

## ABSTRACT

Title of Dissertation:     **ALGORITHMIC DECISION-MAKING  
AND MODEL EVALUATION  
IN SOCIALLY CONSEQUENTIAL DOMAINS**

**Christine Robie Herlihy  
Doctor of Philosophy, 2024**

Dissertation Directed by: **Associate Professor John P. Dickerson  
Department of Computer Science**

Algorithms are increasingly used to create markets, discover and disseminate information, incentivize behaviors, and inform real-world decision-making in a variety of socially consequential domains. In such settings, algorithms have the potential to improve aggregate utility by leveraging previously acquired knowledge, reducing transaction costs, and facilitating the efficient allocation of resources, broadly construed. However, ensuring that the distribution over outcomes induced by algorithmic decision-making renders the broader system sustainable—i.e., by preserving rationality of participation for a diverse set of stakeholders, and identifying and mitigating the costs associated with unevenly distributed harms—remains challenging.

One set of challenges arises during algorithm or model development: here, we must decide how to operationalize sociotechnical constructs of interest, induce prosocial behavior, balance uncertainty-reducing exploration and reward-maximizing exploitation, and incorporate domain-specific preferences and constraints. Common desiderata such as individual or subgroup fairness,

cooperation, or risk mitigation often resist uncontested analytic expression, induce combinatorial relations, or are at odds with unconstrained optimization objectives and must be carefully incorporated or approximated so as to preserve utility and tractability. Another set of challenges arises during model evaluation: here, we must contend with small sample sizes and high variance when estimating performance for intersectional subgroups of interest, and determine whether observed performance on domain-specific reasoning tasks may be upwardly biased due to annotation artifacts or data contamination.

In this thesis, we propose algorithms and evaluation methods to address these challenges and show how our methods can be applied to improve algorithmic acceptability and decision-making in the face of uncertainty in public health and conversational recommendation systems. Our core contributions include: (1) novel resource allocation algorithms to incorporate prosocial constraints while preserving utility in the restless bandit setting; (2) model evaluation techniques to inform harms identification and mitigation efforts; and (3) prompt-based interventions and meta-policy learning strategies to improve expected utility by encouraging context-aware uncertainty reduction in large language model (LLM)-based recommendation systems.

ALGORITHMIC DECISION-MAKING AND MODEL EVALUATION  
IN SOCIALLY CONSEQUENTIAL DOMAINS

by

Christine Robie Herlihy

Dissertation submitted to the Faculty of the Graduate School of the  
University of Maryland, College Park in partial fulfillment  
of the requirements for the degree of  
Doctor of Philosophy  
2024

Advisory Committee:

Associate Professor John P. Dickerson, Chair/Advisor  
Professor Philip Resnik, Dean's Representative  
Professor Hal Daumé III  
Professor Aravind Srinivasan  
Dr. Adith Swaminathan

© Copyright by  
Christine Robie Herlihy  
2024

## Dedication

This work is dedicated to the people who have made my journey possible—my family and friends, my teachers, professors, and mentors, and my doctors and nurses—as well as to the women who have preceded me and succeeded in carving out many possible paths in this field, and all those who will come after—with profound gratitude, and hope for the future.

## Acknowledgments

I am profoundly grateful to have had the opportunity to learn from, collaborate with, and be supported by so many wonderful people and organizations during my doctoral studies. To everyone who helped me get here and stay (mostly) sane while doing it, thank you!

I would like to begin by thanking my advisor, John P. Dickerson. John, it's been such a privilege to learn from and work with you! When I first started this journey, I could never have imagined that I would get to work on such interesting algorithmic topics that combine theoretical rigor with real-world impact. Thank you for making that possible; it's been wonderful to see my econometrics past inform my academic present and future. I deeply appreciate—and would never have made it through without—your intellectual guidance, strategic insights, willingness to talk through ideas, and relentless advocacy on my behalf during my time at UMD.

Next, I would like to thank my committee members: Philip Resnik, Hal Daumé III, Aravind Srinivasan, and Adith Swaminathan. Philip, I'm so grateful to have had the opportunity to learn from and collaborate with you. Thank you for all you have taught me about the importance of centering language and linguistics when doing natural language processing for clinical decision support, even—if not *especially*—in the age of LLMs. I sincerely appreciate your consistent support and mentorship! Hal, I am grateful to have been introduced to the topics of fairness in machine learning and algorithmic harms in your computational linguistics class. I also appreciate and am doing my best to honor the approach to problem selection that you shared with us in

class—namely, that we should focus on problems that we would be happy to see solved, even if we are not the first to solve them, because a solution would be a genuine source of good (or reduced harm) in the world. Thank you for your support during my time at UMD and Microsoft Research FATE. Aravind, I am grateful to have had the chance to learn from and collaborate with you on the probabilistic fairness portion of this work—thank you for your mentorship and support! Adith, I enjoyed working with you and learned so much during my time at Microsoft Research ALR! Thank you for your trust, excellent mentorship, and the helpful push to consider topics at the intersection of decision-making under uncertainty and LLMs.

Next, I would like to thank the collaborators I've been fortunate enough to work with during my studies—my journey would not have been the same without you all, and I am personally and intellectually indebted. To Bill Regli and others at ARLIS: thank you for your mentorship and the opportunity to work on graph-based architectures and algorithms. To Spencer Breiner, Eswaran Subrahmanian, and others at NIST: thank you for your support; I enjoyed and learned so much from our discussions at the intersection of category theory, natural language processing, and computational phenotyping. To Philip Resnik, Carol Espy-Wilson, Deanna Kelly, Nadee Seneviratne, Gowtham Premananth, Yashish Siriwardena, Colin Frazier, and others on the AIM-HI/SCH team: thank you for all you have taught me about constrained resource allocation, ethical multimodal classification, and proactive longitudinal monitoring in the context of clinical decision support for mental health. To Neil Sehgal, Devlon Jackson, Cynthia Baur, Madeline Diep, Naman Molri, and others on the Healthy Me/Mi Salud team: thank you for your mentorship and the opportunity to work on recommendation systems, algorithmic fairness, and public health.

To Aviva Prins, Samuel Dooley, Vedant Nanda, Saptarashmi Bandyopadhyay, Seyed Esmaeili, Yuelin Liu, Pranav Goel, and other UMD collaborators, it's been a privilege getting to

know, learn from, and work with you all! Thank you for being supportive colleagues and friends. To Tom Hurst, Jodie Gray, Migo Gui, and the I4C team: thank you for your logistical and moral support and for all you do to make the department an inclusive and welcoming space!

To Victor Amelkin, Alex Barbosa, Michael Caldara, Joey Durham, Jane Shi, and others at Amazon Robotics: thank you for your guidance and the opportunity to work on graph-based deep reinforcement learning. To Miro Dudík, Alexandra Chouldechova, Kimberly Truong, and others at Microsoft Research FATE: thank you for your mentorship and collaboration, as well as for the opportunity to work on algorithmic fairness and disaggregated model evaluation. To Emmanuel Klu, Donald Martin, Jr., Emi Kolawole, Klara Kaleb, DJ Passey, Georgia Liu, and others at Google Research and X: thank you for your support and the chance to work on algorithmic fairness in dynamic systems, as well as for the many examples of what it looks like to develop algorithmic interventions *with*, rather than *for*, diverse stakeholders. To Adith Swaminathan, Jennifer Neville, Tobias Schnabel, Allen Nie, and others at Microsoft Research Augmented Learning and Reasoning (ALR), thank you for your mentorship and collaboration, as well as for the opportunity to work on query underspecification in chat-based recommendation systems. To Charity Hilton, Trevor Goodyear, James Fairbanks, Micah Halter, Aaron Potvien, Erica Briscoe, Jason Poovey, David Ediger, Evan Stuart, Sheila Isbell, Saul Crumpton, Kristin Morgan, Richard Boyd, Melissa Rost, Jon Duke, Marla Gorges, and others at the Georgia Tech Research Institute: thank you for being excellent colleagues, mentors, and friends during my time at GTRI.

To the teachers and professors I've been fortunate to learn from—growing up and during my time at Georgetown University, Georgia Tech, and the University of Maryland—thank you for encouraging my curiosity and providing an academic foundation that has enabled me to explore diverse fields and view learning as a lifelong source of growth and joy.



Next, I would like to thank the people closest to me. To my family, and in particular to my parents—who have loved and encouraged me and done everything within their power to support my education throughout my life: I deeply appreciate the choices and sacrifices you’ve made to help me reach this goal—and the many goals that preceded it. Thank you for making my path possible and cheering me on even when many steps of this journey took me far from you. I love you and am proud to be your daughter! To my extended family and dearest friends near and far, who have cheered me on from around the world all these years: no matter what city we find ourselves (lost) in or what language we (try our best to) speak when we’re together, I’ll always be grateful to belong to such a wonderful community. Your love, support, and time-zone crossing pep talks have made all the difference on this journey: thank you!

To my partner, Amir Kashani: when I first studied game theory in undergrad, they taught us that the “good” parts of marriage vows are only there for symmetry—I’m happy to report that here, theory and practice diverge. Our own vows were unorthodox, to say the least, and life has shown us that the “hard” parts can come sooner than expected. But through it all, you’ve always shown up for me, believed in me, celebrated my wins, and found just the right cat videos to help me through the inevitable hardships. Thank you for loving me enough to let me pursue something so crazy, and for the wonderful and fulfilling life we’ve built along the way.

Finally, I would like to thank the many healthcare providers who have helped me navigate the unexpected challenges, both personal and pandemic-related, that unfolded during my studies. To say I am grateful is an understatement—I may never find the right words—but I hope to honor the gift of a future I’ve been given through research and advocacy.

# Table of Contents

<b>Dedication</b>	<b>ii</b>
<b>Acknowledgements</b>	<b>iii</b>
<b>Table of Contents</b>	<b>vii</b>
<b>List of Tables</b>	<b>x</b>
<b>List of Figures</b>	<b>xi</b>
<b>List of Abbreviations</b>	<b>xiii</b>
<b>Chapter 1: Introduction and Overview</b>	<b>1</b>
1.1 Sequential Decision-making with Applications to Public Health . . . . .	3
1.2 Model Evaluation for Harms Identification and Mitigation . . . . .	4
1.3 Balancing Exploration & Exploitation in LLM-informed Decision-making . . . . .	6
<b>Chapter 2: Sequential Decision-making with Applications to Public Health</b>	<b>8</b>
2.1 Restless Multi-armed Bandit Model . . . . .	8
2.2 Probabilistically Fair Allocation in the Restless Bandit Setting . . . . .	11
2.2.1 Introduction . . . . .	11
2.2.2 Restless Multi-Armed Bandit Model . . . . .	13
2.2.3 Context, Motivation & Related Work . . . . .	14
2.2.4 Methodological Approach . . . . .	21
2.2.5 Experimental Evaluation . . . . .	27
2.2.6 Summary of Contributions . . . . .	35
2.3 Networked Restless Bandits with Positive Externalities . . . . .	36
2.3.1 Introduction . . . . .	36
2.3.2 Model Preliminaries . . . . .	39
2.3.3 Algorithmic Approach . . . . .	43
2.3.4 Experimental Evaluation . . . . .	51
2.3.5 Summary of Contributions . . . . .	57
<b>Chapter 3: Model Evaluation for Harms Identification and Mitigation</b>	<b>58</b>
3.1 Evaluating Model Performance Across Intersectional Subgroups . . . . .	58
3.1.1 Introduction . . . . .	58
3.1.2 Background and Related Work . . . . .	60

3.1.3	Problem setting . . . . .	63
3.1.4	Structured regression approach . . . . .	69
3.1.5	Confidence intervals . . . . .	73
3.1.6	Goodness-of-fit testing . . . . .	74
3.1.7	Experiments . . . . .	75
3.1.8	Summary of Contributions . . . . .	85
3.2	Identifying and Mitigating Clinical Natural Language Inference Artifacts . . . . .	87
3.2.1	Introduction . . . . .	87
3.2.2	The MedNLI Dataset . . . . .	88
3.2.3	MedNLI Contains Artifacts . . . . .	88
3.2.4	Characteristics of Clinical Artifacts . . . . .	89
3.2.5	Physician-Annotator Heuristics . . . . .	92
3.2.6	Adversarial Filtering . . . . .	94
3.2.7	Discussion . . . . .	96
3.2.8	Ethical Considerations . . . . .	97
<b>Chapter 4: Balancing Exploration &amp; Exploitation in LLM-informed Decision-making</b>		<b>99</b>
4.1	Overcoming Miscalibrated Conversational Priors in LLM-based Chatbots . . . . .	99
4.1.1	Introduction . . . . .	99
4.1.2	Problem Formulation . . . . .	103
4.1.3	Motivating Experiments . . . . .	109
4.1.4	Algorithmic Approach . . . . .	115
4.1.5	Related Work . . . . .	122
4.1.6	Limitations . . . . .	124
4.1.7	Summary of Contributions . . . . .	125
<b>Chapter 5: Conclusion &amp; Future Work</b>		<b>127</b>
5.1	Summary of Contributions . . . . .	127
5.2	Future Research Directions . . . . .	129
<b>Appendix A: Probabilistically Fair Allocation in the Restless Bandit Setting</b>		<b>130</b>
A.1	Notation . . . . .	130
A.2	Empirical Inequity in the Distribution of Actions under Whittle Index Policies . . . . .	132
A.3	Intractability of Alternative Approaches . . . . .	135
A.3.1	Why not use a math-programming approach? . . . . .	135
A.4	PROBFAIR: a Probabilistically Fair Policy . . . . .	147
A.4.1	Proofs . . . . .	147
A.4.2	Dependent Rounding Sampling Approach . . . . .	153
A.4.3	Additional Experimental Details . . . . .	155
<b>Appendix B: Networked Restless Bandits with Positive Externalities</b>		<b>160</b>
B.1	Heuristic Policy . . . . .	160
B.1.1	Pseudocode . . . . .	160
B.1.2	Proofs . . . . .	162
B.2	Additional empirical results . . . . .	169

B.3	Additional experimental details . . . . .	174
<b>Appendix C: Evaluating Model Performance Across Intersectional Subgroups</b>		<b>175</b>
C.1	Confidence intervals for standard estimates . . . . .	175
C.2	Synthetic data generation . . . . .	176
<b>Appendix D: Identifying and Mitigating Clinical Natural Language Inference Artifacts</b>		<b>178</b>
D.1	Hypothesis-only Baseline Analysis . . . . .	178
D.2	Lexical Artifact Analysis . . . . .	178
D.3	Semantic Analysis of Heuristics . . . . .	179
D.4	Adversarial Filtering . . . . .	180
<b>Appendix E: Overcoming Miscalibrated Conversational Priors in LLM-based Chatbots</b>		<b>182</b>
E.1	Helper LLM-based Classifiers . . . . .	182
E.1.1	LLM-based query underspecification classifier . . . . .	182
E.1.2	LLM-based Response Strategy classifier . . . . .	186
E.2	Response-strategy system messages . . . . .	190
E.3	Motivating Experiments . . . . .	191
E.3.1	Synthetic Dataset Construction . . . . .	191
E.3.2	Synthetic Query Templates and Parameter Options . . . . .	191
E.3.3	Stable Diffusion Example: Policy-induced responses . . . . .	195
E.3.4	Simulating User Responses to LLM Questions . . . . .	196
<b>Bibliography</b>		<b>198</b>

## List of Tables

2.1	Expected IB and normalized EMD by policy and fairness bracket. . . . .	32
2.2	$\mathbb{E}[\text{IB}]$ , by $\varphi$ and policy . . . . .	54
3.1	<i>Goodness-of-fit tests on diabetes data.</i> From left to right, we consider increasingly more complex models with a growing set of features and report the $p$ -values of the corresponding goodness-of-fit tests; $p$ -values below 0.05 are in bold. . . . .	80
3.2	<i>Goodness of fit tests on synthetic data.</i> From left to right, we consider increasingly more complex models with a growing set of features and report the $p$ -values of the corresponding goodness-of-fit tests; $p$ -values below 0.05 are in bold. . . . .	82
3.3	Performance (micro F1-score) of the <code>fastText</code> hypothesis-only classifier. . . . .	89
3.4	Confusion matrix for <code>fastText</code> classifier. . . . .	89
3.5	Top 15 tokens by $\text{PMI}(\text{token}, \text{class})$ ; % of <i>class</i> training examples containing the token. . . . .	91
3.6	Average and median hypothesis length by class and entity representation. . . . .	92
3.7	Results of $\chi^2$ test statistic by heuristic, computed using the combined MedNLI dataset ( $\ddagger p < 0.001$ , $\dagger p < 0.01$ , $* p < 0.5$ ). Top class presented with % of heuristic-satisfying pairs. . . . .	93
3.8	Performance (micro F1-score) for the majority class baseline and <code>fastText</code> classifiers, with and without premise, by partition (e.g., <i>full</i> , <i>easy</i> , <i>difficult</i> ). . . . .	95
4.1	Examples from the OpenAssistant dataset tagged by our classifier as critically underspecified (details in Appendix E.1.1). . . . .	110
4.2	For the motivating experiments in Section 4.1.3, we categorize LLM responses into seven response types. . . . .	111
4.3	We evaluated three different policies that encourage different initial response strategies to show the possible room for improvement in multi-step interactions. . . . .	113
A.1	Notation used in Chapter 2.2, with notes on interpretation. . . . .	131
E.1	Classifier performance: over all intents . . . . .	184
E.2	Classifier performance: movie recommendation queries . . . . .	185
E.3	Classifier performance: gift recommendation queries . . . . .	186
E.4	Classifier performance: plant recommendation queries . . . . .	186
E.5	$\tau$ -classifier performance on human-annotated LLM responses to OpenAssistant queries . . . . .	189
E.6	Policy-induced responses to the stable diffusion example query. . . . .	195

## List of Figures

2.1	PROBFAIR: constructing and sampling from $\pi_{PF}$ . . . . .	23
2.2	Expected total reward (left) and unnormalized EMD (right) on a breadth of cohorts. . . . .	33
2.3	Augmented graph example: given graph, $G$ , we construct augmented graph, $G'$ . . . . .	44
2.4	$\mathbb{E}[R]$ by policy and budget . . . . .	53
2.5	Sensitivity results, by varied hyperparameter . . . . .	56
3.1	<i>Point estimates and 95% confidence intervals of selection rate (SEL) and false negative rate (FNR) on diabetes data. Confidence intervals of the standard estimator are calculated using pooled variance (see Equation 3.7). . . . .</i>	68
3.2	<i>Bias–variance trade-off of structured regression estimates of selection rate (SEL) on diabetes data. Averaged across all groups, small groups (size at most 25), and large groups (size above 25), across 100 draws of evaluation dataset. The scale of the MSE is different for different group sizes, but the minimum MSE is attained around the same value of <math>\lambda</math>, thanks to the weighting of the training loss. . . . .</i>	72
3.3	<i>Mean absolute error of estimates of 6 metrics using 4 methods on diabetes data. Averaged across all groups, small groups (size at most 25), and large groups (size above 25), across 20 draws of evaluation dataset. . . . .</i>	79
3.4	<i>Coverage and mean relative width of confidence intervals for 6 metrics constructed by 3 methods on diabetes data. Averaged across all groups and across 20 draws of evaluation dataset. Relative width is with respect to the width of the standard confidence interval. . . . .</i>	79
3.5	<i>Point estimates and 95% confidence intervals of word error rates of five different ASR systems. . . . .</i>	84
4.1	An example failure where a user’s query is under-specified (blue text). Current LLM-based chatbots produce long responses to hedge against uncertainty (purple text). Clarifying the user’s context can avert this failure. . . . .	100
4.2	Real-world users asked severely under-specified queries more than 23% of the time in the OpenAssistant dataset ( $n = 600$ ). . . . .	100
4.3	For a user query such as $q$ : “Is my outfit appropriate for the event I’m attending tonight?”, an LLM-based chatbot can choose different response strategies. These strategies produce responses that differ in their cognitive costs (x-axis) while providing final answers with different, user-specific levels of usefulness (y-axis). A good chatbot should respond so as to maximize overall utility—i.e., by providing useful and low-cost answers for the user. . . . .	101
4.4	PODP plate diagram illustrating user-chatbot interactions, prompt-induced policies ( $\pi^p$ ), and the meta-policy mapping from conversations to prompts ( $\beta$ ). . . . .	103

4.5	$\pi^{\text{RLHF}}$ as an hierarchical probabilistic process. . . . .	108
4.6	Spectrum characterizing the response strategies that a LLM-based chatbot can take. RLHF fine-tuning encourages RESPOND and HEDGE, whereas CLARIFY may be more appropriate when queries are under-specified. . . . .	108
4.7	Even under severe levels of under-specification, GPT-4 prefers to directly answer a user’s query. . . . .	112
4.8	Distribution of accumulated item utilities $\mathcal{U}$ at timesteps $t = 0, 1$ ; grouped by under-specification levels. . . . .	114
4.9	Distribution of response cost at $t = 0$ for each response strategy $\tau_0$ ; grouped by under-specification levels. . . . .	115
4.10	Distribution of the response strategies $\hat{\tau}_0$ induced by the three prompts = {BASELINE, COT, CLARIFYFLEX}; grouped by under-specification levels. . . . .	117
4.11	Distribution of $\bar{\mathcal{U}}$ for each $(\mathbf{p}_0, \mathbf{p}_1)$ sequence; grouped by under-specification levels. . . . .	118
4.12	Our learned meta-policy outperforms baseline across all under-specification buckets, especially when queries are critically under-specified. And it converges to baseline when queries are sufficiently specified. . . . .	121
A.1	The Whittle index values for Arm 1 and 2 can be separated by a horizontal line, meaning that (WLOG) Arm 2 will always be chosen over Arm 1 because its index value dominates. . . . .	133
A.2	The smallest interval $\nu_i$ such that TW satisfies an integer periodicity definition of time-indexed fairness, given $N = 2$ random arms. In over 50% of iterations, no such fairness constraint satisfaction is possible (i.e., $\exists i$ s.t. $\nu_i > T$ ). . . . .	134
A.3	Adherences of PROBFAIR, compared to the IP formulation . . . . .	144
A.4	Illustration of the probability tree for finding the coefficient corresponding to $x_{i,a=0,t=2}$ . . . . .	146
B.1	Edge generation seed 1 . . . . .	172
B.2	Edge generation seed 2 . . . . .	172
B.3	Edge generation seed 3 . . . . .	172
B.4	Edge generation seed 4 . . . . .	173
B.5	Edge generation seed 5 . . . . .	173
B.6	Edge generation seed 6 . . . . .	173
C.1	<i>Comparison of methods for constructing confidence intervals for the standard estimator.</i> Showing coverage of confidence intervals constructed for six metrics on diabetes data, averaged over all groups and over 20 draws of evaluation dataset. Confidence intervals constructed from pooled variance are close to the perfect line (corresponding to coverage equal to confidence level). Confidence intervals derived from separately estimated variances undercover true values. . . . .	176
E.1	Confusion matrix: all intents . . . . .	185
E.2	Confusion matrix: movie recommendation queries . . . . .	185
E.3	Confusion matrix: gift recommendation queries . . . . .	186
E.4	Confusion matrix: plant recommendation queries . . . . .	187
E.5	Confusion matrix: annotated OpenAssistant query responses . . . . .	189

## List of Abbreviations

AAAI	Association for the Advancement of Artificial Intelligence
ACM	Association for Computing Machinery
AI	Artificial Intelligence
ACC	Accuracy
ASR	Automated speech recognition
AUC	Area under the ROC curve
CDS	Clinical decision support
EB	Empirical Bayes
EMD	Earth Mover's Distance
FAccT	Conference on Fairness, Accountability, and Transparency
FNR	False negative rate
FPR	False positive rate
GP	Gaussian Process
HIPAA	Health Insurance Portability and Accountability Act
IB	Intervention benefit
ICU	Intensive care unit
JS	James-Stein (estimator)
KDD	Knowledge Discovery and Data Mining
LLM	Large language model
MAB	Multi-armed bandit
MAE	Mean absolute error
MBM	Model-based metrics
MDP	Markov decision process
MeSH	Medical Subject Headings
MSE	Mean squared error
NLI	Natural language inference
NRMAB	Networked restless multi-armed bandit
PODP	Partially observable decision process
POMDP	Partially observable Markov decision process
PPV	Positive predictive value
RLHF	Reinforcement learning from human feedback
RMAB	Restless multi-armed bandit
ROC	Receiver operating characteristic curve
SEL	Selection rate
SoTA	State of the art
SR	Structured regression



UAI Uncertainty in Artificial Intelligence  
WER Word error rate

## Chapter 1: Introduction and Overview

“If this is the best of all possible worlds, what, then, are the others?”

