - \mathbb{E}_{\tilde{u}\sim p_u}[U(f_{\tilde{u}};u')] + \mathbb{E}_{\tilde{u}\sim p_u}[U(f_{\tilde{u}};u')] - U(f_{p_u};u')\right]\right) \\
&\leq \sup_{p_u}\left(\mathbb{E}_{u'\sim p_u}\left[U(f_{u'};u') - \mathbb{E}_{\tilde{u}\sim p_u}[U(f_{\tilde{u}};u')]\right]\right) \\
&\overset{\text{(i)}}{\leq} \sup_{u_1,u_2\in\mathcal{U}_{|u^*}} \left(U(f_{u_1};u_1) - U(f_{u_2};u_1)\right),
\end{aligned}$$

where the inequality (i) follows by upper bounding the expected deviation with a worst-case deviation. This establishes the required claim. □

Appendix B

Deferred content from Chapter 3

B.1 Proofs

Proof of Theorem 3.1

Without loss of generality, let $\tilde{\pi}$ satisfy $\tilde{\pi}(a \mid s; \theta) = \exp(\Phi(s, a; \theta))/Z(s; \theta)$. Note that this parameterization can be used to express any probability distribution. Also, for any $\delta > 0$, let us define

$$B_\delta(\mathcal{D}) = \bigcup_{t=1}^{n} \left(a_t - \frac{\delta}{2}, a_t + \frac{\delta}{2} \right)$$

as a union of $\frac{\delta}{2}$-balls around the actions that appear in dataset $\mathcal{D}$. Then, for any $\theta^* \in \Theta$, let π^* satisfy

$$\pi^*(a \mid s; \theta) = \begin{cases} \frac{1}{Z'(s;\theta)} \left(\exp(\Phi(s, a; \theta)) \mathbb{1}\{a \notin B_\delta(\mathcal{D})\} + C \; \mathbb{1}\{a \in B_\delta(\mathcal{D})\}\right) & \text{if } \theta = \theta^*, \\ \frac{1}{Z(s;\theta)} \exp(\Phi(s, a; \theta)) & \text{otherwise}, \end{cases}$$

where C is the supremum $C = \sup_{s,a,\theta} \exp(\Phi(s, a; \theta))$. By construction, it is clear that MLE on $\mathcal{D}$ using π^* would yield reward parameter θ^*. Since such π^* can be constructed for any θ^*, we can choose θ^* that satisfies $\|\tilde{\theta} - \theta^*\|_2^2 > \sup_{\theta,\theta'} \|\theta - \theta'\|_2/2$. What remains is showing that there exists δ such that $d_\pi^{\mathsf{wc}}(\pi^*, \tilde{\pi}) < \varepsilon$.

Bounding the worst-case policy divergence. Note that the worst-case divergence is necessarily satisfied at θ^*. Fix any state $s \in \mathcal{S}$. We have,

$$\begin{aligned} D_{\mathrm{KL}}(\pi^*(\cdot \mid s; \theta^*) || \tilde{\pi}(\cdot \mid s; \theta^*)) &= \int_{\mathcal{A}} \pi^*(a \mid s; \theta^*) \log \frac{\pi^*(a \mid s; \theta^*)}{\tilde{\pi}(a \mid s; \theta^*)} da \\ &= \int_{\mathcal{A} \setminus B_\delta(\mathcal{D})} \pi^*(a \mid s; \theta^*) \log \frac{\pi^*(a \mid s; \theta^*)}{\tilde{\pi}(a \mid s; \theta^*)} da + \int_{B_\delta(\mathcal{D})} \pi^*(a \mid s; \theta^*) \log \frac{\pi^*(a \mid s; \theta^*)}{\tilde{\pi}(a \mid s; \theta^*)} da \,. \end{aligned}$$

We consider each term individually. Starting with the first term, we have

$$\int_{\mathcal{A}\setminus B_\delta(\mathcal{D})} \pi^*(a \mid s;\theta^*) \log \frac{\pi^*(a \mid s;\theta^*)}{\tilde{\pi}(a \mid s;\theta^*)} da \leq \log \frac{Z(s;\theta^*)}{Z'(s;\theta^*)} \\ \leq \frac{1}{Z(s;\theta^*)} \left| Z(s;\theta^*) - Z'(s;\theta^*) \right| ,$$

where we use that the policies only differ in their normalizers in the first inequality, and that $\log(t) - \log(s) \leq \frac{1}{\min\{t,s\}} |t - s|$ in the second. Now, using that

$$Z'(s;\theta^*) = \int_{\mathcal{A}\setminus B_\delta(\mathcal{D})} \exp(\Phi(s,a,;\theta^*) da + C \left| B_\delta(\mathcal{D}) \right| ,$$

we have

$$\int_{\mathcal{A}\setminus B_\delta(\mathcal{D})} \pi^*(a \mid s;\theta^*) \log \frac{\pi^*(a \mid s;\theta^*)}{\tilde{\pi}(a \mid s;\theta^*)} da \leq \frac{C}{Z(s;\theta^*)} \left| B_\delta(\mathcal{D}) \right| .$$

Now, let us consider the second term. we have

$$\int_{B_\delta(\mathcal{D})} \pi^*(a \mid s;\theta^*) \log \frac{\pi^*(a \mid s;\theta^*)}{\tilde{\pi}(a \mid s;\theta^*)} da \leq \int_{B_\delta(\mathcal{D})} \pi^*(a \mid s;\theta^*) \left| \log C - \Phi(s,a;\theta^*) \right| da + \log \frac{Z(s;\theta^*)}{Z'(s;\theta^*)} \\ \leq 2 \log C \left| B_\delta(\mathcal{D}) \right| + \frac{C}{Z(s;\theta^*)} \left| B_\delta(\mathcal{D}) \right| ,$$

where we reuse the bound for the first term, and use that $|\phi(s,a;\theta^*)| \leq \log C$. Combining the two bounds yields

$$D_{\mathrm{KL}}(\pi^*(\cdot \mid s;\theta^*) || \tilde{\pi}(\cdot \mid s;\theta^*)) \leq 2 \log C \left| B_\delta(\mathcal{D}) \right| + \frac{2C}{Z(s;\theta^*)} \left| B_\delta(\mathcal{D}) \right| .$$

Using that $|B_\delta(\mathcal{D})| \leq n\delta$ by construction, we can solve for $\delta = \mathcal{O}(\varepsilon/n)$ such that $d_\pi^{\mathsf{wc}}(\pi^*, \tilde{\pi}) < \varepsilon$, as desired. This completes the proof.

Proof of Theorem 3.2

Recall that $L(\theta;\pi,\mathcal{D})$ is the negative log-likelihood of demonstrations $\mathcal{D}$ under policy π and reward parameters θ. Note that we can write

$$L(\theta;\tilde{\pi},\mathcal{D}) = \frac{1}{n}\sum_{t=1}^{n} -\log \tilde{\pi}(a_t \mid s_t;\theta) = \frac{1}{n}\sum_{t=1}^{n} -\log \pi^*(a_t \mid s_t;\theta) + \log \frac{\pi^*(a_t \mid s_t;\theta)}{\tilde{\pi}(a_t \mid s_t;\theta)} \\ = L(\theta;\pi^*,\mathcal{D}) + \frac{1}{n}\sum_{t=1}^{n} \log \frac{\pi^*(a_t \mid s_t;\theta)}{\tilde{\pi}(a_t \mid s_t;\theta)} .$$

By the law of large numbers, we have that under expectation over dataset $\mathcal{D}$,

$$\mathbb{E}_{\mathcal{D}\sim\pi^*}\left[\frac{1}{n}\sum_{t=1}^{n}\log\frac{\pi^*(a_t \mid s_t;\theta)}{\tilde{\pi}(a_t \mid s_t;\theta)}\right] = \mathbb{E}_{s\sim d^*}\left[D_{\mathrm{KL}}(\pi^*(\cdot \mid s;\theta^*) \mid\mid \tilde{\pi}(\cdot \mid s;\theta^*))\right]$$

Using Assumption 3.2 on π^*, for any $\theta \in \Theta$, we also have

$$L(\theta;\pi^*,\mathcal{D}) \geq L(\theta^*;\pi^*,\mathcal{D}) + \nabla_\theta L(\theta^*;\pi^*,\mathcal{D})^\top(\theta^*-\theta) + \frac{cn}{2}\|\theta-\theta^*\|_2^2 .$$

By definition of θ^* and Assumption 3.1, we know that $\nabla_\theta L(\theta^*;\pi^*,\mathcal{D}) = \mathbf{0}$. Substituting $\theta = \tilde{\theta}$ and rearranging yields

$$\|\tilde{\theta}-\theta^*\|_2^2 \leq \frac{2}{c}\left(L(\tilde{\theta};\pi^*,\mathcal{D}) - L(\theta^*;\pi^*,\mathcal{D})\right) .$$

Analogously, using Assumption 3.2 on $\tilde{\pi}$ [1], we have that

$$\|\tilde{\theta}-\theta^*\|_2^2 \leq \frac{2}{c}\left(L(\theta^*;\tilde{\pi},\mathcal{D}) - L(\tilde{\theta};\tilde{\pi},\mathcal{D})\right) .$$

Combining the two bounds yields,

$$\begin{aligned}\|\tilde{\theta}-\theta^*\|_2^2 &\leq \frac{2}{c}\left(L(\theta^*;\tilde{\pi},\mathcal{D}) - L(\theta^*;\pi^*,\mathcal{D}) + L(\tilde{\theta};\pi^*,\mathcal{D}) - L(\tilde{\theta};\tilde{\pi},\mathcal{D})\right)\\ &\leq \frac{2}{c}\left(\frac{1}{n}\sum_{t=1}^{n}\log\frac{\pi^*(a_t \mid s_t;\theta^*)}{\tilde{\pi}(a_t \mid s_t;\theta^*)} - \frac{1}{n}\sum_{t=1}^{n}\log\frac{\pi^*(a_t \mid s_t;\tilde{\theta})}{\tilde{\pi}(a_t \mid s_t;\tilde{\theta})}\right)\end{aligned}$$

Taking an expectation over dataset $\mathcal{D}$ yields the desired result

$$\begin{aligned}\mathbb{E}\left[\|\tilde{\theta}-\theta^*\|_2^2\right] &\leq \frac{2}{c}\mathbb{E}_{s\sim d^*}\left[D_{\mathrm{KL}}(\pi^*(\cdot \mid s;\theta^*) \mid\mid \tilde{\pi}(\cdot \mid s;\theta^*)) - D_{\mathrm{KL}}(\pi^*(\cdot \mid s;\tilde{\theta}) \mid\mid \tilde{\pi}(\cdot \mid s;\tilde{\theta}))\right]\\ &\leq \frac{2}{c}\mathbb{E}_{s\sim d^*}\left[D_{\mathrm{KL}}(\pi^*(\cdot \mid s;\theta^*) \mid\mid \tilde{\pi}(\cdot \mid s;\theta^*))\right] ,\end{aligned}$$

which is the desired result.

Proof of Corollary 3.1

Recall that $\tilde{\pi}, \pi^*$ are parameterized by Q-values $\tilde{Q}, Q^*$ that satisfy the soft Bellman update in (3.4). Fix state s. We have

$$D_{\mathrm{KL}}(\pi^*(\cdot \mid s;\theta^*)||\tilde{\pi}(\cdot \mid s;\theta^*)) = \mathbb{E}_{a\sim\pi^*(\cdot|s;\theta^*)}\left[Q^*(s,a;\theta^*) - \tilde{Q}(s,a;\theta^*)\right] + \log\frac{\sum_{a'}\exp(\tilde{Q}(s,a';\theta^*))}{\sum_{a'}\exp(Q^*(s,a';\theta^*))} .$$

[1] In the statement of Assumption 3.2 in the main paper, we only assume log-concavity for π^*. This will be corrected in a future revision to include both $\pi^*, \tilde{\pi}$

For any action a, we have

$$\begin{aligned}
Q^*(s,a;\theta^*) - \tilde{Q}(s,a;\theta^*) &= \gamma \sum_{s'} P^*(s' \mid s,a) V^*(s';\theta^*) - \gamma \sum_{s'} \tilde{P}(s' \mid s,a) \tilde{V}(s';\theta^*) \\
&= \gamma \sum_{s'} P^*(s' \mid s,a) V^*(s';\theta^*) + \gamma \sum_{s'} P^*(s' \mid s,a) \tilde{V}(s';\theta^*) \\
&\quad - \gamma \sum_{s'} P^*(s' \mid s,a) \tilde{V}(s';\theta^*) - \gamma \sum_{s'} \tilde{P}(s' \mid s,a) \tilde{V}(s';\theta^*) \\
&= \gamma \| P^*(\cdot \mid s,a) - \tilde{P}(\cdot \mid s,a) \|_1 \tilde{V}(s';\theta^*) + \gamma \sum_{s'} P^*(s' \mid s,a)(V^*(s';\theta^*) - \tilde{V}(s';\theta^*)) \\
&\leq \frac{R_{\max}}{1-\gamma} \Delta_P + \gamma \sum_{s'} P^*(s' \mid s,a) \max_{a'} \{ Q^*(s',a';\theta^*) - \tilde{Q}(s',a';\theta^*) \} \\
&\leq \dots \\
&\leq \frac{R_{\max}}{(1-\gamma)^2} \Delta_P \,.
\end{aligned}$$

Now, let us consider the normalization term. We have

$$\begin{aligned}
\log \frac{\sum_{a'} \exp(\tilde{Q}(s,a';\theta^*))}{\sum_{a'} \exp(Q^*(s,a';\theta^*))} &\leq \frac{1}{\sum_{a'} \exp(\tilde{Q}(s,a';\theta^*))} \sum_{a'} \exp(\tilde{Q}(s,a';\theta^*)) \log \frac{\exp(\tilde{Q}(s,a';\theta^*))}{\exp(Q^*(s,a';\theta^*))} \\
&\leq \sum_{a'} (\tilde{Q}(s,a';\theta^*) - Q^*(s,a';\theta^*)) \\
&\leq \frac{|\mathcal{A}| R_{\max}}{(1-\gamma)^2} \Delta_P \,.
\end{aligned}$$

Combining the two bounds and taking an expectation over s yields the desired result.

Proof of Corollary 3.2

The proof follows the format of the proof for Corollary 3.1. Fix state s. We have

$$D_{\mathrm{KL}}(\pi^*(\cdot \mid s;\theta^*) || \tilde{\pi}(\cdot \mid s;\theta^*)) = \mathbb{E}_{a \sim \pi^*(\cdot|s;\theta^*)} \left[Q^*(s,a;\theta^*) - \tilde{Q}(s,a;\theta^*) \right] + \log \frac{\sum_{a'} \exp(\tilde{Q}(s,a';\theta^*))}{\sum_{a'} \exp(Q^*(s,a';\theta^*))} \,.$$

For any action a, we have

$$\begin{aligned}
Q^*(s,a;\theta^*) - \tilde{Q}(s,a;\theta^*) &= \gamma^* \sum_{s'} P(s' \mid s,a) V^*(s';\theta^*) - \tilde{\gamma} \sum_{s'} P(s' \mid s,a) \tilde{V}(s';\theta^*) \\
&= \gamma^* \sum_{s'} P(s' \mid s,a) V^*(s';\theta^*) + \gamma^* \sum_{s'} P(s' \mid s,a) \tilde{V}(s';\theta^*) \\
&\quad - \gamma^* \sum_{s'} P(s' \mid s,a) \tilde{V}(s';\theta^*) - \tilde{\gamma} \sum_{s'} P(s' \mid s,a) \tilde{V}(s';\theta^*) \\
&= (\gamma^* - \tilde{\gamma}) \sum_{s'} P(s' \mid s,a) \tilde{V}(s';\theta^*) + \gamma^* \sum_{s'} P(s' \mid s,a) (V^*(s';\theta^*) - \tilde{V}(s';\theta^*)) \\
&\leq \frac{R_{\max}}{1-\tilde{\gamma}} |\gamma^* - \tilde{\gamma}| + \gamma^* \sum_{s'} P(s' \mid s,a) \max_{a'} \{Q^*(s',a';\theta^*) - \tilde{Q}(s',a';\theta^*)\} \\
&\leq \dots \\
&\leq \frac{R_{\max}}{(1-\tilde{\gamma})(1-\gamma^*)} |\gamma^* - \tilde{\gamma}| \ .
\end{aligned}$$

Now, let us consider the normalization term. We have

$$\begin{aligned}
\log \frac{\sum_{a'} \exp(\tilde{Q}(s,a';\theta^*))}{\sum_{a'} \exp(Q^*(s,a';\theta^*))} &\leq \frac{1}{\sum_{a'} \exp(\tilde{Q}(s,a';\theta^*))} \sum_{a'} \exp(\tilde{Q}(s,a';\theta^*)) \log \frac{\exp(\tilde{Q}(s,a';\theta^*))}{\exp(Q^*(s,a';\theta^*))} \\
&\leq \sum_{a'} (\tilde{Q}(s,a';\theta^*) - Q^*(s,a';\theta^*)) \\
&\leq \frac{|\mathcal{A}| R_{\max}}{(1-\tilde{\gamma})(1-\gamma)} |\tilde{\gamma} - \gamma^*| \ .
\end{aligned}$$

Combining the two bounds yields the desired result.

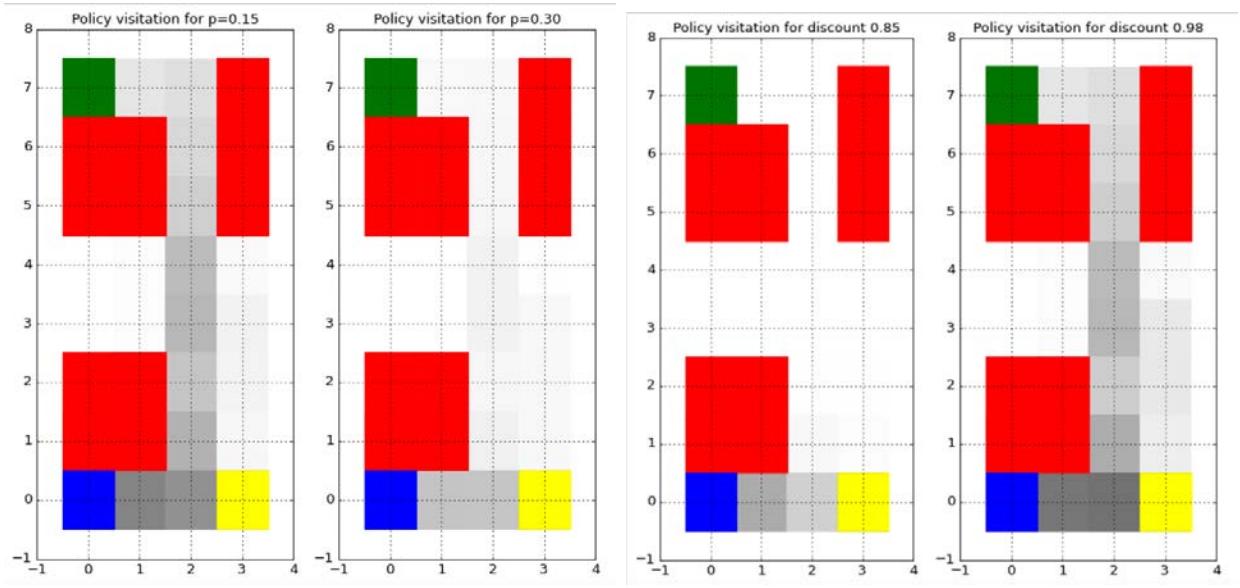


Figure B.1. Visualization of gridworld policies (as state-visitation distributions) with (a) different transition biases (probability p of unintended transitions), and (b) different discount factors.

B.2 Experiment Details

Tabular experiments

Environment and training details Recall that the gridworld environments we consider are described by 8×4 grids, with a start and goal state, and walls, lava, and exactly one waypoint state placed in between. We consider a sparse reward where the agent earns a reward of $\theta = 3$ upon reaching the goal state. Alternatively, if the agent reaches a lava or waypoint state, then its reward is 0 or 1, respectively, for the rest of the trajectory. The agent is able to move in either of the four direction (or choose to stay still), and there is a $p = 30\%$ chance that the agent travels in a different direction than commanded. We choose $\gamma^* = 0.98$ high enough that the goal state is preferred over the closer waypoint state under the optimal policy.

A reward-conditioned policy (model or demonstrator) under each environment is given by $\pi(a \mid s; \theta) \propto \exp(Q(s, a; \theta))$, where $Q(s, a; \theta)$ were derived by value iteration using the an MDP model (can be the true underlying MDP or a biased one) of the environment. During reward inference, we discretize the reward parameter space $\Theta = [1, 4]$ with resolution 64. Because the environment is tabular, instead of sampling demonstrations $\mathcal{D}$ from π, we can instead compute w^π the discounted stationary distribution. Specifically, let ρ be the distribution of the starting state (which in our case, is an indicator vector at the start state

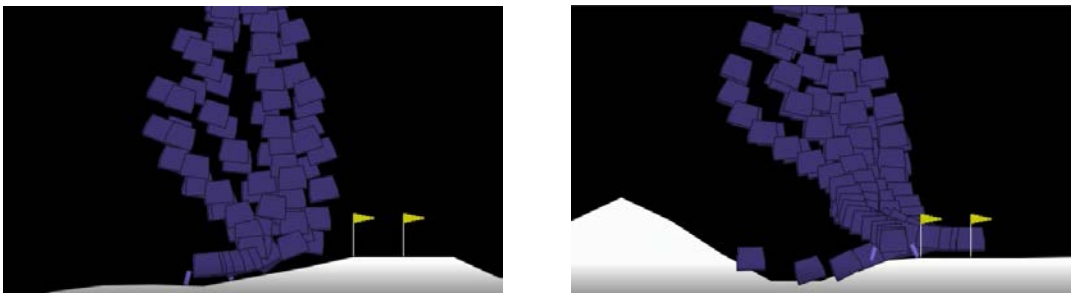


Figure B.2. Visualization of Lunar Lander trajectories for policies with (a) biased internal dynamics that underestimate left-right acceleration and (b) correct internal dynamics.

of each environment), then w^π satisfies:

$$w^\pi(s) = (1-\gamma)\rho(s) + \gamma^* \sum_{s' \in \mathcal{S}} \sum_{a' \in \mathcal{A}} w^\pi(s')\pi(a' \mid s')P(s \mid s', a') \,.$$

We can use this to solve for the true state visitations w^* for any demonstrator policy π^*, which can be used to compute the weighted policy divergence as in (3.3) without explicitly sampling a dataset $\mathcal{D}$ of demonstrations.

Visualization of biased policies In Figure B.1, we visualize the demonstrator policies π^* under the systematic biases considered. We see that in Figure B.1(a), when the demonstrator underestimates the probability of unintended transitions, it heavily prefers the goal state, which has higher reward but is much more dangerous to reach, over the waypoint state Conversely, in Figure B.1(b), when the demonstrator underestimates the discount factor, they strongly prefer the waypoint state that yields lower reward but is much closer.

Continuous control experiments

Environment and training details Recall that the domain we consider is the Lunar Lander game, where an agent needs to navigate a lander onto the landing pad. The reward function yields a large reward for landing on the pad, and a penalty for crashing or going out of bounds. The magnitude of the reward depends on the speed and tilt of the lander upon reaching the landing pad. The physics of the game are deterministic. The reward parameter $\theta \in [0, 1]$ we try to infer is the location of the landing pad (expressed as normalized horizontal displacement).

In this domain, we train a reward-conditioned policy $\pi(a \mid s; \theta)$ by folding the reward parameter θ into the state representation, which is an 8-dimensional vector capturing the lander's current location, velocity, and tilt. The policy is parameterized as a 3-layer fully-connected neural network with hidden dimension of 128, and outputs a squashed Gaussian distribution over actions. Because the state and action space are continuous, we use soft-actor-critic (SAC) [96] with fixed entropy regularization $\alpha = 1$. We train the policy for 600

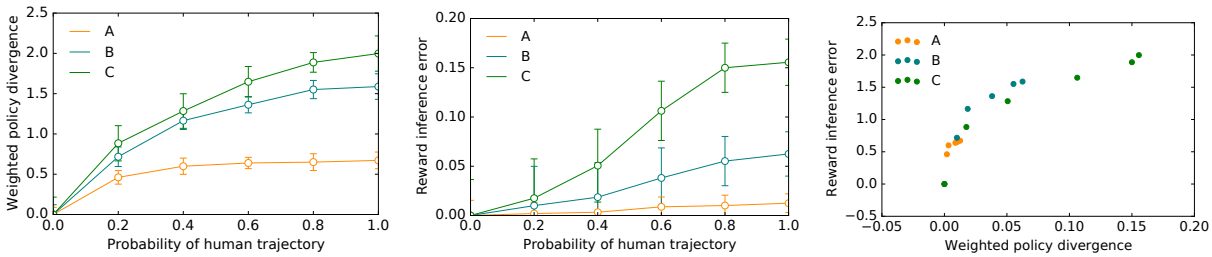


Figure B.3. Effect of human bias (measured by probability of acting according to human policy) on (a) weighted policy divergence and (b) reward inference error on discrete Lunar Lander environments. In (c), we show a scatter plot of the policy and reward errors for fixed probabilities.

episodes of length at most $1,000$, with a batch size of 264, until the policy was able to land on the landing pad with a high success rate. During reward inference, we discretize the reward parameter space $\Theta = [0, 1]$ with resolution 32. We sample datasets $\mathcal{D}$ consisting of $10,000$ observations, and report the mean and standard error of policy and reward error across 10 independent samples of datasets.

Visualization of biased policies In Figure B.2, we visualize the demonstrator policies π^* under different degrees of internal dynamics bias. Recall that parameter p describes how much one unit of power will increase acceleration in the left-right directions. When p is underestimated, the policy will not move right enough to reach the landing pad; in contrast, when p is properly estimated, the policy will reach the landing pad with a high success rate.

Additional experiment with human policies

In line with the experiments with simulated biases, we run an additional experiment similar to the one in Section 3.5, where we instead keep $\tilde{\pi}$ fixed as the optimal policy, and interpolate between the optimal policy and the real human policy for demonstrator policy π^*. We show the effect of the interpolation proportion on policy divergence and reward inference error in Figure B.3. Again, we notice the consistent message that policy error bounds reward error.

Appendix C

Deferred content from Chapter 4

C.1 Blackwell's approachability

Blackwell [29] introduced the concept of approachability as a generalization of the minimax theorem to vector-valued payoffs. Formally, a Blackwell game is an extension of two-player zero-sum games with vector-valued reward functions.

Let $\mathcal{X}, \mathcal{Y}$ denote the action spaces for the two players and $r : \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}^k$ be the corresponding vector-valued reward function. Further, let $S \subseteq \mathbb{R}^k$ denote a target set. The objective of player 1 is to ensure that the reward vector r lies in the set S while that of player 2 is ensure that the reward r lies outside this set S. Following [2], we introduce the notion of satisfiability and response-satisfiability.

Definition C.1 (Satisfiability)**.** *For a Blackwell game parameterized by $(\mathcal{X}, \mathcal{Y}, r, S)$, we say that,*

- *S is* satisfiable *if there exists $x \in \mathcal{X}$ such that for all $y \in \mathcal{Y}$, we have that $r(x, y) \in S$.*
- *S is* response-satisfiable *if for every $y \in \mathcal{Y}$, there exists $x \in \mathcal{X}$ such that $r(x, y) \in S$.*

In the case of scalar rewards, Von Neumann's minimax theorem indicates that any set which is satisfiable is also response-satsifiable. In other words, there exists a strategy for Player 1, oblivious of Player 2's strategy which ensures that the reward belongs to the set S if the set S is response-satisfiable. The existence of such a relation was crucial in obtaining the concept of the Von Neumann winner described in Section 4.2 for the uni-criterion problem.

However, such a statement fails to hold in the general vector-valued case (see [2] for a counterexample). In order to overcome this limitation, Blackwell [29] defined the notion of approachability as follows.

Definition C.2 (Blackwell's Approachability)**.** *Given a Blackwell game $(\mathcal{X}, \mathcal{Y}, r, S)$, we say that a set S is approachable if there exists an algorithm $\mathcal{A}$ which selects points in $\mathcal{X}$ such*

that for any sequence $y_1, \dots y_t \in \mathcal{Y}$,

$$\lim_{T\to\infty} \rho\left(\frac{1}{T}\sum_{t=1}^{T} r(x_t, y_t), S\right) \to 0\ ,$$

where $x_t = \mathcal{A}(y_1, \dots, y_{t-1})$ *is the algorithm's play at time* t *for some distance function* ρ.

The celebrated Blackwell's theorem then claims that any set S is approachable iff it is response-satisfiable. This means that while no single choice of action in the set $\mathcal{X}$ can guarantee a response in the set S, there exists an algorithm which ensures that in the repeated game, the average rewards approach the set S, for any choice of oppenent play.

Note that our definition of *achievability* is a stronger requirement than Blackwell's approachability. While approachability requires the time-averaged payoff in a repeated game to belong to the pre-specified set S, achievability requires the same to be true in a single-shot play of the game. Indeed, as the following lemma shows, one can construct examples of multi-criteria preference problems which are approachable but not achievable.

Proposition C.1 (Approachability does not imply achievability)**.** *There exists a preference tensor* $\mathbf{P} \in \mathcal{P}_{d,k}$ *and a target set* $S \subset [0,1]^k$ *such that*
a) For the Blackwell game given by $(\Delta_d, \Delta_d, \mathbf{P}, S)$*, the set* S *is approachable, and*
b) The set S *is not achievable with respect to* $\mathbf{P}$.

Proof. We will consider an example in a 2-dimensional action space with 2 criteria. Consider the preference matrix given by:

$$\mathbf{P}^1 = \begin{bmatrix} \frac{1}{2} & 1 \\ 0 & \frac{1}{2} \end{bmatrix} \qquad \text{and} \qquad \mathbf{P}^2 = \begin{bmatrix} \frac{1}{2} & 0 \\ 1 & \frac{1}{2} \end{bmatrix}\ , \tag{C.1}$$

along with the convex set $S = [\frac{1}{2}, 1]^2$. The tensor $\mathbf{P}$ represents the strongest possible trade-off between the two objects: Object 1 is preferred over 2 along the first criterion while the reverse is true for the second criterion.

The Blackwell game given by $(\Delta_d, \Delta_d, \mathbf{P}, S)$ can indeed be shown to be approachable. The set S is response-satisfiable since for every strategy $y \in \Delta_d$ chosen by the column player, the choice of $x = y$ would yield a reward vector $\mathbf{P}(x, y) = [\frac{1}{2}, \frac{1}{2}] \in S$. Then, by Blackwell's theorem [29], the set S is approachable.

In contrast, consider any choice of distribution $\pi_1 = [p, 1-p]$ for the multi-criteria preference problem. The corresponding score vectors for responses $i_2 = 1, 2$ are given by:

$$r_1 = \mathbf{P}(\pi_1, i_2 = 1) = \left[\frac{p}{2}, 1 - \frac{p}{2}\right] \qquad \text{and} \qquad r_2 = \mathbf{P}(\pi_1, i_2 = 2) = \left[\frac{1}{2} + \frac{p}{2}, \frac{1}{2} - \frac{p}{2}\right]\ .$$

For any choice of the parameter $p \in [0, 1]$, one cannot have both r_1 and r_2 simultaneously belong to the set S. Hence, we have that the set S is not achievable with respect to $\mathbf{P}$.

This example can be extended to any arbitrary dimension k by extending the tensor to have $\mathbf{P}^j$ equal to the all-half matrix for any criterion $j > 2$ and the target set to be $S = [\frac{1}{2}, 1]^k$. Similarly, in order to extend the example to any dimension, consider the preference tensor (for $k = 2$)

$$\mathbf{P}_d^1 = \begin{bmatrix} \mathbf{P}^1 & \mathbf{P}_{1/2} & \cdots & \mathbf{P}_{1/2} \\ \mathbf{P}_{1/2} & \mathbf{P}^1 & \cdots & \mathbf{P}_{1/2} \\ \vdots & \cdots & \ddots & \vdots \\ \mathbf{P}_{1/2} & \mathbf{P}_{1/2} & \cdots & \mathbf{P}^1 \end{bmatrix} \quad \text{and} \quad \mathbf{P}_d^2 = \begin{bmatrix} \mathbf{P}^2 & \mathbf{P}_{1/2} & \cdots & \mathbf{P}_{1/2} \\ \mathbf{P}_{1/2} & \mathbf{P}^2 & \cdots & \mathbf{P}_{1/2} \\ \vdots & \cdots & \ddots & \vdots \\ \mathbf{P}_{1/2} & \mathbf{P}_{1/2} & \cdots & \mathbf{P}^2 \end{bmatrix},$$

with the smaller matrices $\mathbf{P}^1$ and $\mathbf{P}^2$ from equation (C.1) at the diagonal and $\mathbf{P}_{1/2}$ denoting the all-half tensor of the appropriate dimension. A similar calculation as for the $d = 2$ case yields that the set S is not achievable. This establishes the required claim. □

C.2 Proof of main results

In this section, we provide formal proofs of all the results stated in the main paper. Appendix C.4 to follow collects some additional results and their proofs.

Proof of Proposition 4.1

We establish both parts of the proposition separately.

Proof of part (a)

For any weight vector $w \in \Delta_k$, consider the set

$$S_w = \left\{ r \in [0,1]^k \mid \langle w, r \rangle \geq 1/2 \right\}.$$

The set S_w is clearly convex. Indeed, for any two vectors $r_1, r_2 \in S_w$ and any scalar $\alpha \in [0,1]$, we have

$$\langle w, \alpha r_1 + (1-\alpha) r_2 \rangle = \alpha \langle w, r_1 \rangle + (1-\alpha) \langle w, r_2 \rangle \in \left[\frac{1}{2}, 1 \right].$$

It is straightforward to verify that the set S_w is also monotonic with respect to the orthant ordering.

We now show that a von Neumann winner π^* of the (single-criterion) preference matrix $\mathbf{P}_w := \mathbf{P}(w)$ can be written as $\pi(\mathbf{P}, S_w, \|\cdot\|)$ for an arbitrary choice of norm $\|\cdot\|$. For each $\widetilde{\pi} \in \Delta_d$, we have

$$\langle w, \mathbf{P}(\pi^*, \widetilde{\pi}) \rangle = \sum_{j \in [k]} w_j \mathbf{P}^j(\pi^*, \widetilde{\pi}) = \mathbf{P}_w(\pi^*, \widetilde{\pi})) \overset{\text{(i)}}{\geq} \frac{1}{2},$$

where the inequality (i) follows since π^* is a von Neumann winner for the matrix $\mathbf{P}_w$. Thus, we have the inclusion $\mathbf{P}(\pi^*, \widetilde{\pi}) \in S_w$ for all $\widetilde{\pi} \in \Delta_d$, so that $\max_{\widetilde{\pi} \in \Delta_d} \rho(\mathbf{P}(\pi^*, \widetilde{\pi}), S_w) = 0$ for any distance metric ρ. Consequently, we have

$$\pi^* \in \operatorname*{argmin}_{\pi \in \Delta_k} \max_{\widetilde{\pi} \in \Delta_d} \rho(\mathbf{P}(\pi, \widetilde{\pi}), S_w),$$

which establishes the claim for part (a). □

Proof of part (b)

Consider the multi-criteria preference instance given by target set $S = [\frac{1}{2}, 1]^k$, the ℓ_∞ distance function and the preference tensor $\mathbf{P}$

$$\mathbf{P}^1 = \begin{bmatrix} \frac{1}{2} & 1 \\ 0 & \frac{1}{2} \end{bmatrix}, \quad \mathbf{P}^2 = \begin{bmatrix} \frac{1}{2} & 0 \\ 1 & \frac{1}{2} \end{bmatrix}, \quad \text{and} \quad \mathbf{P}^j = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

The *unique* Blackwell winner for this instance $(\mathbf{P}, S, \|\cdot\|_\infty)$ is given by

$$\underbrace{\pi(\mathbf{P}, S, \|\cdot\|_\infty)}_{\pi^*} = [1/2, 1/2]\,. \tag{C.2}$$

For any weight $w \in [0,1]^k$, consider the von Neumann winners corresponding to the weighted matrices $\mathbf{P}_w$

$$\pi(\mathbf{P}_w, [1/2, 1], |\cdot|) = \begin{cases} [1,0] & \text{for } w \text{ s.t. } \mathbf{P}_w(1,2) > 0.5 \\ [0,1] & \text{for } w \text{ s.t. } \mathbf{P}_w(1,2) < 0.5\,. \\ \pi \in \Delta_2 & \text{otherwise} \end{cases} \tag{C.3}$$

Comparing equations (C.2) and (C.3) establishes the required claim. □

Proof of Theorem 4.1

Let us use the shorthand $\widetilde{\pi} := \pi(\widetilde{\mathbf{P}})$. We begin by decomposing the desired error term as

$$\begin{aligned} &\Delta_{\mathbf{P}}(\widetilde{\pi}, \pi^*) \\ &= \underbrace{v(\widetilde{\pi}; S, \mathbf{P}, \|\cdot\|) - v(\widetilde{\pi}; S, \widetilde{\mathbf{P}}, \|\cdot\|)}_{\text{Perturbation error at } \widetilde{\pi}} + \underbrace{v(\widetilde{\pi}; S, \widetilde{\mathbf{P}}, \|\cdot\|) - v(\pi^*; S, \widetilde{\mathbf{P}}, \|\cdot\|)}_{\leq 0} + \underbrace{v(\pi^*; S, \widetilde{\mathbf{P}}, \|\cdot\|) - v(\pi^*; S, \mathbf{P}, \|\cdot\|)}_{\text{Perturbation error at } \pi^*} \end{aligned}$$

In order to obtain a bound on the perturbation errors, note that for any distribution π, we have

$$\begin{aligned} v(\pi; S, \mathbf{P}, \|\cdot\|) - v(\pi; S, \widetilde{\mathbf{P}}, \|\cdot\|) &= \max_{i_1}[\rho(\mathbf{P}(\pi, i_1), S)] - \max_{i_2}[\rho(\widetilde{\mathbf{P}}(\pi, i_2), S)] \\ &\overset{\text{(i)}}{\leq} \max_{i}[\rho(\mathbf{P}(\pi, i), S) - \rho(\widetilde{\mathbf{P}}(\pi, i), S)], \end{aligned} \tag{C.4}$$

where step (i) follows by setting the i_2 equal to i_1. Noting that the distance is given by the ℓ_q norm, we have

$$\begin{aligned} v(\pi; S, \mathbf{P}, \|\cdot\|) - v(\pi; S, \widetilde{\mathbf{P}}, \|\cdot\|) &\leq \max_i [\min_{z_1 \in S} \|\mathbf{P}(\pi, i) - z_1\|_q - \min_{z_2 \in S} \|\widetilde{\mathbf{P}}(\pi, i) - z_2\|_q] \\ &\overset{\text{(i)}}{\leq} \max_i [\|\mathbf{P}(\pi, i) - \widetilde{\mathbf{P}}(\pi, i)\|_q], \end{aligned}$$

where the inequality (i) follows by setting z_2 equal to z_1. Taking a supremum over all distributions π completes the proof. □

Proof of Corollary 4.1

By Theorem 4.1, it suffices to provide a bound on the quantity $\max_i \|\mathbf{P}(\cdot, i) - \widehat{\mathbf{P}}(\cdot, i))\|_{\infty,\infty}$ for the plug-in preference tensor $\widehat{\mathbf{P}}$. Now by definition, we have

$$\max_i \|\mathbf{P}(\cdot, i) - \widehat{\mathbf{P}}(\cdot, i))\|_{\infty,\infty} = \max_{i_1, i_2, j} |\mathbf{P}^j(i_1, i_2) - \widehat{\mathbf{P}}^j(i_1, i_2)| \, .$$

For each $i = (i_1, i_2, j)$ representing some index of the tensor, let $N_i := \#\{\ell \mid \eta_\ell = i\}$ denote the number of samples observed at that index. Since N_i can be written as a sum of i.i.d. Bernoulli random variables, applying the Hoeffding bound yields

$$\Pr\left\{ \left| N_i - \frac{n}{d^2 k} \right| \geq c \sqrt{\frac{n \log(c/\delta)}{d^2 k}} \right\} \leq \delta \text{ for each } \delta \in (0, 1).$$

Note that we also have $n \geq c_0 d^2 k \log(c_1 d/\delta)$ by assumption. For a large enough choice of the constants (c_0, c_1), applying the union bound yields the sequence of sandwich relations

$$\frac{n}{2d^2 k} \leq N_i \leq \frac{3n}{2d^2 k} \quad \text{for all indices } i \text{ with probability greater than } 1 - \delta. \tag{C.5}$$

Furthermore, conditioned on N_i (for $i = (i_1, i_2, j)$), the Hoeffding bound yields the relation

$$\Pr\left\{ |\mathbf{P}^j(i_1, i_2) - \widehat{\mathbf{P}}^j(i_1, i_2)| \geq c \sqrt{\frac{\log(c/\delta)}{N_i}} \right\} \leq \delta \text{ for each } \delta \in (0, 1).$$

Putting this together with a union bound, we have

$$\Pr\left\{ \max_{i_1, i_2, j} |\mathbf{P}^j(i_1, i_2) - \widehat{\mathbf{P}}^j(i_1, i_2)| \geq c \sqrt{\frac{\log(cd^2 k/\delta)}{\min_i N_i}} \right\} \leq \delta. \tag{C.6}$$

Combining inequalities (C.5) and (C.6) with a final union bound completes the proof. □

Proof of Theorem 4.2

Suppose throughout that $k \geq 2$, and recall the axis-aligned convex target set $S_0 = [\frac{1}{2}, 1]^k$. We split our proof into two cases depending on whether d is even or odd.

Case 1: d **even.** We use Le Cam's method and construct two problem instances with preference tensors given by $\mathbf{P}_0$ and $\mathbf{P}_1$. Two key elements in the construction are the following $2 \times 2 \times 2$ tensors, which we denote by $\mathbf{P}_{\mathsf{cr}}$ and $\widetilde{\mathbf{P}_{\mathsf{cr}}}$, respectively. Their entries are given by

$$\mathbf{P}_{\mathsf{cr}}^1 = \begin{bmatrix} \frac{1}{2} & \frac{1}{2}+\gamma \\ \frac{1}{2}-\gamma & \frac{1}{2} \end{bmatrix} \quad , \quad \mathbf{P}_{\mathsf{cr}}^2 = \begin{bmatrix} \frac{1}{2} & \frac{1}{2}-\gamma \\ \frac{1}{2}+\gamma & \frac{1}{2} \end{bmatrix},$$
$$\widetilde{\mathbf{P}_{\mathsf{cr}}}^1 = \begin{bmatrix} \frac{1}{2} & \frac{1}{2}+\frac{\gamma}{d} \\ \frac{1}{2}-\frac{\gamma}{d} & \frac{1}{2} \end{bmatrix} \quad \text{and} \quad \widetilde{\mathbf{P}_{\mathsf{cr}}}^2 = \begin{bmatrix} \frac{1}{2} & \frac{1}{2}-\frac{\gamma}{d} \\ \frac{1}{2}+\frac{\gamma}{d} & \frac{1}{2} \end{bmatrix}.$$

Note that these tensors are parameterized by a scalar $\gamma \in [0, 1/2]$, whose exact value we specify shortly. Also denote by $\mathbf{P}_{1/2}$ the $2 \times 2 \times 2$ all-half tensor. We are now ready to construct the pair of $d \times d \times k$ preference tensors $(\mathbf{P}_0, \mathbf{P}_1)$.

In order to construct tensor $\mathbf{P}_0$, we specify its entries on the first two criteria according to

$$\mathbf{P}_0^{1:2} = \begin{bmatrix} \mathbf{P}_{1/2} & \mathbf{P}_{1/2} & \cdots & \mathbf{P}_{1/2} \\ \mathbf{P}_{1/2} & \mathbf{P}_{\mathsf{cr}} & \cdots & \mathbf{P}_{1/2} \\ \vdots & \cdots & \ddots & \vdots \\ \mathbf{P}_{1/2} & \mathbf{P}_{1/2} & \cdots & \mathbf{P}_{\mathsf{cr}} \end{bmatrix}, \tag{C.7}$$

and set the entries on the remaining $k-2$ criteria to $1/2$.

On the other hand, the first two criteria of the tensor $\mathbf{P}_1$ are given by

$$\mathbf{P}_1^{1:2} = \begin{bmatrix} \widetilde{\mathbf{P}_{\mathsf{cr}}} & \mathbf{P}_{1/2} & \cdots & \mathbf{P}_{1/2} \\ \mathbf{P}_{1/2} & \mathbf{P}_{\mathsf{cr}} & \cdots & \mathbf{P}_{1/2} \\ \vdots & \cdots & \ddots & \vdots \\ \mathbf{P}_{1/2} & \mathbf{P}_{1/2} & \cdots & \mathbf{P}_{\mathsf{cr}} \end{bmatrix}, \tag{C.8}$$

with the entries on the remaining $k-2$ criteria once again set identically to $1/2$.

Note that the tensors $\mathbf{P}_0$ and $\mathbf{P}_1$ only differ on the first $2 \times 2 \times 2$ block. Furthermore, the following lemma provides an exact calculation of the values $\min_\pi v(\pi; \mathbf{P}_0, S_0, \|\cdot\|_\infty)$ and $\min_\pi v(\pi; \mathbf{P}_1, S_0, \|\cdot\|_\infty)$.

Lemma C.1. *We have*

$$\mathcal{V}_0 := \min_\pi v(\pi; \mathbf{P}_0, S_0, \|\cdot\|_\infty) = 0 \quad \textit{and} \quad \mathcal{V}_1 := \min_\pi v(\pi; \mathbf{P}_0, S_0, \|\cdot\|_\infty) = \frac{\gamma}{3d-2}.$$

Given samples from these two instances, we now use Le Cam's lemma [see 198, Chap 2] to lower bound the minimax risk as

$$\mathfrak{M}_{n,d,k}(S_0, \|\cdot\|_\infty) \geq \frac{|\mathcal{V}_0 - \mathcal{V}_1|}{2}\left(1 - \|\mathbb{P}_0^n - \mathbb{P}_1^n\|_{\text{TV}}\right) = \frac{\gamma}{2(3d-2)}\left(1 - \|\mathbb{P}_0^n - \mathbb{P}_1^n\|_{\text{TV}}\right), \quad \text{(C.9)}$$

where $\mathbb{P}_0^n$ and $\mathbb{P}_1^n$ are the probability distributions induced on sample space by the passive sampling strategy applied to the tensor $\mathbf{P}_0$ and $\mathbf{P}_1$, respectively.

Using Pinsker's inequality, the decoupling property for KL divergence and the fact that that $\text{KL}(P\|Q) \leq \chi^2(P\|Q)$, we have

$$\|\mathbb{P}_0^n - \mathbb{P}_1^n\|_{\text{TV}} \leq \sqrt{\frac{n}{2}\text{KL}(\mathbb{P}_1\|\mathbb{P}_0)} \leq \sqrt{\frac{n}{2}\chi^2(\mathbb{P}_1\|\mathbb{P}_0)}\,. \quad \text{(C.10)}$$

The chi-squared distance between the two distributions $\mathbb{P}_0$ and $\mathbb{P}_1$ is given by

$$\chi^2(\mathbb{P}_1\|\mathbb{P}_0) = \frac{1}{d^2k}\sum_{(i_1,i_2,j)}\left(\frac{\mathbf{P}_1^j(i_1,i_2)}{\mathbf{P}_2^j(i_1,i_2)} - 1\right)^2 \overset{\text{(i)}}{=} \frac{2}{d^2k}\left(\left(\frac{2\gamma}{d}\right)^2 + \left(-\frac{2\gamma}{d}\right)^2\right) = \frac{16\gamma^2}{d^4k}\,,$$

where step (i) follows from the fact that $\mathbf{P}_1$ and $\mathbf{P}_2$ differ only in 4 entries and that the passive sampling strategy samples each index uniformly at random. Putting together the pieces, we have:

$$\mathfrak{M}_{n,d,k}(S_0, \|\cdot\|_\infty) \geq \frac{\gamma}{2(3d-2)}\left(1 - \sqrt{\frac{n}{2}\,\frac{16\gamma^2}{d^4k}}\right) \overset{\text{(ii)}}{=} \frac{1}{48\sqrt{2}}\sqrt{\frac{d^2k}{n}}.$$

where step (ii) follows by setting $\gamma^2 = \frac{d^4k}{32n}$ and using the fact that $3d - 2 \leq 3d$. Note that since we require $\gamma^2 \leq \frac{1}{4}$, the above bound is valid only for $n \gtrsim d^4k$. This concludes the proof for even d.

Case 2: d odd. By assumption, we have $d \geq 5$. In this case, we construct $\mathbf{P}_0$ and $\mathbf{P}_1$ exactly as before, but replace $\mathbf{P}_{\mathsf{cr}}$ in the last two rows of both $\mathbf{P}_0$ and $\mathbf{P}_1$ with the following modified $3 \times 3 \times 2$ tensor:

$$\mathbf{P}_{\mathsf{cr},3}^1 = \begin{bmatrix} \frac{1}{2} & \frac{1}{2}+\gamma & \frac{1}{2}-\gamma \\ \frac{1}{2}-\gamma & \frac{1}{2} & \frac{1}{2}-\gamma \\ \frac{1}{2}+\gamma & \frac{1}{2}+\gamma & \frac{1}{2} \end{bmatrix} \quad \text{and} \quad \mathbf{P}_{\mathsf{cr},3}^2 = \begin{bmatrix} \frac{1}{2} & \frac{1}{2}-\gamma & \frac{1}{2}+\gamma \\ \frac{1}{2}+\gamma & \frac{1}{2} & \frac{1}{2}+\gamma \\ \frac{1}{2}-\gamma & \frac{1}{2}-\gamma & \frac{1}{2} \end{bmatrix}.$$

By mimicking its proof, it can be verified that this modification ensures that the corresponding values $\mathcal{V}_0$ and $\mathcal{V}_1$ still satisfy Lemma C.1. Thus, the lower bound remains unchanged up to constant factors. □

Proof of Lemma C.1

Let us compute the two values separately.

Computing $\mathcal{V}_0$. The choice of distribution $\pi^* = [1, 0, \ldots, 0]$ yields the score vector $[1/2, 1/2, \ldots, 1/2]$, which is in the set S_0. Thus, we have $\mathcal{V}_0 = 0$.

Computing $\mathcal{V}_1$. Note that the optimal distribution π^* achieving the value $\mathcal{V}_1$ will be of the form

$$\pi^* = [p/2, p/2, (1-p)/(d-2), \ldots, (1-p)/(d-2)] \text{ for some } p \in [0, 1].$$

This follows from the symmetry in the preference tensor for row objects ranging from 3 to d. Given such a distribution π^*, the distance of the reward vector from the set S_0 is given by

$$\inf_{z \in S} \|\mathbf{P}(\pi^*, i_2) - z\|_\infty = \begin{cases} \frac{\gamma p}{2d} & i_2 = 1, 2 \\ \frac{\gamma(1-p)}{d-2} & \text{o.w.} \end{cases} .$$

Thus, for any value of $p > 2d/(3d-2)$, the distance is maximized for $i_2 \in \{1, 2\}$, and yields a value $\gamma p/(2d)$. On the other hand, for $p < 2d/(3d-2)$, the maximizing index is $i_2 \geq 3$, and the maximizing value is $\gamma(1-p)/(d-2)$. Optimizing these values for p yields the claim. □

Instance dependent lower bounds

In this section, we give a formal statement of Proposition 4.2 along with its proof.

We begin by defining some notation. For any $\alpha, \beta \in [-\frac{1}{2}, \frac{1}{2}]$ and choice of criteria $j_1, j_2 \in [k]$, we define the tensor $\mathbf{P}^{(j_1,j_2)}_{\alpha,\beta} \in [0, 1]^{2\times 2\times k}$ as

$$\mathbf{P}^{j_1}_{\alpha,\beta} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2}+\alpha \\ \frac{1}{2}-\alpha & \frac{1}{2} \end{bmatrix}, \quad \mathbf{P}^{j_2}_{\alpha,\beta} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2}+\beta \\ \frac{1}{2}-\beta & \frac{1}{2} \end{bmatrix} \quad \text{and} \quad \mathbf{P}^{j}_{\alpha,\beta} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix} \text{ for } j \neq \{j_1, j_2\} .$$

Further, we denote by $\mathbf{P}_{1/2}$ the all-half tensor whose dimensions may vary depending on the context. Any distribution π over the two objects can be parameterized by a value $q \in [0, 1]$ with q being the probability placed on the first object and $1-q$ the probability on the second object. We will consider the distance function given by the ℓ_∞ norm. Given this distance function, we overload our notation for the value

$$v(q; \mathbf{P}^{(j_1,j_2)}_{\alpha,\beta}, S) = \max_i [\rho(\mathbf{P}^{(j_1,j_2)}_{\alpha,\beta}(q, i), S)] \quad \text{and} \quad \mathcal{V}(\mathbf{P}^{(j_1,j_2)}_{\alpha,\beta}; S) = \min_q v(q; \mathbf{P}^{(j_1,j_2)}_{\alpha,\beta}; S) . \tag{C.11}$$

We now state our main assumption for the score set S which allows us to formulate our lower bound.

Assumption C.1. *There exists a pair of criteria (j_1, j_2), values $\alpha_0 \in (0, \frac{1}{2}]$ and $\beta_0 \in [-\frac{1}{2}, 0]$, and a gap parameter $\gamma > 0$ such that*

$$\mathcal{V}(\mathbf{P}_{1/2}; S) + \gamma \leq \mathcal{V}(\mathbf{P}^{(j_1,j_2)}_{\alpha_0,\beta_0}; S)$$

for the all-half tensor $\mathbf{P}_{1/2} \in [0,1]^{2\times 2\times k}$.

The assumption above indicates that there exists a pair of criteria along which one can observe some sort of trade-off when they interact with the underlying score set S. The preference tensor $\mathbf{P}^{(j_1,j_2)}_{\alpha_0,\beta_0}$ captures this trade-off and the gap parameter γ quantifies it. Going forward, we assume without loss of generality that $(j_1, j_2) = (1, 2)$ and drop the dependence of the tensor on these indices, writing $\mathbf{P}_{\alpha_0,\beta_0} \equiv \mathbf{P}^{(1,2)}_{\alpha_0,\beta_0}$. The following lemma indicates the importance of the special values of $(\alpha, \beta) = (0, 0)$ for which $\mathbf{P}_{0,0} = \mathbf{P}_{1/2}$.

Lemma C.2. *For any $\alpha, \beta \in [-\frac{1}{2}, \frac{1}{2}]$, we have $\mathcal{V}(\mathbf{P}_{0,0}; S) \leq \mathcal{V}(\mathbf{P}_{\alpha,\beta}; S)$.*

The above lemma establishes that for any set, the value attained by setting $(\alpha_0, \beta) = (0, 0)$ will be lower than any other setting of the same parameters. For any parameter $\delta \in [0, 1]$, denote by $\mathbf{P}_{\mathsf{wt},\delta}$ the weighted tensor

$$\mathbf{P}_{\mathsf{wt},\delta} := (1 - \delta)\mathbf{P}_{0,0} + \delta\mathbf{P}_{\alpha_0,\beta_0}.$$

In order to understand the value $\mathcal{V}(\mathbf{P}_{\mathsf{wt},\delta}; S)$, we establish the following structural lemma which gives us insight into how this value varies as a function of the parameter $\delta \in [0, 1]$.