— Voltaire, *Candide*

“[Y]es, even though the span between objective and act is most often impassable,

I must believe joy is not impossible.”

— Jean-Paul Pecqueur, “On the Way from Delphi”

Algorithms are increasingly used to create markets, discover and disseminate information, incentivize behaviors, and inform real-world decision-making in a variety of socially consequential domains. In such settings, algorithms have the potential to improve aggregate utility by exploiting previously acquired knowledge, reducing transaction costs, and facilitating the efficient allocation of constrained resources, broadly construed. However, ensuring that the distribution over outcomes induced by algorithmic decision-making renders the broader system sustainable—i.e., by preserving rationality of participation for a diverse set of stakeholders, and identifying and mitigating the costs associated with unevenly distributed harms—remains challenging.

One set of challenges arises during algorithm or model development: here, we must decide how to operationalize sociotechnical constructs of interest, induce prosocial behavior, balance uncertainty-reducing exploration and reward-maximizing exploitation, and incorporate domain-specific preferences and constraints. Common desiderata such as individual or subgroup fairness,

cooperation, or risk mitigation often resist uncontested analytic expression, induce combinatorial relations, or are at odds with unconstrained optimization objectives and must be carefully incorporated or approximated so as to preserve utility and tractability. Another set of challenges arises during model evaluation: here, we must contend with small sample sizes and high variance when estimating performance for intersectional subgroups of interest, and determine whether observed performance on domain-specific reasoning tasks may be upwardly biased due to annotation artifacts or data contamination. In this work, we propose algorithms and evaluation methods to address these challenges and show how our methods can be applied to improve algorithmic acceptability and decision-making in the face of uncertainty in public health and conversational recommendation systems.

This dissertation is organized as follows: in Chapter 2, we begin by considering two instances of *sequential decision-making under uncertainty* with applications to the constrained allocation of public health resources. Next, in Chapter 3, we consider two *model evaluation* tasks with implications for algorithmic fairness, domain-specific reasoning, and model generalizability. In Chapter 4, we consider *large language model (LLM)-informed decision-making* and propose interventions to improve conversation-level outcomes in chat-based recommendation systems. These interventions focus on encouraging context-aware uncertainty reduction when user queries are under-specified. Finally, in Chapter 5, we conclude by identifying open research questions related to LLM-informed decision-making, clinical decision support, and algorithmic fairness in dynamic environments. This dissertation was supported in part by NSF CAREER Award IIS-1846237, NIST MSE Award #20126334, ARL Award W911NF2120076, and NSF SCH Award #2124270.

## 1.1 Sequential Decision-making with Applications to Public Health

In Chapter 2, we specifically consider the *restless multi-armed bandit (RMAB)* setting, where a decision-maker is tasked with determining which subset of individuals (referred to as arms) should receive a beneficial intervention at each timestep, subject to the satisfaction of a budget constraint (see Section 2.1 for details). Each restless arm is formalized as a Markov decision process (MDP), and receipt of the intervention results in an increased probability of a favorable state transition at the next timestep, relative to lack of receipt.

The problem instances we consider are motivated by concrete examples from the clinical domain, where arms represent patients and a binary state space encodes compliance—or lack thereof—with some type of desirable behavior (e.g., taking prescribed medication, exercising, etc.). In such settings, the intervention represents a set of action(s) that make compliance more likely, such as a reminder call from a physician or a supportive message from a peer. Each patient is assumed to benefit from receipt of an intervention; however, the degree of marginal benefit varies in accordance with their underlying state transition function. Resource constraints are assumed to preclude universal allocation. We thus seek a policy that tells us which patients to allocate an intervention to at each time step, such that we maximize total expected discounted reward (i.e., time spent in the “desirable” state for any reward function that is non-decreasing in state  $s$ ), subject to the satisfaction of a budget constraint.

It is PSPACE-hard to pre-compute the optimal policy for a given set of restless arms in the general case [1]. However, as conjectured by Whittle [2] and proven by Weber and Weiss [3], when each arm is indexable, a tractable solution exists that is provably asymptotically optimal: we can decouple the arms and consider a Lagrangian relaxation of the original problem. Our

core contributions include the introduction of novel algorithms to address two limitations of Whittle index-based policies: (1) the *lack of distributive fairness guarantees* (Section 2.2); and (2) the *inability to exploit externalities* when resources are allocated within a community (Section 2.3).

*These works were published in KDD 2023 and AAAI 2023, respectively—see [4] and [5].*

## 1.2 Model Evaluation for Harms Identification and Mitigation

In Section 3.1, we consider the task of *disaggregated evaluation*, which involves assessing the performance of a machine learning model or AI system across a set of *intersectional* subgroups (i.e., subgroups defined by the combination of multiple sensitive attributes, such as race/ethnicity, gender, or age), to determine whether—or to what extent—the deployment of the model will cause any of these subgroups to experience *allocative harms* (i.e., disproportionately low selection rates, or high false negative rates) or *quality-of-service harms* (i.e., disproportionately high task-specific error rates) [6]. The prevailing approach is to compute the performance metric(s) of interest for each subgroup individually. However, small subgroups are common, particularly when higher-order interaction terms are used to define subgroup membership, and the resulting performance estimates for such subgroups are often characterized by high variance, making it difficult to accurately estimate model performance.

We introduce and empirically evaluate a structured regression-based approach that can be used to produce reliable model performance estimates even for very small subgroups. We also provide corresponding inference strategies for constructing confidence intervals and explore how goodness-of-fit testing can yield insight into the structure of fairness-related harms experienced by

intersectional groups. Our empirical results demonstrate that our method is considerably more accurate than the standard approach, especially for small subgroups, and goodness-of-fit testing helps identify the key factors that drive differences in performance.

In Section 3.2, we consider the task of clinical *natural language inference* (NLI), in which a machine learning model is used to classify premise-hypothesis pairs based on whether the hypothesis (i.e., a logical claim about a patient’s diagnosis, prognosis, or behavior) is *entailed by*, *neutral* with respect to, or *contradicted by* the premise (i.e., information extracted from the patient’s deidentified clinical notes). We specifically consider MedNLI, a clinical natural language inference (NLI) dataset created by physician annotators [7], and assess whether, or to what extent, performance estimates of models trained on MedNLI may be upwardly biased due to the presence of annotation artifacts [8].

We empirically demonstrate that MedNLI, like its domain-agnostic, crowd-worker-constructed predecessors—i.e., SNLI [9] and MultiNLI [10]—contains lexical and semantic annotation artifacts that are disproportionately associated with specific classes, allowing a hypothesis-only classifier [11] to significantly outperform a majority-class baseline model, with negative implications for out-of-sample generalizability. We find that entailed hypotheses contain generic versions of specific concepts in the premise, as well as modifiers related to responsiveness, duration, and probability. Neutral hypotheses feature conditions and behaviors that co-occur with, or cause, the medical condition(s) in the premise. Contradiction hypotheses feature explicit negation of the premise and implicit negation via assertion of good health. We conduct adversarial filtering [12] to demonstrate that performance degrades when evaluated on the difficult subset. We conclude by providing partition information and recommendations for alternative dataset construction strategies for knowledge-intensive domains.

These works were published in FAccT 2024 [13] and ACL-IJCNLP 2021 [14], respectively.

### 1.3 Balancing Exploration & Exploitation in LLM-informed Decision-making

In Section 4.1, we formalize goal-directed, conversational interactions between human users and LLM-based agents (i.e., chatbots or copilots) as a *partially observable decision process (PODP)*. In this setting, the ability of the chatbot to deliver well-tailored, query-aligned natural language responses depends on the ability and willingness of users to specify their *true* intent and preferences over the output distribution when issuing natural language queries. However, users may fail to fully specify their preferences due to a variety of reasons, including their epistemic uncertainty or rational desire to reduce their cognitive burden when interacting [15].

In such situations, the copilot may rely on its inductive bias acquired through training and RLHF-based fine-tuning [16] to *respond directly*, or *hedge* by acknowledging the attribute(s) it is uncertain about and producing a set of plausible responses to cover each case. However, in the case of *recommendation*, such response strategies may yield low-utility recs (i.e., items that violate a subset of the user’s latent preferences) or impose a high cognitive cost (due to response length). In contrast, the copilot could reduce uncertainty by asking a small number of clarifying questions, and use the information gained to produce a more tailored response in future timesteps.

We build upon this insight, coupled with the fact that prompts (also referred to as system messages) can be an effective tool for steering the fixed policy an LLM acquires via training and fine-tuning, to propose two interventions designed to make chatbots produce appropriate responses in the face of query underspecification. Our first intervention is inspired by prior research on

the generation of clarification questions [17, 18], and uses a static, “clarification-aware” prompt to nudge the LLM to clarify when appropriate rather than reverting to default response behavior. Our second intervention leverages historical conversation logs (when available) to learn a meta-policy—i.e., a mapping from conversational prefixes to a finite set of prompts. In the online PODP setting, the chatbot first invokes this meta-policy, and then calls the LLM with the resulting prompt to produce a contextually appropriate PODP action (i.e., response). We evaluate each of our proposed interventions on downstream recommendation tasks featuring a synthetic user model. Each intervention achieves higher expected utility relative to baseline when queries are underspecified, and converges to baseline as specification increases.

*This work was published in UAI 2024; see [19].*



## Chapter 2: Sequential Decision-making with Applications to Public Health

Section 2.1 features content from [4], which was done in collaboration with my co-first author, Aviva Prins, as well as Aravind Srinivasan and John P. Dickerson, and [5], which was done in collaboration with John P. Dickerson.

### 2.1 Restless Multi-armed Bandit Model

In Sections 2.2 and 2.3, we consider and extend the restless multi-armed bandit (RMAB) setting; here, we provide an overview. Restless bandits are often used to model budget-constrained resource allocation tasks in which a decision-maker must select a subset of arms (e.g., projects, patients, assets) to receive a beneficial intervention at each timestep, while the state of each arm evolves over time in an action-dependent, Markovian fashion. An RMAB consists of  $N \in \mathbb{N}$  independent arms, each of which evolves over a finite time horizon  $T \in \mathbb{N}$ , according to an associated Markov Decision Process (MDP). Each arm’s MDP is characterized by a 4-tuple  $(\mathcal{S}, \mathcal{A}, P, r)$  where  $\mathcal{S}$  represents the state space,  $\mathcal{A}$  represents the action space,  $P$  represents an  $|\mathcal{S}| \times |\mathcal{A}| \times |\mathcal{S}|$  transition matrix, and  $r : \mathcal{S} \rightarrow \mathbb{R}$  represents a local reward function that maps states to real-valued rewards.

**States, actions, and observability:** We specifically consider a discrete two-state system  $\mathcal{S} := \{0, 1\}$  where 1 (0) represents being in the “good” (“bad”) state. In Section 2.2, we consider

a set of *two* possible actions  $\mathcal{A} := \{0, 1\}$ , where 1 represents the decision to select (“pull”) arm  $i \in [N]$  at time  $t \in [T]$ , and 0 represents the choice to be passive (not pull). In Section 2.3, we consider an *augmented* action space,  $\mathcal{A} := \{0 : \text{no-act}, 1 : \text{message}, 2 : \text{pull}\}$ , where the message action represents an intermediate action with respect to desirability that gets “unlocked” for vertex  $v$  at time  $t$  only when one of  $v$ ’s in-degree neighbors receives a pull at time  $t$  (see Section 2.3.2.3 for details).

In the general RMAB setting, each arm’s state  $s_t^i$  is observable. In Section 2.2, we consider the partially observable extension introduced by Mate et al. [20], where arms’ states are only observable when they are pulled. In the absence of a pull, an arm’s state is replaced with the probabilistic *belief*  $b_t^i \in [0, 1]$  that it is in state 1. Such partial observability captures uncertainty regarding patient status and treatment efficacy associated with outpatient or remotely-administered interventions.

**Transition matrices:** Each arm  $i$  is characterized by a set of transition matrices  $P$ , where  $P_{s,s'}^{a,i}$  represents the probability of transitioning from state  $s$  to state  $s'$  when action  $a$  is taken (the superscript  $i$  is often omitted when clear from context, i.e., for notational convenience). We assume  $P$  to be (a) static and (b) known by the agent at planning time. Assumptions (a) and (b) are likely to be violated in practice; however, they provide a useful modeling foundation, and can be modified to incorporate additional uncertainty, such as the requirement that transition matrices must be learned [21]. Clinical researchers often use longitudinal data to construct risk-adjusted transition matrices that encode cohort-specific transition probabilities. These can guide patient-level decision-making [22].

Consistent with previous literature, we assume strictly positive transition matrix entries, and impose four *structural constraints*: (a)  $P_{0,1}^0 < P_{1,1}^0$ ; (b)  $P_{0,1}^1 < P_{1,1}^1$ ; (c)  $P_{0,1}^0 < P_{0,1}^1$ ; (d)  $P_{1,1}^0 < P_{1,1}^1$ .

[20]. These constraints are application-motivated, and imply that arms are more likely to remain in a “good” state than change from a bad state to a good one, and that a pull is helpful when received. In the absence of such constraints, the effect of the intervention may be superfluous or harmful, rather than desirable. In Section 2.3.2.3, we extend these constraints to a setting featuring more than two actions.

**Objective and constraints:** In the canonical RMAB setting, the agent’s goal is to find a policy  $\pi^*$  that maximizes total expected reward  $\arg \max_{\pi} \mathbb{E}_{\pi}[R(r(s))]$  while satisfying a *budget constraint*,  $k \ll N \in \mathbb{N}$ , which allows the agent to select at most  $k$  arms at each timestep. In Section 2.3, we modify this slightly by considering a real-valued budget constraint,  $B \in \mathbb{R}_{\geq 0}$  to account for our incorporation of an intermediate action with real-valued cost  $\psi \in [0, 1)$  (see Section 2.3.2.3 for details). In both Section 2.2 and 2.3, we consider a cumulative reward function,  $R(\cdot) := \sum_{i \in [N]} \sum_{t \in [T]} \beta^{t-1} r(s_t^i)$ , for some discount rate  $\beta \in [0, 1]$ , and non-decreasing  $r(s)$ .

## 2.2 Probabilistically Fair Allocation in the Restless Bandit Setting

This work was done in collaboration with my co-first author, Aviva Prins, as well as Aravind Srinivasan and John P. Dickerson; see [4]. Non-archival versions of this work have appeared at the Responsible Decision Making in Dynamic Environments workshop (ICML 2022) and at Equity and Access in Algorithms, Mechanisms, and Optimization (EAAMO) 2022.

### 2.2.1 Introduction

Constrained resource allocation problems are common in healthcare, where clinicians may be tasked with monitoring large, distributed patient populations and determining which individuals to expend scarce resources on so as to maximize total welfare. Variants of the restless multi-armed bandit (RMAB) framework introduced in Section 2.1 have been proposed to determine which inmates should be prioritized to receive hepatitis C treatment in U.S. prisons [23], and which tuberculosis patients should receive medication adherence support in India [20].

Current state-of-the-art (SoTA) approaches to solving RMABs rely on the indexing work introduced by Whittle [2]. While the Whittle index solves an otherwise PSPACE-complete problem in an asymptotically optimal fashion by decoupling arms [3], it fails to provide any guarantees about how pulls will be distributed *among arms*.

Though the intervention is canonically assumed to be beneficial for *every* arm, the marginal benefit (i.e., relative increase in the probability of a favorable state transition) varies in accordance with each arm’s underlying state transition function. Consequently, Whittle index-based maximization of total expected reward *without regard for distributive fairness* empirically allocates all available interventions to a small subset of arms, ignoring the rest [24].

There are many application domains where this bimodal distributive outcome may be perceived as unfair or undesirable by beneficiaries and decision-makers, thus motivating efforts to incentivize or guarantee distributive fairness. In the aforementioned healthcare examples, resource constraints and variation in transition dynamics interact. A practical consequence is that a majority of patients will *never* receive the beneficial intervention(s) in question. This, in turn, means that their clinical outcomes will be strictly worse in expectation than they would be under a policy that guaranteed a non-zero probability of receiving the intervention at each timestep.

These considerations are not restricted to healthcare contexts. In animal poaching prevention, the planner must select a subset of areas to patrol (i.e., arms to pull). Concentrating resources on a handful of locations while repeatedly choosing not to patrol other areas may prevent the defenders from learning about attacks in these areas, and/or cause poachers to modify their strategies [25]. In wireless scheduling, multiple processes compete to transmit packets over a shared wireless channel. When scheduling the transmissions, the agent must not only maximize reward but also ensure the performance of critical applications codified in Quality of Service (QoS) guarantees [26, 27, 28].

To improve distributive fairness, we explore whether it is possible to modify the Whittle index to guarantee each arm at least one pull per user-defined time interval, but find this to be intractable. We then introduce PROBFair, a state-agnostic policy that maps each arm to a fairness-constraint satisfying, stationary probability distribution over actions that takes the arm’s transition matrix into account. At each timestep, we then use a dependent rounding algorithm [29] to sample from this probabilistic policy to produce a budget-constraint satisfying discrete action vector.

We evaluate PROBFair on a randomly generated dataset and a realistic dataset derived from obstructive sleep apnea patients tasked with nightly self-administration of continuous positive

airway pressure (CPAP) therapy [30, 31]. Our core contributions include:

- (i) A novel approach that is both efficiently computable and reward maximizing, subject to the guaranteed satisfaction of budget *and* probabilistic fairness constraints.
- (ii) Empirical results demonstrating that PROBFair is competitive vis-à-vis other fairness-inducing policies, and stable over a range of cohort composition scenarios.

### 2.2.2 Restless Multi-Armed Bandit Model

We provide a formal overview of the restless multi-armed bandit (RMAB) model in Section 2.1. Here, we briefly restate the key components for convenience and formalize our proposed extension, which takes the form of a fairness-motivated constraint. Per Section 2.1, a restless multi-armed bandit consists of  $N \in \mathbb{N}$  independent arms, each of which evolves over a finite time horizon  $T \in \mathbb{N}$ , according to an associated Markov Decision Process (MDP). Each arm’s MDP is characterized by a 4-tuple  $(\mathcal{S}, \mathcal{A}, P, r)$  where  $\mathcal{S}$  represents the state space,  $\mathcal{A}$  represents the action space,  $P$  represents an  $|\mathcal{S}| \times |\mathcal{A}| \times |\mathcal{S}|$  transition matrix, and  $r : \mathcal{S} \rightarrow \mathbb{R}$  represents a local reward function that maps states to real-valued rewards. Appendix A.1 summarizes notation; note that  $[N]$  denotes the set  $\{1, 2, \dots, N\}$ .

We extend the RMAB model by introducing a Boolean-valued, distributive fairness-motivated constraint, which may take one of two general forms:

1. *Time-indexed*: A function  $g(\cup_{t \in [T]} \{\vec{a}_t\})$  which is satisfied if each arm is pulled at least once within each user-defined time interval  $\nu \leq T$  (e.g., at least once every seven days), or a minimum fraction  $\psi \in (0, 1)$  of times over the entire time horizon [28].
2. *Probabilistic*: A function  $g'(\vec{p}^i | \vec{a}_t \sim \vec{p}^i \forall t)$  which operates on the stationary probability

vector  $\vec{p}^i$ , from which discrete actions are drawn, by requiring the probability that each arm receives a pull at any given  $t$  to fall within an interval  $[\ell, u]$  where  $0 < \ell \leq \frac{k}{N} \leq u \leq 1$ .

### 2.2.3 Context, Motivation & Related Work

In this section, we motivate our ultimate focus on probabilistic fairness by revisiting the distribution of pulls under Whittle-index based policies. We begin by providing background information on the Whittle index, and then proceed to ask: (1) Which arms are ignored, and why does it matter? (2) Is it possible to modify the Whittle index so as to provide a *time-indexed fairness guarantee* for each arm? In response to the latter, we demonstrate that time-indexed fairness guarantees necessitate the coupling of arms, which undermines the indexability of the problem. We then identify prior work at the intersection of algorithmic fairness, constrained resource allocation, and multi-armed bandits, and identify desiderata that characterize our own approach.

#### 2.2.3.1 Background: Whittle Index-based Policies

Pre-computing the optimal policy for a given set of restless or collapsing arms is PSPACE-hard in the general case [1]. However, as established by Whittle [2] and formalized by Weber and Weiss [3], if the set of arms associated with a problem are *indexable*, we can decouple the arms and efficiently solve the problem using an asymptotically-optimal heuristic index policy.

**Mechanics:** At each timestep  $t \in [T]$ , the value of a pull, in terms of both immediate and expected discounted future reward, is computed for each decoupled arm,  $i \in [N]$ . This value-computation step relies on the notion of a subsidy,  $m$ , which can be thought of as the

opportunity cost of passivity. Formally, the Whittle index is the subsidy required to make the agent indifferent between *pulling* and *not pulling* arm  $i$  at time  $t$ . (Per Section 2.1,  $b$  denotes the probabilistic belief that an arm is in state  $s = 1$ ; for restless arms,  $b_t^i = s_t^i \in \{0, 1\}$ ).

$$W(b_t^i) = \inf_m \{m \mid V_m(b_t^i, a_t^i = 0) \geq V_m(b_t^i, a_t^i = 1)\} \quad (2.1)$$

The value function  $V_m(b)$  represents the maximum expected discounted reward under passive subsidy  $m$  and discount rate  $\beta$  for arm  $i$  with belief state  $b_t^i \in [0, 1]$  at time  $t$ :

$$V_m(b_t^i) = \max \begin{cases} m + r(b_t^i) + \beta V_m(b_{t+1}^i) & \text{passive} \\ r(b_t^i) + \beta [b_t^i V_m(P_{1,1}^1) + (1 - b_t^i) V_m(P_{0,1}^1)] & \text{active} \end{cases} \quad (2.2)$$

Once the Whittle index has been computed for each arm, the agent sorts the indices, and the  $k$  arms with the greatest index values receive a pull at time  $t$ , while the remaining  $N - k$  arms are passive. Weber and Weiss [3] give sufficient conditions for *indexability*:

**Definition 2.2.1.** An arm is indexable if the set of beliefs for which it is optimal to be passive for a given  $m$ ,  $\mathcal{B}^*(m) = \{b \mid \forall \pi \in \Pi_m^*, \pi(b) = 0\}$ , monotonically increases from  $\emptyset$  to the entire belief space as  $m$  increases from  $-\infty$  to  $+\infty$ . An RMAB is indexable if every arm is indexable.

**Indexability** is often difficult to establish, and computing the Whittle index can be complex [32]. Prevailing approaches rely on proving the optimality of a *threshold policy* for a subset of transition matrices [33]. A *forward* threshold policy pulls an arm when its state is at or below a given threshold, and makes the arm passive otherwise; the converse is true for a *reverse* threshold policy. Mate et al. [20] give such conditions for this RMAB setting, when  $r(b) = b$ ,



and provide an algorithm, THRESHOLD WHITTLE, that is asymptotically optimal for forward threshold-optimal arms. Mate et al. [34] expand on this work for any non-decreasing  $r(b)$  and present the RISK-AWARE WHITTLE algorithm.

### 2.2.3.2 Motivation: Individual Welfare & Whittle

**Bimodal allocation:** Existing theory does not offer any guarantees about how the sequence of actions will be distributed over arms under Whittle index-based policies, nor about the probability with which a given arm can expect to be pulled at any particular timestep. Prins et al. [24] demonstrate that Whittle-based policies tend to allocate all pulls to a small number of arms, neglecting most of the population. We present similar findings in Appendix A.2.

This bimodal distribution is a consequence of how the Whittle index prioritizes arms. Whittle favors arms for whom a pull is most beneficial to achieving sustained occupancy in the “good” state, regardless of whether this results in the same subset of arms repeatedly receiving pulls. While the structural constraints in Section 2.1 ensure that a pull is beneficial for every arm, marginal benefit varies. Since reward is a function of each arm’s underlying state, arms whose trajectories are characterized by a relative—*but not absolute*—indifference to the intervention are likely to be ignored.

**Ethical implications:** This zero-valued lower bound on the number of pulls an arm can receive aligns with a *utilitarian* approach to distributive justice, in which the decision-maker seeks to allocate resources so as to maximize total expected utility [35, 36]. This may be incompatible with competing pragmatic and ethical desiderata, including *egalitarian* and *prioritarian* notions of distributive fairness, in which the decision-maker seeks to allocate resources equally among arms

(e.g., ROUND-ROBIN), or prioritize arms considered to be worst-off under the status quo, for some quantifiable notion of *worst-off* that induces a partial ordering over arms [37, 38]. We consider the *worst off* to be arms who would be deprived of algorithmic attention (e.g., not receive any pulls), or, from a probabilistic perspective, would have a *zero-valued lower bound* on the probability of receiving a pull at any given timestep.