Lemma C.3. *Consider a target set S that is given by an intersection of h half-spaces. Then, the value function $\mathcal{V}(\mathbf{P}_{\mathsf{wt},\delta}; S)$ is a piece-wise linear and continuous function of $\delta \in [0, 1]$ with at most $4h$ pieces.*

The above lemma states that the value $\mathcal{V}(\mathbf{P}_{\mathsf{wt},\delta}; S)$ is a piece-wise linear function of δ. Consider the first such piece which has a non-zero slope. Such a line has to exist since $\mathcal{V}(\mathbf{P}_{\mathsf{wt},\delta})$ is continuous in δ and we have $\mathcal{V}(\mathbf{P}_{\mathsf{wt},0}) < \mathcal{V}(\mathbf{P}_{\mathsf{wt},1})$. Also, this slope has to be positive since we know from Lemma C.2 that $\mathcal{V}(\mathbf{P}_{\mathsf{wt},0}) \leq \mathcal{V}(\mathbf{P}_{\mathsf{wt},\delta})$ for any $\delta \in [0, 1]$. Denote the starting point of this line by δ_0 and the corresponding slope by m_0, and observe that the value $\mathcal{V}(\mathbf{P}_{\mathsf{wt},\delta_0}) = \mathcal{V}(\mathbf{P}_{\mathsf{wt},0})$. With this notation, we now proceed to prove the lower bound on sample complexity for any polyhedral target score set S.

Proposition C.2 (Formal)**.** *Suppose that we have a valid polyhedral target set S satisfying Assumption C.1 with parameters (α_0, β_0). Then, there exists a universal constant c such that for all $d \geq 4$, $k \geq 2$, and $n \geq \frac{d^2k}{\delta^2}\frac{(1/2-\delta_0\alpha_0)^2}{\alpha_0^2+\beta_0^2}$, we have*

$$\mathfrak{M}_{n,d,k}(S, \|\cdot\|_\infty) \geq c\frac{m_0(\frac{1}{2} - \delta_0\alpha_0)}{\sqrt{\alpha_0^2 + \beta_0^2}}\sqrt{\frac{d^2k}{n}}\,. \tag{C.12}$$

Proof. For this proof, we focus on the case when the number of criteria k is even. The proof for the case when k is odd can be obtained similar to the proof of Theorem 4.2.

We use Le Cam's method for obtaining a lower bound on the minimax value and construct the lower bound instances using the tensor given by $\mathbf{P}_{\mathsf{wt},\delta}$. For some $\delta \in [0,1]$ (to be fixed later), consider the parameter $\delta_1 = \delta_0 + \delta$. Using these values of δ_0 and δ_1, we create the following two instances $\mathbf{P}_0$ and $\mathbf{P}_1$:

$$\mathbf{P}_0 = \begin{bmatrix} \mathbf{P}_{\mathsf{wt},\delta_0} & \mathbf{P}_{1/2} & \cdots & \mathbf{P}_{1/2} \\ \mathbf{P}_{1/2} & \mathbf{P}_{\alpha_0,\beta_0} & \cdots & \mathbf{P}_{1/2} \\ \vdots & \cdots & \ddots & \vdots \\ \mathbf{P}_{1/2} & \mathbf{P}_{1/2} & \cdots & \mathbf{P}_{\alpha_0,\beta_0} \end{bmatrix} \quad \text{and} \quad \mathbf{P}_1 = \begin{bmatrix} \mathbf{P}_{\mathsf{wt},\delta_1} & \mathbf{P}_{1/2} & \cdots & \mathbf{P}_{1/2} \\ \mathbf{P}_{1/2} & \mathbf{P}_{\alpha_0,\beta_0} & \cdots & \mathbf{P}_{1/2} \\ \vdots & \cdots & \ddots & \vdots \\ \mathbf{P}_{1/2} & \mathbf{P}_{1/2} & \cdots & \mathbf{P}_{\alpha_0,\beta_0} \end{bmatrix},$$

where $\mathbf{P}_{\alpha_0,\beta_0}$ is as given by Assumption C.1. The following lemma now shows that that there exists a small enough $\bar{\delta}$ such that the value function $\mathcal{V}(\mathbf{P}_{\mathsf{wt},\delta}; S)$ is linear in the range $\delta \in [\delta_0, \delta_1]$.

Lemma C.4. *There exists a $\bar{\delta} \in (0,1)$ such that for all $\delta \in [0, \bar{\delta}]$ and $\delta_1 = \delta_0 + \delta$, we have*

a. The value $\mathcal{V}(\mathbf{P}_{\mathsf{wt},\delta_1}; S) = \mathcal{V}(\mathbf{P}_{\mathsf{wt},\delta_0}; S) + \delta m_0$.

b. The minimizer π_1^* for $\mathbf{P}_1^*$ is given by $\pi_1^* = [q_0, 1-q_0, 0 \ldots, 0]$.

We defer the proof of this lemma to the end of the section. Thus, for a small enough value of $\delta \in [0, \bar{\delta}]$, we have $|\mathcal{V}(\mathbf{P}_0) - \mathcal{V}(\mathbf{P}_1)| = \delta m_0$. As was shown in the proof of Theorem 4.2, the minimax rate is lower bounded as

$$\mathfrak{M}_{n,d,k}(S, \|\cdot\|_\infty) \geq \frac{|\mathcal{V}(\mathbf{P}_0) - \mathcal{V}(\mathbf{P}_1)|}{2} (1 - \|\mathbb{P}_0^n - \mathbb{P}_1^n\|_{\mathrm{TV}}) \geq \frac{\delta m_0}{2} \left(1 - \sqrt{\frac{n}{2} \chi^2(\mathbb{P}_1 \| \mathbb{P}_0)}\right), \tag{C.13}$$

where $\mathbb{P}_0^n$ and $\mathbb{P}_1^n$ are the probability distributions induced on sample space by the passive sampling strategy and the preference tensor $\mathbf{P}_0$ and $\mathbf{P}_1$ respectively. In order to obtain the requisite lower bound, we proceed to compute an upper bound on the chi-squared distance between the two distributions $\mathbb{P}_0$ and $\mathbb{P}_1$ as

$$\begin{aligned} \chi^2(\mathbb{P}_1 \| \mathbb{P}_0) &= \frac{1}{d^2 k} \sum_{(i_1,i_2,j)} \left(\frac{\mathbf{P}_1^j(i_1,i_2)}{\mathbf{P}_0^j(i_1,i_2)} - 1 \right)^2 \\ &\overset{\text{(i)}}{\leq} \frac{2}{d^2 k} \left(\left(\frac{\alpha_0^2 \delta^2}{(\frac{1}{2} - \delta_0 \alpha_0)^2} \right) + \left(\frac{\beta_0^2 \delta^2}{(\frac{1}{2} + \delta_0 \beta_0)^2} \right) \right) \\ &\overset{\text{(ii)}}{\leq} \frac{2\delta^2}{d^2 k} \left(\frac{\alpha_0^2 + \beta_0^2}{(\frac{1}{2} - \delta_0 \alpha_0)^2} \right), \end{aligned}$$

where (i) follows from the fact that the instances $\mathbf{P}_0$ and $\mathbf{P}_1$ differ only in 4 entries and (ii) follows from the assumption that $|\alpha_0| \geq |\beta_0|$. Now, substituting the value of $\delta^2 = \frac{d^2k}{4n} \cdot \frac{(\frac{1}{2}-\delta_0\alpha_0)^2}{\alpha_0^2+\beta_0^2}$ and using the above bound with equation (C.13), we have

$$\mathfrak{M}_{n,d,k}(S, \|\cdot\|_\infty) \geq \frac{m_0(\frac{1}{2}-\delta_0\alpha_0)}{8\sqrt{\alpha_0^2+\beta_0^2}}\sqrt{\frac{d^2k}{n}} \ ,$$

which holds whenever we have $\delta \in [0, \bar{\delta}]$ or equivalently $n \geq \frac{d^2k}{4\bar{\delta}^2}\frac{(\frac{1}{2}-\delta_0\alpha_0)^2}{\alpha_0^2+\beta_0^2}$. This establishes the desired claim. □

Proof of Lemma C.2

For any $\alpha, \beta \in [-\frac{1}{2}, \frac{1}{2}]$, consider the value

$$\begin{aligned}\mathcal{V}(\mathbf{P}_{\alpha,\beta}; S) &= \min_{q\in[0,1]} \max_i [\rho(\mathbf{P}_{\alpha,\beta}(q,i), S)] \\ &= \min_{q\in[0,1]} \max_{\tau\in[0,1]} [\rho(\mathbf{P}_{\alpha,\beta}(q,\tau), S)] \\ &\overset{\text{(i)}}{\geq} \rho\left(\left[\frac{1}{2}\right]^k, S\right) = \mathcal{V}(\mathbf{P}_{1/2}; S) \ ,\end{aligned}$$

where (i) follows by setting $\tau = q$ and $\left[\frac{1}{2}\right]^k$ denotes the vector with each entry set to half. This establishes the claim. □

Proof of Lemma C.3

Let us denote by q_0 any minimizer of the value $v(q; \mathbf{P}_{\alpha_0,\beta_0}, S)$ and the two score vectors corresponding to the choices for i in equation (C.11) by $z_{1,i} := \mathbf{P}_{\alpha_0,\beta_0}(q_0, i)$. Observe that for $\mathbf{P}_{\mathsf{wt},\delta}$, the distribution given by q_0 is still a minimizer of its value. Further, the score vectors for the two column choices are given by:

$$z_{\delta,i} = (1-\delta)\left[\frac{1}{2}\right]^k + \delta z_{1,i} \quad \text{for } i = \{1, 2\}.$$

Recall that the distance function is given by $\rho(z_{\delta,i}, S) = \min_{z\in S} \|z_{\delta,i} - z\|_\infty$. Now, the minimizer z will lie on the closest hyperplane(s) to the point $z_{\delta,i}$. In order to establish the claim, it suffices to show that for any fixed hyperplane[1] H, the distance function given by $\rho(z_{\delta,i}, H)$ is a piece-wise linear function for $\delta \in [0, 1]$.

Let us consider a point $z_{\delta,i}$ which does not belong to the half-space given by H, since otherwise, the distance to the half-space is 0. If we have $\rho(z_{\delta,i}, H) = \zeta$, then the vector

[1] we use the hyperplane H and the half-space induced by it interchangeably.

$z_{\delta,i} + \zeta 1_k$ must lie on the hyperplane H. This follows from the monotonicity property of the hyperplane H.

For any $\delta = \frac{1}{2}\delta_1 + \frac{1}{2}\delta_2$ such that $z_{\delta_1,i}$ and $z_{\delta_2,i}$ do not belong to the half-space given by H, we have

$$\rho(z_{\delta,i}) = \frac{1}{2}\underbrace{\rho(z_{\delta_1,i})}_{\zeta_1} + \frac{1}{2}\underbrace{\rho(z_{\delta_2,i})}_{\zeta_2},$$

where the above equality follows since $z_{\delta_1,i} + \zeta_1 1_k$ and $z_{\delta_2,i} + \zeta_2 1_k$ both lie on the hyperplane H and therefore $z_{\delta,i} + \frac{\zeta_1+\zeta_2}{2} 1_k$ also lies on the hyperplane. Combined with the fact that for any point $z_{\delta,i}$ which lies in the half-space given by H, the distance $\rho(z_{\delta,i}, H) = 0$, we have that the function $\rho(z_{\delta,i}, H)$ is a piece-wise linear function with at most 2 linear pieces for $\delta \in [0, 1]$.

Since $\rho(z_{\delta,i}, S)$ is a minimum over h hyperplanes, this function is itself a piece-wise linear function with at most $2h$ pieces. The desired claim now follows from noting that the value function $\mathcal{V}(\mathbf{P}_{\mathsf{wt},\delta}; S)$is a maximum over two piece-wise linear functions each with at most $2h$ pieces. □

Proof of Lemma C.4

Consider $\delta_1 = \delta_0 + \delta$ such that δ_0 and δ_1 share the same linear piece. This can be guaranteed to hold true for all $\delta \leq \bar{\delta}_1$ by the piecewise linear nature of the value $\mathcal{V}(\mathbf{P}_{\mathsf{wt},\delta})$.

For part (b) of the claim, let us consider the tensor $\tilde{\mathbf{P}} = \mathbf{P}_1(3:, 3:)$ formed by removing the first two rows and columns from the tensor $\mathbf{P}_1$. Then, from Assumption C.1, we have that $\mathcal{V}(\tilde{\mathbf{P}}; S) \geq \mathcal{V}(\mathbf{P}_{1/2}; S) + \tilde{\gamma}$ for some $\tilde{\gamma} > 0$. Selecting a value of $\bar{\delta}_2$ such that $\bar{\delta}_2 m_0 \leq \tilde{\gamma}$, we can ensure that condition (b.) is satisfied.

Finally, setting $\bar{\delta} = \min(\bar{\delta}_1, \bar{\delta}_2)$ completes the proof. □

Proof of Theorem 4.3

Let us prove the two claims of the theorem separately. We use the shorthand $v(\pi) := v(\pi; \mathbf{P}, S, \|\cdot\|)$ for convenience.

Establishing convexity. Consider any two distributions $\pi_1, \pi_2 \in \Delta_k$ and a scalar $\alpha \in [0, 1]$. Since the set S is closed and convex, we have

$$\begin{aligned}
v(\alpha\pi_1 + (1-\alpha)\pi_2) &= \max_{i\in[d]} \min_{z\in S} \left[\rho(\mathbf{P}(\alpha\pi_1 + (1-\alpha)\pi_2, i), z)\right] \\
&\overset{\text{(i)}}{=} \max_{i\in[d]} \min_{z_1,z_2\in S} \left[\rho(\alpha\mathbf{P}(\pi_1, i) + (1-\alpha)\mathbf{P}(\pi_2, i), \alpha z_1 + (1-\alpha)z_2)\right] \\
&\overset{\text{(ii)}}{\leq} \max_{i\in[d]} \left(\alpha \cdot \min_{z_1\in S} \left[\rho(\mathbf{P}(\pi_1, i), z_1)\right] + (1-\alpha) \cdot \min_{z_2\in S} \left[\rho(\mathbf{P}(\pi_2, i), z_2)\right]\right) \\
&\leq \alpha v(\pi_1) + (1-\alpha) v(\pi_2),
\end{aligned}$$

where (i) follows from the convexity of S and linearity of the preference evaluation (Eq. (4.2)), (ii) follows from the convexity of the distance function given by ℓ_q norm and (iii) follows from distributing the max over the two terms. This establishes the first part of the theorem.

Establishing the Lipschitz bound. Consider any two distributions $\pi_1, \pi_2 \in \Delta_d$. The difference in their value function can then be upper bounded as

$$\begin{aligned}
|v(\pi_1) - v(\pi_2)| &= |\max_{i_1 \in [d]}[\rho(\mathbf{P}(\pi_1, i_1), S)] - \max_{i_2 \in [d]}[\rho(\mathbf{P}(\pi_2, i_2), S)]| \\
&\overset{\text{(i)}}{\leq} \max_{i \in [d]} |\rho(\mathbf{P}(\pi_1, i), S) - \rho(\mathbf{P}(\pi_2, i), S)| \\
&= \max_{i \in [d]} |\min_{z_1 \in S} \rho(\mathbf{P}(\pi_1, i), z_1) - \min_{z_2 \in S} \rho(\mathbf{P}(\pi_2, i), z_2)| \\
&\overset{\text{(ii)}}{\leq} \max_{i \in [d]} \max_{z \in S} |\rho(\mathbf{P}(\pi_1, i), z) - \rho(\mathbf{P}(\pi_2, i), z)| \ ,
\end{aligned}$$

where (i) follows from using the inequality $|\max_x f(x) - \max_y g(y)| \leq \max_x |f(x) - g(x)|$ and (ii) follows through a similar inequality $|\min_x f(x) - \min_y g(y)| \leq \max_x |f(x) - g(x)|$. Since the distance function ρ is specified by the ℓ_q norm $\|\cdot\|_q$, we have

$$\begin{aligned}
|v(\pi_1) - v(\pi_2)| &\leq \max_{i \in [d]} \|\mathbf{P}(\pi_1, i) - \mathbf{P}(\pi_2, i)\|_q \\
&= \left[\sum_{j=1}^{k} \left(\langle \pi_1 - \pi_2, \mathbf{P}^j(\cdot, i)\rangle\right)^q\right]^{\frac{1}{q}} \\
&\overset{\text{(i)}}{\leq} k^{\frac{1}{q}} \cdot \|\pi_1 - \pi_2\|_1 \ ,
\end{aligned}$$

where (i) follows from an application of Hölder's inequality ($\ell_1 - \ell_\infty$) to the inner product $\langle \pi_1 - \pi_2, \mathbf{P}^j(\cdot, i)\rangle$ and the fact that $\mathbf{P}^j(i_1, i_2) \in [0, 1]$ for any (i_1, i_2, j). This establishes the Lipschitz bound and concludes the proof of the theorem. □

C.3 Local asymptotic analysis for plug-in estimator

In this section, we study the adaptivity properties of the plug-in estimator[2] $\widehat{\pi}_{\mathsf{plug}}$ and derive upper bounds on the error $\Delta_{\mathbf{P}}(\widehat{\pi}_{\mathsf{plug}}, \pi^*)$ which depend on the properties of the underlying problem instance $(\mathbf{P}, S, \rho)$. Contrast this analysis with the upper bounds obtained in Corollary 4.1 and the perturbation result of Dudik et al. [71, Lemma 3] which provides a worst-case upper bound on the error $\Delta_{\mathbf{P}}$ independent of the underlying preference tensor $\mathbf{P}$.

Our focus in this section will be on the uni-criterion setup with $k = 1$ with the target set $S = [\frac{1}{2}, 1]$ in which case the Blackwell winner coincides with the von Neumann winner. Recall

[2]For this section we use the notation $\widehat{\pi}_{\mathsf{plug}}$ and $\widehat{\pi}$ to interchangeably to denote the plug-in estiamtor.

from Section 4.2 that for the uni-criterion setup, the von Neumann winner for a preference matrix $\mathbf{P} \in [0,1]^{d \times d}$ is defined to be the distribution π^* satisfying

$$\pi^* \in \operatorname*{argmax}_{\pi \in \Delta_d} \min_{i \in [d]} \pi^\top \mathbf{P} e_i \ , \tag{C.14}$$

where e_i denotes the basis vector in the i^{th} direction. Observe that the vector π^* corresponds to the mixed Nash equilibirum (NE) strategy of the zero-sum game with pay-off matrix $\mathbf{P}$ for the row player (maximizing player). Given this equivalence, we focus on the more general problem of estimating the Nash distribution of a zero-sum game with pay-offs $\mathbf{A} \in [0,1]^{d \times d}$ given sampled access to the matrix $\mathbf{A}$.

We consider a slighlty modified passive sampling regime introduced in Section 4.3 wherein each sample consists of an observation $y \sim \mathcal{N}(\mathbf{A}_{i_1,i_2}, \sigma^2_{i_1,i_2})$, where the indices $i_1, i_2 \sim \mathsf{Unif}([d])$ are sampled independently. We term this the *Gaussian passive sampling model* in contrast to the Bernoulli sampling model considered in the main text. Note that in the asymptotic regime, the Bernoulli sampling model is equivalent to the Gaussian sampling model with variance $\sigma^2_{i_1,i_2} = \mathbf{A}_{i_1,i_2} \cdot (1 - \mathbf{A}_{i_1,i_2})$. We further assume that the variances satisfy $\max_{i_1,i_2} \sigma^2_{i_1,i_2} \leq 1$. Given access to n samples from this model, we are interested in understanding the performance of the plug-in estimator

$$\widehat{\pi}_{\mathsf{plug}} \in \operatorname*{argmax}_{\pi \in \Delta_d} \min_{i \in [d]} \pi^\top \hat{\mathbf{A}}_n e_i \ ,$$

where $\hat{\mathbf{A}}_n$ is the empirical estimate of the matrix, defined analogous to the estimate $\widehat{\mathbf{P}}$ in equation (4.6). In particular, we will be interested in obtaining a bound on the error

$$\Delta_{\mathbf{A}}(\widehat{\pi}_{\mathsf{plug}}, \pi^*) := \min_{i \in [d]} (\pi^*)^\top \mathbf{A} e_i - \min_{i \in [d]} \widehat{\pi}_{\mathsf{plug}}^\top \mathbf{A} e_i \ ,$$

which measures the gap in the value obtained when distribution $\widehat{\pi}_{\mathsf{plug}}$ is played compared with the value obtained by the Nash distribution π^*. Observe that the optimization problem for obtaining Nash equilibrium in equation (C.14) can be written as the following linear program with decision variables (π, t)

$$\begin{aligned} &\max \ t \\ &\text{such that } \pi^\top \mathbf{A} e_i \geq t \text{ for all } i \in [d], \\ &\sum_i \pi_i = 1 \quad \text{and} \quad \pi_i \geq 0 \text{ for all } i \in [d]. \end{aligned} \tag{Nash}$$

The above linear program has $d + 1$ variables (π, t) and $2d + 1$ constraints including one equality constraint. Similarly, the one can rewrite the objective for the plug-in estaimtor $\widehat{\pi}_{\mathsf{plug}}$ as the solution to a perturbed version of the above linear program

$$\begin{aligned} &\max \ t \\ &\text{such that } \pi^\top \hat{\mathbf{A}}_n e_i \geq t \text{ for all } i \in [d], \\ &\sum_i \pi_i = 1 \quad \text{and} \quad \pi_i \geq 0 \text{ for all } i \in [d]. \end{aligned} \tag{Pert}$$

Before stating our main result concerning the asymptotic distribution of the error $\Delta_{\mathbf{A}}(\widehat{\pi}_{\mathsf{plug}}, \pi^*)$, we introduce some notation first. Let us denote by $x = (\pi, t)$ the variables and by matrix C and vector c_{sim} the set of constraints in the linear program (Nash), that is,

$$C := \begin{bmatrix} \mathbf{A}^\top & -1_d \\ I_d & 0 \end{bmatrix} \quad \text{and} \quad c_{\mathsf{sim}} := [1_d, 0], \tag{C.15}$$

where we have denoted by 1_d the all-ones column vector in d dimension and by I_d the $d \times d$ identity matrix. Using this notation, we can rewrite this LP as

$$\begin{aligned} &\max\ t \\ &\text{such that } Cx \geq 0,\ \ c_{\mathsf{sim}}^\top x = 1 \end{aligned} \tag{C.16}$$

It will also be convenient to define the extended matrix $C_{\mathsf{ext}} := [C; c_{\mathsf{sim}}^\top]$ which contains both the equalit and inequality constraints. For the perturbed version of the linear program (Pert) we denote the analagous matrices respectively by $\hat{C}$ and $\hat{C}_{\mathsf{ext}}$. Observe that the simplex constraint encoded by the vector c_{sim} is deterministic and hence remains the same for both the original and perturbed linear programs.

Observe that the constraint polytope for the LP (Nash) is a closed convex set since the Nash distribution π belongs to the simplex Δ_d and the variable $t \in [0, 1]$. Therefore, the optimal solution $x^* = (\pi^*, t^*)$ will lie on one of faces whose dimension $0 \leq k_f \leq d$. In the special case when $k_f = 0$, we say that the LP admits a unique solution which is a vertex of the constraint polytope. Let us denote by subsets $J_1 \subseteq \{1, \ldots, d\}$ and $J_2 \subseteq \{d+1, \ldots, 2d\}$ the subset of constraints (rows of the constraint matrix C) which are tight for the set of optimal solutions and let us represent their union by $J = J_1 \cup J_2$. Observe that in addition to the equality constraint $c_{\mathsf{sim}}^\top x = 1$, there can be at most d constraints tight, that is, $|J| \leq d$. Further, we denote by $\hat{J}_1, \hat{J}_2$ and $\hat{J}$ the corresponding subsets for the perturbed linear program (Pert).

We first establish a technical lemma which establishes that given enough samples, the active constraints for the original LP (Nash) given by J will be contained in the active constraints $\hat{J}$ for the solution of the perturbed LP (Pert).

Lemma C.5. *Consider the perturbed LP (Pert) for any payoff matrix $\mathbf{A} \in [0, 1]^{d \times d}$ with noise distribution following the Gaussian passive sampling model. Then, for all $n > n_0(\mathbf{A}, \delta)$, we have that the active constraint sets J for the original LP and $\hat{J}$ for the perturbed LP satisfy $J \subseteq \hat{J}$ with probability at least $1 - \delta$.*

We defer the proof of the lemma to the end of the section. Observe that depending on the sampling of the noise variables, the subset $\hat{J}$ can vary with the noise variables. Each of these different subset can be seen as adding additional constraints on top of the $|J|$ constraints which charaterize the set of Nash equilibria for the original LP. Thus, when we look at the constraint matrix $C_{\mathsf{ext},\hat{J}}$, any $x = (\pi, t)$ satisfying $C_{\mathsf{ext},\hat{J}} \cdot x = [0_d, 1]^\top$ will necessarily have π as a Nash equilibrium.

Before stating our main result, we introduce some notation which is essential for the statement. Let use represent by $\Phi := \mathbf{A}^\top_{\hat{J}_1, \hat{J}_2^c}$ the rank r matrix of constraints which are tight in the perturned LP and its singular value decomposition by $\Phi = U\Sigma V^\top$ and the corresponding noisy matrix

$$\hat{\mathbf{A}}^\top_{\hat{J}_1, \hat{J}_2^c} = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^\top \\ V_2^\top \end{bmatrix} + \underbrace{\begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix}}_{Z_n},$$

where the matrix Z represents average zero-mean gaussian noise with n samples obtained from the passive sampling model. Further, we let $\tilde{Z}_n := U^\top Z_n V$ denote the noise matrix rotated by the directions given in U and V. With this, we state our result which characterizes the error $\Delta_{\mathbf{A}}$ for the plug-in estimator $\widehat{\pi}_{\mathsf{plug}}$ in terms of properties of the underlying matrix $\mathbf{A}$ and the noise matrix $\tilde{Z}$.

Theorem C.1. *For any payoff matrix $\mathbf{A} \in [0,1]^{d \times d}$ and for samples $n > n_0(\mathbf{A}, \delta)$ obtained via the Gaussian passive sampling model, we have the error $\Delta_{\mathbf{A}}$ of the plug-in estiamte $\widehat{\pi}_{\mathsf{plug}}$ satisfies*

$$\begin{aligned} \Delta_{\mathbf{A}}(\widehat{\pi}_{\mathsf{plug}}, \pi^*) \leq t^* \max_{i \in \hat{J}_1} e_i^\top U_1 \left(\tilde{Z}_{11} - \tilde{Z}_{12}\tilde{Z}_{22}^{-1}\tilde{Z}_{21} \right) \Sigma_1^{-1} U_1^\top 1_{\hat{J}_1} + O_P(\|\tilde{Z}_{11} - \tilde{Z}_{12}\tilde{Z}_{22}^{-1}\tilde{Z}_{21}\|_2^2) \\ + (\hat{t} - t^*)(1 + O_P(\|\tilde{Z}_{11} - \tilde{Z}_{12}\tilde{Z}_{22}^{-1}\tilde{Z}_{21}\|_2)) \,, \end{aligned} \tag{C.17}$$

with probability at least $1 - \delta$.

A few comments on the theorem are in order. Observe that the upper bound on the error is a stochastic quantity where the randomness is not only in the entries of the matrix $\tilde{Z}$ but also in the matrices U and Σ which depend on the (possibly) random subsets $\hat{J}_1$ and $\hat{J}_2$. The upper bound depends primarily on two terms, up to lower order error factors, one measures the alignment of the Schur complement $\tilde{Z}_{11} - \tilde{Z}_{12}\tilde{Z}_{22}^{-1}\tilde{Z}_{21}$ with the rotated and renormalized ones vector $1_{\hat{J}_1}$, and the second which measures the convergence of the empirical value $\hat{t}$ to the true value t^*. Going forward, we first provide a complete proof this result and then specialize it to the special case when the true Nash π^* is unique and lies in the interior of the simplex Δ_d – this greatly simplifies the above expression and allows us to study the problem dependent adaptivity properties of the plug-in estimator $\widehat{\pi}_{\mathsf{plug}}$.

Remark C.1. *The above analysis can be extended to the multi-criteria preference learning setup $k > 1$ whenever the distance function $\rho = \|\cdot\|_\infty$ and the target set S is a polytope by extending the linear program to handle the additional constraints. Similar to the result above, the final upper bound on the error will then depend only on the constraints which are tight in the original and perturbed programs. It remains an interesting problem to study the asymptotic error for general convex target sets for which the optimization problem can be written as a convex program.*

Proof of Theorem C.1. We will establish the claim by analyzing the structure of the solution $\hat{x} = (\widehat{\pi}_{\mathsf{plug}}, \hat{t})$ output by solving the perturbed linear program (Pert). Recall from our notation that given access to n noisy samples of the matrix $\mathbf{A}$, the set $\hat{J} = \hat{J}_1 \cup \hat{J}_2$ represents the set of constraints which are tight for the perturbed LP with the empirical matrix $\hat{\mathbf{A}}$ where $|\hat{J}| = d$. Also, observe that since these samples come from the Gaussian passive sampling model, we will have that the solution $\hat{x}$ will be unique[3] with probability 1.