**Why algorithmic attention?** This choice is motivated by our desire to improve *equality of opportunity* (i.e., access to the beneficial intervention) rather than *equality of outcomes* (i.e., observed adherence). The agent directly controls who receives the intervention, but has only indirect control (via actions) over the sequence of state transitions an arm experiences. Additionally, proclivity towards adherence may vary widely in the absence of restrictive assumptions about cohort homogeneity, and focusing on equality of outcomes could thus entail a significant loss of total welfare.

**Distributive fairness and algorithmic acceptability:** To realize the benefits associated with an algorithmically-derived resource allocation policy, practitioners tasked with implementation must find the policy to be acceptable (i.e., in keeping with their professional and ethical standards), and potential beneficiaries must find participation to be rational.

With respect to *practitioners*, many clinicians report experiencing mental anguish when resource constraints force them to categorically deny a patient access to a beneficial treatment, and may resort to providing improvised and/or sub-optimal care [39]. Providing fairness-aware decision support can improve acceptability [40, 41] and minimize the loss of reward associated with ethically-motivated deviation to a sub-optimal but equitable approach such as ROUND-ROBIN [42, 43]. For *beneficiaries*, we posit that an arm may consider participation rational when it results in an increase in expected time spent in the adherent state relative to non-participation

(e.g., due to receiving a strictly positive number of pulls in expectation).

### 2.2.3.3 Time-indexed Fairness and Indexability

We now consider whether it is possible to modify the Whittle index to *guarantee* time-indexed fairness (as defined in Section 2.2.2). To begin, recall that the efficiency of Whittle index-based policies stems from our ability to decouple arms when we are only concerned with maximizing total expected reward [2, 3]. However, guaranteeing time-indexed fairness in the planning setting requires time-stamped record keeping. It is no longer sufficient to compute each arm’s infimum subsidy in isolation and order the resulting set of values. Instead, for an optimal index policy to be efficiently computable, it must be possible to modify the value function (Equation 2.11) so as to ensure that the infimum subsidy each arm would require in the absence of fairness constraints is minimally perturbed via augmentation or “donation”, so as to maximize total expected reward while ensuring its own fairness constraint satisfaction or the constraint satisfaction of other arms, respectively, *without* requiring input from other arms.

Plausible modifications include altering the conditions under which an arm receives the subsidy associated with passivity,  $m$ , or introducing a modified reward function,  $r'(b)$  that is capable of accounting for an arm’s fairness constraint satisfaction status in addition to its state at time  $t$ . For example, we might use an indicator function to “turn off” the subsidy until arm  $i$  has been pulled at least once within the interval in question, or increase reward as an arm’s time-since-pulled value approaches the interval cut-off, so as to incentivize a constraint-satisfying pull. When these modifications are viewed from the perspective of a single arm, they *appear* to have the desired effect: if no subsidy is received, it will be optimal to pull for all belief states;

similarly, for a fixed  $m$ , as reward increases it will be optimal to pull for an increasingly large subset of the belief state space.

Recall, however, that structural constraints ensure that when an arm is considered in isolation, the optimal action will *always* be to pull. Whether or not arm  $i$  is *actually* pulled at time  $t$  depends on how the infimum subsidy,  $m$ , it requires to accept passivity at time  $t$  compares to the infimum subsidies required by other arms. Thus, any modification intended to *guarantee* time-indexed fairness constraint satisfaction must be able to alter the ordering *among* arms, such that any arm  $i$  which would otherwise have a subsidy with rank  $> k$  when sorted in descending order will now be in the top- $k$  arms. Even if we were able to construct such a modification for a single arm without requiring system state, if *every* arm had this same capability, then a new challenge would arise: we would be unable to distinguish among arms, and arbitrary tie-breaking could again jeopardize fairness constraint satisfaction. If it is not possible to decouple arms, then we must consider them in tandem—as a result, the tractability of a Whittle index-based approach breaks down.

#### 2.2.3.4 Additional Related Work

While multi-armed bandit problems are canonically framed from the perspective of the decision-maker, interest in individual and group fairness in this setting has grown in recent years [28, 44, 45]. In the *stochastic* multi-armed bandit setting, each arm is characterized by a fixed but unknown average reward rather than by an MDP. The decision-maker thus faces uncertainty about the true utility of each arm and must balance exploration (i.e., pulling arms to gain information about their reward distributions) with exploitation (i.e., pulling the optimal arm(s)) to maximize expected reward. Joseph et al. [44] examine fairness among arms in this

setting, and introduce a definition that requires the decision-maker to favor (i.e., select) arms with higher average reward over arms with lower average reward, even in the face of uncertainty. As the authors note, this definition *is* consistent with reward maximization, but imposes a cost in terms of per-round regret when *learning* the optimal policy, due to the fact that arms with overlapping confidence intervals are chained until they can be separated with high confidence.

Prior work in other non-restless bandit settings demonstrates that alternative definitions—i.e., those which center *distributive fairness among arms* as opposed to the principle that arms with similar average rewards should be treated similarly [46], generally entail deviation from optimal behavior. Li et al. [28] study the combinatorial *sleeping* bandit setting, in which arms are stochastic but may be unavailable at any given timestep. They introduce the minimum selection fraction constraint, which we adapt and refer to as time-indexed fairness (see Section 2.2.2). Chen et al. [45] consider the *contextual* bandit setting, and propose an algorithm that guarantees each arm a minimum probability of selection at each timestep.

In the *restless* setting that we consider, prior works have tended toward opposite ends of the reward-fairness spectrum by either: (1) redistributing pulls without providing arm-level guarantees [34, 47]; or (2) guaranteeing time-indexed fairness without providing optimality guarantees [24]. Recent work has also considered the adjacent problem of fairness among intervention *providers* (i.e., workers) [48]. In contrast to prior work, we aim to *guarantee* rather than incentivize fairness, without incurring an exponential dependency on the time horizon or sacrificing optimality guarantees. We thus seek an efficient policy that is reward maximizing, subject to the satisfaction of both budget and probabilistic fairness constraints.

## 2.2.4 Methodological Approach

Here we introduce PROBFair, an approximately optimal solution to a relaxed version of the allocation task in which we guarantee the satisfaction of *probabilistic* rather than *time-indexed* fairness, along with the budget constraint. This relaxation is necessary for tractability, as it allows us to precompute a stationary, *state-agnostic* probability vector,  $\vec{p}^i$ , from which constraint-satisfying discrete actions are drawn.

PROBFair maps each arm  $i$  to an arm-specific, stationary probability distribution over atomic actions, such that for each timestep  $t$ ,  $P[a_t^i = 1] = p_i$  and  $P[a_t^i = 0] = 1 - p_i$ , where  $p_i \in [\ell, u]$  for all  $i \in [N]$  and  $\sum_i p_i = k$ . Here,  $\ell$  and  $u$  are user-defined fairness parameters satisfying  $0 < \ell \leq \frac{k}{N} \leq u \leq 1$ , per Section 2.2.2. Note that  $\ell T$  and  $uT$  can be interpreted as lower and upper bounds on the expected number of pulls an arm will receive over the time horizon.

In Section 2.2.4.1, we describe how to construct the  $p_i$ 's so as to efficiently approximate our constrained reward-maximization objective within a multiplicative factor of  $(1 - \epsilon)$ , for any given constant  $\epsilon > 0$ . We use a dependent rounding approach detailed in Section 2.2.4.2 to sample from this distribution at each timestep *independently*, to produce a discrete action vector,  $\vec{a}_t \in \{0, 1\}^N$ , which is guaranteed to satisfy the budget constraint,  $k$  [29].

To motivate our approach, note that when we take the union of each arm's stationary probability vector, we obtain a system-level policy,  $\pi_{PF} : \{i \mid i \in N\} \rightarrow [1 - p_i, p_i]^N$ . Regardless of the system's initial state, repeated application of this policy will result in convergence to a steady-state distribution in which (WLOG) arm  $i$  is in the adherent state (i.e., state 1) with probability  $x_i \in [\ell, u]$ , and the non-adherent state (i.e., state 0) with probability  $1 - x_i \in [0, 1]$ .

By definition, for any arm  $i$ ,  $x_i$  will satisfy the equation:

$$x_i [(1 - p_i)P_{1,1}^0 + p_iP_{1,1}^1] + (1 - x_i)[(1 - p_i)P_{0,1}^0 + p_iP_{0,1}^1] = x_i. \quad (2.3)$$

Thus,  $x_i = f_i(p_i)$ , where

$$f_i(p_i) = \frac{(1 - p_i)P_{0,1}^0 + p_iP_{0,1}^1}{1 - (1 - p_i)P_{1,1}^0 - p_iP_{1,1}^1 + (1 - p_i)P_{0,1}^0 + p_iP_{0,1}^1} \quad (2.4)$$

We seek the policy which maximizes total expected reward, where reward is non-decreasing in  $s$  (i.e., with time spent in the adherent state). Thus, PROBFair is defined as:

$$\pi_{PF} = \arg \max_{p_i \in [\ell, u]} \sum_i f_i(p_i) \text{ s.t. } \sum_i p_i \leq k \quad (2.5)$$

Solving this constrained maximization problem is thus consistent with maximizing the expected number of timesteps each arm will spend in the adherent state, subject to satisfying the budget *and* probabilistic fairness constraints. We emphasize that our construction process takes the transition matrices of each arm  $i$  into account via  $f_i$  (Equation 2.4).

### 2.2.4.1 Computing the $p_i$ 's: Algorithmic Approach

**Overview:** To construct  $\pi_{PF}$ , we: (1) partition the arms based on the shapes of their respective  $f_i$  functions (Equation 2.4); (2) perform a grid search over possible ways to allocate the budget,  $k$ , between the two subsets of arms; (2a) solve each sub-problem to produce a probabilistic policy for the arms in that subset; (2b) compute the total expected reward of the policy; (3) take the argmax over this set of grid search values to determine the approximately optimal budget

allocation; and (4) form  $\pi_{PF}$  by taking the union over the policies produced by evaluating each sub-problem at its approximately optimal share of the budget. Figure 2.1 visualizes; the remainder of this section provides technical details.

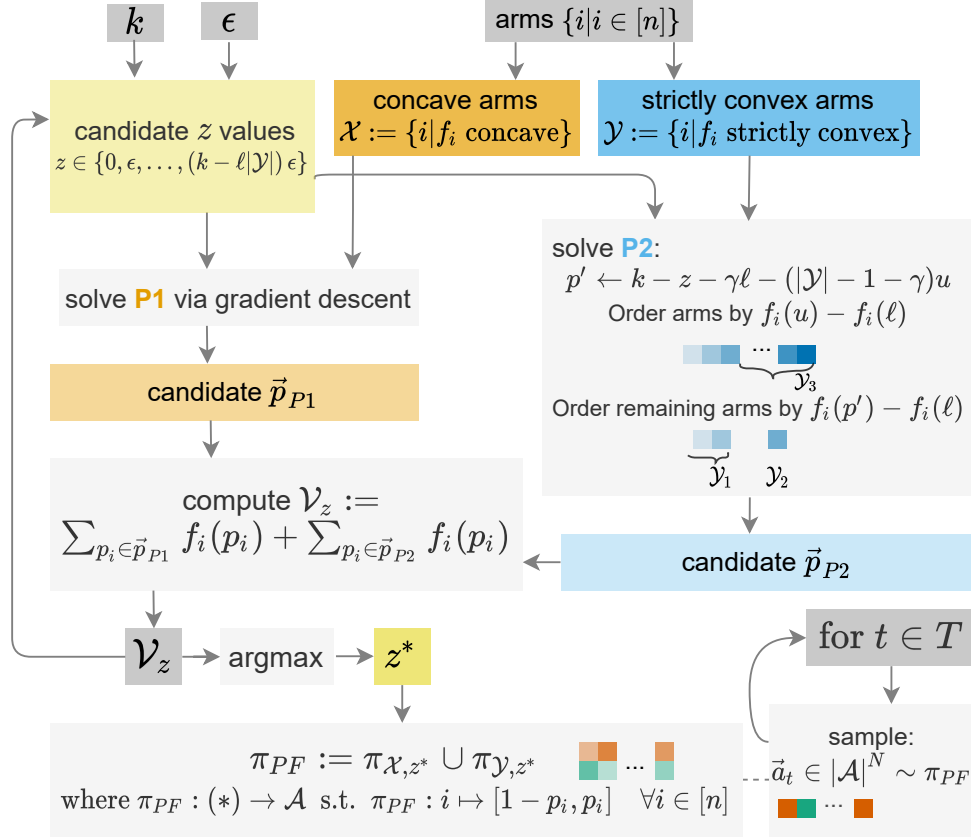


Figure 2.1: PROBFair: constructing and sampling from  $\pi_{PF}$

We begin by introducing two theorems (see Appendix A.4.1 for full proofs):

**Theorem 2.2.2.** For every arm  $i \in [N]$ ,  $f_i(p_i)$  is either *concave* or *strictly convex* in all of  $p_i \in [0, 1]$ .

*Proof Sketch.* WLOG, fix an arm  $i \in [N]$ . The second derivative of  $f_i$  is given by:

$$f_i''(p_i) = \frac{2c_4^2 \left( c_1 - \frac{c_2 c_3}{c_4} \right)}{(c_3 + c_4 p_i)^3} \quad (2.6)$$



For notational convenience,  $c_1$ ,  $c_2$ ,  $c_3$ , and  $c_4$  are constants derived from an arm's transition matrix,  $P^i$ . For all  $p_i \in [0, 1]$ ,  $c_3 + c_4 p_i \in (0, 1)$ . Thus, the sign of  $f_i''(p_i)$  is determined by  $c_1 - \frac{c_2 c_3}{c_4}$ , which does not depend on  $p_i$ .  $\square$

**Theorem 2.2.3.** *For each arm  $i \in [N]$ , the structural constraints introduced in Section 2.1 ensure that  $f_i(p_i)$  is monotonically non-decreasing in  $p_i$  over the interval  $[0, 1]$ .*

By Theorem 2.2.2, the arms can be partitioned into two disjoint sets:  $\mathcal{X} = \{i \mid f_i \text{ is concave}\}$  and  $\mathcal{Y} = \{i \mid f_i \text{ is strictly convex}\}$ . Then,  $\pi_{PF}$  is the union of the solutions to **P1** and **P2** at the optimal grid search value  $z^* = \arg \max_z \sum_{i \in \mathcal{X}} f_i(p_i) + \sum_{i \in \mathcal{Y}} f_i(p_i)$ :

**(P1)** maximize  $\sum_{i \in \mathcal{X}} f_i(p_i)$  subject to:  $p_i \in [\ell, u]$  for all  $i \in \mathcal{X}$ , and  $\sum_{i \in \mathcal{X}} p_i = z$

**(P2)** maximize  $\sum_{i \in \mathcal{Y}} f_i(p_i)$  subject to:  $p_i \in [\ell, u]$  for all  $i \in \mathcal{Y}$ , and  $\sum_{i \in \mathcal{Y}} p_i = k - z$

Algorithm 1 provides pseudocode.

---

**Algorithm 1** PROBFair

---

```

1: procedure PROBFair( $[N], k, \epsilon, \ell, u$ )
2:    $\mathcal{X} \leftarrow \{i \mid f_i \text{ is concave in all of } p_i \in [0, 1]\}$ 
3:    $\mathcal{Y} \leftarrow \{i \mid f_i \text{ is strictly convex in all of } p_i \in [0, 1]\}$ 
4:    $\text{grid\_search\_vals} \leftarrow \{\epsilon j \mid j \in [0, k - \ell|\mathcal{Y}|]\}$ 
5:   for  $z \in \text{grid\_search\_vals}$  do
6:      $\pi_{\mathcal{X}, z} \leftarrow \text{SOLVEP1}(\mathcal{X}, k, z, \ell, u)$ 
7:      $\pi_{\mathcal{Y}, z} \leftarrow \text{SOLVEP2}(\mathcal{Y}, k, z, \ell, u)$ 
8:      $\mathcal{V}_z \leftarrow \sum_{p_i \in \pi_{\mathcal{X}, z}} f_i(p_i) + \sum_{p_i \in \pi_{\mathcal{Y}, z}} f_i(p_i)$ 
9:    $z^* \leftarrow \arg \max_z \mathcal{V}_z$ 
10:   $\pi_{PF} \leftarrow \pi_{\mathcal{X}, z^*} \cup \pi_{\mathcal{Y}, z^*}$ 
11:  return  $\pi_{PF}$ 

```

---

**P1** is a concave-maximization problem that can be solved efficiently via gradient descent. The computational complexity is  $O\left(\frac{|\mathcal{X}|}{\epsilon^2}\right)$  [49]. To solve **P2**, we begin by introducing a lemma that we prove in Appendix A.4.1:

**Lemma 2.2.4.** *P2 has an optimal solution in which  $p_i \in (\ell, u)$  for at most one  $i \in \mathcal{Y}$ .*

*Proof Sketch.* Suppose for contradiction there exists some optimal solution  $\vec{p}$  with distinct indices  $i, j \in \mathcal{Y}$  such that  $p_i, p_j \in (\ell, u)$ . Let us compare  $\vec{p}$  with a perturbed solution  $p_i := p_i + \epsilon$  and  $p_j := p_j - \epsilon$ . Using a Taylor series expansion, the change in objective must be  $(\epsilon^2/2) \cdot (f_i''(p_i) + f_j''(p_j)) + O(\epsilon^3)$ . Since  $f_i$  and  $f_j$  are strictly convex,  $f_i''(p_i) + f_j''(p_j) > 0$ . Thus, the objective increases regardless of the sign of (tiny)  $\epsilon$ , a contradiction.  $\square$

Given this structure, an optimal solution  $\{p_i^* \mid i \in \mathcal{Y}\}$  will set some number of arms  $\gamma \in \mathbb{Z}^+$  to  $\ell$ , at most one arm to  $p' \in (\ell, u]$ , and the remaining  $|\mathcal{Y}| - \gamma - 1$  arms to  $u$ . We represent these subsets by  $\mathcal{Y}_1, \mathcal{Y}_2$ , and  $\mathcal{Y}_3$ , respectively. Let  $\gamma = \left\lfloor \frac{|\mathcal{Y}|u - (k-z)}{u-\ell} \right\rfloor$ , and  $p' = k - z - |\mathcal{Y}_1|\ell - |\mathcal{Y}_3|u \in (\ell, u]$ . Intuitively, when the remaining budget  $k - z$  allows us to set all arms in  $\mathcal{Y}$  to  $u$ ,  $\gamma = |\mathcal{Y}_1| = 0$ . Conversely, when there is only enough budget left to satisfy the fairness constraint for arms in  $\mathcal{Y}$ ,  $\gamma = |\mathcal{Y}_1| = |\mathcal{Y}|$ . With the cardinality of each subset thus established, per Theorem 2.2.5 (see below), we use Algorithm 2 to optimally partition the arms in  $\mathcal{Y}$ .

---

**Algorithm 2** SOLVEP2

Note: all sorts are ascending; arrays are zero-indexed.

---

```

1: procedure SOLVEP2( $\mathcal{Y} \subseteq N, k, z, \ell, u$ )
2:    $\gamma \leftarrow \left\lfloor \frac{|\mathcal{Y}|u - (k-z)}{u-\ell} \right\rfloor$ 
3:    $p' \leftarrow k - z - \gamma\ell - (|\mathcal{Y}| - 1 - \gamma)u$ 
4:   if  $|\mathcal{Y}| - \gamma - 1 > 0$  then
5:      $\Delta_{\mathcal{Y}} = \text{sort}([f_i(u) - f_i(\ell) \mid \forall i \in \mathcal{Y}])$ 
6:      $\mathcal{Y}_3 \leftarrow \{(\Delta_{\mathcal{Y}})[(|\mathcal{Y}| - \gamma - 1) :]\}$ 
7:   else  $\mathcal{Y}_3 \leftarrow \emptyset$ 
8:    $\Delta_{\mathcal{Y} \setminus \mathcal{Y}_3} = \text{sort}([f_i(p') - f_i(\ell) \mid \forall i \in \mathcal{Y} \setminus \mathcal{Y}_3])$ 
9:    $\mathcal{Y}_1 \leftarrow \{(\Delta_{\mathcal{Y} \setminus \mathcal{Y}_3})[:\gamma]\}$ 
10:   $\mathcal{Y}_2 \leftarrow \{(\Delta_{\mathcal{Y} \setminus \mathcal{Y}_3})[\gamma]\}$ 
11:   $\pi_{\mathcal{Y}} := i \mapsto (\ell \mid i \in \mathcal{Y}_1) \vee (p' \mid i \in \mathcal{Y}_2) \vee (u \mid i \in \mathcal{Y}_3)$ 
12:  return  $\pi_{\mathcal{Y}}$ 

```

---

**Theorem 2.2.5.** *Alg. 2 yields the mapping from arms in  $\mathcal{Y}$  to subsets in  $\{\mathcal{Y}_1, \mathcal{Y}_2, \mathcal{Y}_3\}$  which maximizes  $\sum_{i \in \mathcal{Y}} f_i(p_i)$  s.t.  $\sum_{i \in \mathcal{Y}} p_i = k - z$ . (See Appendix A.4.1 for the complete proof).*

*Proof Sketch.* By Lemma 2.2.4, there exists at most one arm with optimal value  $p_i^* \in (\ell, u)$ . By Lemma A.4.1,  $\gamma := |\mathcal{Y}_1| = \left\lfloor \frac{|\mathcal{Y}|u - (k-z)}{u-\ell} \right\rfloor$  and  $p' = k - z - |\mathcal{Y}_1|\ell - |\mathcal{Y}_3|u \in (\ell, u)$ . Then, we can rewrite Equation 2.5 as an optimization problem over set assignment:

$$\begin{aligned} & \arg \max_{\{\mathcal{Y}_1, \mathcal{Y}_2, \mathcal{Y}_3\}} \sum_{i \in \mathcal{Y}_1} f_i(\ell) + f_j(p') + \sum_{i'' \in \mathcal{Y}_3} f_{i''}(u) \\ & \text{s.t. } |\mathcal{Y}_1| = \gamma, \mathcal{Y}_2 = \{j\}, \bigcap_{x=1}^3 \mathcal{Y}_x = \emptyset, \text{ and } \bigcup_{x=1}^3 \mathcal{Y}_x = \mathcal{Y}. \end{aligned}$$

By algebraic manipulation, assigning the  $|\mathcal{Y}| - \gamma - 1$  arms with maximal values of  $f_i(u) - f_i(\ell)$  to  $\mathcal{Y}_3$  produces a maximal solution. Similarly, we assign  $j \in \mathcal{Y}_2$  if  $f_j(p') - f_j(\ell)$  is maximal among the remaining arms. By definition,  $\mathcal{Y}_1^* = \mathcal{Y} \setminus (\mathcal{Y}_2 \cup \mathcal{Y}_3^*)$ , which completes the proof.  $\square$

**Corollary 2.2.6.** *Alg. 2 has time complexity  $O(|\mathcal{Y}| \log |\mathcal{Y}|)$ .*

With our solutions to **P1** and **P2** so defined, the cost of finding our probabilistic policy in this way is  $O\left(\frac{k-\ell|\mathcal{Y}|}{\varepsilon} \left(\frac{|\mathcal{X}|}{\varepsilon^2} + |\mathcal{Y}| \log |\mathcal{Y}|\right)\right)$ , which is at worst  $O\left(\frac{kN}{\varepsilon^3}\right)$  when all  $N$  arms are in  $\mathcal{X}$ .

### 2.2.4.2 Sampling Approach

For problem instances with feasible solutions, Algorithm 1 returns  $\pi_{PF}$ , a mapping from the set of arms to a set of stationary probability distributions over actions, such that for each arm  $i$ , the probability of receiving a pull at any given timestep is in  $[\ell, u]$ . By virtue of the fact that  $\ell > 0$ , this policy guarantees probabilistic fairness constraint satisfaction for all arms. We use a

linear-time algorithm introduced by Srinivasan [29] and detailed in Appendix A.4.2 to sample from  $\pi_{PF}$  at each timestep, such that the following properties hold: (1) with probability one, we satisfy the budget constraint by pulling exactly  $k$  arms; and (2) any given arm  $i$  is pulled with probability  $p_i$ . Formally, each time we draw a vector of binary random variables  $(X_1, X_2 \dots X_N)$  from the distribution  $\pi_{PF}$ ,  $\Pr [i : X_i = 1 | = k] = 1$  and  $\forall i, \Pr[X_i = 1] = p_i$ .

## 2.2.5 Experimental Evaluation

In this section, we empirically demonstrate that PROBFAIR enforces the probabilistic fairness constraint introduced in Section 2.2.2 with minimal loss in total expected reward, relative to fairness-aware alternatives. We begin by identifying our comparison policies, evaluation metrics, and datasets. We then present results from three experiments: (1) PROBFAIR versus fairness-inducing alternative policies, holding the cohort fixed and considering fairness-aligned sets of hyperparameters; (2) PROBFAIR evaluated on a breadth of cohorts representing different types of patient populations; and (3) PROBFAIR when fairness is *not* enforced (i.e.,  $\ell = 0$ ), to examine the cost of state agnosticism.

### 2.2.5.1 Experimental Setup

**Policies:** In each of our experiments, we compare PROBFAIR against a subset of the following baseline<sup>§</sup> and fairness- $\{\text{inducing}^\dagger, \text{guaranteeing}^\ddagger, \text{and agnostic}^*\}$  policies:

We specifically consider three **THRESHOLD WHITTLE**-based heuristics:  $H_{\text{FIRST}}$ ,  $H_{\text{LAST}}$ , and  $H_{\text{RAND}}$ . These heuristics partition the  $k$  pulls available at each timestep into (un)constrained subsets, where a pull is *constrained* if it is executed to satisfy a time-indexed fairness constraint.

RANDOM <sup>§</sup>	Select $k$ arms uniformly at random at each $t$ .
ROUND-ROBIN <sup>§, ‡</sup>	Select $k$ arms at each $t$ in fixed, sequential order.
TW-BASED HEURISTICS <sup>‡</sup>	Select top- $k$ arms based on Whittle index values. Available arms vary based on time-indexed fairness constraint satisfaction [24].
RISK-AWARE TW (RA-TW) <sup>†</sup>	Select top- $k$ arms based on Whittle index values. Incentivizes fairness via concave reward function [34].
THRESHOLD WHITTLE (TW) <sup>*</sup>	Select top- $k$ arms based on Whittle index values [2, 20].

During constrained pulls, only arms that have not yet been pulled the required number of times within a  $\nu$ -length interval are available; other arms are excluded from consideration, unless *all* arms have already satisfied their constraints.  $H_{\text{FIRST}}$ ,  $H_{\text{LAST}}$ , and  $H_{\text{RAND}}$  position constrained pulls at the beginning, end, or randomly within each interval of length  $\nu$ , respectively. Appendix A.4.3.1 provides pseudocode.

**Objective:** In all experiments, we assign equal value to the adherence of a given arm over time. Thus, we set our objective to reward occupancy in the “good” state: a simple local reward  $r_t(s_t^i) := s_t^i \in \{0, 1\}$  and undiscounted cumulative reward function,  $R(r(s)) := \sum_{i \in [N]} \sum_{t \in [T]} r(s_t^i)$ .

**Evaluation metrics:** We are interested in comparing policies along two dimensions: reward maximization and fairness (i.e., with respect to the distribution of algorithmic attention). To this end, we rely on two performance metrics: (a) intervention benefit and (b) earth mover’s distance.

*Intervention benefit (IB)* is the total expected reward of an algorithm, normalized between the reward obtained with no interventions (0% intervention benefit) and the asymptotically optimal but fairness-agnostic THRESHOLD WHITTLE algorithm (100%) [20]. Formally,

$$\text{IB}_{\text{NoAct, TW}}(\text{ALG}) := \frac{\mathbb{E}_{\text{ALG}}[R_{\text{ALG}}(\cdot)] - \mathbb{E}_{\text{NoAct}}[R(\cdot)]}{\mathbb{E}_{\text{TW}}[R(\cdot)] - \mathbb{E}_{\text{NoAct}}[R(\cdot)]} \quad (2.7)$$

Note that per Lemma A.4.3 (Appendix A.4.3.2), the price of fairness (PoF) [50] is inversely proportional to intervention benefit. We thus report IB.

*Earth mover’s distance (EMD)* is a metric that allows us to compute the minimum cost required to transform one probability distribution into another [51]. We use it to compare algorithms with respect to fairness—i.e., how evenly a set of pulls are allocated among arms. (Other metrics that may measure individual distributive fairness are discussed in Appendix A.4.3.2).

For each algorithm, we consider a discrete distribution  $F$  of observed pull counts, where each bucket,  $j \in \{0 \dots T\}$ , corresponds to a feasible number of total pulls that an arm could receive, and  $F[j] \in \{0 \dots N\}$  corresponds to the number of arms whose observed pull count is equal to  $j$ . Each algorithm produces  $kT$  total pulls, so the distributions have the same total mass.

We use ROUND-ROBIN as a fair reference algorithm since it distributes pulls evenly among arms. We then compute the minimum cost required to transform each algorithm’s distribution,  $F_{\text{ALG}}$ , into that of ROUND-ROBIN’s,  $F_{\text{RR}}$ .

For our application, this is equivalent to:

$$\text{EMD}_{\text{RR}}(\text{ALG}) := \left| \sum_{h=0}^T \sum_{j=0}^h F_{\text{ALG}}[j] - F_{\text{RR}}[j] \right| \quad (2.8)$$

Unless otherwise noted, we normalize EMD such that the maximum distance we encounter, that of TW, is one:

$$\frac{\text{EMD}_{\text{RR}}(\text{ALG}) - \text{EMD}_{\text{RR}}(\text{RR})}{\text{EMD}_{\text{RR}}(\text{TW}) - \text{EMD}_{\text{RR}}(\text{RR})} = \frac{\text{EMD}_{\text{RR}}(\text{ALG})}{\text{EMD}_{\text{RR}}(\text{TW})} \quad (2.9)$$

**Datasets:** We evaluate performance on two datasets: (a) a realistic patient adherence behavior model and (b) a set of randomly generated synthetic transition matrices.

*CPAP Adherence.* Obstructive sleep apnea (OSA) is a common condition that causes

interrupted breathing during sleep [52]; when used throughout the entirety of sleep, continuous positive airway pressure therapy (CPAP) eliminates nearly 100% of obstructive apneas for the majority of treated patients [53]. However, poor adherence behavior in using CPAP reduces its beneficial outcomes. CPAP non-adherence affects an estimated 30-40% of patients [54].

We derive the CPAP dataset that we use in our experiments from the work of Kang et al. [30, 31], who model the dynamics and patterns of patient adherence behavior as a basis for designing effective and economical interventions. In particular, we adapt their Markov model of CPAP adherence behavior (a three-state system based on hours of nightly CPAP usage) to a two-state system using the clinical standard for adherence—at least four hours of CPAP machine usage per night [53]. Kang et al. [30] find, via expectation-maximization on CPAP usage patterns, that patients can be divided into two groups based on this clinical standard. Though patients in the first cluster occasionally miss a night, these patients utilize a CPAP machine for more than four hours every night without assistance, while patients in the second cluster do not. We refer to the latter cluster as the *non-adherent* cohort in our analysis.

Kang et al. [31] consider many intervention effects. We specifically consider an intervention effect,  $\alpha_{\text{interv}} = 1.1$ , that broadly characterizes supportive interventions such as telemonitoring and phone support, which are associated with a moderate 0.70 hours (95% CI  $\pm 0.35$ ) increase in device usage per night [55]. We also add random  $\sigma = 1$  logistic noise to the transition matrices so that there is some variance in individual arm dynamics. To prevent overlap with the general cohort we consider for contrast, added noise can only *hinder* the probability of adherence in the non-adherent cohort.

*Synthetic.* In addition, we construct a synthetic dataset of randomly generated arms such that the structural constraints outlined in Section 2.1 are preserved. We conjecture that forward (reverse)

threshold-optimal arms are a subset of concave (strictly convex) arms (see Appendix A.4.3.3).

### 2.2.5.2 PROBFair vs. Fairness-aware Alternatives

Here we compare PROBFair to policies which either *induce* or *guarantee* fairness. The former includes RISK-AWARE WHITTLE (RA-TW), which incentivizes fairness via concave reward  $r(b)$  [34]. We use the authors’ suggested reward function  $r(b) = -e^{\lambda(1-b)}$ ,  $\lambda = 20$ . This imposes a large negative utility on lower belief values, which motivates preemptive intervention. However, RA-TW does not *guarantee* time-indexed or probabilistic fairness for individual arms. The latter includes ROUND-ROBIN and the FIRST, LAST, and RANDOM heuristics, which guarantee time-indexed fairness but do *not* provide any optimality guarantees.

In Table 2.1, we report average results for each policy, along with margins of error for 95% confidence intervals, computed over 100 simulation seeds for a synthetic cohort of 100 collapsing arms, with  $k = 20$  and  $T = 180$ . To facilitate meaningful comparisons between PROBFair and the heuristics, we consider combinations of values for  $\ell$  and  $\nu$  that produce equivalent, integer-valued lower bounds on the number of pulls any arm can expect to receive—i.e.,  $\min_i \mathbb{E}[\sum_t \mathbb{1}(a_t^i = 1)] = \ell \times T = \frac{T}{\nu}$ . Key findings from this experiment include:

- *Fairer* hyperparameter values ( $\ell \uparrow$ ,  $\nu \downarrow$ ), correspond to decreases in  $\mathbb{E}[\text{IB}]$  and  $\mathbb{E}[\text{EMD}]$ , reflecting improved individual fairness at the expense of total reward.
- PROBFair is competitive with respect to RA-TW, outperforming when  $\ell = 0.056$ , and incurring a slight loss in  $\mathbb{E}[\text{IB}]$  but improvement in  $\mathbb{E}[\text{EMD}]$  for  $\ell = 0.1$ .
- For each  $(\ell, \nu)$  combination, PROBFair performs competitively with respect to the best-performing heuristic (which, like TW, are state-aware, see Section 2.2.5.4).



$\min_i \mathbb{E}[\# \text{ pulls}]$	Policy		$\mathbb{E}[\text{IB}]$ (%)	$\mathbb{E}[\text{EMD}]$ (%)
10 $\ell = 0.056$ $\nu = 18$	PF	$\ell$	$88.73 \pm 0.26$	$81.78 \pm 0.18$
	H <sub>FIRST</sub>	$\nu$	$86.11 \pm 0.26$	$71.53 \pm 0.13$
	H <sub>LAST</sub>	$\nu$	$87.37 \pm 0.28$	<b><math>70.48 \pm 0.12</math></b>
	H <sub>RAND</sub>	$\nu$	<b><math>90.79 \pm 0.22</math></b>	$74.12 \pm 0.15$
18 $\ell = 0.1$ $\nu = 10$	PF	$\ell$	$80.80 \pm 0.30$	$59.96 \pm 0.19$
	H <sub>FIRST</sub>	$\nu$	$76.62 \pm 0.30$	$49.54 \pm 0.09$
	H <sub>LAST</sub>	$\nu$	$77.95 \pm 0.30$	<b><math>49.26 \pm 0.08</math></b>
	H <sub>RAND</sub>	$\nu$	<b><math>81.53 \pm 0.30</math></b>	$52.73 \pm 0.10$
30 $\ell = 0.167$ $\nu = 6$	PF	$\ell$	<b><math>66.12 \pm 0.35</math></b>	$23.61 \pm 0.12$
	H <sub>FIRST</sub>	$\nu$	$63.58 \pm 0.31$	<b><math>18.98 \pm 0.03</math></b>
	H <sub>LAST</sub>	$\nu$	$64.63 \pm 0.34$	$19.47 \pm 0.04$
	H <sub>RAND</sub>	$\nu$	$65.21 \pm 0.32$	$19.64 \pm 0.04$
comparison	RA-TW		$85.12 \pm 0.42$	<b><math>95.80 \pm 0.42</math></b>
	TW		<b><math>100.00 \pm 0.00</math></b>	$100.00 \pm 0.00$
baseline	RANDOM		$50.02 \pm 0.35$	$10.08 \pm 0.10$
	NOACT		$0.00 \pm 0.00$	$73.48 \pm 0.13$
	RR		<b><math>56.96 \pm 0.33</math></b>	<b><math>0.00 \pm 0.00</math></b>

Table 2.1: Expected IB and normalized EMD by policy and fairness bracket.

### 2.2.5.3 PROBFair on a Breadth of Cohorts

In this section we conduct sensitivity analysis with respect to cohort composition. For each dataset, we identify a transition matrix characteristic that can be modified during the generation process to produce a subset of arms that will exhibit less favorable transition dynamics than their peers. For the synthetic dataset, this characteristic is *strict convexity*. For the CPAP dataset, it is *non-adherence*, a mnemonic coined by Kang et al. [30] to characterize a cluster of study participants, and contrast this to a model fit on the general patient population.

For each dataset, we generate ten different cohorts, each of which is characterized by the percentage of unfavorable arms that it contains. We use a seed to control the generation process such that each cohort contains 100 collapsing arms in total. A sliding window of the unfavorable arms we can generate with this seed are included as we increase the cardinality of the unfavorable

subset. For ease of interpretation, we present unnormalized results over 100 simulation seeds with  $k = 20$  and  $T = 180$  in Figure 2.2, and then summarize normalized performance below.

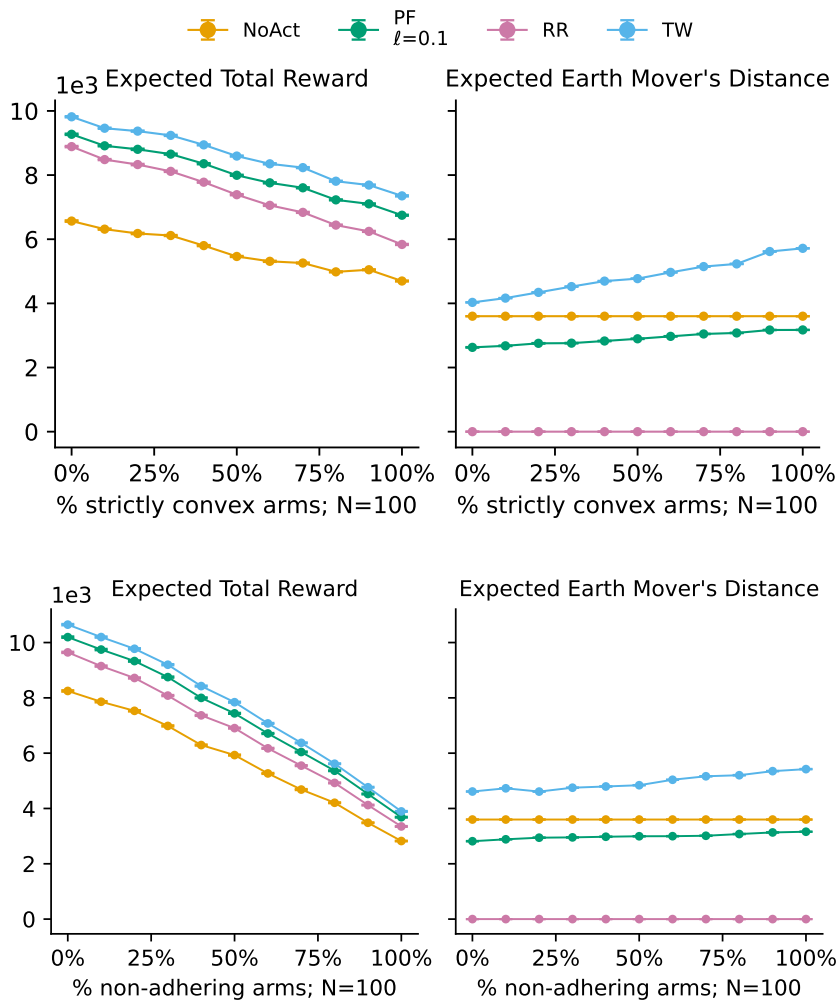


Figure 2.2: Expected total reward (left) and unnormalized EMD (right) on a breadth of cohorts.

- Per Figure 2.2, for each dataset, expected total reward predictably declines for all policies as the percentage of unfavorable arms increases, while unnormalized EMD increases for TW and PROBFair.
- *Synthetic*: As the proportion of strictly convex arms increases, PROBFair’s allocation of resources tends towards the bimodality of TW.

- *CPAP*: As the proportion of non-adherent arms increases, the level of intervention required to improve trajectories rises, but the budget is static.
- For each dataset, PROBFair’s normalized performance remains stable even as cohort composition is varied:
  - *Synthetic*: With respect to IB (EMD), PROBFair achieves an average (over all cohorts) of averages (over 100 simulations per cohort) of  $80.69\% \pm 1.42\%$  ( $58.98\% \pm 1.29\%$ ).
  - *CPAP*: The corresponding values for IB (EMD) are:  $79.84\% \pm 0.68\%$  ( $59.68\% \pm 1.08\%$ ).

#### 2.2.5.4 PROBFair: Price of State Agnosticism

Here, we investigate the cost associated with PROBFair’s state agnosticism, relative to state-aware THRESHOLD WHITTLE. To ensure a fair comparison, we set  $\ell = 0$  and  $u = 1$ , effectively constructing a version of PROBFair in which probabilistic fairness is *not* enforced. (Recall that TW is fairness-agnostic; in the previous results, we do not expect PROBFair to obtain the same total reward as TW).

While PROBFair incorporates each arm’s structural information (i.e., transition matrices), it produces a set of *stationary* probability distributions over actions from which all discrete actions are subsequently drawn. TW, in contrast, ingests each arm’s current state at each timestep, and is thus able to exploit *realized* sequences of state transitions.

While we thus expect PROBFair to incur some loss in intervention benefit, our results (computed over 100 simulation seeds, with  $k = 20$ ,  $N = 100$ , and  $T = 180$ ) indicate that this loss is acceptable rather than catastrophic. Relative to TW,  $\text{PROBFair}_{\ell=0}$  obtains  $97.41\% \pm 0.26$  of  $\mathbb{E}[\text{IB}]$  and incurs an increase of only  $4.56\% \pm 0.19$  with respect to  $\mathbb{E}[\text{EMD}]$ .

## 2.2.6 Summary of Contributions

In Chapter 2.2, we introduce PROBFair, a novel, probabilistically fair algorithm for constrained resource allocation. Our theoretical results prove that this policy is reward-maximizing, subject to the guaranteed satisfaction of both budget and tunable probabilistic fairness constraints. Our empirical results demonstrate that PROBFair preserves utility while providing fairness guarantees. Promising future directions include: (1) extending PROBFair to address larger state and/or action spaces; and (2) relaxing the requirement for stationarity in the construction of  $\pi_{PF}$ .

## 2.3 Networked Restless Bandits with Positive Externalities

This work was done in collaboration with John P. Dickerson; see [5]. A related, non-archival version featured Pranav Goel as an additional collaborator and appeared at the Disinformation Countermeasures and Machine Learning (DisCoML) workshop (ICML 2022); see [56].

### 2.3.1 Introduction

Here, we study the planning task of allocating budget-constrained indivisible resources so as to maximize the expected amount of time that members of a cohort will spend in a desirable state (e.g., adherent to a prescribed exercise regimen). Restless multi-arm bandits (RMABs) are well-suited for such tasks, as they represent each individual as a Markov decision process (MDP) whose stochastic state transitions are governed by an action-dependent transition function.

Conventionally, an arm must receive the resource at time  $t$  to derive any benefit from it, where benefit takes the form of an increased probability of transitioning to the desirable state at time  $t + 1$  (i.e., relative to non-receipt). However, many resource allocation tasks *occur within communities*, and can be characterized by *positive externalities* that allow arms to derive partial, indirect benefit when their neighbor(s) receive the resource. We consider chronic disease management programs as a motivating example. These programs often combine resource-constrained physician-support with less cost-intensive, more scalable peer support to encourage participants to make lifestyle modifications. To this end, we introduce *networked restless bandits*, a novel multi-armed bandit setting in which arms are both restless and embedded within a directed graph. We then present a graph-aware, Whittle-based heuristic algorithm that is constrained reward-maximizing in this setting.

Our core contributions include:

- (i) Our networked restless bandit model, which lets us represent topological relationships between arms, and associate arm  $i$ 's receipt of a pull with positive externalities for its neighbors.
- (ii) GRETA, a graph-aware, Whittle index-based heuristic algorithm that lets us efficiently construct a constrained reward-maximizing mapping from arms to actions at each timestep.
- (iii) Empirical results which demonstrate that GRETA outperforms comparison policies across a range of hyperparameter values and graph topologies.

### 2.3.1.1 Related Work

**Restless bandits:** The restless multi-armed bandit (RMAB) framework was introduced by Whittle [2] as a way to model the sequential allocation of a budget-constrained, indivisible resource over a population of  $N$  dynamic arms, where: (1) at most  $k \ll n$  arms can receive the resource (i.e., a pull) at any given timestep; and (2) the state of each arm evolves over time, regardless of whether or not it is pulled. We provide a formal description in Section 2.3.2.1.

**Indexability:** In the general case, it is PSPACE-hard to pre-compute the optimal policy for a given cohort of restless arms [1]. However, as conjectured by Whittle [2] and proven by Weber and Weiss [3], when each arm is indexable, a tractable solution exists that is provably asymptotically optimal: we can decouple the arms and consider a Lagrangian relaxation of the original problem. In this context, the Whittle index can be thought of as the infimum subsidy required to make an arm indifferent between a pull and passivity, given its current state. Whittle-index based policies use these index values to rank arms when selecting which  $k$  arms to pull.

Proving indexability can be difficult and often requires the problem instance to satisfy specific structural properties, such as the optimality of threshold policies [32]. Additionally, much of the foundational work in this space focuses on the two-action setting, and cannot be directly extended to the multi-action setting that we consider.

Glazebrook et al. [57] do consider the multi-action setting, but for divisible rather than indivisible resources; they also require an arm to consume this resource at a level that is decreasing in the resource charge. Killian et al. [58] study multi-action restless bandits and do not make any of the structural assumptions required to verify indexability, but instead develop a Lagrangian bound-minimization approach; however, they do not consider relationships among arms.

Mate et al. [20] introduce the collapsing bandit model, and demonstrate that this problem is indexable when forward or reverse threshold policies are optimal. They also introduce an efficient, closed-form approach to computing the Whittle index called THRESHOLD WHITTLE (TW), and empirically demonstrate that this approach performs well even when optimality conditions are not satisfied. We leverage TW as a subroutine within GRETA.

**Bandits and graphs:** Prior work at the intersection of multi-armed bandits and graphs has tended to focus on *stochastic*, rather than restless arms, and on *graph-structured feedback* (i.e., rewards), rather than the embedding of arms within a directed graph, and/or the spillover effects associated with allocation in the face of adjacency. For example, Valko [59] examines a graph structure among *actions* in *stochastic* bandits, and Lu et al. [60] examines a graph structure over *rewards*. However, we examine a graph structure among *arms* in the restless bandit setting.

In recent work, Ou et al. [61] look at a mobile intervention setting. Similarly to our model, they combine the traditional restless bandit setting with network externalities; however, their model and goal are fundamentally different. Their arms represent locations on a network with

pulls impacting a mixture of subpopulations who are located at or near that pull, probabilistically. In contrast, in our model, vertices represent individual arms, and our algorithm exploits—when advantageous—the propensity for allocating a high-cost, high-benefit resource to one arm to unlock potential lower-cost, intermediate-benefit resources for the arm’s neighbors.

## 2.3.2 Model Preliminaries

### 2.3.2.1 Restless Multi-arm Bandits

The restless bandit (RMAB) setting features an agent with  $n \in \mathbb{N}$  arms. The state of each arm evolves over time and in response to the agent’s actions, in a way that is governed by the arm’s underlying Markov decision process (MDP). Each MDP is defined by a state space,  $\mathcal{S}$ , an action space,  $\mathcal{A}$ , a cost function,  $C : \mathcal{A} \rightarrow \mathbb{R}$ , a local reward function,  $r : \mathcal{S} \rightarrow \mathbb{R}$ , and a transition function,  $P : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ . The objective is to find a policy,  $\pi : \mathcal{S} \rightarrow \mathcal{A}$ , that maximizes total expected discounted reward over a finite time horizon,  $T$ —i.e.,  $\pi^* = \arg \max_{\pi} E_{\pi} [R(\cdot)]$ . The agent must select exactly one action per arm at each timestep, and the associated costs must not exceed the per-timestep budget,  $B \in \mathbb{R}_{\geq 0}$ .

### 2.3.2.2 Motivating Example

For ease of exposition, we ground our networked restless bandit model in a *motivating example*: let arms represent patients striving to adhere to a chronic disease management program, such as an exercise regimen. A patient’s “state” on any given day is thus determined by whether they adhere (i.e., exercise), or fail to adhere to their regimen. To encourage adherence, many such programs feature a combination of resource-constrained physician- and peer support [62].