Given this uniqueness, we can express the solution $\hat{x} = (\hat{\pi}, \hat{t})$ as the solution to the following set of linear equations

$$\begin{bmatrix} \hat{\mathbf{A}}_{\hat{J}_1}^\top & -1_{|\hat{J}_1|} \\ I_{\hat{J}_2} & 0_{|\hat{J}_2|} \\ 1_d & 0 \end{bmatrix} \cdot \begin{bmatrix} \widehat{\pi}_{|\hat{J}_1|} \\ \widehat{\pi}_{|\hat{J}_1|} \\ \hat{t} \end{bmatrix} = \begin{bmatrix} 0_{|\hat{J}_1|} \\ 0_{|\hat{J}_2|} \\ 1 \end{bmatrix}.$$

Let us denote by the vector $b_{\hat{J}} := [-1_{|\hat{J}_1|}, 0_{|\hat{J}_2|}]^\top$ and by the matrix $\tilde{C}_{\hat{J}} := [\hat{\mathbf{A}}_{\hat{J}_1}^\top; I_{\hat{J}_2}]$. Using a standard block matrix inversion formula, we have that the output solution

$$\widehat{\pi} = \hat{t}\tilde{C}_{\hat{J}}^{-1} b_{\hat{J}} \quad \text{and} \quad \hat{t} = \frac{1}{1_d^\top \tilde{C}_{\hat{J}}^{-1} b_{\hat{J}}}.$$

In order to further simplify the above expression, let us denote by $\hat{\mathbf{A}}_{\hat{J}_1, \hat{J}_2^c}$ the matrix formed by selecting the rows $\hat{J}_1$ and the columns $\hat{J}_2^c := [d] \setminus \hat{J}_2$ from the matrix $\hat{\mathbf{A}}$. The estimate $\widehat{\pi}$ is then given by

$$\widehat{\pi}_{\hat{J}_2^c} = -\hat{t}\hat{\mathbf{A}}_{\hat{J}_1, \hat{J}_2^c}^{-T} \cdot 1_{|\hat{J}_1|} \quad \text{and} \quad \widehat{\pi}_{\hat{J}_2} = 0.$$

Plugging in the above value of the estimate $\widehat{\pi}$ into the error term $\Delta_{\mathbf{A}}(\widehat{\pi}, \pi^*)$, we get,

$$\begin{aligned} \Delta_{\mathbf{A}}(\widehat{\pi}, \pi^*) &= \min_{i \in [d]} e_i^\top \mathbf{A}^\top \pi^* - \min_{i' \in [d]} e_{i'}^\top \mathbf{A}^\top \widehat{\pi} \\ &\overset{\text{(i)}}{=} \max_{i' \in [d]} \min_{i \in [d]} e_i^\top \mathbf{A}_{\cdot, \hat{J}_2^c}^\top \pi^*_{\hat{J}_2^c} - e_{i'}^\top \mathbf{A}_{\cdot, \hat{J}_2^c}^\top \widehat{\pi}_{\hat{J}_2^c} \\ &\overset{\text{(ii)}}{\leq} \max_{i \in \hat{J}_1} e_i^\top \mathbf{A}_{\hat{J}_1, \hat{J}_2^c}^\top \left(\pi^*_{\hat{J}_2^c} - \widehat{\pi}_{\hat{J}_2^c} \right) \\ &\overset{\text{(iii)}}{=} \max_{i \in \hat{J}_1} e_i^\top \mathbf{A}_{\hat{J}_1, \hat{J}_2^c}^\top \left(\hat{t}\hat{\mathbf{A}}_{\hat{J}_1, \hat{J}_2^c}^{-T} - t^* (\mathbf{A}_{\hat{J}_1, \hat{J}_2^c}^\top)^\dagger \right) 1_{|\hat{J}_1|} \end{aligned}$$

where equality (i) follows from noting that one of the Nash equilibria will have the components $\pi^*_{\hat{J}_2} = 0$ from Lemma C.5, (ii) follows from upper bounding the min and noting that the only columns of $\mathbf{A}$ that can be minimizers are those in $\hat{J}_1$ for large enough samples n, and (iii) follows by substituting the values of $\widehat{\pi}$ and the nash distribution π^*. We can further

[3]For the case when an entire column of matrix $\mathbf{A}$ is determinisitc, those constraints (if tight) can be combined with the other deterministic constraints and the analysis can proceed from there.

split the error term into two components, one which looks at the error in value $\hat{t}$, and the other corresponding to the error in matrix $\hat{\mathbf{A}}$.

$$\Delta_{\mathbf{A}}(\widehat{\pi}, \pi^*) \leq t^* \max_{i \in \hat{J}_1} e_i^\top \mathbf{A}^\top_{\hat{J}_1, \hat{J}_2^c} \left(\hat{\mathbf{A}}^{-\top}_{\hat{J}_1, \hat{J}_2^c} - (\mathbf{A}^\top_{\hat{J}_1, \hat{J}_2^c})^\dagger \right) 1_{|\hat{J}_1|} + (\hat{t} - t^*) \max_{i \in \hat{J}_1} e_i^\top \mathbf{A}^\top_{\hat{J}_1, \hat{J}_2^c} \hat{\mathbf{A}}^{-\top}_{\hat{J}_1, \hat{J}_2^c} 1_{|\hat{J}_1|} \tag{C.18}$$

Let us denote by $\Phi := \mathbf{A}^\top_{\hat{J}_1, \hat{J}_2^c}$. Then, we can rewrite the matrix $\hat{\mathbf{A}}^\top_{\hat{J}_1, \hat{J}_2^c} = \Phi + Z_n$ where the matrix Z_n represents the zero-mean noise from the Gaussian passive sampling model. Further, let $\Phi = U \Sigma V^\top$ denote the SVD of the matrix Φ. With this, the first term in the above decomposition for any fixed value of i is given by

$$e_i^\top \Phi((\Phi + Z_n)^{-1} - \Phi^\dagger) 1_{|\hat{J}_1|} = e_i^\top U \Sigma \left((\Sigma + U^\top Z_n V)^{-1} - \Sigma^\dagger \right) U^\top 1_{|\hat{J}_1|}.$$

Let us denote by $\tilde{Z}_n := U^\top Z_n V$ the effective noise matrix. Using the block matrix inversion formula, the above expression can be written as

$$\begin{aligned} e_i^\top \Phi((\Phi + Z_n)^{-1} - \Phi^\dagger) 1_{|\hat{J}_1|} &= e_i^\top \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \left(\begin{bmatrix} \Sigma_1 + \tilde{Z}_{11} & \tilde{Z}_{12} \\ \tilde{Z}_{21} & \tilde{Z}_{22} \end{bmatrix}^{-1} - \begin{bmatrix} \Sigma_1^{-1} & 0 \\ 0 & 0 \end{bmatrix} \right) \begin{bmatrix} U_1^\top \\ U_2^\top \end{bmatrix} 1_{\hat{J}_1} \\ &\overset{\text{(i)}}{=} e_i^\top U_1 \Sigma_1 \left((\Sigma_1 + \tilde{Z}_{11} - \tilde{Z}_{12} \tilde{Z}_{22}^{-1} \tilde{Z}_{21})^{-1} - \Sigma_1^{-1} \right) U_1^\top 1_{\hat{J}_1}. \end{aligned} \tag{C.19}$$

where Σ_1 is the diagonal matrix with non-zero singular value of Φ and equality (i) follows from the fact that $U_2^\top 1_{\hat{J}_1} = 0$. To see this, recall that U represents the column space of the matrix $\Phi = \mathbf{A}^\top_{\hat{J}_1, \hat{J}_2^c}$, that is the row space of the matrix $\mathbf{A}^\top_{\hat{J}_1, \hat{J}_2^c}$ with U_2 representing the null space of this matrix. Since we know that $(\pi^*)^\top \mathbf{A}^\top_{\hat{J}_1, \hat{J}_2^c} = t^* 1_{|\hat{J}_1|}$, all vectors in the null space will necessarily have to be orthogonal to the vector $1_{|\hat{J}_1|}$. Combining the above error bound with equation (C.18), we have,

$$\begin{aligned} \Delta_{\mathbf{A}}(\widehat{\pi}, \pi^*) &\leq t^* \max_{i \in \hat{J}_1} e_i^\top U_1 \Sigma_1 \left((\Sigma_1 + \tilde{Z}_{11} - \tilde{Z}_{12} \tilde{Z}_{22}^{-1} \tilde{Z}_{21})^{-1} - \Sigma_1^{-1} \right) U_1^\top 1_{\hat{J}_1} \\ &\quad + (\hat{t} - t^*) \max_{i \in \hat{J}_1} e_i^\top U_1 \Sigma_1 \left((\Sigma_1 + \tilde{Z}_{11} - \tilde{Z}_{12} \tilde{Z}_{22}^{-1} \tilde{Z}_{21})^{-1} \right) U_1^\top 1_{\hat{J}_1} \\ &\leq t^* \max_{i \in \hat{J}_1} e_i^\top U_1 \left(\tilde{Z}_{11} - \tilde{Z}_{12} \tilde{Z}_{22}^{-1} \tilde{Z}_{21} \right) \Sigma_1^{-1} U_1^\top 1_{\hat{J}_1} + O_P(\|\tilde{Z}_{11} - \tilde{Z}_{12} \tilde{Z}_{22}^{-1} \tilde{Z}_{21}\|_2^2) \\ &\quad + (\hat{t} - t^*)(1 + O_P(\|\tilde{Z}_{11} - \tilde{Z}_{12} \tilde{Z}_{22}^{-1} \tilde{Z}_{21}\|_2)) \,, \end{aligned}$$

where the final inequality follows from the Taylor series expansion $(I + X)^{-1} = I - X + O(\|X\|_2^2)$ whenever $\|Z\|_2 < 1$. This establishes the desired claim. □

Asymptotic error under uniqueness assumption. Having established an upper bound on the error for the general setup in Theorem C.1, we now consider the specific scenario where the payoff matrix $\mathbf{A}$ is a preference matrix and has a unique von Neumann winner π^*. This is formalized in the following assumption.

Assumption C.2 (Unique Nash equilibrium)**.** *The payoff matrix* $\mathbf{A}$ *belongs to the set of preference matrices* $\mathcal{P}_{d,1}$ *and has a unique mixed Nash equilibrium* π^**, that is,* $\pi^*_i > 0$ *for all* $i \in [d]$*.*

For any preference matrix $\mathbf{A} \in \mathcal{P}_{d,1}$ and the Bernoulli passive sampling model discussed in Section 4.3, the asymptotic variance for the Gaussian passive sampling model is $\sigma^2_{i,j} = \mathbf{A}_{i,j} \cdot (1 - \mathbf{A}_{i,j})$. Let us represent by Σ_i the diagonal matrix corresponding to the variances along the i^{th} column of the matrix $\mathbf{A}$ with

$$\Sigma_i = \mathrm{diag}(\mathbf{A}_{1,i} \cdot (1 - \mathbf{A}_{1,i}), \ldots, \mathbf{A}_{d,i} \cdot (1 - \mathbf{A}_{d,i}).$$

Given this notation, we now state a corollary which specializes the result of Theorem C.1 to payoff matrices satisfying the above assumption.

Corollary C.1. *For any payoff matrix* $\mathbf{A}$ *satisfying Assumption C.2 and for samples* $n > n_0(\mathbf{A}, \delta)$*, we have that the error* $\Delta_{\mathbf{A}}$ *of the plug-in estimate* $\widehat{\pi}_{\mathsf{plug}}$ *satisfies*

$$\begin{aligned}\Delta_{\mathbf{A}}(\widehat{\pi}_{\mathsf{plug}}, \pi^*) &\leq \|Z_n \pi^*\|_\infty + O_P(\|Z_n\|_2^2) \\ &\leq c \cdot \sqrt{\frac{\sigma^2_{\mathbf{A}} d^2}{n} \log\left(\frac{d}{\delta}\right)} + O_P(\|Z_n\|_2^2)\,, \end{aligned} \tag{C.20}$$

with probability at least $1 - \delta$ *and the variance* $\sigma^2_{\mathbf{A}} := \max_{i \in [d]} (\pi^*)^\top \Sigma_i \pi^*$*.*

We make a few remarks on the above corollary. Observe that the above is a high probability bound on the error $\Delta_{\mathbf{A}}$ of the plug-in estimator $\widehat{\pi}_{\mathsf{plug}}$. Compared with the upper bounds of Corollaries 4.1 and C.2, the asymptotic bound on the error above is instance dependent – the effective variance $\sigma^2_{\mathbf{A}}$ depends on the underlying preference matrix $\mathbf{A}$. In particular, this variance measures how well does the underlying von Neumann winner π^* align with each variance associated with each column of the matrix $\mathbf{A}$. In the worst case, since each entry of $\mathbf{A}$ is bounded above by 1, the variance $\sigma^2_{\mathbf{A}} = 1$ and we recover back the upper bounds from Corollaries 4.1 and C.2 for the uni-criterion case. The second term in the upper bound comprising the operator norm of the sampling noise, $\|Z_n\|_2^2$, can be shown to be $O_d(\frac{1}{n})$ with high probability, and therefore contributes as a lower order term.

Proof of Corollary C.1. Observe that Assumption C.2 implies that the set of tight constraints for the LP (Nash) are the ones corresponding to payoff matrix A. That is, the set $J_1 = [d]$ and $J_2 = \phi$. Following Lemma C.5, we have, for n large enough, the subset of tight constraints for the perturbed LP (Pert) satisfy $\hat{J}_1 = J_1$ and $\hat{J}_2 = J_2$. Further, the uniqueness assumption guarantees that the matrix $\mathbf{A}$ is full rank and hence, invertible.

Since the matrix $\hat{\mathbf{A}}$ is itself a preference matrix (by construction), the value $\hat{t} = t^* = \frac{1}{2}$ and therefore, using the upper bound on the error from Theorem C.1, we have,

$$\begin{aligned}\Delta_{\mathbf{A}}(\widehat{\pi}, \pi^*) &\leq t^* \max_{i \in [d]} \left[e_i^\top Z_n \mathbf{A}^{-\top} 1_d\right] + O_P(\|Z_n\|_2^2) \\ &\leq \|Z_n \pi^*\|_\infty + O_P(\|Z_n\|_2^2)\end{aligned}$$

where the final inequality follows by noting that $\pi^* = t^* \mathbf{A}^{-\top} 1_d$ and recall that Z_n denotes the zero mean noise-matrix obtained by the passive Gaussian sampling model with (asymptotic) variance $\sigma^2_{i,j} = \mathbf{A}_{i,j} \cdot (1 - \mathbf{A}_{i,j})$. Let us denote by $\Sigma_{\mathbf{A}}$ the diagonal matrix measuring the alignment of the Nash equilibrium π^* with the variance of the i^{th} column of the underlying matrix $\mathbf{A}$, that is,

$$\Sigma_{\mathbf{A}}(i,j) = \begin{cases} (\pi^*)^\top \Sigma_i \pi^* & \text{for } i = j \\ 0 & \text{otherwise} \end{cases}.$$

Following a similar calculation as in the proof of Corollary C.2, we have that each entry of the matrix $Z_{i,j}$ will have samples $N_{i,j} = \Theta(\frac{n}{d^2})$. Combined with a standard bound for the maximum of sub-Gaussian random variables [209], we have for $n > n_0(\mathbf{A}, \delta)$, with probability at least $1 - \delta$,

$$\Delta_{\mathbf{A}}(\widehat{\pi}, \pi^*) \leq c \cdot \sqrt{\frac{\sigma^2_{\mathbf{A}} d^2}{n} \log\left(\frac{d}{\delta}\right)} + O_P(\|Z_n\|_2^2) \ ,$$

for some universal constant $c > 0$ and where the variance $\sigma^2_{\mathbf{A}} = \max_{i \in [d]} \Sigma_{\mathbf{A}}(i,i)$. □

Example: Generalized Rock-Papers-Scissor. While in the worst-case, the variance determining the sample complexity of learning Nash from samples is $\sigma^2_{\mathbf{A}} = \Theta(1)$, we will now construct a family of preference matrices $\mathbf{A}^{(d)}$, for different values of dimension d, and show that $\sigma^2_{\mathbf{A}} = O(\frac{1}{d})$. This exhibits that the plug-in estimator $\widehat{\pi}_{\mathsf{plug}}$ can indeed adapt to the problem complexity and has a sample complexity of $\tilde{O}(\frac{d}{\epsilon^2})$ for these class of easier problems compared to the worst-case complexity of $\tilde{O}(\frac{d^2}{\epsilon^2})$.

Our example is a high-dimensional generalization of the classical Rock-Papers-Scissors (RPS) game. Recall, that the pay-off matrix for the RPS game is

$$\mathbf{A}^{\mathsf{RPS}} = \begin{array}{c|ccc} & \text{R} & \text{P} & \text{S} \\ \hline \text{R} & 0.5 & 0 & 1 \\ \text{P} & 1 & 0.5 & 0 \\ \text{S} & 0 & 1 & 0.5 \end{array}.$$

Observe that the above payoff matrix encodes a deterministic game: Rock beats Scissor, Scissor beats Paper, and Paper beats Rock. Similar to this, we define a randomized version of the above RPS game with payoffs where we allow a small probability 0.25 with which the lesser preferred item in a match-up can defeat the other, for example, Scissor against Rock. Explicilty, such a payoff matrix $\mathbf{A}^{(3)}$ is given by

$$\mathbf{A}^{(3)} = \begin{array}{c|ccc} & \text{R} & \text{P} & \text{S} \\ \hline \text{R} & 0.50 & 0.25 & 0.75 \\ \text{P} & 0.75 & 0.50 & 0.25 \\ \text{S} & 0.25 & 0.75 & 0.50 \end{array}.$$

Similar to the deterministic RPS game, the above randomized game can be seen to have a unique Nash equilibrium with $\pi^* = [\frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$.

We now describe a d-dimensional generalization of the above payoff matrix, for any odd value of $d = 2d' + 1$. For the element e_1, the first entry will be set to 0.50, the next d' entries to be 0.25 and the final d' entries to be 0.75 – game-theoretically, this means that the element e_1 loses to elements $e_2 - e_{d'+1}$ and is preferred over elements $e_{d'+2} - e_{2d'+1}$, both with probability 0.75. Similarly, for the i^{th} element, the row $\mathbf{A}_i^{(d)}$ is given by

$$\mathbf{A}^{(d)}(i,j) = \begin{cases} 0.50 & \text{for } j = i \\ 0.25 & \text{for } j \in [i+1 \pmod d, i+d' \pmod d] \\ 0.75 & \text{for } j \in [i+d'+1 \pmod d, i+2d' \pmod d] \end{cases}.$$

It is easy to see from the form of the pay-off matrix that each element e_i is preferred over d' elements and has a lower preference than d' elements. By the symmetry of the payoff matrix, the unique Nash equilibrium is given by the distribution $\pi^* = \frac{1}{d}1_d$ which lies in the interior of the simplex Δ_d and hence satisfies Assumption C.2. Further, we can compute the variance $\sigma^2_{\mathbf{A}^{(d)}}$ as

$$\sigma^2_{\mathbf{A}^{(d)}} = \max_{i\in[d]} (\pi^*)^\top \Sigma_i \pi^* = \max_i \left(\frac{1}{4d^2} + \sum_{j\neq i} \frac{3}{16d^2} \right) \leq \frac{1}{d^2} + \frac{3}{16d},$$

Plugging this variance in the upper bound obtained in Corollary C.1, for $n > n_0(\mathbf{A}, \delta)$ we have

$$\Delta_{\mathbf{A}^{(d)}}(\widehat{\pi}_{\mathsf{plug}}, \pi^*) \leq c\sqrt{\frac{d}{n}\log\left(\frac{d}{\delta}\right)} \tag{C.21}$$

with probability greater than $1 - \delta$. Thus, to obtain an ϵ-accurate solution for the payoff matrix $\mathbf{A}^{(d)}$, the plug-in estimator $\widehat{\pi}_{\mathsf{plug}}$ requires $\tilde{O}(\frac{d}{\epsilon^2})$ samples, a factor d less than the worst-case sample complexity of $\tilde{O}(\frac{d^2}{\epsilon^2})$.

Proof of Lemma C.5

Recall from our discussion above that the the constraint set for the LP (Nash), the constraint polytope is a closed convex set and has a finite number $|V| = O((2d+2)^{\frac{d+1}{2}})$, which follows from McMullen's theorem [142]. Let us denote each vertex of the polytope by $x_v = (\pi_v, t_v)$ and the corresponding set of d constraints which define the vertex by J_v. Further, let V^* denote the set of optimal vertices.

Because of the random Gaussian noise, for $n = \Omega(d^2)$, we have that the solution $\hat{x} = (\widehat{\pi}_{\mathsf{plug}}, \hat{t})$ will be unique with probability 1. In order to establish the claim of the lemma, we can equivalently show that for any vertex $x_v \notin V^*$, the corresponding vertex $\hat{x}_v$ will not be output by the perturbed LP. This follows from the observation that for $n > O(d^2 \log(1/\delta))$

and for any vertices x_v, we have

$$\mathbb{P}\left(|\hat{t}_v - t_v| \geq c\sqrt{\frac{d^2}{n}\log\left(\frac{1}{\delta}\right)}\right) \leq \delta,$$

for some universal constant[4] $c > 0$. Taking a union bound over all $|V|$ vertices, we have

$$\mathbb{P}\left(\exists\, x_v \text{ s.t. } |\hat{t}_v - t_v| \geq c\sqrt{\frac{d^2}{n}\log\left(\frac{|V|}{\delta}\right)}\right) \leq \delta.$$

Therefore, whenever $\max_v t_v \leq t^* - \gamma$ for some $\gamma > 0$, we have that after $n > c\frac{d^2}{\gamma^2}\log\left(\frac{|V|}{\delta}\right)$, with probability at least $1 - \delta$, we have,

$$\max_{v \notin V^*} \hat{t}_v < \min_{v \in V^*} \hat{t}_v \ , \tag{C.22}$$

and therefore the perturbed LP will have active constraint set satisfying $J \subseteq \hat{J}$. This proves the desired claim.

C.4 Additional results and their proofs

This section covers additional sample complexity results as well as optimization algorithms for finding the Blackwell winner of a multi-criteria preference learning instance.

Sample complexity bounds for ℓ_1 norm

Corollary C.2. *Suppose that the distance ρ is induced by the ℓ_1 norm $\|\cdot\|_1$. Then there exists a universal constant $c > 0$ such that given a sample size $n > cd^2k\log(\frac{cdk}{\delta})$, we have for each valid target set S*

$$\Delta_{\mathbf{P}}(\widehat{\pi}_{\mathsf{plug}}, \pi^*) \leq ck\sqrt{\frac{d^2k}{n}\log\left(\frac{cdk}{\delta}\right)} \tag{C.23}$$

with probability exceeding $1 - \delta$.

Proof. Being somewhat more explicit with our notation, let $N_{(i_1,i_2,j)}$ denote the number of samples observed under the passive sampling model at index (i_1, i_2, j) of the tensor. Proceeding as in equation (C.6), we have

$$\Pr\left\{\|\mathbf{P}^j(\cdot, i_2) - \widehat{\mathbf{P}}^j(\cdot, i_2)\|_\infty \geq c\sqrt{\frac{\log(cd/\delta)}{\min_{i_1 \in [d]} N_{(i_1,i_2,j)}}}\right\} \leq \delta.$$

[4] the constant c can change values across lines, but will always remain a universal constant independent of problem parameters.

Summing over all criteria $j \in [k]$ along with a union bound, we obtain

$$\Pr\left\{\|\mathbf{P}(\cdot, i_2) - \widehat{\mathbf{P}}(\cdot, i_2))\|_{\infty,1} \geq ck\sqrt{\frac{\log(cdk/\delta)}{\min_{i_1,j} N_{(i_1,i_2,j)}}}\right\} \leq \delta.$$

Finally, in order to obtain a bound on the maximum deviation in the $(\infty, 1)$-norm, we take a union bound over all d choices of the index i_2, and apply inequality (C.5) to obtain

$$\max_{i_2} \|\mathbf{P}(\cdot, i_2) - \widehat{\mathbf{P}}(\cdot, i_2))\|_{\infty,1} \leq ck\sqrt{\frac{d^2k}{n}\log\left(c\frac{dk}{\delta}\right)}$$

with probability exceeding $1 - \delta$. □

A few comments regarding the corollary are in order. The above corollary suggests that the sample complexity required for obtaining an ϵ-accurate solution with respect to the ℓ_1 norm is $n = \widetilde{O}(\frac{d^2k^3}{\epsilon^2})$. Observe that this bound is a factor of k^2 worse than the corresponding one for ℓ_∞ norm established in Corollary 4.1. This additional sample complexity occurs since for any vector $v \in \mathbb{R}^k$, we have $\|v\|_1 \leq k\|v\|_\infty$. This implies that the error when measured with respect to ℓ_1 can be upto k times larger; since the sample complexity scales as $\frac{1}{\epsilon^2}$, the corresponding increase with respect to the number of criteria k is quadratic.

Optimization algorithms

Recall that Theorem 4.3 established that the objective function $v(\pi; \mathbf{P}, S, \|\cdot\|_q)$ is convex in π and Lipschitz with respect to the ℓ_1 norm. This implies that one could compute the plug-in solution $\widehat{\pi}_{\mathsf{plug}}$ as a solution to a constrained optimization problem. In this section, we discuss a few specific algorithms based on zeroth-order and first-order methods for obtaining such a solution.

Zeroth-order optimization

Zeroth-order methods for minimizing a function $f(x)$ over $x \in \mathcal{X}$ work with a function query oracle. That is, at each time step, the algorithm has access to an oracle which returns the value $f(x)$ for any point $x \in \mathcal{X}$. In our setup, since we are interested in minimizing the value function $v(\pi; \mathbf{P}, S, \rho)$ over $\pi \in \Delta_d$, such a function query requires access to the target set S via an oracle $\mathcal{O}_S^0$ such that

$$\mathcal{O}_S^0(z) \to \min_{z_1 \in S} \rho(z, z_1)\ ,$$

for the underlying distance function $\rho(\cdot)$. The oracle $\mathcal{O}_S^0$ essentially takes as input a score vector $z \in [0, 1]^k$ and outputs the distance of this point to the target set S. Given this oracle, it is easy to see that for any π, one can compute the corresponding value function $v(\pi; \mathbf{P}, S, \rho)$.

Algorithm 5: Zeroth-order method for multi-criteria preference learning

Input: Time steps T, step size η, smoothing radius δ
Initialize: $\theta_1 = 0$
for $t = 1, \ldots, T$ **do**
  $\pi_t = \text{argmax}_{\pi \in \Delta_d} \langle \theta_t, \pi \rangle - r(\pi)$ where $r(\pi) = \sum_i \pi_i \log(\pi_i)$
  Sample u_t uniformly from the Euclidean unit sphere $\{u \mid \|u\|_2 = 1\}$
  For every $i \in [d]$, query points $z_{1,i} = \mathbf{P}(\pi_t + \delta u_t, i)$ and $z_{2,i} = \mathbf{P}(\pi_t + \delta u_t, i)$
  Set $v(\pi_t + \delta u_t; \mathbf{P}, S, \rho) = \max_i \rho(z_{1,i}, S)$ and $v(\pi_t - \delta u_t; \mathbf{P}, S, \rho) = \max_i \rho(z_{2,i}, S)$
  Set sub-gradient estimate $\hat{g}_t = \frac{d}{2\delta} \left(v(\pi_t + \delta u_t; \mathbf{P}, S, \rho) - v(\pi_t - \delta u_t; \mathbf{P}, S, \rho) \right) u_t$
  Update $\theta_{t+1} = \theta_t - \eta \hat{g}_t$
Output: $\bar{\pi}_T = \frac{1}{T} \sum_{t=1}^{T} \pi_t$

There have been several algorithms proposed for optimization with such oracles when the underlying function f is convex [76, 4, 181, 69, 149, 180] or non-convex, smooth [89]. The key idea in the proposed algorithms is to utilize the zeroth-order oracle to constuct estimates of the (sub-)gradient of the function f using a class of techniques called *randomized smoothing*. The algorithms then differ in the construction of these estimates depending on the underlying randomness as well as on the number of oracle calls during each time step.