Examples include, but need not be limited to, a reminder call from a physician, a supportive message from a fellow participant, or the provision of awareness-raising outreach materials. Thus, a coordinator seeking to maximize the number of patients who exercise over the program’s duration might select a small subset of patients each day to receive a call from a physician, and ask these people to message a handful of their peers in-turn, or pass along an educational pamphlet to their caregiver(s). In each case, the lower-cost, easier-to-scale information dissemination option amplifies physician outreach, allowing a broader subset of individuals to receive partial benefit.

### 2.3.2.3 Networked Restless Bandits

With this motivating example in mind, we now introduce our networked restless bandit model, which allows us to model directed relationships among arms. Given a set of  $n$  arms, let  $G = (V, E)$  be a directed graph, and let there exist a bijective mapping  $\varphi$  from arms to vertices — i.e.,  $\forall v \in V, \exists! i \in [n]$  s.t.  $\varphi(i) = v$ . Let a directed edge,  $e \in E$ , exist between arms  $u$  and  $v$  if it is possible for  $v$  to benefit indirectly when  $u$  receives a pull. Let  $\mathcal{N}_{in}(u) = \{v \in V \mid \exists e_{v,u} \in E\}$  and  $\mathcal{N}_{out}(u) = \{v \in V \mid \exists e_{u,v} \in E\}$  represent  $u$ ’s one-hop indegree and outdegree neighborhoods, respectively. This graph is assumed to be constructed or operated by the agent; as such, it is assumed to be observable. Real-life examples with this property include mentoring programs and online social networks.

**State space:** We consider a discrete state space,  $\mathcal{S} := \{0, 1\}$ , where the states admit a total ordering by desirability, such that state 1 is more “desirable” than state 0. In our example, state 0 represents non-adherence to the exercise regimen, while state 1 represents adherence. We assume that each arm’s state is observable (e.g., via fitness tracker data).

**Action space:** The traditional restless bandit setting considers a binary action space,  $\mathcal{A} := \{0, 1\}$ , where 1 (or 0) represents the decision to pull (or not pull) arm  $i$  at time  $t$ . To model positive externalities, we define an extended action space,  $\mathcal{A} := \{0 : \text{no-act}, 1 : \text{message}, 2 : \text{pull}\}$ . Here, actions 0 and 2 correspond to the actions *don't pull* and *pull* respectively. We note that our message action need not represent a literal message. Instead, it represents an intermediate action with respect to desirability that gets “unlocked” as an available action for vertex  $v$  at time  $t$  only when some vertex  $u \in \mathcal{N}_{\text{in}}(v)$  receives a pull at time  $t$ .

**Transition function:** For each arm  $i \in [n]$ , let  $P_{s,s'}^{a,i}$  represent the probability that arm  $i$  will transition from state  $s$  to  $s'$  given action  $a$ . In the offline setting, these transition matrices are assumed to be static and known to the agent at planning time. This assumption is reasonable when historical data from the same or similar population(s) provides a source for informative priors, as is common in many domains, including healthcare and finance [22, 63]. Extension to the online setting where transition matrices must be learned is possible via Thompson sampling [21, 64, 65, 66].

We assume nonzero transition matrix entries, and impose two sets of domain-motivated **structural constraints** [20]:

$$(i) \forall a \in \mathcal{A}, P_{0,1}^a < P_{1,1}^a \text{ and } (ii) \forall (a, a') \in \mathcal{A} \times \mathcal{A}, a < a' \rightarrow P_{0,1}^a < P_{0,1}^{a'}; P_{1,1}^a < P_{1,1}^{a'}.$$

Constraint set (i) implies that each arm is more likely to stay in the desirable state (i.e.,  $s = 1$ ) than transition there from the undesirable state (i.e.,  $s = 0$ ). Constraint set (ii) implies that messages and pulls are beneficial when received, and that a strict preference relation over actions can be defined for each arm, such that  $\text{no-act} \prec \text{message} \prec \text{pull}$ .

**Cost function:** We map our action space to the cost vector  $\vec{c} = [0, \psi, 1]$ , where  $0 \leq \psi < 1$ .

Intuitively, this mapping preserves the standard notion that no cost is incurred when an arm does not receive any form of intervention. It also encodes the idea that the more beneficial an action is, the more expensive it is to provide, which motivates us to exploit positive externalities. Additionally, when there are no edges, i.e.,  $E = \emptyset$ , and no messages can be sent, the unit cost of a pull lets us recover the original restless bandit setting, where we must choose which  $k \ll n$  arms to pull at each timestep.

**Objective and constraints:** It is possible, though not tractable at scale, to take a constrained optimization-based approach to solving for the optimal policy,  $\pi^*$ . We build on Killian et al. [58]’s approach below to show how our constrained setting can be modeled. To begin, let  $\mathbf{s}$  represent a vector containing the state of each arm, i.e.  $[s^i \in \mathcal{S} | i \in [n]]$ , and let  $\mathbf{X}$  represent a matrix containing binary decision variables, one for each of  $n$  arms and  $|\mathcal{A}|$  actions. We require our local reward function,  $r : \mathcal{S} \rightarrow \mathbb{R}$  to be non-decreasing in  $s$ , which is consistent with our goal of maximizing the expected time that each arm spends in the “desirable” state. Equation 2.10 formalizes our task:

$$\begin{aligned}
J(\mathbf{s}) &= \max_{\mathbf{X}} \left\{ \sum_{i=0}^{n-1} r^i(s^i) + \beta \mathbb{E}[J(\mathbf{s}'), \mathbf{X}] \right\} \\
\text{subject to } & \sum_{i=0}^{n-1} \sum_{j=0}^{|\mathcal{A}|-1} x_{i,j} \cdot c_j \leq B \\
& x_{i,1} \leq \bigvee_{i' \in \mathcal{N}_{\text{in}}(i)} x_{i',2} \quad \forall i \in [n] \\
& \sum_{j=0}^{|\mathcal{A}|-1} x_{i,j} = 1 \quad \forall i \in [n] \\
& \mathbf{X} \in \{0, 1\}^{n \times |\mathcal{A}|}
\end{aligned} \tag{2.10}$$

Our goal is to find assignments of the decision variables contained in  $\mathbf{X}$  such that expected discounted reward is maximized, subject to a series of feasibility constraints: (i) across all actions and arms, do not expend more than  $B$  budget; (ii) ensure that if `message` is chosen for an arm  $i$ , then that arm has at least one indegree neighbor  $i'$  such that `pull` was chosen; and, (iii) ensure that each arm receives exactly one action at each timestep. However, two challenges arise: (1) a direct solution via value iteration is exponential in  $n$  and (2) Lagrangian relaxation-based approaches rely on the decoupling of arms, which jeopardizes the satisfaction of our neighborhood constraint on actions. This motivates us to propose a graph-aware, Whittle-based heuristic algorithm.

### 2.3.3 Algorithmic Approach

Here, we introduce GRETA, a graph-aware, Whittle-index based heuristic algorithm that can be used to efficiently construct a constrained reward-maximizing policy. A key insight that GRETA exploits is that while we cannot decouple arms in the networked setting, since we must know whether any of an arm’s indegree neighbors will receive a pull at time  $t$  to know if the arm is eligible to receive a message, we *can* compute two sets of Whittle indices for each arm, by considering each active action as a separate instance of a two-action problem. Note that the structural constraints ensure that for a given state, an arm will require a higher subsidy to forgo a pull as opposed to a message. We can then construct an augmented graph which allows us to compare the cumulative subsidy required for the arms represented by directed edge  $(u, v)$  to forgo a *pull* and *message*, respectively to those required by other directed edges  $\in G$  (including, importantly, the inverse action-pair implied by edge  $(v, u)$ ).

### 2.3.3.1 GRETA: A Graph-aware Heuristic

**Set-up:** We begin by building an augmented graph,  $G'$ . This graph contains every vertex and edge in  $G$ , along with a dummy vertex,  $-1$ , and directed edge  $(u, -1) \forall u \in V$ . This lets us map each directed edge  $(u, v)$  in  $G$  to the action pair  $(\text{pull}, \text{message})$ , and  $(u, -1)$  to  $(\text{pull}, \text{no-act})$ . Figure 2.3 visualizes this process for an example graph,  $G$ , and corresponding augmented graph,  $G'$ ; Appendix B.1.1.1 gives pseudocode. We also construct an augmented arm set,  $[n] \cup \{-1\}$ , and extend our bijective mapping from arms to vertices such that  $\varphi : -1 \mapsto -1$ .

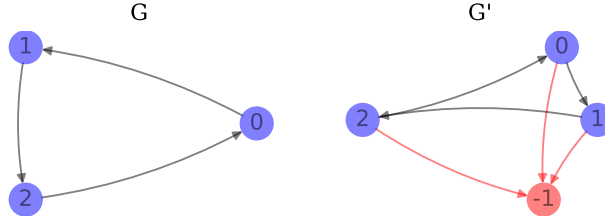


Figure 2.3: Augmented graph example: given graph,  $G$ , we construct augmented graph,  $G'$ .

Next, we pre-compute the Whittle index for each vertex-active action combination  $(v, \alpha) \in V' \times \mathcal{A} \setminus \{0\}$ . When we compute the Whittle index for a given  $(v, \alpha)$  pair, we seek the infimum subsidy,  $m$ , required to make arm  $i$  (i.e.,  $\varphi^{-1}(v)$ ) indifferent between passivity (i.e.,  $\text{no-act}$ ) and receipt of action  $\alpha$  at time  $t$  [2]. We cannot compute the Whittle index for our placeholder  $-1$  vertex because it is not attached to an MDP, so we map it to 0.

---

**Algorithm 3** Compute Whittle indices for  $V' \times \mathcal{A} \setminus \{0\}$

---

- 1: **procedure** WHITTLE( $V', \alpha \in \{1, 2\}, \varphi$ )
  - 2:    $\lambda := i \mapsto \begin{cases} 0, & \text{if } i = -1 \\ \inf_m \{m \mid V_m(s_t^i, a_t^i = 0) \geq V_m(s_t^i, a_t^i = \alpha)\}, & \text{otherwise} \end{cases}$
  - 3:   **return**  $W_\alpha \leftarrow \{\lambda \circ \varphi^{-1}(v) \mid v \in V'\}$
- 

Per Equation 2.11, the value function in line 2 of Algorithm 3 represents the maximum

expected discounted reward that arm  $i \in [n]$  with state  $s_t^i$  can receive at time  $t$  given a subsidy  $m$ , discount rate  $\beta$ , and active action  $\alpha \in \{1, 2\}$ :

$$V_m(s_t^i) = \max \begin{cases} m + r(s_t^i) + \beta V_m(s_{t+1}^i) & \text{no-act} \\ r(s_t^i) + \beta [s_t^i V_m(P_{1,1}^\alpha) + (1 - s_t^i) V_m(P_{0,1}^\alpha)] & \alpha \end{cases} \quad (2.11)$$

**GRETA:** With our augmented graph and Whittle index values in hand, we now present our algorithm. We provide pseudocode in Algorithm 4, and structure our exposition sequentially. At each timestep  $t \in T$ , GRETA takes as *input*: (1) an augmented set of restless arms,  $[n] \cup \{-1\}$  embedded in an augmented directed graph,  $G' = (V', E')$ ; (2) a budget,  $B \in \mathbb{R}$ ; (3) a cost function,  $C : \mathcal{A} \rightarrow \mathbb{R}$ ; (4) a message cost,  $\psi \in [0, 1)$ ; and (5) a set of Whittle index values per active action  $\alpha \in \{1, 2\}$ , denoted by  $W_1$  and  $W_2$ , respectively. Given these inputs, GRETA *returns* a reward-maximizing, constraint-satisfying action vector,  $\vec{a}_t$ .

---

**Algorithm 4** GRETA: graph-aware, Whittle-based heuristic  
 Note: all sorts are descending; arrays are zero-indexed.

---

```

1: procedure GRETA( $G', V', E', B, C, \psi, W_1, W_2$ )
2:    $\vec{a}_t \leftarrow 0^{|V|}$ 
3:    $B' \leftarrow B$ 
4:   while  $\bigvee_{e \in E'} \text{GETCOST}(u, v, \vec{a}_t, C) \leq B' \wedge E' \neq \emptyset$  do
5:      $b \leftarrow \min(B', 2)$ 
6:      $\hat{a}_2, \nu_2 \leftarrow \text{PULLONLY}(E', [b], W_2)$ 
7:      $\hat{a}_{(1,2)}, \nu_{(1,2)}, E'_\emptyset \leftarrow \text{MP}(G', b, C, \psi, \vec{a}_t, W_1, W_2)$ 
8:     if  $\nu_2 \geq \nu_{(1,2)}$  then
9:        $\vec{a}_t, B' \leftarrow \text{MODACTSB}(G', C, \hat{a}_2, \vec{a}_t, B')$ 
10:       $E', G' \leftarrow \text{UPDATEG}(V', E', \hat{a}_2, \emptyset)$ 
11:     else
12:        $\vec{a}_t, B' \leftarrow \text{MODACTSB}(G', C, \hat{a}_{(1,2)}, \vec{a}_t, B')$ 
13:       $E', G' \leftarrow \text{UPDATEG}(V', E', \hat{a}_{(1,2)}, E'_\emptyset)$ 
14:   return  $\vec{a}_t$ 

```

---

In lines 2-3 of Algorithm 4, we initialize  $\vec{a}_t$  such that each vertex is mapped to 0 (no-act),

and set our *remaining budget* variable,  $B'$ , equal to the per-timestep budget,  $B$ .

In lines 4-13, we iteratively update our action vector  $\vec{a}_t$  until we have insufficient remaining budget to afford any available edge-action pair, or our augmented edge set,  $E' = \emptyset$ . The termination check in line 4 requires us to: (1) check if we've already incurred the cost of a *pull* or *message* (*message*) for vertex  $u$  ( $v$ ); and (2) offset accordingly when we compute the cost of  $(a_t^u = 2, a_t^v = 1)$ , per Algorithm 5.

---

**Algorithm 5** Compute cost to pull  $u$  and message  $v$

---

```

1: procedure GETCOST( $u, v, \vec{a}_t, C$ )
2:    $c_u \leftarrow C(2)(1 - \mathbb{1}(a_t^u > 0)) + \mathbb{1}(a_t^u = 1)(C(2) - C(1))$ 
3:    $c_v \leftarrow C(1)(1 - \mathbb{1}(a_t^v = 1 \vee v = -1))$ 
4: return  $c_u + c_v$ 

```

---

The subroutines called in lines 6-7 of GRETA serve to ensure that we will only deviate from the pull-assignment choices of graph-agnostic THRESHOLD WHITTLE—i.e., by considering a combination of pulls *and* messages—when it is strictly beneficial to do so.

Since pulls have unit cost, and  $\psi \in [0, 1)$ , we consider our per-timestep budget in sequential chunks of 2. We have two options for allocating each chunk over actions: (1) considering *only* pulls, and selecting the two arms with highest  $W_2$  index values; or (2) considering messages *and* pulls, and selecting the set of directed  $(u, v)$  edges with highest edge-level subsidies such that each  $u$  receives a pull, and each  $v$  (excluding  $-1$ ) receives a message. In lines 8-13, we select the candidate action set with the highest cumulative subsidy, and update  $\vec{a}_t$ ,  $B'$ , and  $G'$  accordingly.

**Pulls only:** Allocation option (1) maps arms who have yet to receive a pull at time  $t$  to candidate actions  $\in \{0, 2\}$  by sorting their  $W_2$  index values in descending order and selecting the top-2 arms to receive pulls. Appendix B.1.1 gives pseudocode (see Algorithm 13).

**Messages and pulls:** Allocation option (2) maps arms to candidate actions by computing an

edge index value for each directed edge  $e \in E'$ . Algorithm 6 provides pseudocode.

---

**Algorithm 6** Cumulative subsidy of max pull-message set

Note: all sorts are descending; arrays are zero-indexed.

---

```

1: procedure MP( $G', b \in \mathbb{R}, C, \psi, \vec{a}_t, W_1, W_2$ )
2:    $G'' = (V'', E'') \leftarrow G'$ 
3:    $\hat{a}_{(1,2)} : v \in V'' \mapsto \vec{a}_t^v$ 
4:    $f : (u, v) \in E'' \mapsto \mathbb{R}$ 
5:    $E'_{\emptyset} \leftarrow \emptyset$ 
6:    $\nu_{(1,2)} \leftarrow 0$ 
7:   while  $\exists_{e \in E''} \text{GETCOST}(u, v, \hat{a}, C) \leq b \wedge E'' \neq \emptyset$  do
8:     for  $u \in V'' \setminus \{-1\}$  do
9:        $\mathcal{N}'_{\text{out}}(u) \leftarrow \{v \mid (u, v) \in E'' \wedge \hat{a}_{(1,2)}^v = 0\}$ 
10:      EDGEINDICES( $f'', u, \mathcal{N}'_{\text{out}}(u), b, \psi, W_1, W_2$ )
11:      values  $\leftarrow \text{SORT}(\{f((u, v)) \mid (u, v) \in E''\})$ 
12:      if |values| = 0 then
13:        break
14:      for  $f((u, v)) \in \text{values}$  do
15:         $\text{cost}_{u,v} \leftarrow \text{COMPUTECOST}(u, v, \hat{a}_{(1,2)}, C)$ 
16:        if  $\text{cost}_{u,v} \leq b$  then
17:           $h : u \mapsto 2; v \mapsto 1$ 
18:           $\hat{a}_{(1,2)}, b \leftarrow \text{MODACTSB}(G'', C, h, \hat{a}_{(1,2)}, b)$ 
19:           $E'', G'' \leftarrow \text{UPDATEG}(V'', E'', \hat{a}_{(1,2)}, \emptyset)$ 
20:           $\nu_{(1,2)} += f((u, v))$ 
21:           $E'_{\emptyset} \leftarrow E'_{\emptyset} \cup \{(u, v)\}$ 
22:          break
23:       $\triangleright$  Return best arm-actions, cumulative subsidy,  $E'_{\emptyset}$ 
23:   return  $\hat{a}_{(1,2)}, \nu_{(1,2)}, E'_{\emptyset}$ 

```

---

In line 2 of Algorithm 6, we start by defining  $G''$  to be a local copy of our augmented graph,  $G'$ . We then create a function,  $\vec{a}_{(1,2)}$  to map each vertex  $v \in V''$  to its candidate action, which we initialize to be  $\vec{a}_t^v$  (line 3). We do this because we require the current  $G''$  to determine which ( $\text{pull}_{u, \text{message}_v}$ ) edge-action combinations are possible, and for  $\vec{a}_t$  to correctly compute the cost of these hypothetical actions, but we don't want to modify  $\vec{a}_t$  or  $G'$  in-place. Next, in lines 4-5, we define a function,  $f$  that maps each edge  $(u, v) \in E'$  to a real-valued edge index value, and a set,  $E'_{\emptyset}$ , to hold the edges we will need to remove from  $G'$  if we select the candidate actions returned by Algorithm 6. In line 6, we initialize  $\nu_{(1,2)} = 0$  to represent the cumulative subsidy of our candidate action set.



In lines 7-22 of Algorithm 6, we iteratively update our candidate action function,  $\hat{a}_{(1,2)}$ , until we run out of (small- $b$ ) budget, or  $E'' = \emptyset$ . Inside each iteration of the WHILE-loop, we begin by computing an edge index value for each directed edge  $(u, v) \in E'$  (lines 8-10). To do this, we loop over vertices in  $V' \setminus \{-1\}$  (line 8), and for each vertex  $u$ , let  $\mathcal{N}'_{\text{out}}(u) \subseteq \mathcal{N}_{\text{out}}(u)$  represent the subset of  $u$ 's one-hop out-degree neighbors currently slated to receive a `no-act` at time  $t$ .

For each edge  $(u, v) \in \mathcal{N}'_{\text{out}}(u)$ , our edge index value represents the cumulative subsidy required to forgo a *pull* for arm  $u$  (i.e.,  $W_2^u$ ) and a *message* for arm  $v$  (i.e.,  $W_1^v$ ). Note: if we pull  $u$ , message  $v$ , and have budget left over, we can message up to  $|\mathcal{M}_t^u|$  vertices  $v' \in \mathcal{N}'_{\text{out}}(u)$  at time  $t$  *without* incurring additional pull costs, where  $|\mathcal{M}_t^u| = |\mathcal{N}'_{\text{out}}(u)|$  if  $\psi = 0$  and  $\min(\lfloor b/\psi \rfloor, |\mathcal{N}'_{\text{out}}(u)|)$  for  $\psi \in (0, 1)$ . To exploit this diminishing marginal cost, we sort  $u$ 's, neighbors by their index-values and let the max-valued edge represent the cumulative, cost-feasible value of  $\mathcal{N}'_{\text{out}}(u)$ , rather than just  $(u, v)$ . Algorithm 7 provides pseudocode.

---

**Algorithm 7** Compute edge index values

Note: all sorts are descending; arrays are zero-indexed.

---

```

1: procedure EDGEINDICES( $f, u, \mathcal{N}'_{\text{out}}(u), b, \psi, W_2, W_1$ )
2:    $n\_msgs \leftarrow |\mathcal{N}'_{\text{out}}(u)|$  if  $\psi = 0$  else  $\min(\lfloor b/\psi \rfloor, |\mathcal{N}'_{\text{out}}(u)|)$ 
3:    $msg\_values \leftarrow \text{SORT}(g : v \in \mathcal{N}'_{\text{out}}(u) \mapsto W_1^v)$ 
4:    $max\_edge \leftarrow (u, \arg \max_v msg\_values)$ 
5:   for  $v \in \mathcal{N}'_{\text{out}}(u)$  do
6:     if  $(u, v) = max\_edge$  then
7:        $f((u, v)) \leftarrow W_2^u + \sum_{i=0}^{n\_msgs-1} msg\_values_i$ 
8:     else
9:        $f((u, v)) \leftarrow W_2^u + W_1^v$ 
10:  return

```

$\triangleright f$  is updated in-place

---

Then, in lines 11-13 of Algorithm 6, we sort the edge-index values in descending order. Note that we break if there are no values to be sorted; this corresponds to the scenario in which no additional pulls are available/cost-feasible, and every arm not receiving a pull is already receiving a message, but we still have budget left—i.e., when  $\psi = 0$ . In lines 14-22, we choose the top

cost-feasible edge-action pair from our sorted list, and update our candidate action function,  $\vec{a}_{(1,2)}$  and local budget,  $b$  accordingly. Note that if  $\psi = 0$  and arm  $u$  receives a pull, we message every  $v \in \mathcal{N}'_{\text{out}}(u)$ . Appendix B.1.1.3 provides pseudocode for the MODACTSB subroutine (see Algorithm 14).

Finally, we update our local copy of the augmented graph by removing  $(u, v)$ , as well as any directed edge terminating in  $u$ , and the placeholder edge,  $(u, -1)$ . This is because: (a) we do not want to reconsider the edge-action pair we've selected; and (b) by virtue of how we select  $(u, v)$ ,  $f((u, v)) \geq f((\cdot, u))$  or any such  $u$ -terminating edge is cost-prohibitive. Appendix B.1.1 provides pseudocode for the UPDATEG subroutine (see Algorithm 15). We conclude the MP subroutine (Algorithm 6) by returning our candidate action function,  $\hat{a}_{(1,2)}$ , the associated cumulative subsidy value,  $\nu_{(1,2)}$ , and the set of candidate edges to be removed from  $G', E'_{\emptyset}$ .

**Putting the pieces together:** With the exposition of each of GRETA's subroutines complete, we now return to lines 8-13 of Algorithm 4. We compare the cumulative subsidy values returned by the PULLONLY and MP subroutines, and use the candidate action function associated with the maximum cumulative subsidy to update our action vector,  $\vec{a}_t$ , remaining budget,  $B'$ , and augmented graph,  $G'$ . When the WHILE-loop terminates, we return  $\vec{a}_t$ . By virtue of how this action vector is constructed, it is reward-maximizing and guaranteed to satisfy the budget constraint.

### 2.3.3.2 Theoretical Analysis

**Bounding expected reward:** Per Theorem 2.3.1, the expected cumulative reward of GRETA with message cost,  $\psi > 0$ , will be lower-bounded by that of graph-agnostic THRESHOLD WHITTLE, and upper-bounded by GRETA with  $\psi = 0$ . See Appendix B.1.2 for a complete proof.

**Theorem 2.3.1.** For a given set of  $[n]$  restless or collapsing arms with transition matrices satisfying the structural constraints outlined in Section 2.3.2.3, corresponding directed graph,  $G = (V, E)$ , budget  $B \in \mathbb{R}_{\geq 0}$ , non-decreasing local reward function,  $r : \mathcal{S} \rightarrow \mathbb{R}$ , cumulative reward function,  $R$ , and cost vector  $\vec{c} = [0, \psi, 1]$  such that  $\psi \in [0, 1)$ , we have:  $\mathbb{E}_{\text{TW}}[R] \leq \mathbb{E}_{\text{GH}, \psi > 0}[R] \leq \mathbb{E}_{\text{GH}, \psi = 0}[R]$

*Proof Sketch.* The first inequality follows from how GRETA constructs each  $\vec{a}_t$ . The second inequality follows from the fact that: (a) per our structural constraints and choice of  $r$ ,  $E[r_t^i | s_t^i, a_t^i]$  is strictly increasing with  $a_t^i \forall i, t$ ; and (b) for  $\psi = 0$ , we can message at least as many arms as when  $\psi > 0$ .  $\square$

**Computational complexity:** Per Theorem 2.3.2, GRETA is efficiently computable in time polynomial in its inputs; see Appendix B.1.2 for a complete proof.

**Theorem 2.3.2.** For convenience, let:  $\xi = \mathbb{1}(\psi > 0) \times \min(|E'|^2, \lfloor \frac{B}{\psi} \rfloor |E'|) + \mathbb{1}(\psi = 0) \times |V'| |E'|$ . Then, for  $\psi \in [0, 1)$  and time horizon,  $T$ , the time complexity of GRETA is:

$$\begin{cases} O(\max(\xi^2 |V'|^2 \log |V'|, \xi^2 |V'| |E'|^2) T), & \text{if } \psi > 0 \\ O(\max(\xi^2 |V'|^2 \log |V'|, \xi^2 |V'| |E'|^2, \xi^2 |V'|^2 |E'|) T), \\ \text{otherwise} \end{cases}$$

These bounds indicate that GRETA is well-suited for sparse graphs and values of  $\psi = 0$  or  $\psi \rightarrow 0.5$  ( $\psi > 0.5$  will also improve runtime, but may reduce opportunities to exploit externalities). Conversely, pathological cases will include large-scale dense graphs and values of the message

cost,  $\psi$ , which approach but do not equal 0. We consider improving scale to be a valuable direction for future work. The combinatorial nature of the problem we consider suggests that sampling and/or distributed methods will be critical in this regard [67, 68].

### 2.3.4 Experimental Evaluation

In this section, we demonstrate that GRETA consistently outperforms a set of robust graph-agnostic and graph-aware comparison policies. We begin by identifying the set of policies we compare against, as well as our evaluation metrics, graph generation, and mapping of arms to vertices. We proceed to present results from three experiments: (1) GRETA versus the optimal policy (for small  $n$ ); (2) GRETA versus comparison policies for a fixed cohort and graph; and (3) GRETA evaluated on a series of different budgets, message costs, and graph topologies.

#### 2.3.4.1 Experimental Setup

**Policies:** In our experiments, we compare the policy produced by GRETA against a subset of the following graph- $\{\text{agnostic}^\circledast$  and  $\text{aware}^\dagger\}$  policies:

THRESHOLD WHITTLE (TW) $^\circledast$	Compute Whittle index values using pull as (only) active action. Pull $\lfloor B \rfloor$ arms with highest Whittle index values; all others get no-act [2, 20].
RANDOM $^\dagger$	Construct $G'$ ; select budget-feasible edge-action pairs uniformly at random until budget exhausted.
CENTRALITY-WEIGHTED RANDOM $^\dagger$	Construct $G'$ ; select budget-feasible edge-action pairs weighted by out-degree centrality of <code>src</code> vertex until budget exhausted.
MYOPIC $^\dagger$	Construct $G'$ ; sort edge-action pairs by expected reward at $t + 1$ . Select cost-feasible pairs until budget exhausted.
VALUE ITERATION (VI) $^\dagger$	Find the optimal policy via value iteration for <i>system-level</i> MDP (intractable at scale, but computable for small $ V $ and $ E $ ).

We note that in the restless (but graph-agnostic) setting: (1) RANDOM and MYOPIC are

common baselines. Here, we have extended them to the networked setting. (2) THRESHOLD WHITTLE represents a state-of-the-art approach. To the best of our knowledge, no additional (efficiently computable) graph-aware policies exist for the novel networked restless bandit setting we propose.

**Objective:** Our optimization task is consistent with assigning equal value to each timestep that any arm spends in the “desirable” state. This motivates our choice of a local reward function  $r_t(s_t^i) := s_t^i \in \{0, 1\}$  and undiscounted cumulative reward function  $R(r(s)) := \sum_{i \in [N]} \sum_{t \in [T]} r(s_t^i)$ .

*Intervention benefit (IB):* For each policy, we compute total expected reward,  $\mathbb{E}_\pi[R(\cdot)]$ , by taking the average over 50 simulation seeds. We then compute the intervention benefit as defined in Equation 2.12, where NOACT represents a policy in which no pulls or messages are executed, and GH represents the policy produced by GRETA.

$$\mathbf{IB}_{\text{NoAct, GH}}(\pi) := \frac{\mathbb{E}_\pi[R_\pi(\cdot)] - \mathbb{E}_{\text{NoAct}}[R(\cdot)]}{\mathbb{E}_{\text{GH}}[R(\cdot)] - \mathbb{E}_{\text{NoAct}}[R(\cdot)]} \quad (2.12)$$

**Graph generation:** For each cohort of  $n$  restless arms that we consider in our experiments, we use a stochastic block model (SBM) to generate a graph with  $|V| = n$  vertices [69]. This generator partitions the vertices into blocks and stochastically inserts directed edges, with hyperparameter  $p_{in}(p_{out}) \in [0, 1]$  controlling the probability that a directed edge will exist between two vertices in the same (different) block(s).

We consider two options for  $\varphi : [n] \rightarrow V$ : (1) *random*; and (2) *by cluster*. For mapping (1), we generate  $\lceil \frac{n}{10} \rceil$  blocks of uniform size, and map arms to vertices—and, by extension, to blocks—uniformly at random. This mapping represents allocation settings with a peer support component where participants are randomly assigned to groups, without regard for their behavioral

similarity.

For mapping (2), we use an off-the-shelf K-MEANS algorithm to cluster the arms in flattened transition-matrix vector space [70]. We use the cardinality of the resulting clusters to determine the size of each block, and map arms to vertices based on cluster membership. This mapping represents intervention allocation settings with a peer support component where participants with similar transition dynamics are grouped together.

### 2.3.4.2 GRETA vs. the Optimal Policy

In this experiment, we compare GRETA to  $\pi_{VI}^*$ , where  $\pi_{VI}^*$  denotes the optimal policy obtained via value iteration for the *system-level* MDP [71]. This system-level MDP has state space  $\mathcal{S}' := \{\mathcal{S}\}^n$ , action space  $\mathcal{A}' := \{\mathcal{A}\}^n$ , a transition function,  $P : \mathcal{S}' \times \mathcal{A}' \rightarrow \mathcal{S}'$ , and reward function,  $R' = \sum_{i \in [n]} s^i$ . To ensure budget and neighborhood constraint satisfaction, only cost- and topologically feasible actions,  $\mathcal{A}'' \subseteq \mathcal{A}'$  are considered.

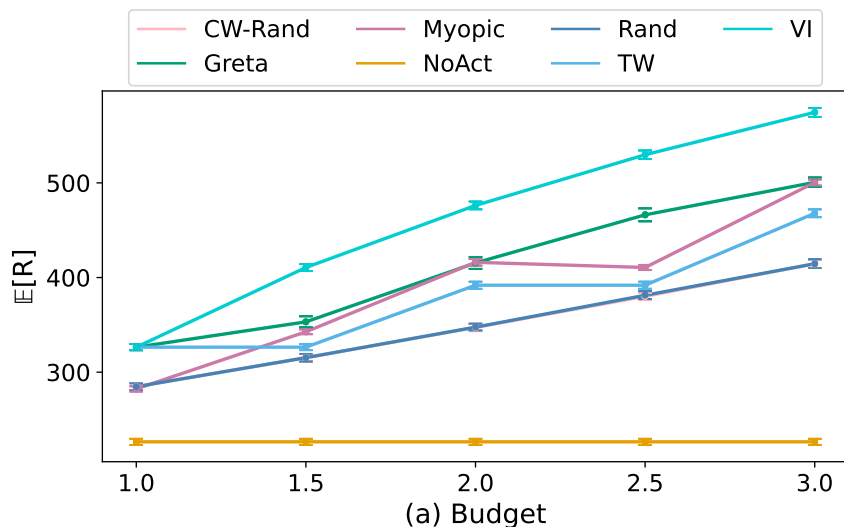


Figure 2.4:  $\mathbb{E}[R]$  by policy and budget

Figure 2.4 reports results for a synthetic cohort of 8 arms embedded in a fully connected

graph (i.e.,  $p_{\text{in}} = p_{\text{out}} = 1.0$ ). We let  $T = 120$ ,  $\psi = 0.5$ , and report unnormalized  $\mathbb{E}_{\pi}[R]$ , along with margins of error for 95% confidence intervals computed over 50 simulation seeds for values of  $B \in \{1, 1.5, 2, 2.5, 3\}$ . GRETA outperforms TW for each value of  $B$  (with predictably larger gaps for values with remainders  $= \psi$  that graph-agnostic TW cannot exploit), and is competitive with respect to  $\pi_{\text{VI}}^*$ .

### 2.3.4.3 GRETA vs. Alternative Policies

Here we compare GRETA to the graph-agnostic and graph-aware comparison policies outlined in Section 2.3.4.1. We consider a synthetic cohort of  $n = 100$  restless arms whose transition matrices are randomly generated in such a way so as to satisfy the structural constraints introduced in Section 2.3.2. We use a stochastic block model (SBM) generator with  $p_{\text{in}} = 0.2$  and  $p_{\text{out}} = 0.05$ , and consider both the *random* and *by cluster* options for  $\varphi$ . We let  $T = 120$ ,  $B = 10$ , and  $\psi = 0.5$ . In Table 2.2, we report results for each mapping-policy combination, along with margins of error for 95% confidence intervals computed over 50 simulation seeds.

Key findings from this experiment include:

- The policy produced by GRETA achieves significantly higher  $\mathbb{E}_{\pi}[\text{IB}]$  than each comparison.
- The gap in  $\mathbb{E}_{\pi}[\text{IB}]$  between GRETA and MYOPIC (the best-performing alternative) is larger for the *by cluster* mapping than the *random* mapping. This suggests that in assortative networks, relatively homogeneous transition dynamics within blocks facilitate exploitation of

$\varphi(i)$	Policy	$\mathbb{E}[\text{IB}]$ (%) ( $\pm$ )
randomly	RANDOM	75.82 $\pm$ 0.890
	CWRANDOM	74.79 $\pm$ 1.068
	MYOPIC	87.83 $\pm$ 1.115
	TW	83.57 $\pm$ 0.779
	GRETA	<b>100.00 <math>\pm</math> 0.000</b>
by cluster	RANDOM	64.19 $\pm$ 0.786
	CWRANDOM	63.59 $\pm$ 0.804
	MYOPIC	76.24 $\pm$ 0.921
	TW	72.65 $\pm$ 0.684
	GRETA	<b>100.00 <math>\pm</math> 0.000</b>

Table 2.2:  $\mathbb{E}[\text{IB}]$ , by  $\varphi$  and policy

the diminishing marginal costs associated with the pull-message dynamic.

#### 2.3.4.4 Sensitivity Analysis

We conduct sensitivity analysis with respect to: (1) the budget,  $B$ ; (2) the message cost,  $\psi$ ; and (3) the underlying graph topology, via the  $p_{\text{in}}$  and  $p_{\text{out}}$  hyperparameters of our stochastic block model graph generator. As we vary each of the aforementioned hyperparameters, we consider a fixed cohort size of  $n = 100$  randomly-generated, structural constraint-satisfying arms, a time horizon,  $T = 120$ , and a mapping  $\varphi : i \in [n] \mapsto v \in V$  from arms to vertices that is determined by cluster. We report unnormalized  $\mathbb{E}_{\pi}[R]$ , along with margins of error for 95% confidence intervals computed over 50 simulation seeds, for GRETA, THRESHOLD WHITTLE, NOACT, and MYOPIC, which is the best-performing non-TW alternative. We describe each task below, and present results in Figure 2.5.

**Budget:** We hold message cost fixed at  $\psi = 0.5$ , let  $p_{\text{in}} = 0.25$ ,  $p_{\text{out}} = 0.05$ , and consider values of  $B \in \{5\%, 10\%, 15\%\}$  of  $n$ . As Figure 2.5(a) illustrates,  $\mathbb{E}_{\pi}[R]$  intuitively rises with  $B$  for each policy considered. For each value of  $B$ , we find that GRETA achieves higher  $\mathbb{E}_{\pi}[R]$  than the comparison policies, and that the gap between GRETA and the best-performing alternative also increases with  $B$ .

**Message cost:** Here, we hold the budget fixed at 6, let  $p_{\text{in}} = 0.25$ ,  $p_{\text{out}} = 0.05$ , and consider values of  $\psi \in \{0.0, 0.25, 0.5, 0.75, 0.9\}$ . As Figure 2.5(b) illustrates,  $\mathbb{E}_{\pi}[R]$  decreases as the message cost,  $\psi$ , increases for GRETA and MYOPIC, while it remains constant for active-action agnostic NOACT and message-agnostic TW. For each value of  $\psi$  that we consider, GRETA achieves higher  $\mathbb{E}_{\pi}[R]$  than each of the comparison policies. This gap is intuitively largest when



$\psi = 0$ , and decreases until GRETA converges with TW—notably, without suffering loss in total expected reward due to divisibility issues with respect to  $B$ , when  $\psi = 0.75$ .

**Graph topology:** We hold the budget fixed at  $B = 10$ , let message cost,  $\psi = 0.5$ , and consider two sets of increasingly *assortative* (*disassortative*)  $(p_{in}, p_{out})$  ordered pairs. In each case, we start with  $E = \emptyset$ —i.e.,  $(0.0, 0.0)$ , and then hold  $p_{out}$  ( $p_{in}$ ) fixed at 0.1 and steadily increase  $p_{in}$  ( $p_{out}$ ). Figure 2.5(c) and (d) present results. For GRETA, while  $\mathbb{E}_{\pi}[R]$  is generally increasing in the number of edges, the rate of growth levels off as assortativity rises, but remains

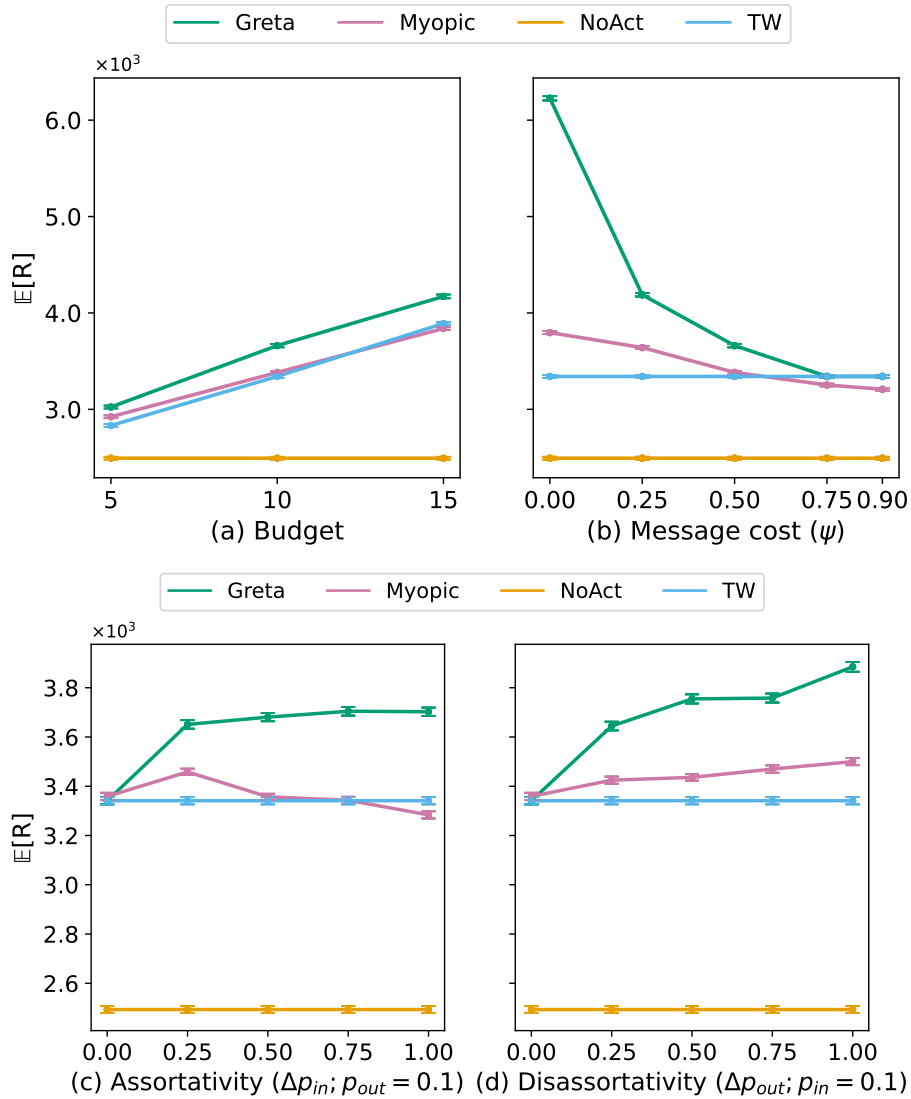


Figure 2.5: Sensitivity results, by varied hyperparameter

robust as disassortativity rises. This suggests that homophilic clustering of arms with respect to transition dynamics may undermine total welfare by inducing competition within neighborhoods, while heterophilic clustering can help to smooth out subgroups' relative demand for constrained resources over time.

### 2.3.5 Summary of Contributions

In Chapter 2.3, we introduce networked restless bandits, a novel multi-armed bandit setting in which arms are restless and embedded in a directed graph. We show that this framework can be used to model constrained resource allocation in community settings, where receipt of the resource by an individual can result in spillover effects that benefit their neighbor(s). We also present GRETA, a graph-aware, Whittle-based heuristic algorithm which is constrained reward-maximizing and budget-constraint satisfying in our networked restless bandit setting. Our empirical results demonstrate that the policy produced by GRETA outperforms a set of graph-agnostic and graph-aware comparison policies for a range of different budgets, message costs, and graph topologies. Future directions include: (1) relaxing the assumption of perfect observability of transition matrices and/or graph topology; (2) considering individual and/or group fairness; and (3) incorporating sampling and/or distributed methods to improve scalability.

## Chapter 3: Model Evaluation for Harms Identification and Mitigation

### 3.1 Evaluating Model Performance Across Intersectional Subgroups

This work was done in collaboration with my co-first author, Kimberly Truong, as well as Alexandra Chouldechova and Miroslav Dudík; see [13].

#### 3.1.1 Introduction

A core task when assessing the fairness of an AI system is measuring its performance across different subgroups defined by combinations of demographic or other sensitive attributes. Many of the best-known studies of algorithmic bias are grounded in this type of analysis. This includes the [Buolamwini and Gebru](#)'s Gender Shades study [72], which found that commercial gender classifiers have much higher error rates for darker-skinned women than other groups, and the [Obermeyer et al.](#)'s study [73] finding bias in commercial algorithms used to guide health-care decisions, as well as many others [74, 75, 76, 77].

In their work formalizing this type of analysis, Barocas et al. [6] introduce the term *disaggregated evaluation* to refer to this task. The authors draw attention to the many decisions that often implicitly go into shaping any given disaggregated evaluation: from who will be involved, to what data will be used, to the statistical approach taken, to drawing inferences from the data. In

our work, we focus on the question of statistical methodology given an available dataset and pre-determined subgroups and performance metrics of interest. Specifically, we introduce a method for estimating performance across subgroups that we show (i) is more accurate than approaches taken in standard practice; and (ii) can provide greater insight into which factors drive observed variation in performance.

The “standard approach” to disaggregated evaluation proceeds by stratifying the evaluation data across subgroups and then conducting inference (i.e., computing performance metrics, confidence intervals, or other statistics) separately for each group. The primary challenge when applying this approach are small sample sizes. Even for moderately-sized evaluation datasets, sample sizes quickly get small once considering intersectional subgroups. For instance, in a medical diabetes mellitus dataset we use later, we have a 5000-patient evaluation dataset, of which 2689 patients are female, 620 are female *and* over age 80, but only 6 are female, over age 80, *and* Hispanic. Indeed, of the 32 distinct gender-age-race/ethnicity subgroups that can be formed in the data, 8 (i.e., 25%) have fewer than 10 observations, and nearly half have fewer than 25 observations. Inference based on so few observations is often uninformative, and may be unreliable. In practice, subgroups that are too small tend to be either excluded from analysis or merged with other small but potentially heterogeneous subgroups to form higher-level “catch-all” categories (e.g., “other”). These practices greatly limit the extent to which intersectional groups are even considered in many disaggregated evaluations. As a consequence, standard assessments may fail to surface fairness-related harms that could disproportionately affect intersectional subgroups [78], which in turn means that steps to mitigate those harms will not be taken.

In this work, we introduce a structured regression approach to disaggregated evaluation that we demonstrate can yield reliable system performance estimates even for very small subgroups

(e.g., for groups with fewer than 25 observations). We also provide corresponding inference strategies for constructing confidence intervals for the subgroup-level performance estimates. We then demonstrate how goodness-of-fit testing can provide insight into the structure of fairness-related harms experienced by intersectional groups and also identify situations where observed variation in performance is attributable to benign factors. Lastly, we present results on two publicly available datasets, and several variants of semi-synthetic data. The results show that our method is considerably more accurate than the standard approach, especially for small subgroups. They further show that our method outperforms more statistically sophisticated baselines, including the *model-based metrics* method introduced by Miller et al. [79], while also offering additional advantages. We conclude by discussing limitations and future directions.

### 3.1.2 Background and Related Work

In their taxonomy of sociotechnical harms of algorithmic systems, Shelby et al. [80] identify five high-level categories of harm: representational, allocative, quality of service, interpersonal, and social system. Our work contributes to the broader literature characterizing and assessing *allocative* and *quality-of-service* harms that can result from the use of algorithmic systems. Allocative harms, first discussed by Barocas et al. [81], occur when systems produce an inequitable distribution of information, opportunity, or resources across groups. As a running example, we consider a hypothetical setting in which a model trained to predict 30-day hospital readmission is used to prioritize high-risk patients for more intensive post-discharge care. Allocative harms might occur in this setting if certain subgroups of patients are disproportionately under-prioritized for more intensive care (i.e., have low selection rates) or are under-selected relative to their observed rate of

readmission (i.e., have high false negative rates).

Quality-of-service harms occur when algorithmic systems underperform for certain socially salient groups of users [80, 82]. We examine quality-of-service harms across race and gender groups in the context of commercial automated speech recognition (ASR) systems using data previously analysed by Koenecke et al. [74]. Specifically, we assess whether there is significant variation in the word error rate (WER) of the ASR systems across intersectional race and gender subgroups.

The term “intersectionality” was introduced by Crenshaw [78] to describe the distinct patterns of discrimination and disadvantage experienced by Black women, which she argued cannot be understood in terms of race or gender discrimination alone. In recent years, algorithmic fairness research has examined intersectional bias from many perspectives. This includes work introducing quantitative metrics intended to capture notions of intersectional fairness, such as subgroup fairness [83], differential fairness [84, 85], and multi-calibration [86], along with learning algorithms for estimating and achieving these criteria. Wang et al. [87] study “predictivity differences” across intersectional subgroups, and discuss limitations of existing summary statistics (such as the maximum disparity across all groups) in capturing meaningful notions of intersectional harm. Our work differs from this literature because we are specifically interested in the task of disaggregated evaluation. This entails estimating and reporting system performance for each intersectional subgroup, rather than computing a particular fairness metric or learning a fairness-constrained model.

Our work most directly contributes to the growing literature introducing more sample-efficient methods for conducting disaggregated evaluations. This literature includes methods that leverage unlabelled data in model evaluation [88, 89, 90]; methods that bound or approximate performance for intersectional subgroups using marginal statistics [91]; and synthetic data aug-

mentation approaches [92]. In work more closely related to the spirit of our structured regression approach, Piratla et al. [93] introduce the Attributed Accuracy Assay (AAA) method, which models the accuracy of a model as a function of sensitive attributes and other features via a Gaussian Process (GP). While we do not rely on GPs, we do proceed similarly by modeling the accuracy (or error) of a given model. Whereas we are specifically concerned with fairness and disaggregated evaluation, Piratla et al. [93] aim to produce an “accuracy surface” model that clients can use to estimate the performance of an existing model on their data.