Given the results of Theorem 4.3, we can restrict our focus on algorithms for the class of convex Lipschitz function f. To this end, Shamir [180] proposed an algorithm for optimizing such functions which required *two* function evaluations at each time. The algorithm, adapted to the multi-criteria preference learning problem, is detailed in Algorithm 5. For our setup, we select the negative entropy regularization, $r(\pi) = \sum_i \pi_i \log(\pi_i)$ to suit the geometry of our domain $\mathcal{X} = \Delta_d$.

The proposed algorithm, maintains an estimate of the distribution, π_t, and at each time step t, queries the function value $v(\cdot; \mathbf{P}, S, \rho)$ at the following two points: $\pi_t + \delta u_t$ and $\pi_t - \delta u_t$, where u is sampled uniformly from the Euclidean unit sphere and $\delta > 0$ represents the smoothing radius. Given these queries, the sub-gradient estimate, $\hat{g}_t$ is then obtained as:

$$\hat{g}_t := \frac{d}{2\delta} \left(v(\pi_t + \delta u_t; \mathbf{P}, S, \rho) - v(\pi_t - \delta u_t; \mathbf{P}, S, \rho) \right) u_t \ .$$

The sub-gradient estimate is then used to update the parameter estimate π_{t+1} using the mirror descent algorithm with the specified regularization function. The zeroth-order method in Algorithm 5 does not require the underlying function to be smooth and hence works for our problem setup with arbitrary non-differentiable distance functions. We can now obtain the following convergence result, based on Theorem 1 from the work of Shamir [180].

Proposition C.3. *Suppose the conditions of Theorem 4.3 hold, and that Algorithm 5 is run for T iterations with step-size $\eta_t = \frac{c}{k^{1/q}\sqrt{dT}}$ and smoothing radius $\delta = \frac{c \log d}{\sqrt{T}}$, and produces a*

sequence $\pi_1, \pi_2, \ldots, \pi_T$. *Then we have*

$$v\left(\bar{\pi}_T; \mathbf{P}, S, \|\cdot\|_q\right) \leq \min_{\pi \in \Delta_d} v(\pi; \mathbf{P}, S, \|\cdot\|_q) + ck^{\frac{1}{q}} \cdot \sqrt{\frac{d \log^2 d}{T}}$$

where $\bar{\pi}_T = \frac{1}{T}\sum_{t=1}^{T} \pi_t$.

Proof. By Theorem 4.3, the value function $v(\pi; \mathbf{P}, S, \|\cdot\|_q)$ is convex and $L_{\mathsf{v}} = k^{\frac{1}{q}}$-Lipschitz with respect to $\|\cdot\|_1$. Also, the choice of the regularizer $r(\pi) = \sum_i \pi_i \log(\pi_i)$ is 1-strongly convex with respect to the $\|\cdot\|_1$. Plugging in the above values in Theorem 1 from [180] establishes the above convergence rate. □

Thus, in order to obtain a distribution $\hat{\pi}$ that is ϵ-close to π^* in function value, we need to run Algorithm 5 for $T = O\left(\frac{k^{\frac{2}{q}} d \log^2 d}{\epsilon^2}\right)$ iterations. Also, note that each iteration of the algorithm requires d calls to the oracle $\mathcal{O}_S^0$. Therefore the total oracle complexity of the procedure is $O\left(\frac{k^{\frac{2}{q}} d^2 \log^2 d}{\epsilon^2}\right)$.

First-order optimization

In this section, we look at first-order methods to compute the plug-in estimator. Let us denote by $\partial v(\pi)$ the set of sub-differentials of the function $v(\cdot; \mathbf{P}, S, \|\cdot\|)$ evaluated at π. Further, let the set $\Gamma(\pi)$ denote the set of maximizers for a policy π, that is,

$$\Gamma(\pi) = \left\{ \tilde{\pi} \in \Delta_d \mid \tilde{\pi} \in \operatorname*{argmax}_{\pi_2 \in \Delta_d} \min_{z \in S} \left[\|\mathbf{P}(\pi, \pi_2) - z\|\right] \right\} . \tag{C.24}$$

Note that both of these quantities depend implicitly on the tuple $(S, \mathbf{P}, \|\cdot\|)$, but we have dropped this dependence in the notation. Given the setup above, Lemma C.6 below characterizes this set $\partial v(\pi)$ for any smooth ℓ_q norm (with $1 < q < \infty$).

Lemma C.6. *Suppose that the distance is induced by a smooth* ℓ_q *norm for* $1 < q < \infty$. *Then the set of sub-differentials of* v *at* π *is given by:*

$$\partial v(\pi) = conv\left\{ \frac{\mathbf{P}(\cdot, \pi_2)\left[\mathbf{P}(\pi, \pi_2) - \Pi_S(\mathbf{P}(\pi, \pi_2))\right]}{\|\mathbf{P}(\pi, \pi_2) - \Pi_S(\mathbf{P}(\pi, \pi_2))\|_q} \mid \pi_2 \in \Gamma(\pi) \right\} ,$$

where $\Pi_S(z)$ *denotes the unique projection of the point* z *onto set* S *along* $\|\cdot\|_q$.

We defer the proof of the above lemma to later in the section. Note that in order to access such a sub-gradient, we need access to an oracle $\mathcal{O}_S^1$ that provides projection queries of the form

$$\mathcal{O}_S^1(z) \to \operatorname*{argmin}_{z_1 \in S} \rho(z, z_1).$$

Algorithm 6: First-order method for multi-criteria preference learning

Input: Time steps T, step size η

Initialize: $\theta_1 = 1_k$

for $t = 1, \ldots, T$ **do**

Set the distribution $\pi_t = \frac{\theta_t}{\|\theta_t\|_1}$

Obtain $g_t \in \text{conv}\left\{ \frac{\mathbf{P}(\cdot,\pi_2)[\mathbf{P}(\pi_t,\pi_2) - \Pi_S(\mathbf{P}(\pi_t,\pi_2))]}{\|\mathbf{P}(\pi_t,\pi_2) - \Pi_S(\mathbf{P}(\pi_t,\pi_2))\|_q} \mid \pi_2 \in \Gamma(\pi_t) \right\}$ [See eq.(C.24) for $\Gamma(\pi_t)$]

Update $\theta_{t+1,i} = \pi_{t,i} \exp(-\eta g_{t,i})$

Output: $\bar{\pi}_T = \frac{1}{T} \sum_{t=1}^{T} \pi_t$

The oracle $\mathcal{O}_S^1$ takes in a point z and outputs the closest point in the set S to this point. Given such an oracle, we can compute the sub-gradient of the function $v(\pi; \mathbf{P}, S, \rho)$ using Lemma C.6 by evaluating it at the point given by $\mathbf{P}(\pi, \pi_2)$ for some $\pi_2 \in \Gamma(\pi)$.

Given access to such a projection oracle $\mathcal{O}_S^1$, we detail out a procedure based on a standard implementation of mirror descent with entropic regularization (or Exponentiated gradient method) in Algorithm 6 to minimize the objective $v(\pi; G)$. Note that we select the negative entropy function, $r(\pi) = \sum_i \pi_i \log(\pi_i)$, as the regularization function for the mirror descent procedure since our parameter space is given by the simplex Δ_k and the negative entropy function is known to be 1-strongly convex with respect to $\|\cdot\|_1$ over this space.

The algorithm works by maintaining at each time instance a distribution π_t over the set of objects and updates it via an exponentiated gradient update. That is, the sub-gradient g_t is evaluated at the current point π_t using access to both $\mathcal{O}_S^1$ and $\mathcal{O}_S^0$, and is used to update each coordinate of the variable θ_t. The updated distribution π_{t+1} is obtained via a KL-projection of θ_t onto the simplex Δ_k, which can be shown to be equivalent to the normalization $\theta/\|\theta\|_1$. We now proceed to prove a convergence result for this gradient-based Algorithm 6, based on a standard analysis of the mirror descent procedure (for example, see [41, Theorem 4.2]).

Proposition C.4. *Suppose the conditions of Theorem 4.3 hold and consider any ℓ_q-norm for $1 < q < \infty$. Suppose that running Algorithm 5 for T iterations with step-size $\eta_t = \frac{1}{k^{1/q}} \sqrt{\frac{2 \log d}{T}}$ produces a sequence $\pi_1, \pi_2, \ldots, \pi_T$. Then we have*

$$v(\bar{\pi}_T; \mathbf{P}, S, \|\cdot\|_q) \leq \min_{\pi \in \Delta_d} v(\pi; \mathbf{P}, S, \|\cdot\|_q) + k^{\frac{1}{q}} \cdot \sqrt{\frac{2 \log d}{T}}$$

where $\bar{\pi}_T = \frac{1}{T} \sum_{t=1}^{T} \pi_t$.

Proof. Note that the function $v(\pi; \mathbf{P}, S, \|\cdot\|_q)$ is convex and $k^{\frac{1}{q}}$-Lipschitz with respect to the ℓ_1 norm from Theorem 4.3. Further, the mirror map given by negative entropy function is 1-strongly convex with respect to $\|\cdot\|_1$. Plugging in these values in Theorem 4.2 from [41] establishes the required convergence rate. □

In order to obtain an ϵ-accurate solution in function value, it suffices to run the above algorithm for $T = O\left(\frac{k^{\frac{2}{q}} \log d}{\epsilon^2}\right)$ iterations, with each iteration using 1 call to the oracle $\mathcal{O}_S^1$ and d calls to the oracle $\mathcal{O}_S^0$ (to obtain the set Γ). Thus, we see that the total oracle complexity changes as $\mathcal{O}_S^1 : O\left(\frac{k^{\frac{2}{q}} \log d}{\epsilon^2}\right)$ calls and $\mathcal{O}_S^0 : O\left(\frac{k^{\frac{2}{q}} d \log d}{\epsilon^2}\right)$ calls – effectively, an $O(d \log d)$ decrease in the calls to $\mathcal{O}_S^0$ is compensated by a corresponding increase of $O(\frac{\log d}{\epsilon^2})$ calls to the stronger oracle $\mathcal{O}_S^1$.

Proof of Lemma C.6. Consider the function $\phi(\pi_1, \pi_2) = \max_{z \in S} \|\mathbf{P}(\pi_1, \pi_2) - z\|$ over the domain $\pi_2 \in \Delta_d$. For any fixed π_2, we have that the function $\phi(\pi_1, \pi_2)$ is convex in π_1. Thus, by Danskin's theorem, we have that the subdifferential set is given by:

$$\partial v(\pi) = \text{conv}\left\{ \frac{\partial \phi(\pi, \pi_2)}{\partial \pi} \mid \pi_2 \in \Gamma(\pi) \right\}, \tag{C.25}$$

where conv represents the convex hull of the set. Let us now focus on the partial derivative $\frac{\partial \phi(\pi, \pi_2)}{\partial \pi}$ for any π_2 which is a maximizer. This partial derivative involves differentiation of a metric projection onto a convex set, which has been studied extensively in the literature of convex analysis [160, 220, 7]. Recently, Balestro et al. [20] established that for distance functions given by smooth norms, the derivative of metric projection for any $z \notin S$ is given by:

$$\nabla \rho(z, S) = \nabla \min_{z_2 \in S} \|z - z_2\| = \frac{z - \Pi_S(z)}{\|z - \Pi_S(z)\|},$$

where $\Pi_S(z)$ denotes the unique projection of the point z onto set S. Combining this with the chain rule of differentiation, we have that:

$$\frac{\partial \phi(\pi, \pi_2)}{\partial \pi} = \frac{\mathbf{P}(\cdot, \pi_2) \left[\mathbf{P}(\pi, \pi_2) - \Pi_S(\mathbf{P}(\pi, \pi_2))\right]}{\|\mathbf{P}(\pi, \pi_2) - \Pi_S(\mathbf{P}(\pi, \pi_2))\|_q}.$$

The above, in conjunction with equation (C.25) establishes the desired claim. □

C.5 Details of user study

In this section, we provide the deferred details of the user study from Section 4.4.

Self-driving environment. The self-driving environment consists of an autonomous car which can be controlled by providing real-valued inputs acceleration and angular acceleration at every time step. We allow the policies to have access to the dynamics of this environment. Observe that there is no explicit reward function in the environment and each policy differs in the way it optimizes a chosen reward function to drive the car forward in a safe manner.

Policies. The MPC based Policies A-E were constructed by optimizing linear rewards comprising features F1-F9 as

F1 Distance from the starting point along y-axis.

F2 Velocity of the autonomous car.

F3 Distance from the center of each lane.

F4 Gaussian collision detector for nearby objects.

F5 Collision detector which works at smaller radii than F4.

F6 Over-speeding feature which penalizes higher speeds.

F7 Reward for over-taking vehicles in the front.

F8 Gaussian off-road detector.

F9 Reward to promote speeding up near obstacles.

For each of the base policy, we set the weights of the features to encode different driving behaviors.

Pol A programmed to prefer the right-most lane and progress forward at a slow speed.

Pol B programmed to prefer the left-most lane and move forward as fast as possible.

Pol C programmed to be conservative, avoids collision and proceeds forward.

Pol D programmed to get attracted towards other cars and obstacles.

Pol E programmed to prefer center lane and exhibit opportunistic behavior by moving ahead of other cars.

Details of target set and linear weights. We selected the two data-oblivious sets to trade-off between the criteria C1-C5 as

$$\begin{aligned} S_1 &= \{z \mid z \in [0,1]^5, z_1 \geq 0.3, z_2 \geq 0.3, z_3 \geq 0.2, z_4 \geq 0.3, z_5 \geq 0.4\}, \\ S_2 &= \{z \mid z \in [0,1]^5, z_1 \geq 0.25, z_2 \geq 0.25, z_3 \geq 0.25, z_4 \geq 0.25, z_5 \geq 0.25, z_1 + z_5 \geq 0.9\}. \end{aligned} \tag{C.26}$$

In addition, we selected 9 set of weights $w_{1:9}$ for linearly combining the different criteria.

w_1: Average of the users' self-reported weights.

w_2: Weight vector obtained by regressing the overall criterion on C1-C5 with squared loss as

$$w_2 \in \operatorname*{argmin}_{w \in \Delta_5} \sum_{i_1, i_2} (\mathbf{P}_{\mathsf{ov}}(i_1, i_2) - \sum_j w(j) \mathbf{P}^j(i_1, i_2))^2.$$

w_3: Weight obtained by regressing Bradley-Terry-Luce (BTL) scores. The BTL parametric model assumes a real-valued score v_i for each policy and posits that $\Pr(\text{Pol } i \succeq \text{Pol } j) = \exp(v_i)/\exp(v_i) + \exp(v_j)$. Denoting the scores obtained from the overall preferences by v^{ov} and those obtained from the individual criteria by v^j for $j \in [5]$, the weight

$$w_2 \in \operatorname*{argmin}_{w \in \Delta_5} \sum_i (v_i^{\mathsf{ov}} - \sum_j w(j) v_i^j)^2.$$

w_4: Data-oblivious weight $w_4 = [0.2, 0.2, 0.2, 0.2, 0.2]$.

w_5: Data-oblivious weight $w_5 = [0.25, 0.5/3, 0.5/3, 0.5/3, 0.25]$.

w_6: Data-oblivious weight $w_6 = [0.30, 0.4/3, 0.4/3, 0.4/3, 0.30]$.

w_7: Data-oblivious weight $w_7 = [0.5/3, 0.5/3, 0.25, 0.5/3, 0.25]$.

w_8: Data-oblivious weight $w_8 = [0.4/3, 0.4/3, 0.3, 0.4/3, 0.30]$.

w_9: Data-oblivious weight $w_9 = [0.3, 0.1/2, 0.3, 0.1/2, 0.3]$.

The set of data oblivious weights were chosen to account for different trade-offs along the criteria C1-C5 including the uniform weight w_4.

Data Collection. Table C.1 shows the comparison data collected from the Mturk users in both the phases of the experiment. The entry i, j of the comparison matrices represents the fraction of users which preferred Policy i over Policy j. The top 5 rows and columns of each matrix correspond to the baseline policies while the bottom rows correspond to the two randomized policies R1 and R2 obtained as the Blackwell winner corresponding to sets S_1 and S_2 respectively.

In addition, we would like to highlight some details from an experiment design perspective. Since the experiment was run in two phases, we could not guarantee the same set of subjects to participate in both parts of the experiment. In order to limit distribution shifts, we restricted the nationality of the subjects to United States and began both the phases on the same time and day of the week. Also, in order to prevent biased evaluations, the ordering of the policy pairs as well as the ordering policies within a comparison was randomized across the users.

Figures C.1, C.2 and C.3 shows the experiment setup we used for obtaining comparison data from Amazon Mechanical Turk users consisting of the instructions, the policy comparison page and the questionnaire that the users were asked to fill out.

	A	B	C	D	E
A	0.50	0.64	0.45	0.41	0.39
B	0.36	0.50	0.30	0.30	0.25
C	0.55	0.70	0.50	0.55	0.57
D	0.59	0.70	0.45	0.50	0.52
E	0.61	0.75	0.43	0.48	0.50
R1	0.49	0.80	0.22	0.46	0.29
R2	0.49	0.88	0.66	0.61	0.41

(a) C1: Aggressiveness

	A	B	C	D	E
A	0.50	0.57	0.50	0.50	0.41
B	0.43	0.50	0.30	0.39	0.45
C	0.50	0.70	0.50	0.43	0.59
D	0.50	0.61	0.57	0.50	0.57
E	0.59	0.55	0.41	0.43	0.50
R1	0.46	0.71	0.32	0.51	0.39
R2	0.51	0.71	0.61	0.59	0.51

(b) C2: Predictability

	A	B	C	D	E
A	0.50	0.16	0.25	0.32	0.30
B	0.84	0.50	0.89	0.82	0.68
C	0.75	0.11	0.50	0.73	0.61
D	0.68	0.18	0.27	0.50	0.41
E	0.70	0.32	0.39	0.59	0.50
R1	0.73	0.22	0.76	0.78	0.76
R2	0.90	0.24	0.44	0.66	0.66

(c) C3: Quickness

	A	B	C	D	E
A	0.50	0.59	0.45	0.57	0.39
B	0.41	0.50	0.32	0.34	0.32
C	0.55	0.68	0.50	0.48	0.59
D	0.43	0.66	0.52	0.50	0.50
E	0.61	0.68	0.41	0.50	0.50
R1	0.44	0.80	0.20	0.39	0.24
R2	0.41	0.80	0.71	0.59	0.39

(d) C4: Conservativeness

	A	B	C	D	E
A	0.50	0.52	0.41	0.50	0.43
B	0.48	0.50	0.32	0.55	0.55
C	0.59	0.68	0.50	0.55	0.57
D	0.50	0.45	0.45	0.50	0.50
E	0.57	0.45	0.43	0.50	0.50
R1	0.54	0.68	0.32	0.49	0.41
R2	0.63	0.73	0.59	0.61	0.54

(e) C5: Collision Risk

	A	B	C	D	E
A	0.50	0.39	0.25	0.43	0.34
B	0.61	0.50	0.30	0.50	0.50
C	0.75	0.70	0.50	0.57	0.61
D	0.57	0.50	0.43	0.50	0.48
E	0.66	0.50	0.39	0.52	0.50
R1	0.66	0.76	0.29	0.59	0.39
R2	0.66	0.73	**0.66**	0.56	0.51

(f) Overall Preferences

Table C.1. Each matrix consists of pairwise comparisons between policies elicited from a user study with around 50 participants on Mturk. An entry i, j of the comparison matrices represents the fraction of users which preferred Policy i over Policy j. Policies A-E comprise the base set of policies while Policies R1-R2 are the randomized Blackwell winners obtained from the sets in equation (C.26). While Policy C is the overall von Neumann winner, Policy R2 is preferred over it by 66% of the users.

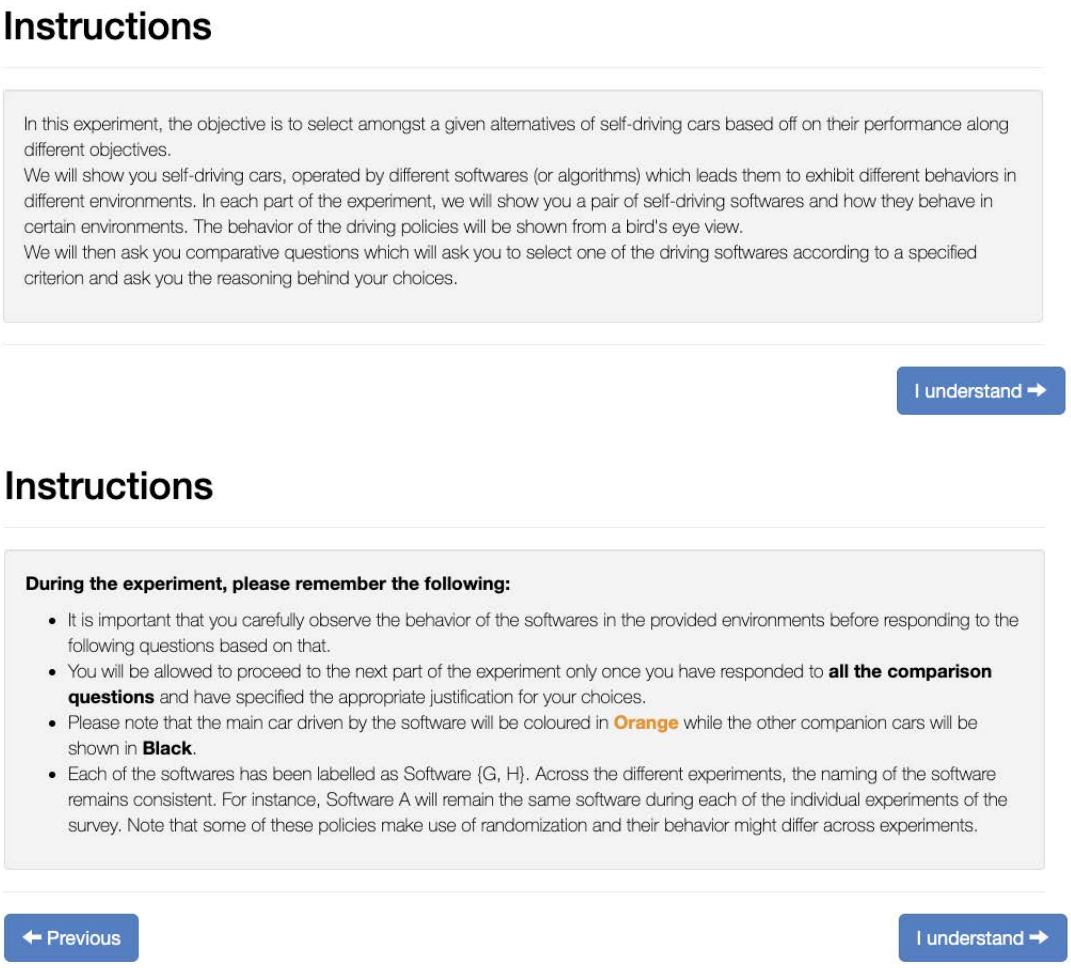


Figure C.1. Instructions provided to the users before the experiment began. The users were asked to compare behavior of policies and were told to expect some policies to exhibit a randomized behavior.

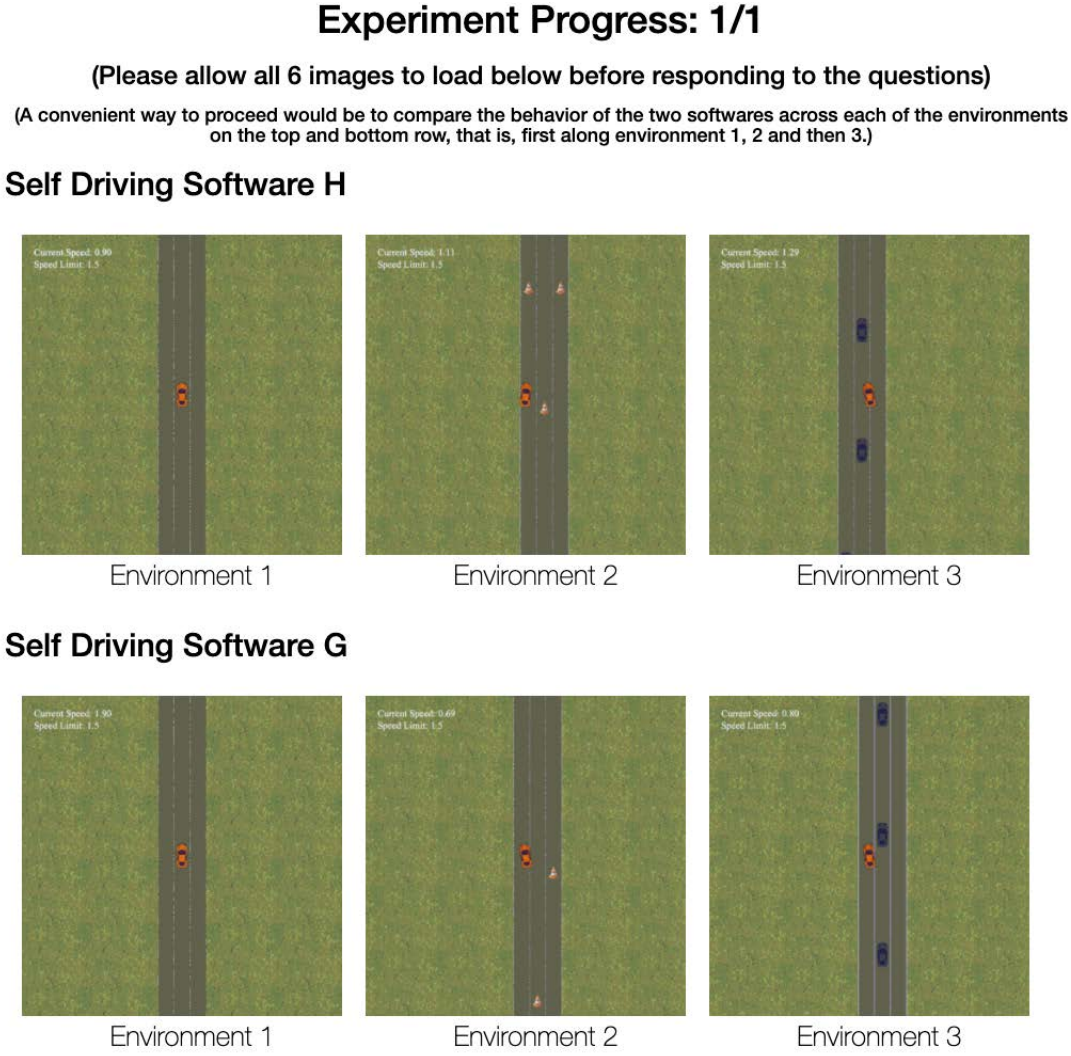


Figure C.2. Layout of the experiment where each panel shows a GIF exhibiting a Policy controlling the autonomous vehicle in one of the worlds of the environment. The users were instructed to compare behaviors across each of the columns before proceeding to answer the questions.