The most closely related work in recent literature is that of Miller et al. [79], who introduce a Bayesian structured regression approach that they call *model-based metrics* (MBM). Their method applies to AI models that produce a score (say to predict a risk of hospital readmission). By modeling the distribution of scores given select features and the observed outcome, they are able to make inference on any performance metric of interest, but the approach is not directly applicable to the evaluation of models that do not produce classification scores (e.g., MBM does not directly apply to the evaluation of WER in ASR systems). Unlike the MBM approach, we model the target metric directly and fit separate models for each performance metric of interest. Our experiments show that our method yields more accurate estimates than MBM (see Section 3.1.7.1).

Our approach is also related to the classical line of research on normal means estimation, originating with the James-Stein (JS) estimator [94, 95]. The JS estimator works by shrinking standard estimates towards zero (or some other constant), which leads to a substantial decrease in variance, while only a moderate increase in bias. This favorable bias–variance trade-off in turn leads to a more accurate estimator. Our estimator also works by optimizing bias–variance trade-off, but it enjoys additional advantages: availability of confidence interval procedures and flexibility to incorporate information in the form of covariates. In our experiments, we match and

sometimes outperform the JS estimator (see Section 3.1.7.1).

Many important challenges lie outside the scope of this work. We focus on improving the accuracy of disaggregated evaluation, especially on small groups, assuming that relevant sensitive attributes and performance metrics have been determined and a suitable evaluation dataset collected. However, many key sociotechnical challenges arise during the evaluation conception and dataset construction phases [6, 96]. For instance, as Barocas et al. [6] discuss, the sensitive attributes often include socially constructed—and potentially contested—features (like race and gender), which makes the task of mapping people to attributes and corresponding subgroups potentially fraught, particularly when it involves inference or use of proxy variables, or poses a risk for members of already-marginalized subgroups. Another set of challenges arises when deciding on a performance metric. In many high-stakes applications (like education and health-care), we are not able to directly measure who might benefit, so we need to rely on proxies. This step is critical, since a poor choice of a proxy may further exacerbate existing inequities, as is the case, for instance, when predicting risk of re-offense from arrest records [97] or predicting health-care needs based on health-care expenditures [73].

### 3.1.3 Problem setting

We would like to assess the fairness-related harms of an AI system by evaluating its performance on intersectional subgroups of users specified by  $k \geq 2$  sensitive attributes (like race and gender), taking values in finite sets  $\mathcal{A}_1, \dots, \mathcal{A}_k$ . The set of all possible  $k$ -tuples of sensitive-attribute values is denoted  $\mathcal{A} = \mathcal{A}_1 \times \dots \times \mathcal{A}_k$ .

We assume that we have access to an evaluation dataset  $S$ , consisting of individuals described



by tuples of the form  $(X, A, Y, \hat{Y})$  sampled i.i.d. from some underlying distribution  $\mathcal{D}$ , where  $X$  contains application-relevant information about the individual (e.g., the health history of a patient),  $A \in \mathcal{A}$  is a  $k$ -tuple of sensitive attributes,  $Y$  is an observed outcome variable (e.g., whether the patient was readmitted within 30 days of discharge), and  $\hat{Y}$  is an output produced by the AI system (e.g., a score used for prioritizing patients into post-discharge care).

For any  $k$ -tuple  $a \in \mathcal{A}$ , we write  $a[1], \dots, a[k]$  to denote its components. In the ASR example below, we consider two sensitive attributes, *race* and *gender*, with domains  $\mathcal{A}_1 = \{\text{Black, white}\}$  and  $\mathcal{A}_2 = \{\text{male, female}\}$ . In that case, for example, if  $a = (\text{Black, female})$ , then  $a[1] = \text{Black}$  and  $a[2] = \text{female}$ . When possible, we use mnemonic indices for components of  $a$  and write  $a[\textit{race}]$  and  $a[\textit{gender}]$  to mean  $a[1]$  and  $a[2]$ , and similarly  $\mathcal{A}_{\textit{race}}$  and  $\mathcal{A}_{\textit{gender}}$  to mean  $\mathcal{A}_1$  and  $\mathcal{A}_2$ .

For each  $a \in \mathcal{A}$ , we define  $\mathcal{D}_a$  to be the distribution of individuals with  $A = a$ , so  $\mathcal{D}_a$  is the conditional distribution  $\mathcal{D}(X, A, Y, \hat{Y} \mid A = a)$ , representing an intersectional group. Let  $\Delta$  denote the set of all probability distributions over tuples  $(X, A, Y, \hat{Y})$ , so  $\mathcal{D} \in \Delta$  and also  $\mathcal{D}_a \in \Delta$  for all  $a \in \mathcal{A}$ . A *performance metric* is a function  $m : \Delta \rightarrow \mathbb{R}$  that maps a probability distribution over tuples  $(X, A, Y, \hat{Y})$  into a real number. For example, if the underlying AI system performs binary classification, so  $Y, \hat{Y} \in \{0, 1\}$ , we could measure its performance using *accuracy*, defined, for any  $p \in \Delta$ , as

$$\text{ACC}(p) = \mathbb{P}_p[Y = \hat{Y}],$$

where  $\mathbb{P}_p[\cdot]$  is the probability of an event with respect to  $p$ . The overall system performance is then quantified by  $\text{ACC}(\mathcal{D})$  and the performance on the group  $a \in \mathcal{A}$  by  $\text{ACC}(\mathcal{D}_a)$ .

Given a performance metric  $m$ , the goal of *disaggregated evaluation* is to estimate the values

$m(\mathcal{D}_a)$  for all  $a \in \mathcal{A}$ . We denote these values as

$$\mu_a = m(\mathcal{D}_a).$$

Our only source of information about  $\mathcal{D}$  is the evaluation dataset  $S$  of size  $n = |S|$ , sampled i.i.d. from  $\mathcal{D}$ . The standard approach to disaggregated evaluation splits the dataset  $S$  into groups

$$S_a = \{(X, A, Y, \hat{Y}) \in S : A = a\}$$

of size  $n_a = |S_a|$ , and then evaluates  $m$  on each  $S_a$  (or, more precisely, on the probability distribution that puts an equal probability mass on each data point in  $S_a$ ). We denote the resulting *standard estimates* as

$$Z_a = m(S_a). \tag{3.1}$$

For example, if  $m$  is accuracy, then

$$Z_a = \text{ACC}(S_a) = \frac{1}{n_a} \sum_{(X,A,Y,\hat{Y}) \in S_a} \mathbf{1}\{Y = \hat{Y}\},$$

where  $\mathbf{1}\{\cdot\}$  is an indicator equal to 1 if its argument is true and 0 if it is false.

We next connect this abstract framework to two concrete scenarios already mentioned in [Section 3.1.2](#).

*Example 1* (Diabetes). We consider an AI system that refers high-risk patients into a post-discharge care program. We wish to assess the allocative harms of this system. To explore this scenario, we use a publicly available dataset of diabetes patients developed by Strack et al. [98]. The dataset

contains information about patient hospital visits, including whether each patient was readmitted within 30 days after discharge. We use the readmission as a proxy for whether the patient should be recommended for the care program.

Each data point corresponds to a patient admission, where  $X$  describes the patient history and hospital tests;  $A$  describes the patient’s *race*, *gender*, and (binned) *age*;  $Y \in \{0, 1\}$  indicates whether the patient was readmitted within 30 days after discharge; and  $\hat{Y} \in [0, 1]$  is the score produced by the AI system that has been trained to predict  $Y$ . We assume that the hospital uses a threshold  $r$ , and patients with  $\hat{Y} \geq r$  are automatically referred into the care program.

One type of allocative harm occurs when a subgroup of patients is disproportionately under-prioritized, i.e., if a subgroup has a low *selection rate*, denoted as

$$\text{SEL}(\mathcal{D}_a) = \mathbb{P}_{\mathcal{D}_a}[\hat{Y} \geq r].$$

We also consider a second type of harm, which occurs when a subgroup of patients experiences a disproportionately large rate of false negatives (i.e., many of those patients that should be recommended are not), measured by the *false negative rate*

$$\text{FNR}(\mathcal{D}_a) = \mathbb{P}_{\mathcal{D}_a}[\hat{Y} < r \mid Y = 1].$$

*Example 2 (ASR).* To assess quality-of-service harms of an ASR system, we use a dataset from Koenecke et al. [74], consisting of audio snippets (of length between 5s and 50s) spoken by various speakers. In the dataset,  $X$  describes properties of the snippet (like duration in seconds),  $A$  has two components corresponding to the speaker’s *race* and *gender*,  $Y$  is the ground-truth transcription of

the snippet, and  $\hat{Y}$  is the transcription provided by the AI system.

The quality-of-service harms occur when the system underperforms for a subgroup of users.

The performance is evaluated by the *word error rate*

$$\text{WER}(\mathcal{D}_a) = \mathbb{E}_{\mathcal{D}_a}[\text{wer}(\hat{Y}, Y)],$$

where *wer* is a snippet-level word error rate defined as

$$\text{wer}(\hat{Y}, Y) = \frac{\text{subst} + \text{del} + \text{ins}}{|Y|},$$

where *subst*, *del*, and *ins* is the number of word substitutions, deletions, and insertions in  $\hat{Y}$  compared with the ground truth  $Y$ , and  $|Y|$  is the number of words in  $Y$ .

To quantify the accuracy of an estimator, like the standard estimator introduced above, we often use *mean squared error* (MSE). We will use a modified definition of MSE that accounts for the fact that estimates like  $Z_a = m(S_a)$  are sometimes undefined, for instance, when the metric  $m$  is defined as a conditional probability, like FNR in Example 1, and the set  $S_a$  has no samples that satisfy the condition (e.g., no samples with  $Y = 1$  in case of FNR). For an estimator  $\hat{\mu}$  of a quantity  $\mu$ , let  $\mathcal{E}$  denote the event that  $\hat{\mu}$  is defined. The *bias*, *variance*, and *mean squared error* (MSE) of  $\hat{\mu}$  are defined as

$$\text{Bias}(\hat{\mu}) = \mathbb{E}[\hat{\mu} \mid \mathcal{E}] - \mu, \quad \text{Var}(\hat{\mu}) = \mathbb{E}[(\hat{\mu} - \mathbb{E}[\hat{\mu} \mid \mathcal{E}])^2 \mid \mathcal{E}], \quad \text{MSE}(\hat{\mu}) = \mathbb{E}[(\hat{\mu} - \mu)^2 \mid \mathcal{E}], \quad (3.2)$$

where the expectations are with respect to the data-generating process giving rise to the dataset

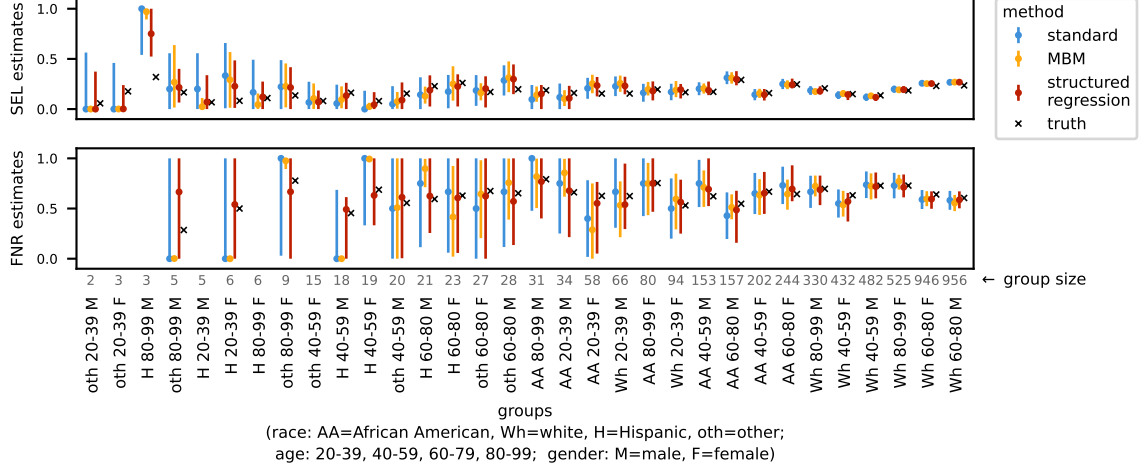


Figure 3.1: *Point estimates and 95% confidence intervals of selection rate (SEL) and false negative rate (FNR) on diabetes data. Confidence intervals of the standard estimator are calculated using pooled variance (see Equation 3.7).*

used to calculate  $\hat{\mu}$  (which is itself a random variable). An estimator with bias equal to zero is called *unbiased*.

Mean squared error decomposes into bias and variance terms as

$$\text{MSE}(\hat{\mu}) = [\text{Bias}(\hat{\mu})]^2 + \text{Var}(\hat{\mu}), \quad (3.3)$$

so for unbiased estimators, mean squared error is equal to variance.

Throughout this chapter, we assume that the standard estimates  $Z_a$  are unbiased. Writing this condition in terms of the metric  $m$ , we assume that for all  $\mathcal{D} \in \Delta$  and all  $n \geq 1$ , the performance metric  $m$  satisfies

$$\mathbb{E}_{S \sim \mathcal{D}^n} [m(S) \mid m(S) \text{ is defined}] = m(\mathcal{D}), \quad (3.4)$$

which is true for all the metrics in Section 3.1. Substituting  $\mathcal{D}_a$  for  $\mathcal{D}$  and  $n_a$  for  $n$  in Eq. (3.4) implies that  $\mathbb{E}[Z_a \mid Z_a \text{ is defined}] = \mu_a$ . In the rest of this chapter, we drop conditioning on the events like “ $Z_a$  is defined,” and just write  $\mathbb{E}[Z_a] = \mu_a$  for simplicity.

Since the standard estimates  $Z_a$  are unbiased, their MSE is equal to their variance, which typically scales as  $O(1/n_a)$ . Thus, standard estimates are accurate when  $n_a$  is large, but less accurate when  $n_a$  is small. Unfortunately, even for moderately sized evaluation datasets, the sizes of intersectional groups can be quite small. In Figure 3.1, we show standard estimates of SEL and FNR on diabetes data (alongside estimates produced by methods introduced later in this chapter). Although the evaluation dataset has 5000 data points, the intersectional groups are as small as size 2, and almost half of the groups are of size less than 25, leading to substantial errors in the standard estimates.

### 3.1.4 Structured regression approach

We next develop a structured regression (SR) approach, which seeks to overcome the main shortcoming of the standard estimator: its large variance for small groups. Our approach builds on two main ideas. First, we leverage information across *all data points*, not just data points in  $S_a$ , to estimate  $\mu_a$ , by pooling the data across related groups, for example, across intersectional groups that agree in one of their attributes (like *age*), and by using additional explanatory variables (like  $X$ ). This is accomplished by fitting a regression model for  $\mu_a$ s, with  $Z_a$ s viewed as observations. Second, we make the regression model sufficiently expressive, so that it can express standard estimates. Regularization is used to optimize the bias–variance trade-off between the high-variance standard estimator and a high-bias (but low-variance) constant estimator.

To start, since the standard estimates are unbiased, that is,  $\mathbb{E}[Z_a] = \mu_a$ , we can write

$$Z_a = \mu_a + \varepsilon_a$$

for all  $a \in \mathcal{A}$ , where  $\varepsilon_a$ 's are independent random variables with  $\mathbb{E}[\varepsilon_a] = 0$ . We denote the variance of  $Z_a$  as  $\sigma_a^2 = \text{Var}(Z_a) = \mathbb{E}[\varepsilon_a^2]$ . In order to estimate  $\mu_a$ , we consider a linear model of the form

$$\mu_a = \theta_0 + \boldsymbol{\theta} \cdot \boldsymbol{\phi}^a$$

for all  $a \in \mathcal{A}$ , where  $\boldsymbol{\phi}^a \in \mathbb{R}^d$  is the feature vector describing the group  $a$ , and  $\theta_0 \in \mathbb{R}$ ,  $\boldsymbol{\theta} \in \mathbb{R}^d$  are the parameters of the linear model. It remains to specify how to define  $\boldsymbol{\phi}^a$ , how to fit the parameters  $\theta_0$  and  $\boldsymbol{\theta}$ , and how to estimate  $\sigma_a$ .

### Defining feature vectors $\boldsymbol{\phi}^a$ .

The coordinates of  $\boldsymbol{\phi}^a$  are referred to as features and denoted as  $\phi_j^a$  for  $j$  from some suitable index set. We allow features to be linearly dependent. We consider the following types of features:

1. *Sensitive features.* These are derived directly from  $a$ . We always include group-identity indicators for all the groups  $a' \in \mathcal{A}$ , yielding features of the form  $\phi_{a'}^a = \mathbf{1}\{a = a'\}$ . This allows the linear model to express any combination of values  $\mu_a$ . Additionally, in order to pool information across related groups, we also define indicators for individual attribute values, that is, features of the form  $\phi_{i,v}^a = \mathbf{1}\{a[i] = v\}$  for  $i \in \{1, \dots, k\}$  and  $v \in \mathcal{A}_i$ . In our diabetes example, there are three sensitive attributes: *race*, *age*, and *gender*, with  $|\mathcal{A}_{\text{race}}| = 4$ ,  $|\mathcal{A}_{\text{age}}| = 4$ , and  $|\mathcal{A}_{\text{gender}}| = 2$ , so  $|\mathcal{A}| = 4 \cdot 4 \cdot 2 = 32$ . We use a total of 42 sensitive features: 32 group-identity indicators, 4 indicators of *race*, 4 indicators of *age*, and 2 indicators of *gender*. An example of a group-identity indicator is  $\phi_{(\text{Hispanic}, 80-99, \text{female})}^a$  and an example of a sensitive-attribute indicator is  $\phi_{\text{race}, \text{Hispanic}}^a$ .

2. *Explanatory features.* These are derived from  $X$ ,  $Y$ , and possibly  $\hat{Y}$ . We first featurize

$X$  using some real-valued functions  $f_j(X)$ ,  $j = 1, \dots, \ell$ , and then define explanatory features  $\phi_j^a = \mathbb{E}_{X \sim S_a}[f_j(X)]$ . Additionally, when  $Y$  is categorical, we define features  $\phi_y^a = \mathbb{P}_{Y \sim S_a}[Y = y]$  measuring rates of different outcomes in the group  $a$ . In our diabetes example, we use 7 explanatory features: 5 are derived from individual-level features  $f_j$ , including, for example, the number of inpatient days of a given patient in the prior year; and there are 2 features  $\phi_y^a$  corresponding to 2 possible values of  $Y$ .

3. *Interaction terms.* Finally, it is also possible to consider various interaction terms, both among features of the same type (such as interactions between *gender* and *age* indicators), or of different types (like interactions between the outcome  $Y$  and *age*).

### Fitting the linear model.

We fit  $(\theta_0, \boldsymbol{\theta})$  by lasso regression [99], minimizing an  $\ell_1$ -penalized square loss. To improve the statistical efficiency of the estimator, loss for each group  $a$  is weighted inversely proportional to the variance of  $Z_a$ . Intuitively, since our model can express true  $\mu_a$ , we expect the square loss on each group to be on the order of the variance of  $Z_a$ , so inverse weighting “equalizes the scale” of losses across groups. The penalized loss is then

$$L_\lambda(\theta_0, \boldsymbol{\theta}) = \sum_{a \in \mathcal{A}} \frac{1}{\sigma_a^2} \left( \theta_0 + \boldsymbol{\theta} \cdot \boldsymbol{\phi}^a - Z_a \right)^2 + \lambda \|\boldsymbol{\theta}\|_1, \quad (3.5)$$

where  $\lambda$  is the regularization hyperparameter. Denoting the minimizer of  $L_\lambda$  (for a given  $\lambda$ ) as  $(\hat{\theta}_0, \hat{\boldsymbol{\theta}})$ , we obtain the estimates  $\hat{\mu}_a = \hat{\theta}_0 + \hat{\boldsymbol{\theta}} \cdot \boldsymbol{\phi}^a$ .

Tuning  $\lambda$  allows us to navigate the bias–variance tradeoff. When  $\lambda = 0$ , the loss is minimized by  $\hat{\mu}_a = Z_a$ , which can always be expressed by suitable  $\hat{\theta}_0$  and  $\hat{\boldsymbol{\theta}}$ , because sensitive features



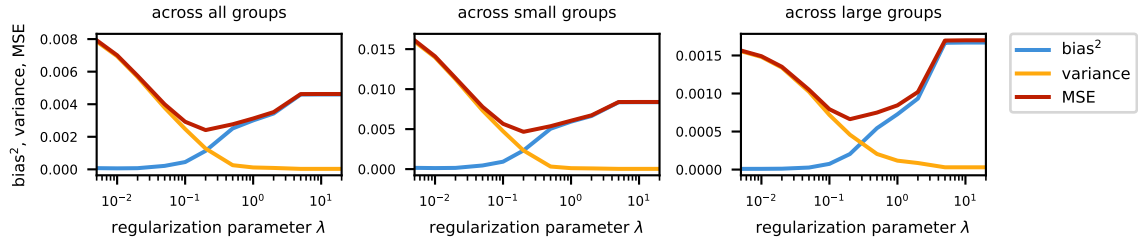


Figure 3.2: *Bias–variance trade-off of structured regression estimates of selection rate (SEL) on diabetes data.* Averaged across all groups, small groups (size at most 25), and large groups (size above 25), across 100 draws of evaluation dataset. The scale of the MSE is different for different group sizes, but the minimum MSE is attained around the same value of  $\lambda$ , thanks to the weighting of the training loss.

include indicators of all values  $a \in \mathcal{A}$ . As  $\lambda \rightarrow \infty$ , the optimization returns  $\hat{\theta} \approx \mathbf{0}$ . Fixing  $\hat{\theta} = \mathbf{0}$  and optimizing only over the intercept term yields the constant solution

$$\hat{\mu}_a = \hat{\mu}_0, \quad \text{with} \quad \hat{\mu}_0 = \frac{\sum_{a \in \mathcal{A}} Z_a / \sigma_a^2}{\sum_{a \in \mathcal{A}} 1 / \sigma_a^2},$$

corresponding to a weighted average of  $Z_a$ s. This solution has a small variance, but it may suffer from a large bias when the true values  $\mu_a$  are far from identical. By tuning  $\lambda$ , we thus move from the standard estimate to the constant estimate, decreasing the variance while increasing the bias. The mean squared error is typically minimized at some intermediate value of  $\lambda$  (see Figure 3.2). We tune  $\lambda$  by 10-fold cross-validation, where the individual folds are obtained by stratified sampling of the dataset  $S$  with respect to the sensitive attribute tuple  $A$ .

### Estimating variance $\sigma_a^2$ .

Variances  $\sigma_a^2$  are needed to determine weights in our optimization procedure. A simple approach is to estimate  $\sigma_a^2$  separately on each dataset  $S_a$  by using standard variance estimators (when available), or, more generically, by bootstrap. Unfortunately, for small sample sizes, these

variance estimates themselves might be inaccurate.

To overcome this limitation, we posit a parametric model for variance, namely,  $\sigma_a^2 = \sigma^2/n_a$ , for some parameter  $\sigma$ . To estimate  $\sigma$ , we proceed in two stages. We first use bootstrap on each set  $S_a$  to obtain the initial estimate of  $\sigma_a^2$ , which we denote  $(\hat{\sigma}_a^{\text{boot}})^2$ . Thus,  $n_a(\hat{\sigma}_a^{\text{boot}})^2$  is the initial estimate of  $\sigma^2$ . We expect the variance of this estimate to be on the order  $O(1/n_a)$ . Taking a weighted average across groups, with weighting inversely proportional to  $(1/n_a)$ , yields our final estimator of  $\sigma^2$ , which translates into an estimator of  $\sigma_a^2$ :

$$\hat{\sigma}^2 = \frac{\sum_{a \in \mathcal{A}} n_a \cdot [n_a(\hat{\sigma}_a^{\text{boot}})^2]}{\sum_{a \in \mathcal{A}} n_a} \quad \text{and} \quad \hat{\sigma}_a^2 = \hat{\sigma}^2/n_a \text{ for all } a \in \mathcal{A}. \quad (3.6)$$

We refer to these as the *pooled estimates* of variance. In our preliminary experiments, these performed better than the initial estimates  $(\hat{\sigma}_a^{\text{boot}})^2$ , particularly on small datasets.