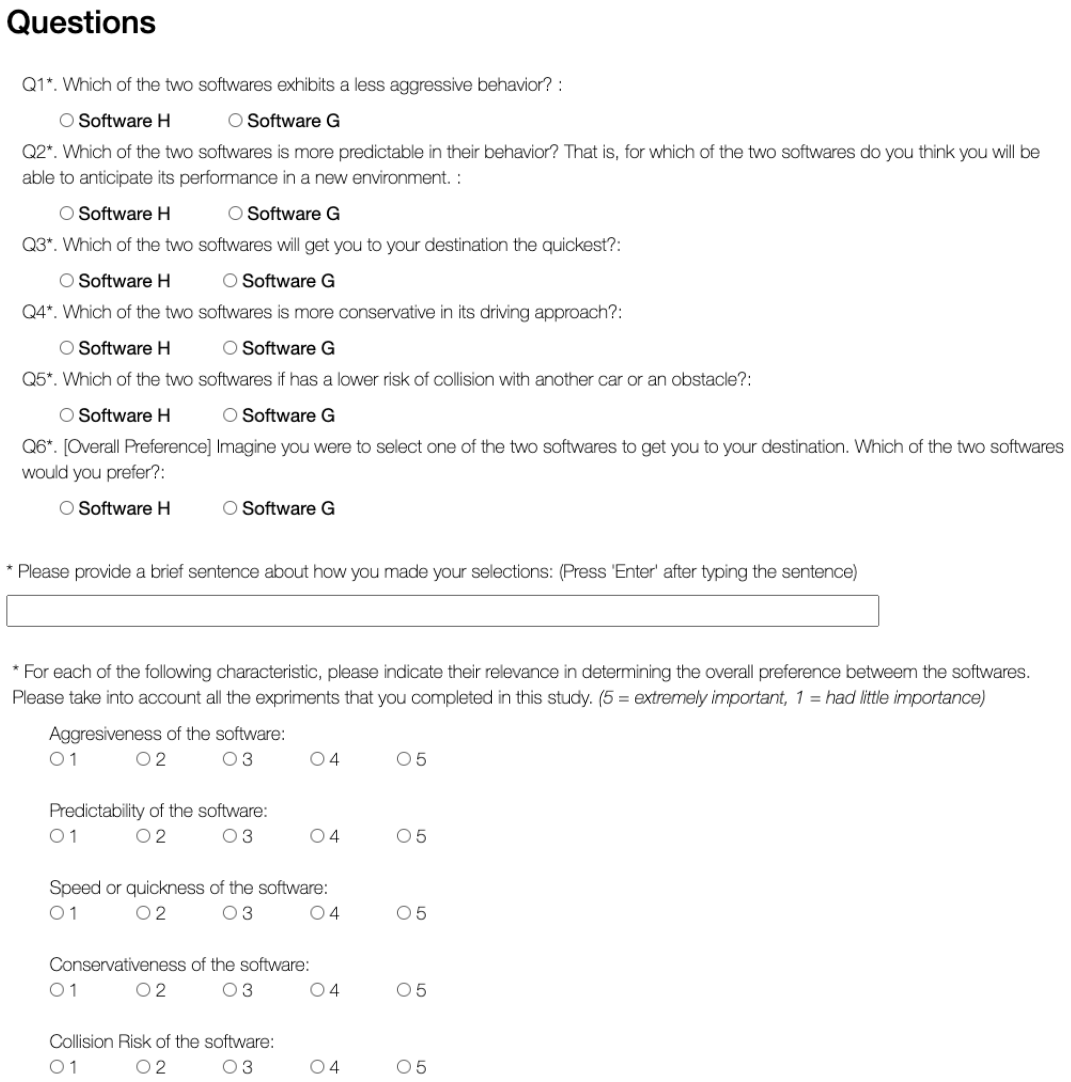
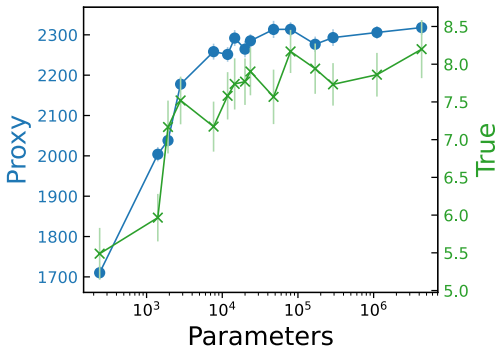


Figure C.3. Layout of the questions panel comprising the 6 comparison questions and the form for reporting the relevance of each criterion in the overall evaluation.

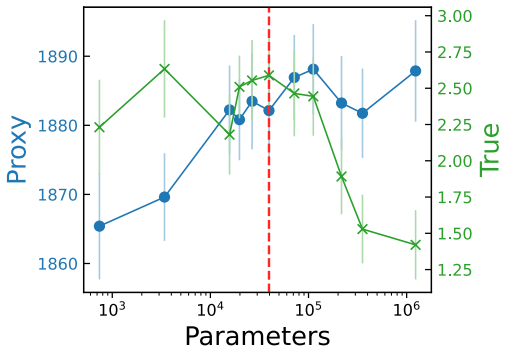
Appendix D

Deferred content from Chapter 5

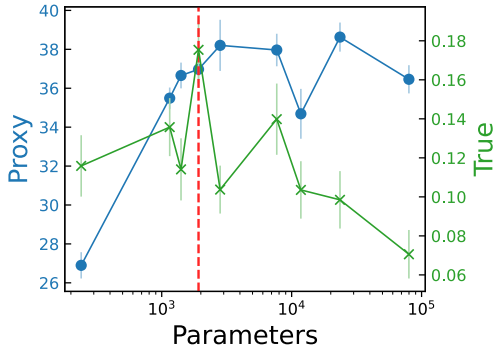
D.1 Mapping The Effects of Reward Misspecification



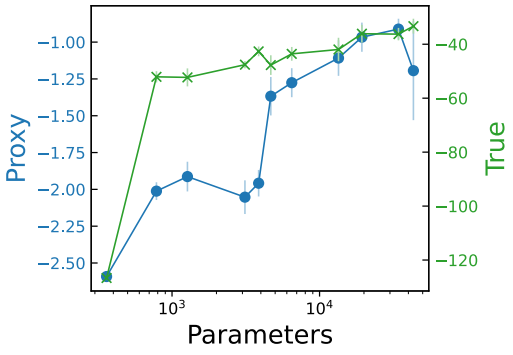
(a) *traffic_merge* - Misweighting



(b) *traffic_bottle* - Misweighting]

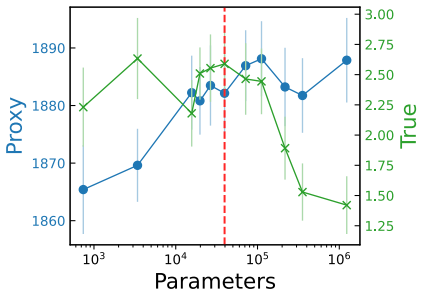


(c) *traffic_merge* - Space

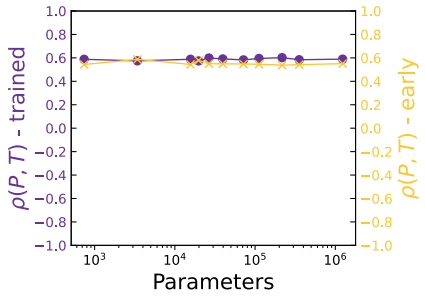


(d) COVID - Misweighting

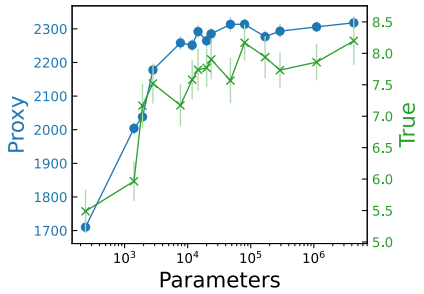
Figure D.1. Additional model size scatter plots. Observe that not all misspecifications cause misalignment. We plot the proxy reward with "•" and the true reward with "×". The proxy reward is measured on the left-hand side of each figure and the true reward is measured on the right hand side of each figure.



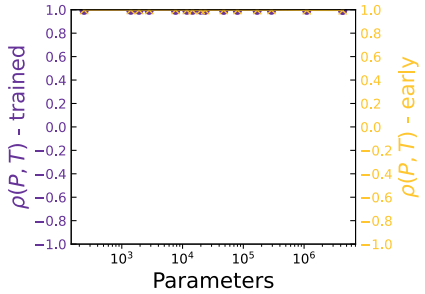
(a) *traffic_bottle* - Misweighting



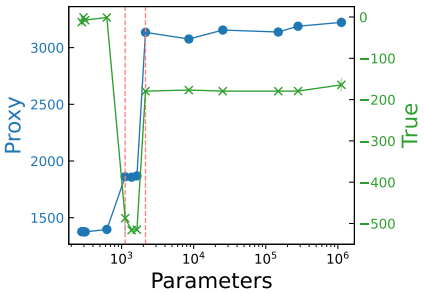
(b) Correlation for Figure D.2a



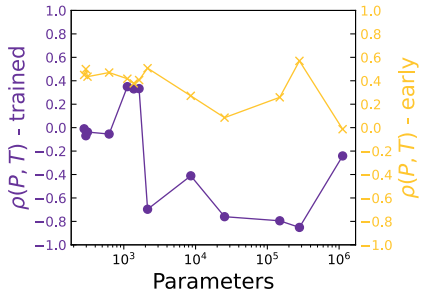
(c) *traffic_merge* - Misweighting



(d) Correlation for Figure D.2c



(e) *traffic_merge* - Ontological



(f) Correlation for Figure D.2e

Figure D.2. Correlations between the proxy and true rewards, along with the reward hacking induced. In the left column, we plot the proxy reward with "•" and the true reward with "×". In the right column, we plot the trained checkpoint correlation and the randomly initialized checkpoint correlation.

Effect of Model Size

We plot the proxy and true reward vs. model size in Figure D.1, following the experiment described in Section 5.3.

Correlation between Proxy and True Rewards

We plot the correlation between proxy and true rewards, following the experiment described in Section 5.3. Interestingly, we see that reward hacking still occurs when there is positive correlation between the true and proxy rewards, e.g., in Figures D.2a/D.2b. Unsurprisingly,

Env. - Misspecification	# Policies	# Problematic	Rollout length	Trusted policy size
Traffic-Mer - misweighting	10	7	270	[96, 96]
Traffic-Mer - scope	16	9	270	[16, 16]
Traffic-Mer - ontological	23	7	270	[4]
Traffic-Bot - misweighting	12	9	270	[64, 64]
COVID - ontological	13	6	200	[16, 16]

Table D.1. Benchmark statistics. We average over 5 rollouts in traffic and 32 rollouts in COVID.

proxy-true pairs which are highly correlated, e.g., Figure D.2c/D.2d do not exhibit reward hacking. Finally, proxy-true pairs which are negatively correlated, e.g., Figure D.2e/D.2f exhibit the most reward hacking.

D.2 Polynomaly

Benchmark Statistics

See Table D.1 for Polynomaly's statistics.

Receiver Operating Characteristic Curves

We plot the ROC curves for the detectors described in Section 5.4. Our detectors are calculated as follows.

Let P and Q represent two probability distributions with $M = \frac{1}{2}(P + Q)$. Then the Jensen-Shannon divergence and the Hellinger distance between them is given by

$$\begin{aligned} \mathrm{JSD}(P||Q) &\coloneqq \frac{1}{2}\mathrm{KL}(P||M) + \frac{1}{2}\mathrm{KL}(Q||M) \\ \mathrm{Hellinger}(P,Q) &\coloneqq \frac{1}{2}\int \left(\sqrt{dP} - \sqrt{dQ}\right)^2. \end{aligned} \tag{D.1}$$

Our proposed detectors estimate the distance $\mathcal{D}(\pi_{\text{trusted}}, \pi_{\text{unknown}})$ between the trusted policy π_{trusted} and unknown policy π_{unknown} as follows: We generate r rollouts of π_{unknown}, where $r = 5$ in the traffic environment and $r = 32$ in the COVID environment. Every s steps of a rollout, where $s = 10$ in the traffic environment and $s = 1$ in the COVID environment, we set P to be the action distribution of π_{unknown} given the unknown agent's state at that timestep in the rollout and Q to be the action distribution of π_{trusted} given the unknown agent's state at that timestep in the rollout. Intuitively, if P and Q are far apart, then the trusted agent would have performed a different action than the unknown agent at that given timestep, indicating a possible case of reward hacking. We then compute either $\mathrm{JSD}(P\|Q)$ or $\mathrm{Hellinger}(P, Q)$ following Equation (D.1). These distances are collected every s steps over

the entire rollout, and we calculate metrics on these distances (range, mean, etc.) to assign an anomaly score to the untrusted policy.

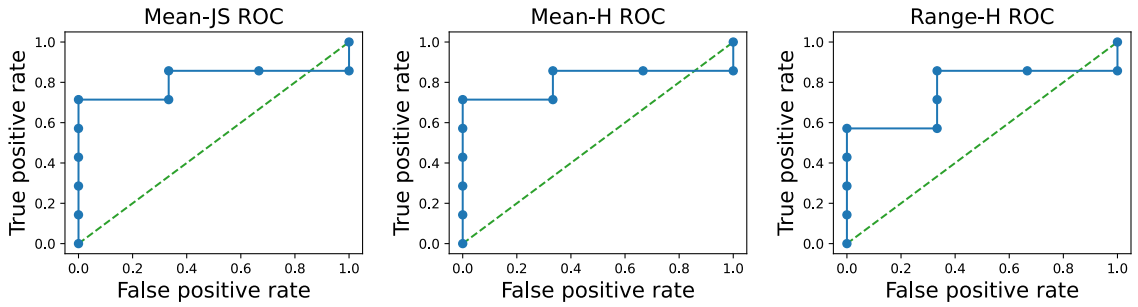


Figure D.3: ROC curves for Traffic-Mer - misweighting.

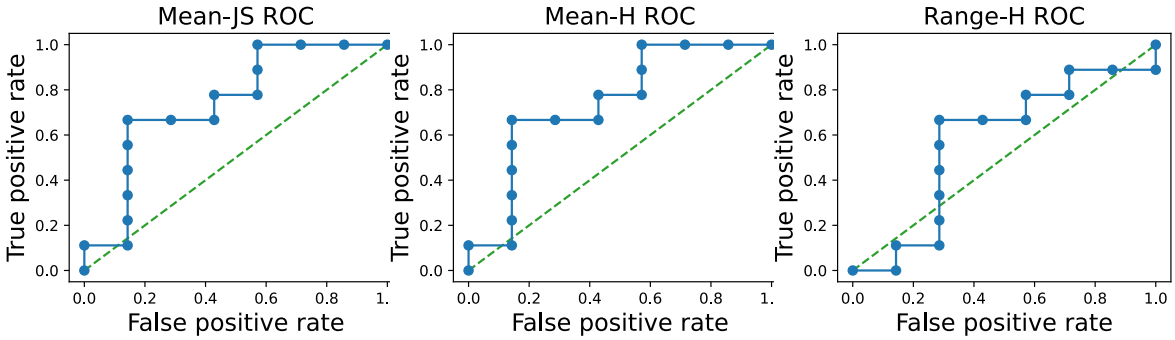


Figure D.4: ROC curves for Traffic-Mer - scope.

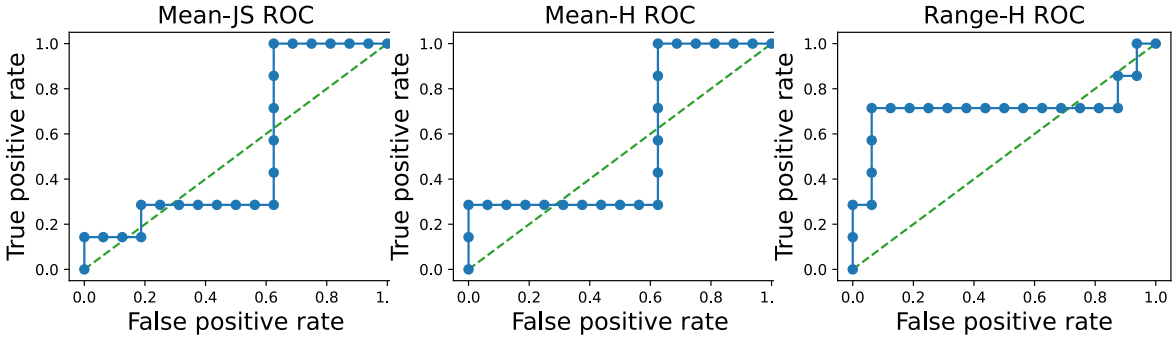


Figure D.5: ROC curves for Traffic-Mer - ontological.

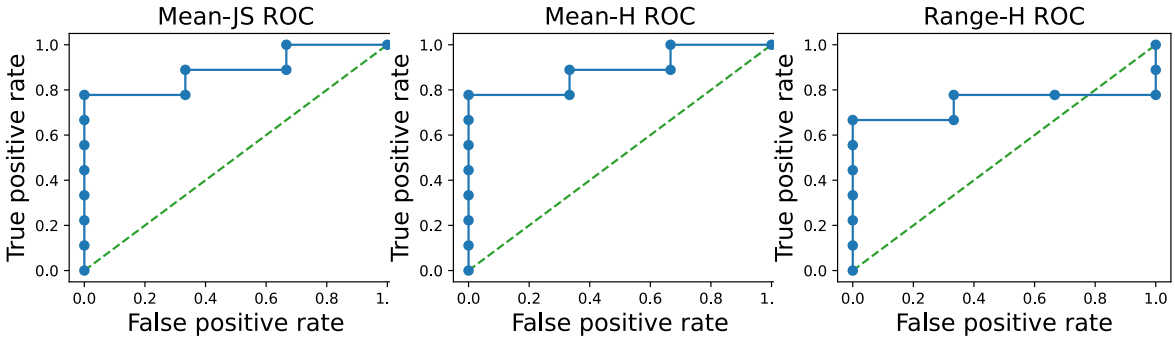


Figure D.6: ROC curves for Traffic-Bot - misweighting.

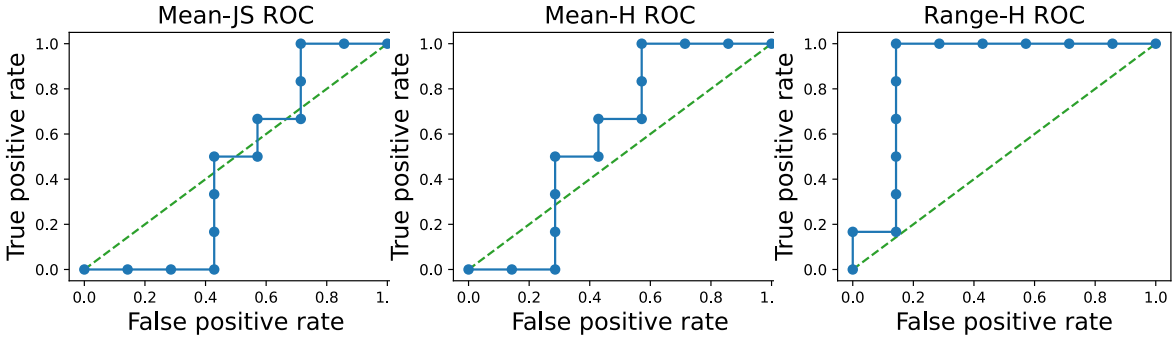


Figure D.7: ROC curves for COVID - ontological.

Appendix E

Deferred content from Chapter 6

E.1 Technical details for proposed framework

RKHS assumption

The Hilbert spaces $\mathbb{H}_\pi$ and $\mathbb{H}_r$ are Reproducing Kernel Hilbert Spaces defined by kernel functions $\mathcal{K}_\pi, \mathcal{K}_r : \mathcal{X} \times \mathcal{X} \mapsto [0, 1]$ respectively defined over a compact instance space $\mathcal{X}$. Further, the kernels $\mathcal{K}_\pi$ and $\mathcal{K}_r$ satisfy the Hilbert-Schmidt condition

$$\int_{\mathcal{X}\times\mathcal{X}} \mathcal{K}_i(x, z)^2 d\mathbb{P}(x) d\mathbb{P}(z) \leq \infty \quad \text{for } i = \{\pi, r\} \,, \tag{E.1}$$

for some distribution $\mathbb{P}$ over space $\mathcal{X}$. Mercer's theorem [143] implies that such kernel functions have an associated set of eigenfunctions (with corresponding eigenvalues) that form an orthonormal basis for $L^2(\mathcal{X}, \mathbb{P})$. We restate a version of this theorem below [209].

Theorem E.1 (Mercer's theorem)**.** *Suppose that the space $\mathcal{X}$ is compact and the positive semi-definite kernel $\mathcal{K}$ satisfies the Hilbert-Schmidt condition (E.1). Then there exists a sequence of eigenfunctions $(\phi_j)_{j=1}^\infty$ that form an orthonormal basis of $L^2(\mathcal{X}, \mathbb{P})$ and non-negative eigenvalues $(\mu_j)_{j=1}^\infty$ such that*

$$\int_{\mathcal{X}} \mathcal{K}(x, z)\phi_j(z) d\mathbb{P}(z) = \mu_j \phi_j(x) \quad \textit{for all } j = 1, 2, \ldots. \tag{E.2}$$

Furthermore, the kernel function has the expansion

$$\mathcal{K}(x, z) = \sum_{j=1}^{\infty} \mu_j \phi_j(x)\phi_j(z) \,, \tag{E.3}$$

where the convergence of the sequence holds absolutely and uniformly.

Conditions for reward boundedness

For learning to be feasible in the proposed framework, we would require that the evaluation functional $F(\pi, r^*)$ is bounded for any policy $\pi \in \mathbb{H}_\pi$. Using the fact that $\|r^*\|_{\mathbb{H}_r} \leq 1$ and $\|\pi\|_{\mathbb{H}_\pi} \leq 1$, we have

$$F(\pi, r^*) = \langle r^*, M\pi \rangle_{\mathbb{H}_r} = (r^*)^\top S_r M \pi \leq \|S_r^{\frac{1}{2}} M S_\pi^{-\frac{1}{2}}\|_{\mathsf{op}} \,. \tag{E.4}$$

Thus one sufficient condition for the reward functional to be bounded is to ensure that the operator norm $\|S_r^{\frac{1}{2}} M S_\pi^{-\frac{1}{2}}\|_{\mathsf{op}}$ is finite. In the special case when the map is diagonal with $M = \mathrm{diag}(\nu_j)$, the above condition simplifies to

$$F(\pi, r^*) \leq \sup_{j \geq 1} \left[\frac{\nu_j \mu_{\pi,j}^{\frac{1}{2}}}{\mu_{r,j}^{\frac{1}{2}}} \right] \,. \tag{E.5}$$

Regularity assumptions on map M

We assume that the map M is a compact bounded operator from the policy space $\mathbb{H}_\pi$ to the reward space $\mathbb{H}_r$. By Schauder's theorem, the adjoint M^* is also a compact operator. Thus, the map $M^*M : \mathbb{H}_\pi \to \mathbb{H}_\pi$ is a compact self-adjoint operator. This allows us to use the spectral theorem for compact self-adjoint operators which guarantees the existence of eignevalues and eignefunctions for the operator M^*M and a corresponding singular value decomposition for the map M [125].

Non-aligned RKHSs

As mentioned in the Section 4.2, if the eigenvectors of the spaces $\mathbb{H}_r$ and $\mathbb{H}_\pi$ are not aligned, one can consider the following simple transformation which resolves this. Let Φ_π and Φ_r represent the eigenvectors.

$$\tilde{r} = \Phi_r r, \quad \tilde{\pi} = \Phi_\pi^\top \pi, \quad \text{and} \quad \tilde{M} = \Phi_r^\top M \Phi_\pi \,. \tag{E.6}$$

The above transformation implies that $\|\tilde{r}\|_{\mathbb{H}_r} \leq 1$ and $\|\tilde{\pi}\|_{\mathbb{H}_\pi} \leq 1$.

E.2 Proof of main results

In this section we provide the proofs for the main results of this work. Appendix E.4 to follow contains the proofs for the other results.

Proof of Theorem 6.1

We begin by proving the result for the special case when the policy set C_π consists of the entire unit ball and then generalize the analysis to arbitrary policy sets.

Case 1: C_π is unit ball in $\mathbb{H}_\pi$. For this special case, observe that the the optimal policy π^* and the plug-in policy $\hat{\pi}_{\mathsf{plug}}$ for any reward estimate $\hat{r}$ can be written as

$$\pi^* = \frac{M^* r^*}{\|M^* r^*\|_{\mathbb{H}_\pi}} \quad \text{and} \quad \hat{\pi}_{\mathsf{plug}} = \frac{M^* \hat{r}}{\|M^* \hat{r}\|_{\mathbb{H}_\pi}} \,, \tag{E.7}$$

where the operator M^* is the adjoint of of the map M. To prove a bound on the excess risk using the plug-in estimate, we use the following lemma which bounds this error in terms a deviation of the estimated and true rewards.

Lemma E.1. *Consider any vectors x and y with finite non-zero norm under some inner product $\langle \cdot, \cdot \rangle$. Then, we have*

$$\langle x, \frac{x}{\|x\|} - \frac{y}{\|y\|} \rangle \leq \frac{\|x - y\|^2}{2\|y\|} \,. \tag{E.8}$$

The proof of the above lemma is presented in Section E.2. Taking the above as given, we can upper bound the excess risk

$$\begin{aligned} \Delta(\hat{\pi}; r^*) &= \langle M^* r^*, \frac{M^* r^*}{\|M^* r^*\|_{\mathbb{H}_\pi}} - \frac{M^* \hat{r}}{\|M^* \hat{r}\|_{\mathbb{H}_\pi}} \rangle_{\mathbb{H}_\pi} \\ &\leq \frac{\|M^*(r^* - \hat{r})\|^2_{\mathbb{H}_\pi}}{2\|M^* \hat{r}\|_{\mathbb{H}_\pi}} \,. \end{aligned} \tag{E.9}$$

Case 2: Arbitrary set C_π. For this case, consider the excess risk of plug-in estimator $\hat{\pi}_{\mathsf{plug}}$ obtained by maximizing reward estimate $\hat{r}$

$$\begin{aligned} \Delta(\hat{\pi}; r^*) &= \langle M^* r^*, \pi^* - \hat{\pi}_{\mathsf{plug}} \rangle_{\mathbb{H}_\pi} \\ &= \langle M^*(r^* - \hat{r}), \pi^* \rangle_{\mathbb{H}_\pi} + \langle M^* \hat{r}, \pi^* - \hat{\pi}_{\mathsf{plug}} \rangle_{\mathbb{H}_\pi} + \langle M^*(\hat{r} - r^*), \hat{\pi}_{\mathsf{plug}} \rangle_{\mathbb{H}_\pi} \\ &\overset{\text{(i)}}{\leq} 2\|M^*(r^* - \hat{r})\|_{\mathbb{H}_\pi} \,, \end{aligned} \tag{E.10}$$

where the final inequality follows from the fact that $\hat{\pi}_{\mathsf{plug}}$ maximizes $F(\pi; \hat{r})$ over the set C_π.

Thus, we see that for both the cases above, we can upper bound the excess risk of the plug-in estimator in terms of the norm $\|M^*(r^* - \hat{r})\|_{\mathbb{H}_\pi}$. Next, we evaluate this for the ridge regression based reward estimator for any set of n queries $\mathcal{Q} = \{\pi_1, \ldots, \pi_n\}$ with covariance matrix $\Sigma = \frac{1}{n} \sum_i \pi_i \pi_i^\top$. For any regularization parameter $\lambda_{\mathsf{reg}} > 0$, we have,

$$\begin{aligned} \hat{r} &= \arg\min_{r \in \mathbb{H}_r} \frac{1}{n} \sum_{i=1}^n (y_i - \langle r, M\pi_i \rangle_{\mathbb{H}_r})^2 + \lambda_{\mathsf{reg}} \|r\|^2_{\mathbb{H}_r} \\ &\overset{\text{(i)}}{=} (M\Sigma M^\top S_r + \lambda_{\mathsf{reg}} I)^{-1} \cdot \frac{1}{n} \sum_{i=1}^n y_i M \pi_i \\ &= r^* - \lambda_{\mathsf{reg}} (M\Sigma M^\top S_r + \lambda_{\mathsf{reg}} I)^{-1} r^* + (M\Sigma M^\top S_r + \lambda_{\mathsf{reg}} I)^{-1} \left(\frac{M}{n} \sum_{i=1}^n \epsilon_i \pi_i \right) , \end{aligned} \tag{E.11}$$

where and equality (i) follows by substituting the value of $y_i = F(\pi_i, r^*) + \epsilon_i$. Let us denote by matrix $A = M\Sigma M^\top S_r + \lambda_{\text{reg}} I$. Therefore, the error in reward estimation

$$\begin{aligned}\hat{r} - r^* &= \lambda_{\text{reg}} A^{-1} r^* + A^{-1}\left(\frac{M}{n}\sum_{i=1}^{n}\epsilon_i \pi_i\right) \\ &\sim \mathcal{N}\left(\lambda_{\text{reg}} A^{-1} r^*, \frac{\tau^2}{n} A^{-1} M\Sigma M^\top A^{-\top}\right) , \end{aligned} \tag{E.12}$$

where the final distribution follows from our assumption on the noise variables $\epsilon_i \sim \mathcal{N}(0, \tau^2)$. Using this above distributional form, we have

$$\begin{aligned}\mathbb{E}[\|M^*(r^* - \hat{r})\|_{\mathbb{H}_\pi}^2] &= \lambda_{\text{reg}}^2 \cdot \langle M^* A^{-1} r^*, M^* A^{-1} r^* \rangle_{\mathbb{H}_\pi} + \frac{\tau^2}{n} \cdot \text{tr}\left[S_\pi M^* A^{-1} M \Sigma_n M^\top A^{-\top} (M^*)^\top\right] \\ &= \lambda_{\text{reg}}^2 \cdot \text{tr}\left[(r^*)^\top A^{-\top} (M^*)^\top S_\pi M^* A^{-1} r^*\right] + \frac{\tau^2}{n} \cdot \text{tr}\left[S_\pi M^* A^{-1} M\Sigma M^\top A^{-\top} (M^*)^\top\right] . \end{aligned} \tag{E.13}$$

The final bound for the general policy set C_π follows from using the above bound with a an application of Jensen's inequality. In order to convert the above bound to a high probability bound, we require an infinite dimensional analog of the Hanson-Wright concentration inequality. Using Theorem 2.6 from Chen and Yang [50] along with equation (E.12), we obtain

$$\Pr(\Delta(\hat{\pi}; r^*) \geq \mathbb{E}[\Delta(\hat{\pi}; r^*)] + t) \leq 2\exp\left(-C\min\left(\frac{t^2}{\|\Gamma\|_{\text{HS}}^2}, \frac{t}{\Gamma\|_{\text{op}}}\right)\right)$$

where the covariance matrix $\Gamma = S_\pi^{\frac{1}{2}} M^* A^{-1} M\Sigma M^\top A^{-\top} (M^*)^\top S_\pi^{\frac{1}{2}}$. □

Proof of Lemma E.1

Let the vector $y = x + \delta_x$ for some difference vector δ_x. Using this, we have

$$\begin{aligned}\langle x, \frac{x}{\|x\|} - \frac{y}{\|y\|}\rangle &= \langle x, \frac{x}{\|x\|} - \frac{x+\delta_x}{\|x+\delta_x\|}\rangle \\ &= \frac{\|x\|}{\|x+\delta_x\|}\left(\|x+\delta_x\| - \|x\| - \frac{\langle x, \delta_x\rangle}{\|x\|}\right) \\ &\overset{\text{(i)}}{\leq} \frac{\|x\|}{\|x+\delta_x\|}\left(\|x\| + \frac{\langle x, \delta_x\rangle}{\|x\|} + \frac{\|\delta_x^2\|}{2\|x\|} - \|x\| - \frac{\langle x, \delta_x\rangle}{\|x\|}\right) \\ &= \frac{\delta_x^2}{2\|x+\delta_x\|} , \end{aligned} \tag{E.14}$$

where (i) follows from using the inequality $\sqrt{a^2 + z} \leq a + \frac{z}{2a}$. This establishes the result. □

Proof of Proposition 6.1

Let us denote the the map $M = \text{diag}(\nu_j)$ and the covariance matrix $\Sigma = \text{diag}(\sigma_j)$. From the upper bound obtained in Theorem 6.1, we have,

$$\begin{aligned}
\mathbb{E}[\|M^*(r^* - \hat{r})\|^2_{\mathbb{H}_\pi}] &= \lambda^2_{\mathsf{reg}} \cdot \|M^* A^{-1} r^*\|^2_{\mathbb{H}_\pi} + \frac{\tau^2}{n^2} \cdot \sum_{i=1}^{n} \|M^* A^{-1} M \pi_i\|^2_{\mathbb{H}_\pi} \\
&\overset{\text{(i)}}{\leq} \lambda^2_{\mathsf{reg}} \cdot \|S_\pi^{\frac{1}{2}} M^* A^{-1} S_r^{-\frac{1}{2}}\|^2_{\mathsf{op}} + \frac{\tau^2}{n} \cdot \text{tr}\left[S_\pi M^* A^{-1} M \Sigma M^\top A^{-\top} (M^*)^\top\right] \\
&\overset{\text{(ii)}}{\leq} \lambda^2_{\mathsf{reg}} \cdot \sup_{j \geq 1} \left[\frac{\nu_j^2 \mu_{r,j} \mu_{\pi,j}}{\nu_j^4 \sigma_j^2 + \lambda^2_{\mathsf{reg}} \mu^2_{r,j}}\right] + \frac{\tau^2}{n} \cdot \sup_{j \geq 1} \left[\frac{\nu_j^4 \mu^2_{\pi,j}}{\nu_j^4 \sigma_j^2 + \lambda^2_{\mathsf{reg}} \mu^2_{r,j}}\right], \qquad \text{(E.15)}
\end{aligned}$$

where inequality (i) follows from using the fact that $\|r^*\|_{\mathbb{H}_r} \leq 1$ and inequality (ii) uses the diagonal structure of the map M as well as the fact that each policy $\pi_i \in \mathcal{Q}$ has unit $\mathbb{H}_\pi$-norm.

Recall that the choice of querying strategy queries each scaled eigenfunction $\sqrt{\mu_{\pi,j}}\phi_{\pi,j}$ of the policy space $n^{1-\alpha}$ times. Therefore the j^{th} entry of the covariance matrix Σ is given by

$$\sigma_j = \begin{cases} \frac{\mu_{\pi,j}}{n^\alpha} & \text{for } j \leq n^\alpha \\ 0 & \text{otherwise} \end{cases}. \qquad \text{(E.16)}$$

Plugging the above value of σ_j into equation (E.15), we obtain

$$\begin{aligned}
\mathbb{E}[\|M^*(r^* - \hat{r})\|^2_{\mathbb{H}_\pi}] \leq \max &\left\{\sup_{j \leq n^\alpha} \frac{\lambda^2_{\mathsf{reg}} n^{2\alpha} \zeta_j}{\zeta_j^2 + \lambda^2_{\mathsf{reg}} n^{2\alpha}}, \sup_{j > n^\alpha} \zeta_j\right\} \\
&+ \frac{\tau^2}{n} \cdot \max\left\{\sup_{j \leq n^\alpha} \frac{n^{2\alpha} \zeta_j^2}{\zeta_j^2 + \lambda^2_{\mathsf{reg}} n^{2\alpha}}, \sup_{j > n^\alpha} \frac{\zeta_j^2}{\lambda^2_{\mathsf{reg}}}\right\} \qquad \text{(E.17)}
\end{aligned}$$

This concludes the proof of the proposition. □

Proof of Corollary 6.1

We now derive explicit finial sample rates for the case when the spectrum of the map $M^\top S_r M S_\pi^{-1}$ satisfies a power law decay for some parameter $\beta > 0$. In the notation used in Proposition 6.1, we have the quantity

$$\zeta_j \asymp j^{-\beta}. \qquad \text{(E.18)}$$

Our proof strategy will be to instantiate the bias and variance terms for this setting of ζ_j and finally select a setting for the exploration parameter α and regularization parameter λ_{reg}.

Bounding Bias. The bias term in the proposition is a max over two terms

$$\text{Bias}^2 = \max\left\{\sup_{j\le n^\alpha} \frac{\lambda_{\mathsf{reg}}^2 n^{2\alpha}\zeta_j}{\zeta_j^2 + \lambda_{\mathsf{reg}}^2 n^{2\alpha}}, \sup_{j>n^\alpha} \zeta_j\right\}. \tag{E.19}$$

We consider the two terms in the analysis here separately. For the first term,

$$\sup_{j\le n^\alpha} \frac{\lambda_{\mathsf{reg}}^2 n^{2\alpha}\zeta_j}{\zeta_j^2 + \lambda_{\mathsf{reg}}^2 n^{2\alpha}} = \lambda_{\mathsf{reg}}^2 \sup_{j\le n^\alpha}\left[\frac{1}{\frac{j^{-\beta}}{n^{2\alpha}} + \lambda_{\mathsf{reg}}^2 j^\beta}\right] \le \lambda_{\mathsf{reg}} n^\alpha\ , \tag{E.20}$$

where the final inequality follows from using $a^2 + b^2 \ge 2ab$. For the second term, we have

$$\sup_{j\ge n^\alpha} \zeta_j = \sup_{j\ge n^\alpha} j^{-\beta} = n^{-\alpha\beta}. \tag{E.21}$$

Bounding Variance. Recall that the variance term (assuming $\tau = 1$) is given by

$$\text{Variance} = \frac{1}{n}\cdot\max\left\{\sup_{j\le n^\alpha} \frac{n^{2\alpha}\zeta_j^2}{\zeta_j^2 + \lambda_{\mathsf{reg}}^2 n^{2\alpha}}, \sup_{j>n^\alpha} \frac{\zeta_j^2}{\lambda_{\mathsf{reg}}^2}\right\}\ . \tag{E.22}$$

We again consider both terms of the maximum separately. For the first term,

$$\frac{1}{n}\cdot\sup_{j\le n^\alpha} \frac{n^{2\alpha}\zeta_j^2}{\zeta_j^2 + \lambda_{\mathsf{reg}}^2 n^{2\alpha}} \le n^{2\alpha-1}\ , \tag{E.23}$$

where the inequality follows from ignoring the term $\lambda_{\mathsf{reg}}^2 n^{2\alpha}$ in the denominator. For the second variance term,

$$\sup_{j>n^\alpha} \frac{\zeta_j^2}{n\lambda_{\mathsf{reg}}^2} = \frac{n^{-2\alpha\beta}}{\lambda_{\mathsf{reg}}^2 n}\ . \tag{E.24}$$

Setting regularization parameter. By setting $\lambda_{\mathsf{reg}} > n^{-\alpha\beta-\alpha}$, we can have that the bias term is dominated by $\lambda_{\mathsf{reg}} n^\alpha$. Similarly, the above setting also implies that the variance term is dominated by $n^{2\alpha-1}$. Combing these observations, we have that the expected error is upper bounded by

$$\Delta(\hat{\pi}_{\mathsf{plug}}; r^*) \le \lambda_{\mathsf{reg}} n^\alpha + n^{2\alpha-1} \quad \text{where } \lambda_{\mathsf{reg}} > n^{-\alpha\beta-\alpha}. \tag{E.25}$$

Setting $\lambda_{\mathsf{reg}} = n^{-\alpha(\beta+1)}$ and then $\alpha = \frac{1}{\beta+2}$, we get that

$$\Delta(\hat{\pi}_{\mathsf{plug}}; r^*) \le n^{-\frac{\beta}{\beta+2}}. \tag{E.26}$$

This completes the proof of the corollary. □

Proof of Theorem 6.2

In order to prove the general theorem, we exhibit a transformation which allows us to reduce the problem to that with the diagonal structure described in Proposition 6.1.

We will consider orthogonally diagonalizable matrices S_r and S_π which represent the eigenvectors and eigenvalues of the Hilbert spaces $\mathbb{H}_r$ and $\mathbb{H}_\pi$. Consider the following set of transformations for any reward $r \in C_r$ and policy $\pi \in C_\pi$.

$$\tilde{r} = S_r^{\frac{1}{2}} r, \quad \tilde{\pi} = S_\pi^{\frac{1}{2}} \pi, \quad \tilde{M} = S_r^{\frac{1}{2}} M S_\pi^{-\frac{1}{2}}. \tag{E.27}$$

With this transformation, we can rewrite the objective function above

$$\max_{\tilde{\pi}} \langle \tilde{r}, \tilde{M}\tilde{\pi} \rangle \quad \text{s.t.} \quad \langle \tilde{\pi}, \tilde{\pi} \rangle = 1 \text{ and } \langle \tilde{r}, \tilde{r} \rangle = 1 \ ,$$

where the inner product $\langle \cdot, \cdot \rangle$ denotes the standard ℓ_2 inner product. Observe that we have overloaded notation to denote by $\tilde{r}^* = \tilde{r}$. Further, using these above transformations, we can rewrite the adjoint operator

$$M^* = S_\pi^{-1} M^\top S_r = S_\pi^{-\frac{1}{2}} (S_r^{\frac{1}{2}} M S_\pi^{-\frac{1}{2}})^\top S_r^{\frac{1}{2}} = S_\pi^{-\frac{1}{2}} \tilde{M}^\top S_r^{\frac{1}{2}} \ . \tag{E.28}$$

Recall from Theorem 6.1, the matrix

$$A = M \Sigma M^\top S_r + \lambda_{\mathsf{reg}} I = S_r^{-\frac{1}{2}} \left[\tilde{M} \tilde{\Sigma} \tilde{M}^\top + \lambda_{\mathsf{reg}} I \right] S_r^{\frac{1}{2}} \ , \tag{E.29}$$

where the covariance matrix $\tilde{\Sigma} = \frac{1}{n} \sum_i \tilde{\pi} \tilde{\pi}^\top$. We have used the fact here that the matrices S_π and S_r are orthogonally diagonalizable and hence symmetric. Finally, we will denote the singular value decomposition of the compact map M in the matrix form as

$$\tilde{M} = U_{\tilde{M}} \Lambda_{\tilde{M}} V_{\tilde{M}}^\top \ .$$

The existence of such a decomposition is guaranteed by the regularity assumptions we consider on the map M in Appendix E.1. We will now analyze the bias and the variance terms from the upper bound on $\mathbb{E}[\|M^*(r^* - \hat{r}\|_{\mathbb{H}_\pi}^2]$ from Theorem 6.1.

Bound on bias. The squared bias term is given by

$$\begin{aligned}
\lambda_{\mathsf{reg}}^{-2} \cdot \text{Bias}^2 &= r^\top A^{-\top} (M^*)^\top S_\pi M^* A^{-1} r \\
&= r^\top S_r^{\frac{1}{2}} S_r^{-\frac{1}{2}} \cdot S_r^{\frac{1}{2}} (\tilde{M} \tilde{\Sigma} \tilde{M}^\top + \lambda_{\mathsf{reg}} I)^{-1} S_r^{-\frac{1}{2}} \cdot S_r^{\frac{1}{2}} \tilde{M} S_\pi^{-\frac{1}{2}} \cdot S_\pi \cdot M^* A^{-1} r \\
&= \tilde{r}^\top (\tilde{M} \tilde{\Sigma} \tilde{M}^\top + \lambda_{\mathsf{reg}} I)^{-1} \tilde{M} \cdot S_\pi^{\frac{1}{2}} S_\pi^{-\frac{1}{2}} \tilde{M}^\top S_r^{\frac{1}{2}} \cdot S_r^{-\frac{1}{2}} (\tilde{M} \tilde{\Sigma} \tilde{M}^\top + \lambda_{\mathsf{reg}} I)^{-1} S_r^{\frac{1}{2}} r \\
&= \tilde{r}^\top (\tilde{M} \tilde{\Sigma} \tilde{M}^\top + \lambda_{\mathsf{reg}} I)^{-1} \tilde{M} \cdot \tilde{M}^\top (\tilde{M} \tilde{\Sigma} \tilde{M}^\top + \lambda_{\mathsf{reg}} I)^{-1} \tilde{r} \\
&= \tilde{r}^\top U_{\tilde{M}} (\Lambda_{\tilde{M}} V_{\tilde{M}}^\top \tilde{\Sigma} V_{\tilde{M}} \Lambda_{\tilde{M}} + \lambda_{\mathsf{reg}} I)^{-1} \Lambda_{\tilde{M}}^2 (\Lambda_{\tilde{M}} V_{\tilde{M}}^\top \tilde{\Sigma} V_{\tilde{M}} \Lambda_{\tilde{M}} + \lambda_{\mathsf{reg}} I)^{-1} U_{\tilde{M}}^\top \tilde{r} \ ,
\end{aligned} \tag{E.30}$$

where we have used the SVD decomposition for the matrix $\tilde{M}$ in the last step.

Bound on variance. The variance term is given by

$$\begin{aligned}\text{Var} &= \frac{\tau^2}{n} \cdot \text{tr}\left[S_\pi M^* A^{-1} M \Sigma_n M^\top A^{-\top} (M^*)^\top\right] \\ &= \frac{\tau^2}{n} \cdot \text{tr}\left[\tilde{M}^\top (\tilde{M}\tilde{\Sigma}\tilde{M}^\top + \lambda_{\mathsf{reg}} I)^{-1} \tilde{M}\tilde{\Sigma}\tilde{M}^\top (\tilde{M}\tilde{\Sigma}\tilde{M}^\top + \lambda_{\mathsf{reg}} I)^{-1} \tilde{M}\right] \\ &= \frac{\tau^2}{n} \cdot \text{tr}\left[\Lambda_{\tilde{M}} (\Lambda_{\tilde{M}} V_{\tilde{M}}^\top \tilde{\Sigma} V_{\tilde{M}} \Lambda_{\tilde{M}} + \lambda_{\mathsf{reg}} I)^{-1} \Lambda_{\tilde{M}} V_{\tilde{M}}^\top \tilde{\Sigma} V_{\tilde{M}} \Lambda_{\tilde{M}} (\Lambda_{\tilde{M}} V_{\tilde{M}}^\top \tilde{\Sigma} V_{\tilde{M}} \Lambda_{\tilde{M}} + \lambda_{\mathsf{reg}} I)^{-1} \Lambda_{\tilde{M}}\right] .\end{aligned} \tag{E.31}$$

Finally, by making a substitution for reward $\tilde{r} = U_{\tilde{M}}^\top \tilde{r}$ and policy $\tilde{\pi} = V_{\tilde{M}}^\top \tilde{\pi}$ in equations (E.30) and (E.31), we recover back the bias variance expressions used in the analysis for Proposition 6.1. What remains to be shown is that our particular choice of query policies correspond to basis vectors in this transformed space. For this, observe that the sampling policies

$$\pi_j = \sum_{i=1}^{\infty} \sqrt{\mu_{\pi,i}} \cdot \langle \phi_{\tilde{M},j}, \phi_{\pi,i} \rangle \phi_{\pi,i} \quad \text{for } j \leq n^\alpha ,$$

is such that the transformed policies

$$\tilde{\pi}_j = V_{\tilde{M}}^\top S_\pi^{\frac{1}{2}} \pi_j = V_{\tilde{M}}^\top S_\pi^{\frac{1}{2}} \cdot S_\pi^{-\frac{1}{2}} V_{\tilde{M}} e_j = e_j , \tag{E.32}$$

indeed correspond to the basis vector. This finishes the proof of the desired claim. □

Proof of Corollary 6.2

The proof of this corollary follows similar to that of Corollary 6.1 in terms of bounding the bias and the variance. The final rate follows by an application of Jensen's inequality to conclude

$$\mathbb{E}[\|M^*(r^* - \hat{r})\|_{\mathbb{H}_\pi}] \leq (\mathbb{E}[\|M^*(r^* - \hat{r})\|_{\mathbb{H}_\pi}^2])^{\frac{1}{2}} . \tag{E.33}$$

The final rate that we get in this case is thus upper bounded by the square root of the rate observed in Corollary 6.1. This concludes the proof. □

E.3 Gaussian process bandit optimization

In this section, we discuss in detail the application of our framework to the problem of frequentist Gaussian process bandit optimization, also known as Kernelized multi-armed bandits (MAB) problem. Recall the reduction of the Kernel MAB problem to our setup required us to define three elements.

Reward space $\mathbb{H}_r$. Given the RKHS $\mathbb{H}$ as well as the elements of the cover $\mathcal{C}_\epsilon$, we view the reward function as a map from $\mathcal{C}_\epsilon$ to $\mathbb{R}$, or equivalently as a vector in $\mathbb{R}^{N_{\mathsf{cov}}(\epsilon)}$. More precisely, letting $\tilde{f} = [f(x_1), \ldots, f(x_{N_{\mathsf{cov}}})]$ denote the vector of evaluations of a function f, we define

$$\begin{aligned} \mathbb{H}_r &:= \operatorname{span}\{\tilde{f} \mid f \in \mathbb{H}\} \\ \text{with } \langle \tilde{f}_1, \tilde{f}_2 \rangle_{\mathbb{H}_r} &:= \tilde{f}_1^\top K^{-1} \tilde{f}_2 \end{aligned}, \tag{E.34}$$

where $\langle \cdot, \cdot \rangle$ represents the standard ℓ_2 inner product. With this notation, let us define the true reward $r^* := \tilde{f}^* = [f^*(x_1), \ldots, f^*(x_{N_{\mathsf{cov}}})]$.

Policy Space $\mathbb{H}_\pi$. For the policy space $\mathbb{H}_\pi$ in our setup, we let

$$\begin{aligned} \mathbb{H}_\pi &:= \operatorname{span}\{k_x = [\mathcal{K}(x, x_1), \ldots, \mathcal{K}(x, x_{N_{\mathsf{cov}}})] \in \mathbb{R}^{N_{\mathsf{cov}}} \mid x \in \mathcal{C}_\epsilon\} \\ &\text{with } \langle k_1, k_2 \rangle_{\mathbb{H}_\pi} := \langle k_1, K^{-2} k_2 \rangle \ . \end{aligned} \tag{E.35}$$

The choice of the above norm ensures that

$$\langle k_i, k_j \rangle_{\mathbb{H}_\pi} = \langle K^{-1} k_i, K^{-1} k_j \rangle = \langle e_i, e_j \rangle = \delta_{i,j} \quad \text{for all} \quad (x_i, x_j) \in \mathcal{C}_\epsilon \times \mathcal{C}_\epsilon \ .$$

For the policy space $\mathbb{H}_\pi$, we have created an orthonormal embedding of the set of vectors $\{k_x\}_{x \in \mathcal{C}}$. Observe that this policy set that we construct satisfies the regularity Assumption 6.1 because each vector k is an eigenvector of the space $\mathbb{H}_\pi$.

Map M. By our assumption that the kernel $\mathcal{K}$ is a Mercer's kernel, we have that $\mathbb{H}_\pi \subseteq \mathbb{H}_r$, that is, for all $x \in \mathcal{C}$, the vector $k_x \in \mathbb{H}_r$. Furthermore, both $\mathbb{H}_r$ and $\mathbb{H}_\pi$ are sub-spaces of $\mathbb{R}^{N_{\mathsf{cov}}}$ and we can take the map $M = I_{N_{\mathsf{cov}}}$.

With these definitions, we now explicitly establish a correspondence between our doubly nonparameteric bandit problem and the Kernel MAB problem.

Connecting the problems

Given an RKHS $\mathbb{H}$ with an associated Mercer's kernel $\mathcal{K}$, the objective of the zeroth-order bandit optimization problem is

$$\max_{x \in \mathcal{X}} f^*(x) \quad \text{such that} \quad \|f^*\|_{\mathbb{H}} \leq 1 \ , \tag{P1}$$

with access to oracle

$$\mathcal{O}_{f^*} : x \mapsto f^*(x) + \eta \quad \text{where } \eta \sim \mathcal{N}(0, \tau^2) \ .$$

Equivalently, the objective in our reward learning framework is

$$\max_{\pi \in \mathbb{H}_\pi} \langle r^*, \pi \rangle_{\mathbb{H}_r} \quad \text{such that} \quad \|r^*\|_{\mathbb{H}_r} \leq 1 \text{ and } \|\pi\|_{\mathbb{H}_\pi} \leq 1 \ , \tag{P2}$$

with the corresponding spaces and inner products are defined in the previous section. The oracle required in our setup responds with

$$\mathcal{O}_{r^*} : \pi \mapsto \langle r^*, \pi \rangle_{\mathbb{H}_r} + \eta \quad \text{where } \eta \sim \mathcal{N}(0, \tau^2) \ ,$$

for any policy $\pi \in \mathbb{H}_\pi$ such that $\|\pi\|_{\mathbb{H}_\pi} \leq 1$. Our first lemma below states that obtaining such a n oracle is indeed feasible if we are able to restrict our queries π to include only points k_x for which the vector $k_x \in \mathcal{C}_\epsilon$.

Lemma E.2. *Given access to oracle $\mathcal{O}_{f^*}$ for a function f^*, the corresponding oracle $\mathcal{O}_{r^*}$ can be implemented when the query set consists of $\{k_x\}_{x \in \mathcal{C}_\epsilon}$.*

Proof. For any query point k, the oracle $\mathcal{O}_{r^*}$ needs to compute the value $\langle r^*, k \rangle_{\mathbb{H}_r} = f^*(x)$. Thus, these two oracles on the provided query set are exactly identical. □

Lemma E.3. *For any $f^* \in \mathbb{H}$ satisfying $\|f^*\|_{\mathbb{H}} \leq 1$, we have that $\|r^*\|_{\mathbb{H}_r} \leq 1$.*

Proof. Observe that an alternate way to define the RKHS norm is given by

$$\|f\|_{\mathbb{H}} := \sup_{S \subseteq \mathcal{X}; |S| \leq \infty} f|_S K_S^{-1} f|_S \ .$$

The fact that $\|r^*\|_{\mathbb{H}_r}$ is computed on $\mathcal{C}_\epsilon \subset \mathcal{X}$ establishes the desired claim. □

Finally, we turn to establishing a relation between the solutions obtained from solving the relaxed problem (P2) as compared to solving the original problem (P1). We denote the corresponding maximizers for both problems

$$x^* \in \arg\max_{x \in \mathcal{X}} f^*(x) \quad \text{and} \quad x_\pi^* \in \arg\max_{x \in \mathcal{C}_\epsilon} \langle r^*, k_x \rangle_{\mathbb{H}_r} \ , \tag{E.36}$$

The following lemma now relates both these maximizers together.

Lemma E.4. *For an RKHS $\mathbb{H}$ with kernel $\mathcal{K}$ satisfying Assumption 6.2 with constant $L_{\mathcal{K}} > 0$ and any function $f^* \in \mathbb{H}$, let $x^* \in \mathcal{X}$ and $x_\pi^* \in \mathcal{C}_\epsilon$ be the maximizers as defined in equation (E.36), we have*

$$f^*(x_\pi^*) \geq f^*(x^*) - \sqrt{2cL_{\mathcal{K}}\epsilon} \ . \tag{E.37}$$

Proof. Denote by $\Pi_{\mathcal{C}_\epsilon}(x^*) := \arg\min_{x \in \mathcal{C}_\epsilon} \|x^* - x\|_2$ the projection of the point x^* onto the set $\mathcal{C}_\epsilon$. Then, we have

$$\begin{aligned} f^*(x^*) - f^*(x_\pi^*) &= f^*(x^*) - f^*(\Pi_{\mathcal{C}_\epsilon}(x^*)) + f^*(\Pi_{\mathcal{C}_\epsilon}(x^*)) - f^*(x_\pi^*) \\ &\leq \sqrt{2cL_{\mathcal{K}}\epsilon} \ . \end{aligned}$$

This completes the proof of the lemma. □

The above lemma shows that solving Problem P2 is equivalent to solving Problem P1 up to an additive factor of $\sqrt{2cL_{\mathcal{K}}\epsilon}$ when we are working with an ϵ-cover over the domain space.

Analysis for bandit optimization

Recall from the previous section that the quantity which determines the rate of decay is the ratio of eigenvalues

$$\zeta_j = \frac{\hat{\mu}_{\pi,j}}{\hat{\mu}_{r,j}} = \frac{\hat{\mu}^2_{r,j}}{\hat{\mu}_{r,j}} = \hat{\mu}_{r,j} \ ,$$

where $\hat{\mu}_{r,j}$ is the j^{th} eigenvalue of the kernel matrix K. Let us denote by $\mathbb{P}$ denote the uniform distribution over the input space $\mathcal{X}$ and let us suppose that the cover N_{cov} is formed using random samples from this distribution. Let us denote by $\{\mu_j\}$ the eigenvalues and by ϕ_j the corresponding eigen vectors of the Mercer kernel $\mathcal{K}$. For every point $x \in \mathcal{X}$, let us denote by

$$\Phi(x) := \left(\sqrt{\mu_j}\phi_j(x)\right)_{j=1}^{\infty} \ ,$$

the corresponding featurization of the point x. Then, for $S := \mathbb{E}_{x\sim\mathbb{P}}[\Phi(x)\Phi(x)^\top]$, we have

$$[S]_{j,k} = [\mathbb{E}_{x\sim\mathbb{P}}[\Phi(x)\Phi(x)^\top]]_{j,k} = \mathbb{E}_{x\sim\mathbb{P}}[\sqrt{\mu_j}\sqrt{\mu_k}\phi_j(x)\phi_k(x)] = \mu_j\delta_{j,k} \ . \tag{E.38}$$

Observe that the kernel matrix K and the (scaled) sample covariance matrix $N_{\mathsf{cov}} \cdot \hat{S} = \sum_{x\in\mathcal{C}} \Phi(x)\Phi(x)^\top$ are similar matrices and thus have the same eigenvalues. The following lemma, adapted from Koltchinskii and Lounici [121, Theorem 9] relates the eigenvalues of the sample covariance matrix $\hat{S}$ to those of the underlying kernel $\mathcal{K}$.

Lemma E.5. *For any $\lambda_S > 0$ and size of the cover satisfying $N_{\mathsf{cov}}(\epsilon) > c \cdot \frac{\mathrm{tr}(S(S+\lambda_S I)^{-1})}{\epsilon_S^2} + \frac{1}{\epsilon_S^2}\log\left(\frac{1}{\delta}\right)$, we have,*

$$\hat{\mu}_j \leq (1+\epsilon_S)\mu_j + \lambda_S\epsilon_S \quad \textit{for all } j \ , \tag{E.39}$$

with probability at least $1-\delta$.

The following corollary of Lemma E.5 provides us with a way to control the deviation of the eigenvalues $\hat{\mu}_j$ from the corresponding μ_j in a multiplicative manner.

Corollary E.1. *For any value of decay parameter $\beta > 1$ and $\gamma < \beta$, we have, for all j, the eigenvalues*

$$\hat{\mu}_j \leq \frac{3}{2}\mu_j + \frac{N_{\mathsf{cov}}^{-\gamma}}{2} \ , \tag{E.40}$$

with high probability.

Proof. Let us understand the condition $N_{\mathsf{cov}}(\epsilon) \gg \frac{\mathrm{tr}(S(S+\lambda_S I)^{-1})}{\epsilon_S^2}$ and see what restrictions it puts on the value of the covering number. Lets suppose that the true eigen values $\mu_j \asymp j^{-\beta}$

and we set the value of $\lambda_S \asymp N_{\mathsf{cov}}^{-\gamma}$. Therefore, the sum

$$\sum_j \frac{j^{-\beta}}{j^{-\beta}+\lambda_S} \lesssim N_{\mathsf{cov}}^{\frac{\gamma}{\beta}} + \frac{1}{N_{\mathsf{cov}}^{-\gamma}} \sum_{j > N_{\mathsf{cov}}^{\frac{\gamma}{\beta}}} j^{-\beta}$$
$$\lesssim N_{\mathsf{cov}}^{\frac{\gamma}{\beta}} + \frac{N_{\mathsf{cov}}^{\frac{\gamma}{\beta}}}{\beta-1} .$$

Thus, if we set $\epsilon_S = \frac{1}{2}$, then for any $\beta > 1$ and $\gamma < \beta$, the above condition on the covering number will be satisfied and we get desired bound on the deviation of the empirical eigenvalues from population eigenvalues. □

The above corollary is essential to our argument because often times we have a good understanding of the decay of the eigenvalues of the kernel $\mathcal{K}$ associated with the RKHS and this allows us to relate the set of empirical eigenvalues to these.

We now present a proof of Theorem 6.3, restated below, which upper bounds the excess risk for this setup. We will then use a batch to online conversion bound to convert this to a regret bound and specialize to the Matérn kernel later.

Theorem E.2 (Restated Theorem 6.3)**.** *Suppose that the eigenvalues of a $L_{\mathcal{K}}$-Lipschitz kernel $\mathcal{K}$ with respect to a distribution $\mathbb{P}$ over $\mathcal{X}$ satisfy the power-law decay $\mu_j \asymp j^{-\beta}$. Let $\hat{x}_{\mathsf{plug}}$ be the output of Algorithm 3 using n queries to the oracle $\mathcal{O}_{f^*}$. Then, for any value of $\gamma \in (1 + \frac{1}{d}\frac{\log(1/\epsilon)}{\log(L_{\mathcal{K}}/\epsilon^2)}, \beta)$ and $\epsilon \in (0,1)$, the excess risk*

$$\max_x f^*(x) - f^*(\hat{x}_{\mathsf{plug}}) \lesssim N_{\mathsf{cov}}^{\frac{1}{\beta+2}}(\epsilon) \cdot n^{\frac{-\beta}{2(\beta+2)}} + N_{\mathsf{cov}}^{\frac{1-\gamma}{2}}(\epsilon) + \sqrt{L_{\mathcal{K}}\epsilon} ,$$

with high probability.

Proof. Our strategy, as before, will be to explore n^α directions and assume $\tau^2 = 1$. Recall, that for symmetric matrices, Theorem 6.2, the excess error of the plug-in estimator can be upper bounded as

$$\mathbb{E}[\Delta(\hat{\pi}_{\mathsf{plug}}; r^*)]^2 \le \lambda_{\mathsf{reg}}^2 \sup_{j\ge 1} \left[\frac{1}{\frac{\nu_j^2\sigma_j^2}{\mu_{\pi,j}\mu_{r,j}} + \frac{\lambda_{\mathsf{reg}}^2\mu_{r,j}}{\mu_{\pi,j}\nu_j^2}} \right] + \frac{1}{n} \sup_{j\ge 1} \left[\frac{\nu_j^4\mu_{\pi,j}^2}{\nu_j^4\sigma_j^2 + \lambda_{\mathsf{reg}}^2\mu_{r,j}^2} \right] .$$

Bounding Bias. We will split the analysis into two cases.

Case 1: $j > n^\alpha$. For this case, we have that $\sigma_j = 0$ and therefore

$$\lambda_{\mathsf{reg}}^2 \sup_{j>n^\alpha} \frac{\hat{\mu}_{\pi,j}}{\lambda_{\mathsf{reg}}^2\hat{\mu}_{r,j}} = \sup_{j>n^\alpha} N_{\mathsf{cov}}\hat{\mu}_j \lesssim \sup_{j>n^\alpha} N_{\mathsf{cov}}(\mu_j + N_{\mathsf{cov}}^{-\gamma}) \le N_{\mathsf{cov}} n^{-\alpha\beta} + N_{\mathsf{cov}}^{1-\gamma} , \tag{E.41}$$

with the above holding with high probability from an application of Corollary E.1 for any $1 < \gamma < \beta$.

Case 2: $j \leq n^\alpha$. For this case, we have $\sigma_j = \frac{\mu_{\pi,j}}{n^\alpha}$. The bias can then be upper bounded as

$$\lambda_{\mathsf{reg}}^2 \sup_{j \leq n^\alpha} \left[\frac{1}{\frac{\nu_j^2 \mu_{\pi,j}}{n^{2\alpha}\mu_{r,j}} + \frac{\lambda_{\mathsf{reg}}^2 \mu_{r,j}}{\mu_{\pi,j}\nu_j^2}} \right] \leq \lambda_{\mathsf{reg}} n^\alpha \,, \tag{E.42}$$

where the final inequality follows from using $a^2 + b^2 \geq 2ab$.

Bounding variance. As we did in the section above, let us split the analysis into two cases.

Case 1: $j > n^\alpha$. For this case, the variance term simplifies to

$$\frac{1}{n} \sup_{j > n^\alpha} \left[\frac{\mu_{\pi,j}^2}{\lambda_{\mathsf{reg}}^2 \mu_{r,j}^2} \right] = \frac{1}{\lambda_{\mathsf{reg}}^2 n} \sup_{j > n^\alpha} \left[N_{\mathsf{cov}}^2 \hat{\mu}_j^2 \right] \leq \frac{N_{\mathsf{cov}}^2}{\lambda_{\mathsf{reg}}^2 n} \sup_{j > n^\alpha} \left[\hat{\mu}_j^2 \right] \lesssim \frac{N_{\mathsf{cov}}^2 n^{-2\alpha\beta} + N_{\mathsf{cov}}^{2(1-\gamma)}}{\lambda_{\mathsf{reg}}^2 n} \,. \tag{E.43}$$

Case 2: $j \leq n^\alpha$. For the second case, we can upper bound the variance term

$$\frac{1}{n} \sup_{j \leq n^\alpha} \left[\frac{\nu_j^4 \mu_{\pi,j}^2}{\frac{\nu_j^4 \mu_{\pi,j}^2}{n^{2\alpha}} + \lambda_{\mathsf{reg}}^2 \mu_{r,j}^2} \right] \leq \frac{n^{2\alpha}}{n} \,, \tag{E.44}$$

where the last inequality follows from ignoring the second term in the denominator.

Setting regularization parameter. From the analysis in the above paragraphs, we have

$$\text{Bias}^2 \leq \max\{N_{\mathsf{cov}} n^{-\alpha\beta} + N_{\mathsf{cov}}^{1-\gamma}, \lambda_{\mathsf{reg}} n^\alpha\} \leq \max\{N_{\mathsf{cov}} n^{-\alpha\beta}, \lambda_{\mathsf{reg}} n^\alpha\} + N_{\mathsf{cov}}^{1-\gamma} \,, \tag{E.45}$$

$$\text{Variance} \leq \max\{\frac{N_{\mathsf{cov}}^2 n^{-2\alpha\beta} + N_{\mathsf{cov}}^{2(1-\gamma)}}{\lambda_{\mathsf{reg}}^2 n}, \frac{n^{2\alpha}}{n}\} \leq \max\{\frac{N_{\mathsf{cov}}^2 n^{-2\alpha\beta}}{\lambda_{\mathsf{reg}}^2 n}, \frac{n^{2\alpha}}{n}\} + \frac{N_{\mathsf{cov}}^{2(1-\gamma)}}{\lambda_{\mathsf{reg}}^2 n} \,. \tag{E.46}$$

For regularization parameter $\lambda_{\mathsf{reg}} > N_{\mathsf{cov}} n^{-\alpha\beta-\alpha}$ and $\gamma > \frac{\alpha\beta}{\log_n N_{\mathsf{cov}}}$, we have

$$\text{Bias}^2 \leq \lambda_{\mathsf{reg}} n^\alpha + N_{\mathsf{cov}}^{1-\gamma} \,,$$
$$\text{Variance} \leq \frac{n^{2\alpha}}{n} \,.$$

Excess risk bound. To obtain the final excess risk bound, we set $\alpha = \frac{1+\log_n N_{\mathsf{cov}}}{\beta+2}$

$$\begin{aligned} \mathbb{E}[\Delta(\hat{\pi}_{\mathsf{plug}}; r^*)]^2 &\leq \lambda_{\mathsf{reg}} n^\alpha + \frac{n^{2\alpha}}{n} + N_{\mathsf{cov}}^{1-\gamma} \\ &\leq N_{\mathsf{cov}} n^{-\alpha\beta} + n^{2\alpha-1} + N_{\mathsf{cov}}^{1-\gamma} \\ &\overset{\text{(i)}}{\lesssim} N_{\mathsf{cov}}^{\frac{2}{\beta+2}} n^{\frac{-\beta}{\beta+2}} + N_{\mathsf{cov}}^{1-\gamma} \,, \end{aligned} \tag{E.47}$$

where inequality (i) follows from our particular choice of α. Combining the above bound with Lemma E.4 completes the proof. □

The following corollary instantiates the above theorem for the case when the input space is the unit ball, that is, $\mathcal{X} = \mathbb{B}_d(1)$.

Corollary E.2. *Let the input space $\mathcal{X} = \mathbb{B}_d(1)$ and the kernel $\mathcal{K}$ satisfy Assumption 6.2. Then, for any $\beta > 1 + \frac{2}{d}$, we have*

$$\max_x f^*(x) - \mathbb{E}_{x\sim\hat{\pi}_{\mathsf{plug}}} f^*(x) \lesssim L_{\mathcal{K}}^{\frac{d}{\beta+2+2d}} n^{\frac{-\beta}{2(\beta+2+2d)}} \ . \tag{E.48}$$

Proof. From the bound in Theorem 6.3, we have,

$$\begin{aligned}
\max_x f^*(x) - \mathbb{E}_{x\sim\hat{\pi}_{\mathsf{plug}}} f^*(x) &\lesssim N_{\mathsf{cov}}^{\frac{1}{\beta+2}}(\epsilon) \cdot n^{\frac{-\beta}{2(\beta+2)}} + N_{\mathsf{cov}}^{\frac{1-\gamma}{2}}(\epsilon) + \sqrt{L_{\mathcal{K}}\epsilon} \\
&\overset{\text{(i)}}{\leq} N_{\mathsf{cov}}^{\frac{1}{\beta+2}}\left(\frac{\epsilon^2}{L_{\mathcal{K}}}\right) \cdot n^{\frac{-\beta}{2(\beta+2)}} + N_{\mathsf{cov}}^{\frac{1-\gamma}{2}}\left(\frac{\epsilon^2}{L_{\mathcal{K}}}\right) + \epsilon \\
&\overset{\text{(ii)}}{\leq} L_{\mathcal{K}}^{\frac{d}{\beta+2}} \cdot \left(\frac{1}{\epsilon}\right)^{\frac{2d}{\beta+2}} \cdot n^{\frac{-\beta}{2(\beta+2)}} + \left(\frac{L_{\mathcal{K}}}{\epsilon^2}\right)^{\frac{d(1-\gamma)}{2}} + \epsilon \\
&\overset{\text{(iii)}}{\leq} L_{\mathcal{K}}^{\frac{d}{\beta+2}} \cdot \left(\frac{1}{\epsilon}\right)^{\frac{2d}{\beta+2}} \cdot n^{\frac{-\beta}{2(\beta+2)}} + 2\epsilon \ ,
\end{aligned} \tag{E.49}$$

where inequality (i) follows from substituting $\epsilon \to \epsilon^2/L_{\mathcal{K}}$, (ii) follows from the fact that $N_{\mathsf{cov}}(\epsilon) \asymp \left(\frac{1}{\epsilon}\right)^d$, and (iii) follows from using the assumption that $\beta > \gamma > 1 + \frac{2}{d}\frac{\log(1/\epsilon)}{\log(L_{\mathcal{K}}/\epsilon^2)}$.

Finally, setting $\epsilon \asymp L_{\mathcal{K}}^{\frac{d}{\beta+2+2d}} n^{\frac{-\beta}{2(\beta+2+2d)}}$, we get

$$\max_x f^*(x) - \mathbb{E}_{x\sim\hat{\pi}_{\mathsf{plug}}} f^*(x) \lesssim L_{\mathcal{K}}^{\frac{d}{\beta+2+2d}} n^{\frac{-\beta}{2(\beta+2+2d)}} \ .$$

This establishes the desired claim. □

Regret bound for Matérn Kernel

In this section, we specialize the bound from Theorem 6.3 for the special class of Matérn kernels. Recall that the Matern kernel is a distanced based kernel with $\mathcal{K}(x, y) = f(\|x-y\|)$. Denote by $r = \|x - y\|$, the exact form for the kernel is given by

$$\mathcal{K}_{\mathsf{Matern},\nu}(r) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}r}{l}\right)^{\nu} K_\nu \left(\frac{\sqrt{2\nu}r}{l}\right) \ , \tag{E.50}$$

with parameters ν and l and where K_ν is the modified Bessel function of the second kind. Going forward, lets fix the scale parameter $l = 1$ without loss of generality.

The following lemma then bounds the Lipschitz constant for this class of kernels when the distance function is the ℓ_2 norm.

Lemma E.6 (Lipschitz Matérn Kernel)**.** *Consider the Matérn kernel with parameter $\nu > \frac{3}{2}$. The Lipschitz constant of this kernel is bounded by*

$$L_{\mathcal{K}} \leq \sup_{r \in (0,1)} [\frac{e 2^{2-\nu} \nu K_{\nu-1}(1)}{\Gamma(\nu)} \cdot r e^{-\sqrt{2\nu} r}]. \tag{E.51}$$

Proof. The approach will be to show that the kernel function $\mathcal{K}_{\mathsf{Matern},\nu}$ is a Lipschitz function of the distance r and then cover the ℓ_2 ball in the d dimensional space appropriately. We now look at the derivative of the function $\mathcal{K}_{\mathsf{Matern},\nu}(r)$ with respect to r.

$$\begin{aligned}
\partial \mathcal{K}_{\mathsf{Matern},\nu}(r) &= \frac{2^{1-\nu}(\sqrt{2\nu})^{\nu}}{\Gamma(v)} \left(\nu r^{\nu-1} K_{\nu}(\sqrt{2\nu} r \partial r + r^{\nu} \partial K_{\nu}(\sqrt{2\nu} r \right) \\
&\overset{\text{(i)}}{=} \frac{2^{1-\nu}(\sqrt{2\nu})^{\nu}}{\Gamma(v)} \left(\nu r^{\nu-1} K_{\nu}(\sqrt{2\nu} r) - r^{\nu} \left(\sqrt{2\nu} K_{\nu-1}(\sqrt{2\nu} r) + \frac{\nu \sqrt{2\nu}}{\sqrt{2\nu} r} K_{\nu}(\sqrt{2\nu} r) \right) \right) \partial r \\
&= -\frac{2^{1-\nu}(\sqrt{2\nu})^{\nu}}{\Gamma(v)} \left(r^{\nu} \sqrt{2\nu} K_{\nu-1}(\sqrt{2\nu} r) \right) \partial r \ ,
\end{aligned} \tag{E.52}$$

where (i) follows from the identity $\partial K_{\nu}(z) = (-K_{\nu-1}(z) - \frac{\nu}{z} K_{\nu}(z)) \partial z$.

For any $\nu > \frac{1}{2}$, we have the inequality

$$\frac{K_{\nu}(x)}{K_{\nu}(y)} < \exp^{y-x} \left(\frac{y}{x} \right)^{\nu} \quad \text{for } 0 < x < y. \tag{E.53}$$

Instantiating the above with $y = 1$ and $\nu > \frac{3}{2}$, we have

$$\begin{aligned}
|\partial \mathcal{K}_{\mathsf{Matern},\nu}(r)| &\leq \frac{2^{1-\nu}(\sqrt{2\nu})^{\nu}}{\Gamma(v)} \left(r^{\nu} \sqrt{2\nu} \cdot \frac{e^{-\sqrt{2\nu} r}}{(\sqrt{2\nu} r)^{\nu-1}} \cdot e K_{\nu-1}(1) \right) \\
&\leq \frac{e 2^{2-\nu} \nu K_{\nu-1}(1)}{\Gamma(\nu)} \cdot r e^{-\sqrt{2\nu} r} \ .
\end{aligned} \tag{E.54}$$

The Lipschitz constant for this case can now be obtained by taking a sup over $r \in (0, 1)$. □

While our upper bound was in terms of sample complexity, in order to compete with the cumulative regret formulation, we adapt an explore-then-commit strategy. The following lemma relates the sample complexity bound to a cumulative regret bound.

Lemma E.7 (Batch to online conversion)**.** *Suppose an algorithm has sample complexity $O(n^{-\alpha})$ in the passive learning setup, the explore then commit strategy based on this learning algorithm would have regret $O(T^{\frac{1}{1+\alpha}})$.*

Proof. For some parameter $\gamma > 0$, let the explore then commit algorithm explore for T^{γ} steps and the commit to the strategy obtained post this exploration for the remaining $T - T^{\gamma}$ time steps. The cumulative regret for such an algorithm is

$$\mathfrak{R}_T = T^{\gamma} + T^{-\alpha\gamma}(T - T^{\gamma}) \leq T^{\gamma} + T^{1-\alpha\gamma} \ . \tag{E.55}$$

Setting $\gamma = \frac{1}{1+\alpha}$ finishes the proof. □

We now proceed to prove Corollary 6.3 which instantiates the bound in Theorem 6.3 for the class of Matérn kernels.

Corollary E.3 (Restated Corollary 6.3)**.** *Consider the family of Matérn kernels with parameter $\nu > \frac{3}{2}$ defined with the euclidean norm over $\mathbb{R}^d$. The T-step regret of the explore-then-commit algorithm is*

$$\mathfrak{R}_{\mathsf{Mat},T} \lesssim O\left(L_{\mathcal{K}}^{\frac{d^2}{2\nu+d(3+2d)}} \cdot T^{\frac{4\nu+d(6+4d)}{6\nu+d(7+4d)}}\right) .$$

with high probability.

Proof. First, observe that excess risk bound in Corollary E.2 can be converted to a corresponding T-step regret bound by an application of Lemma E.7 such that

$$\mathfrak{R}_T \lesssim O\left(L_{\mathcal{K}}^{\frac{d}{\beta+2+2d}} \cdot T^{\frac{2\beta+4+4d}{3\beta+4+4d}}\right). \tag{E.56}$$

For the class of Matérn kernels, the decay parameter $\beta = 1 + \frac{2\nu}{d}$ [112, Theorem 9]. Using this wit the above regret bound, we get,

$$\mathfrak{R}_{\mathsf{Mat},T} \lesssim O\left(L_{\mathcal{K}}^{\frac{d^2}{2\nu+d(3+2d)}} \cdot T^{\frac{4\nu+d(6+4d)}{6\nu+d(7+4d)}}\right) .$$

This completes the proof. □

E.4 Adaptive sampling via GP-UCB

In this section, we prove an upper bound on the expected risk of the Gaussian process upper confidence bound algorithm (GP-UCB) algorithm of Srinivas et al. [185]. In order to adapt their algorithm for our setup, consider the function

$$f_r(x) := \langle r, Mx \rangle_{\mathbb{H}_r} \quad \text{such that } D = \{x \mid \|x\|_{\mathbb{H}_\pi} \leq 1\}. \tag{E.57}$$

We have used x to denote policies in this setup to be consistent with the notation in Srinivas et al. [185]. Observe that the domain defined above is not compact – a necessary condition for the algorithm to work. One work around this is to truncate the unit ball after a finite number of dimensions and bound this truncation error. The excess risk incurred by this truncation can be made arbitrary small. Going forward, we ignore this truncation. The regret for the UCB algorithm is shown to be upper bounded by $\tilde{O}(\gamma_T\sqrt{T})$ where γ_T is the information gain with

$$\gamma_T := \max_{x_1,\ldots,x_T \in D} \frac{1}{2} \log\det(I + [\mathcal{K}(x_i, x_j)]_{i,j=1}^T) , \tag{E.58}$$

where we have assumed without loss of generality that the noise variance $\tau = 1$. For our setup, the kernel function $\mathcal{K}(x_i, x_j) = \langle Mx_i, Mx_j \rangle_{\mathbb{H}_r}$. We additionally require that the reward function r belongs to the RKHS spanned by the set $\{Mx \mid x \in D\}$. Denote by $S = S_\pi^{\frac{1}{2}} M^\top S_r^{-1} S_\pi^{\frac{1}{2}}$ and suppose that its eigenvalues satisfy a power law decay with $\sigma_j(S) = \zeta_j = j^{-\beta}$. The following lemma upper bounds the information gain for this setup in terms of the power law parameter $\beta > 0$.

Lemma E.8 (Information Gain.)**.** *The information gain γ_T for the above setup is bounded as*

$$\gamma_T = O(\log(T) \cdot T^{\frac{1}{\beta+1}}) \ . \tag{E.59}$$

Proof. The quantity of interest here is the information gain

$$\gamma_T := \max_{x_1,\ldots,x_T} \frac{1}{2} \log\det(I + XSX^\top) \quad \text{such that } \forall j \ \|x_j\|_2 \leq 1 \ , \tag{E.60}$$

where the matrix $X = [x_1^\top; \ldots; x_T^\top]$ and we have assumed that the noise variance is 1. From the setup described above, we have that the eigen values of S decay as $\lambda_j \asymp j^{-\beta}$. It is easy to see that

$$F_{\mathsf{ig}}(\{x_t\}) := \frac{1}{2} \log\det(I + XSX^\top) \tag{E.61}$$

is a monotonic sub-modular function. Thus, the value of γ_T can be upper bounded by $(1-1/e)^{-1}$ times the value of the greedy maximization algorithm. The greedy maximization algorithm is equivalent to picking

$$x_t = \arg\max_x F_{\mathsf{ig}}(X_{t-1} \cup \{x\}) \ .$$

It is easy to see that at each time t, the unit vector x_t will be an eigen vector of the matrix S. Given this observation, we can finally upper bound the value of the info gain

$$\gamma_T \leq c \cdot \max_{m_1,\ldots,m_T} \sum_{j=1}^{T} \log(1 + m_j \lambda_j) \quad \text{such that } m_j \geq 0 \text{ and } \sum_j m_j = T.$$

Solving the above optimization problem, the optimal choice of the variables

$$m_j = \max\left\{\frac{1}{\lambda} - \frac{1}{\lambda_j}, 0\right\} \quad \text{and} \quad \sum_j m_j = T \ . \tag{E.62}$$

Setting $\lambda = T^{-\frac{\beta}{\beta+1}}$ ensures that there are $T^{\frac{1}{\beta+1}}$ active directions. Substituting the above values of m_j in the expression for γ_T, we get

$$\begin{aligned}\gamma_T &\leq c \cdot \sum_{j=1}^{\infty} \log(1 + \max(\frac{\lambda_j}{\lambda} - 1, 0)) \\ &\leq c \cdot \log\left(\frac{\lambda_1}{\lambda}\right) \cdot \sum_{j=1}^{\infty} \mathbb{I}[\lambda_j > \lambda] \\ &\overset{\text{(i)}}{=} O(\log(T) \cdot T^{\frac{1}{\beta+1}}) \,,\end{aligned}$$

where (i) follows from setting $\lambda = T^{-\frac{\beta}{\beta+1}}$. This establishes the required claim. □

We are now ready to state this our sample complexity bound for GP-UCB for this subclass of problems.

Proposition E.1 (Sample complexity for GP-UCB)**.** *Suppose that the police space $\mathbb{H}_\pi$, reward space $\mathbb{H}_r$ and the map M satisfy the power law decay assumption with exponent $\beta > 0$. The estimator $\hat{\pi}_{\mathsf{ucb}}$ output by the GP-UCB algorithm satisfies*

$$\mathbb{E}[\Delta(\hat{\pi}_{\mathsf{ucb}}; r^*)] \leq \tilde{O}(n^{-\frac{\beta-1}{2(\beta+1)}}) \,. \tag{E.63}$$

The proof of the sample complexity bound in Proposition E.1 now follows the regret bound of $\tilde{O}(\gamma_T\sqrt{T})$ along with using the upper bound on the information gain from Lemma E.8.

$$\mathbb{E}[\Delta(\hat{\pi}_{\mathsf{plug}}; r^*)] = \tilde{O}(n^{\frac{1}{\beta+1}-\frac{1}{2}}) = \tilde{O}(n^{-\frac{\beta-1}{2(\beta+1)}}) \,. \tag{E.64}$$

More recently, [44] extended the analysis of [203] to show that the SupKernelUCB algorithm achieves a regret bound $\tilde{O}(\sqrt{\gamma_T T})$. Using this modified bound, one can improve the above analysis to obtain excess risk

$$\mathbb{E}[\Delta(\hat{\pi}_{\mathsf{plug}}; r^*)] = \tilde{O}(n^{\frac{1}{2(\beta+1)}-\frac{1}{2}}) = \tilde{O}(n^{-\frac{\beta}{2(\beta+1)}}) \,, \tag{E.65}$$

which is still worse than those obtained by the bounds by our proposed ridge regression estimator.

E.5 Further details on experimental evaluation

In the simulation study, we work with d dimensional RKHSs $\mathbb{H}_r$ and $\mathbb{H}_\pi$. In order to simulate the nonparmeteric regime, we typically use value of n which are less or at most a constant times the dimension d. We set the matrices $S_\pi = \mathrm{diag}(j^{-1.75})$, $S_r = \mathrm{diag}(j^{-1})$ and the map $M = I$. This is allowed since the policy space is smaller than the reward space. With this, the effective decay parameter $\beta = \beta_\pi - \beta_r = 0.75$. We sampled the true reward r^* uniformly at random from the unit ball in $\mathbb{H}_r$. We further sampled the oracle noise $\epsilon \sim \mathcal{N}(0, 0.01)$. All plots were averaged over 10 runs.