### 3.1.5 Confidence intervals

So far we have focused on obtaining point estimates  $\hat{\mu}_a$ . However, in order for these estimates to be useful in practice, we also need to quantify our uncertainty about their values. We do so by using confidence intervals. For unbiased estimators, like the standard estimator  $Z_a$ , confidence intervals can be derived by estimating the variance and then using normal approximation, which works quite well for  $Z_a$  with the pooled estimates of variance (see Appendix C.1).

However, variance-based approach does not work with lasso estimates, because they are biased—in fact, they achieve their improved accuracy *by being biased*—and so a simple approach of using variance-based confidence intervals or bootstrap percentiles yields confidence intervals that are too narrow. Fortunately, there is a rich literature on lasso-based confidence

intervals [100, 101, 102]. We use the *residual bootstrap lasso+partial ridge* (rBLPR) approach of Liu et al. [103]. As the name suggests, it is based on a two-stage *lasso+partial ridge* (LPR) point estimator, which first runs lasso as a feature-selection method, and then fits a ridge regression model, which only penalizes the features that were not selected by lasso. The rBLPR method calculates confidence intervals for the LPR estimate by residual bootstrap (see [103] for details).

### 3.1.6 Goodness-of-fit testing

When presenting the results of disaggregated evaluations, the most common approach is to display point estimates and (sometimes) confidence intervals for every subgroup, as we see, for example, in Figure 3.1. While this type of a plot can be helpful in identifying groups that may experience poor performance or allocation, it does not provide a narrative for understanding how these harms accrue. Goodness-of-fit testing can complement disaggregated evaluations by allowing us to answer questions such as:

1. *Do intersectional groups experience additive, sub-additive, or super-additive fairness-related harms?* For example, when a model is found to perform poorly for Black women, is this explained by the model performing poorly for Black people and women, or are there additional sources of error specific to the intersectional group of Black women? An answer to this question can, for example, inform future collection of training data.
2. *Are there benign factors that explain a significant amount of the observed performance variation across groups?* For example, are observed differences in the performance of an ASR system attributable to systematically worse audio quality in the recordings for speakers from certain groups? Presence of such benign factors *does not lessen the harm*,

but the knowledge of the factors that drive performance differences can be used to design mitigations (for example, denoising algorithms targeted at specific types of sensors or noise characteristics).

These types of questions can be framed as goodness-of-fit tests. We consider goodness-of-fit tests that compare two linear models:  $M_0$ , with fewer features, and  $M_1$ , with some additional features. Such a test asks whether the additional features included in model  $M_1$  improve the goodness of fit compared with model  $M_0$ , where the goodness-of-fit is measured using the square loss as in Eq. (3.5). To answer the first question above, we can compare a model  $M_0$ , which includes only indicators of *race* and *gender*, with a model  $M_1$ , which also includes interaction terms. To answer the second question, we can compare a model  $M'_0$ , which only includes benign factors, with a model  $M'_1$ , which additionally includes indicators of *race*, *gender*, and *age*.

While there are goodness-of-fit tests that have been designed for lasso regression [104], in this chapter, we use standard  $F$ -tests designed for unregularized linear regression. In contrast to the foregoing discussion, we do not include features corresponding to the indicators of  $a$  (because these would trivially yield standard estimates with perfect goodness-of-fit, which in this case corresponds to overfitting).

### 3.1.7 Experiments

In this section, we evaluate the accuracy of point estimates and calibration of confidence intervals produced by our structured regression (SR) approach. We also demonstrate how goodness-of-fit tests can be used to provide insights about what drives the variation of performance across groups.

In our evaluation, we compare SR with several baselines. First, there is the **standard estimator**  $Z_a = m(S_a)$ . We construct confidence intervals for  $Z_a$  using normal approximation with pooled variance estimates  $(\hat{\sigma}_a)^2$  from Eq. (3.6). Given a confidence level  $\gamma$  (say 95%), or a significance level  $\alpha = 1 - \gamma$  (say 5%), we use the confidence interval

$$[Z_a + q_{\alpha/2}\hat{\sigma}_a, Z_a + q_{1-\alpha/2}\hat{\sigma}_a], \quad (3.7)$$

where  $q_p$  is the  $p$ -th quantile of the standard normal distribution.

Our second baseline is the **model-based metrics (MBM)** approach [79]. As mentioned in Section 3.1.2, MBM is a Bayesian approach to structured regression that models the scores produced by an AI system (like  $\hat{Y}$  in the diabetes example). However, it is not directly applicable to performance metrics that are not based on scores, so we do not use it in the ASR experiments. Similar to SR, MBM uses linear modeling, and so requires specifying features for each data point. It comes with a bootstrapping procedure for constructing confidence intervals.

And finally, we compare our point estimates with the classical **James-Stein (JS)** estimator [94, 95]. The estimator works by shrinking standard estimates towards zero (or some other constant). We use a variant due to Bock [105], which is adapted to unequal variances (in our case, pooled estimates  $\hat{\sigma}_a^2 = \hat{\sigma}^2/n_a$ ), giving rise to

$$\hat{\mu}_a^{\text{JS}} = \hat{\mu}_0 + \left(1 - \frac{(|\mathcal{A}| - 3)\hat{\sigma}^2}{\sum_{a' \in \mathcal{A}} n_{a'}(Z_{a'} - \hat{\mu}_0)^2}\right)_+ (Z_a - \hat{\mu}_0),$$

where  $\hat{\mu}_0 = (\sum_{a \in \mathcal{A}} n_a Z_a)/n$  is a weighted average of  $Z_a$ 's. Compared with Bock's original estimator [105], we use  $|\mathcal{A}|$  in the numerator, as this has been previously observed to lead to better

performance [106]. This method does not have a standard procedure for confidence intervals.

### 3.1.7.1 Diabetes experiments

In our first set of experiments, we explore the scenario from Example 1 using the dataset developed by Strack et al. [98], and previously used in an AI fairness tutorial [107] and to evaluate the MBM approach [79]. The dataset contains hospital admission records from 130 hospitals in the U.S. over a ten-year period (1998–2008) for patients who were admitted with a diabetes diagnosis and whose hospital stay lasted one to fourteen days. It is a tabular dataset with 47 features describing each encounter, including patient demographics and clinical information (see Strack et al. [98] for more details).

Following Miller et al. [79], we filter out records with missing demographic information and those with age below 20. We preprocess clinical features as in [107]. To emulate an AI system that scores patients for a post-discharge care program, we use 25% of the data to train a logistic regression model to predict whether the patient will be readmitted into hospital within 30 days. The remaining 75% of the data, consisting of 73,988 hospital admissions across 55,157 individuals, is used as the ground truth  $\mathcal{D}$  in all of our evaluation experiments.

We consider three sensitive attributes, *race*, *age*, and *gender*, with  $\mathcal{A}_{race} = \{\text{African American, Hispanic, white, other}\}$ ,  $\mathcal{A}_{age} = \{20\text{--}39, 40\text{--}59, 60\text{--}79, 80\text{--}99\}$ , and  $\mathcal{A}_{gender} = \{\text{male, female}\}$ . Hospital admissions are represented as tuples  $(X, A, Y, \hat{Y})$ , where  $X$  includes the clinical features,  $A = (\text{race}, \text{age}, \text{gender})$ ,  $Y \in \{0, 1\}$  indicates whether the patient was readmitted within 30 days of discharge, and  $\hat{Y} \in [0, 1]$  is the readmission probability predicted by the logistic regression model. From the ground truth we then sample an evaluation dataset  $S$  of size 5000 by stratified

sampling according to  $A$ .

As in Example 1, we assume that the hospital uses a threshold  $r$ , and patients with  $\hat{Y} \geq r$  are automatically referred into the care program. We set the threshold  $r$  so that  $\mathbb{P}_{\mathcal{D}}[\hat{Y} \geq r] = 0.2$ , meaning that only 20% of patients are referred, and write  $\pi(\hat{Y}) = \mathbf{1}\{\hat{Y} \geq r\}$  to denote this decision rule. We consider 6 performance metrics (including those already introduced earlier), defined for any  $p \in \Delta$  as

$$\begin{aligned} \text{SEL}(p) &= \mathbb{P}_p[\pi(\hat{Y}) = 1], & \text{ACC}(p) &= \mathbb{P}_p[\pi(\hat{Y}) = Y], \\ \text{FNR}(p) &= \mathbb{P}_p[\pi(\hat{Y}) = 0 \mid Y = 1], & \text{FPR}(p) &= \mathbb{P}_p[\pi(\hat{Y}) = 1 \mid Y = 0], \\ \text{PPV}(p) &= \mathbb{P}_p[Y = 1 \mid \pi(\hat{Y}) = 1], & \text{AUC}(p) &= \mathbb{P}_{(Y, \hat{Y}) \sim p, (Y', \hat{Y}') \sim p}[\hat{Y} < \hat{Y}' \mid Y = 0, Y' = 1]. \end{aligned}$$

The first five metrics (selection rate, accuracy, false positive rate, false negative rate, and positive predictive value) are derived from the confusion matrix. The final metric is the area under the ROC curve;  $(Y, \hat{Y})$  and  $(Y', \hat{Y}')$  in its definition are sampled independently according to  $p$ .

In order to apply SR, we need to specify features  $\phi^a$ . As sensitive features, we use indicators of *race*, *age*, *gender*, as well as indicators of the triple  $(\textit{race}, \textit{age}, \textit{gender})$ . We use 7 explanatory features: indicators for 2 possible values of  $Y$ , and 5 additional clinical features describing the number of inpatient visits, outpatient visits, and emergency visits in the preceding year, number of diagnoses at admission, and whether any of the diagnoses was congestive heart failure. For MBM, we use the same set of features, but without the triple indicators.

In Figure 3.1 from earlier, point estimates obtained by SR appear to be closer to the ground truth than those obtained by the standard method and MBM. Confidence intervals constructed by SR are of similar size as the standard confidence intervals, and occasionally smaller. MBM

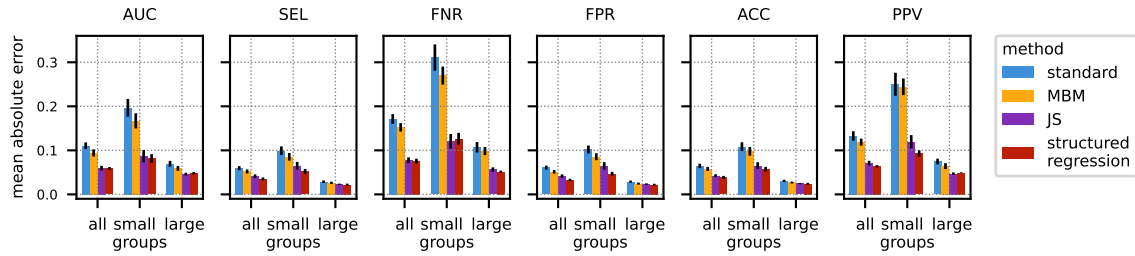


Figure 3.3: Mean absolute error of estimates of 6 metrics using 4 methods on diabetes data. Averaged across all groups, small groups (size at most 25), and large groups (size above 25), across 20 draws of evaluation dataset.

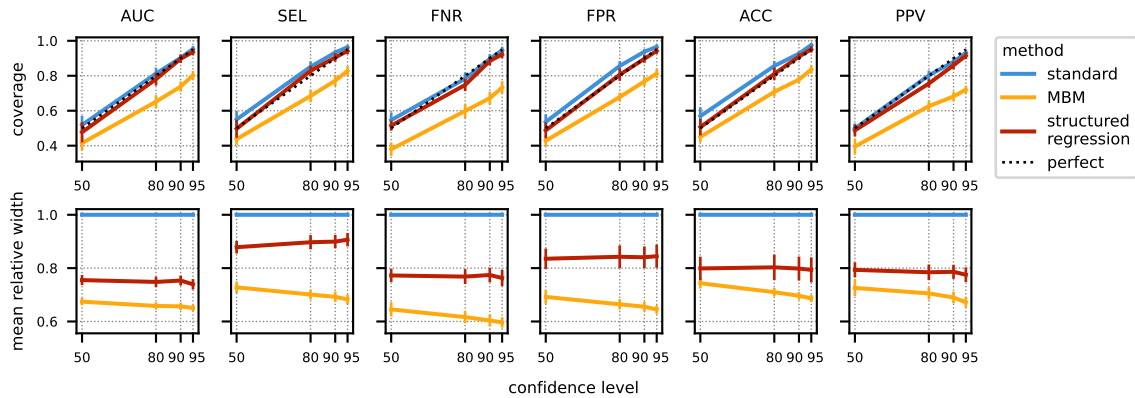


Figure 3.4: Coverage and mean relative width of confidence intervals for 6 metrics constructed by 3 methods on diabetes data. Averaged across all groups and across 20 draws of evaluation dataset. Relative width is with respect to the width of the standard confidence interval.

appears to produce smaller confidence intervals than SR, but they seem to miss the ground-truth metric values more often. We next evaluate these anecdotal observations more systematically.

In Figure 3.3, we evaluate the quality of point estimates using *mean absolute error* (MAE), which is the mean deviation of the point estimate from the truth, averaged across 20 draws of evaluation dataset, and over all groups, or separately over the groups of size at most 25 (which we call *small*) and groups of size greater than 25 (which we call *large*). We see that JS and SR yield substantially more accurate point estimates than the standard method and MBM. The improvement is particularly dramatic for small groups. JS and SR exhibit similar performance, but in several cases SR is somewhat better on small groups (e.g., for FPR and PPV). (We use MAE instead of MSE, because MAE values are easier to interpret, but MSE results are qualitatively similar.)



In Figure 3.4, we shift attention to confidence intervals. In the top plots, we evaluate *coverage*, that is, how often the ground truth lies in the confidence intervals (across 20 draws of evaluation dataset and across all groups). We show coverage as a function of the confidence level. We see that both standard method as well as SR are well-calibrated, with their coverage close to the confidence level, whereas MBM is over-confident, with coverage well below the confidence level. In the bottom plots, we evaluate the *mean relative width* of confidence intervals, meaning the mean of the ratio between the width of a confidence interval and the width of the standard confidence interval. We see that MBM has the narrowest intervals, but this is at the expense of coverage. On the other hand, SR is able to maintain well-calibrated coverage while still decreasing the confidence intervals by up to 20% compared with the standard method.

Finally, in Table 3.1, we demonstrate the use of goodness-of-fit tests:

Estimated metric	Goodness-of-fit test $p$ -values (comparing a more expressive vs a less expressive model)			
	<i>expl</i> vs $\emptyset$	<i>sens</i> vs $\emptyset$	<i>expl + sens</i> vs <i>expl</i>	<i>expl + sens + Y · sens</i> vs <i>expl + sens</i>
AUC	0.281	0.438	0.726	0.543
SEL	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	0.842
FNR	0.093	0.473	0.735	0.565
FPR	<b>0.001</b>	<b>0.000</b>	<b>0.000</b>	0.431
ACC	<b>0.000</b>	<b>0.000</b>	<b>0.005</b>	0.182
PPV	0.316	0.493	0.470	0.874

*Model abbreviations:  $\emptyset$ =intercept only, expl=explanatory features, sens=sensitive features, ·=interactions*

Table 3.1: *Goodness-of-fit tests on diabetes data.* From left to right, we consider increasingly more complex models with a growing set of features and report the  $p$ -values of the corresponding goodness-of-fit tests;  $p$ -values below 0.05 are in bold.

From left to right, we consider increasingly more complex models with a growing set of features. Beginning with just the intercept, adding explanatory features, then sensitive features (just

the indicators of *race*, *age*, and *gender*, but not of their combination), and eventually interaction terms between the outcome  $Y$  and sensitive features.

There is no evidence to go beyond the intercept-only model when estimating AUC, FNR, PPV. This does not necessarily mean that there is no disparity in performance, but we might not have enough data to tell. Indeed, for instance, confidence intervals for FNR in Figure 3.1 are large for a vast majority of the groups, so in this case using SR is not sufficient to reduce uncertainty, and additional data collection may be required. On the other hand, the table shows that both explanatory and sensitive features help with modeling SEL, FPR, and ACC. In fact, sensitive features improve the fit *after* the explanatory features have already been added, meaning that differences in performance across the groups cannot be explained by the “benign” explanatory features alone.

### 3.1.7.2 Experiments with synthetic data

We next provide a few more examples of goodness-of-fit analysis on synthetic data. We continue to use the diabetes dataset as described in the previous section, but with different values  $\hat{Y}$ . We consider the performance metric  $m(p) = \mathbb{E}_p[\hat{Y}]$  (this is quite similar to selection rate or word error rate) and generate  $\hat{Y}$  in such a way that ground-truth metric values  $\mu_a$  have a specific structure. We consider 4 different ground-truth structures, titled  $\text{model}_{age}$ ,  $\text{model}_{age+rc}$ ,  $\text{model}_{age\cdot rc}$ , and  $\text{model}_{expl}$ , according to the variables they depend on, with “+” denoting an additive dependence and “.” presence of interactions (see Appendix C.2 for details).

In Table 3.2, moving from left to right, we test goodness-of-fit of more and more complex models. In the first row, ground-truth depends only on *age*, but there is a significant improvement in

goodness-of-fit from  $\emptyset$  to  $expl$ , because of correlation between  $age$  and  $expl$ . After the explanatory features have been included,  $age$  still helps (the improvement from  $expl$  to  $expl + age$  is significant), so the variation in the performance metric cannot be explained by the “benign factors” alone. On the other hand, if the data is drawn from  $model_{expl}$ , there is no evidence that  $age$  or  $race$  help after explanatory features have been added.

Data-generating model	Goodness-of-fit test $p$ -values (comparing a more expressive vs a less expressive model)						
	$expl$ vs $\emptyset$	$age$ vs $\emptyset$	$expl + age$ vs $expl$	$expl + rc$ vs $expl$	$expl + age + rc$ vs $expl + age$	$expl + age + rc$ vs $expl + rc$	$expl + age + rc + age \cdot rc$ vs $expl + age + rc$
$model_{age}$	<b>0.025</b>	<b>0.000</b>	<b>0.000</b>	0.487	0.153	<b>0.000</b>	0.576
$model_{expl}$	<b>0.000</b>	<b>0.000</b>	0.323	0.366	0.608	0.551	<b>0.000</b>
$model_{age+rc}$	<b>0.013</b>	0.661	0.142	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	0.089
$model_{age \cdot rc}$	<b>0.003</b>	<b>0.000</b>	<b>0.000</b>	<b>0.040</b>	<b>0.015</b>	<b>0.000</b>	<b>0.002</b>

Model abbreviations:  $\emptyset$ =intercept only,  $expl$ =explanatory features,  $rc$ =race,  $\cdot$ =interactions

Table 3.2: *Goodness of fit tests on synthetic data.* From left to right, we consider increasingly more complex models with a growing set of features and report the  $p$ -values of the corresponding goodness-of-fit tests;  $p$ -values below 0.05 are in bold.

In the last two rows, we demonstrate how goodness-of-fit tests handle data from additive models versus models with interactive terms. The former corresponds to the situation when harms experienced by intersectional groups combine additively, the latter when there is an additional intersectional effect. For the additive ground truth ( $model_{age+rc}$ ), tests suggest a sequence of variable additions  $expl + rc + age$ , but then show no support for including interaction terms. For the data from  $model_{age \cdot rc}$ , tests correctly provide support for an inclusion of interactions.

### 3.1.7.3 Experiments with ASR data

Finally, we explore the scenario from Example 2 using the data provided by Koenecke et al. [74] as a supplement to their paper finding racial disparities in commercial ASR systems. Similar

to Koenecke et al. [74], we use the *matched* dataset, which contains 4282 snippets across 105 distinct speakers. (Matching ensures that there is the same number of snippets from Black and white speakers and that the marginal distributions of various descriptive statistics match.)

For each audio snippet, we are provided with various statistics (like duration and word count), an anonymized speaker id, speaker demographics, and word error rates (WERs) on that snippet by five ASR systems, developed by Google, IBM, Amazon, Microsoft, and Apple. This information is encoded as a tuple  $(X, A, W_1, \dots, W_5)$ , where  $X$  contains the identity of the speaker, the duration of the snippet in seconds, and word count,  $A$  contains two sensitive attributes, *gender* and *race*, with  $\mathcal{A}_{gender} = \{\text{male, female}\}$  and  $\mathcal{A}_{race} = \{\text{Black, white}\}$ , and finally, instead of  $Y$  (human transcription) and  $\hat{Y}_1, \dots, \hat{Y}_5$  (transcriptions by five ASR systems), we directly have the corresponding word error rates  $W_i = \text{wer}(\hat{Y}_i, Y)$ . The performance metric for the system  $i$  is thus  $m(p) = \mathbb{E}_p[W_i]$  for any  $p \in \Delta$ .

Although there appears to be a large number of samples ( $n = 4282$ ), there are only 105 distinct speakers. We expect there to be a substantial amount of correlation between WERs of the same individual, so an analysis that treats the WERs as independent is likely to overstate the statistical significance of findings, and may arrive at incorrect conclusions, in particular, when some speakers have many more snippets than others. In our experiments, we therefore present results both from a snippet-level analysis that treats the WERs across all snippets as independent (as done in [74]), and a speaker-level analysis that first reduces the data to speaker-level WERs by taking an average of WERs across the speaker’s snippets.

We first compare disaggregated evaluation results obtained by SR versus the standard method. To apply SR, we need to specify features  $\phi^a$ . As sensitive features, we use indicators of *race* and *gender*, as well as indicators of the pair  $(\text{race}, \text{gender})$ . We use only one explanatory

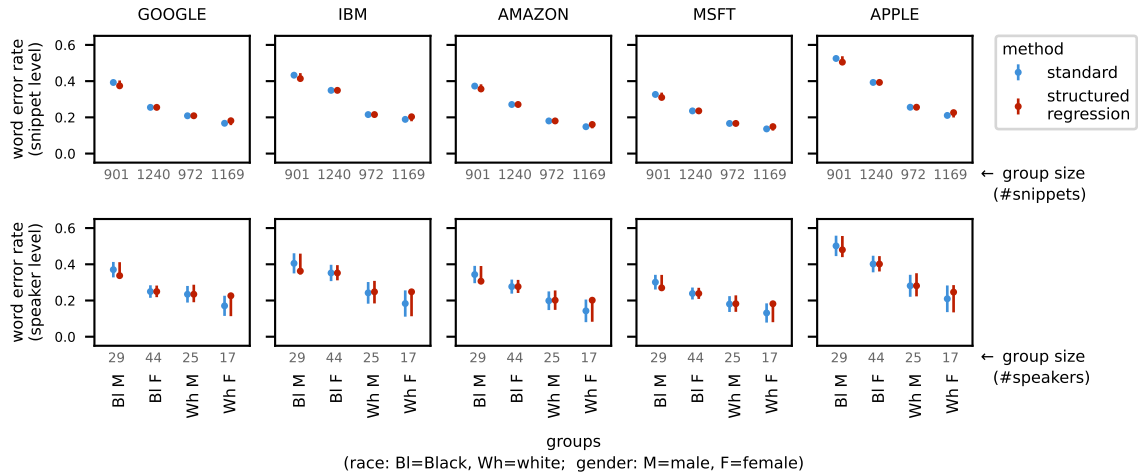


Figure 3.5: Point estimates and 95% confidence intervals of word error rates of five different ASR systems.

feature, equal to the log duration of the snippet.

In Figure 3.5, we report the results. At the snippet level, both methods generally replicate the results of Koenecke et al. [74]: Black male speakers have the largest WER, followed by Black female speakers, white male speakers, and white female speakers. The main difference is that SR systematically shrinks the WER values of the extreme groups (Black male speakers and white female speakers) towards the mean. Results at the speaker level have substantially larger confidence intervals than the snippet-level results, reflecting smaller group sizes. Also, due to smaller group sizes, the SR point estimates are shrunk towards the mean more aggressively.

We also carry out the goodness-of-fit analysis of structure of intersectional harms. At the speaker level, we find that the variation of performance of all systems is well-explained by the additive model  $expl + race + gender$  (the  $p$ -values of adding each variable in turn are below 0.003), but not by a model with interactions. This is in contrast with the snippet-level analysis, which supports the model with interactions (with  $p$ -values below 0.001). We interpret this conservatively and conclude that there is evidence for an additive structure of intersectional harms, but not for an interaction term. This does not mean that there are no interaction effects, just that we cannot

conclude that from the data at hand.

### 3.1.8 Summary of Contributions

In Chapter 3.1, we have introduced a structured regression (SR) approach to disaggregated evaluation and compared its performance with a variety of baselines. We have seen that SR and James-Stein (JS) estimator both substantially improve the accuracy of point estimates compared with the standard approach and a more sophisticated MBM baseline. Both methods are simple to implement, and are also close in terms of performance, so the choice among them should be driven by their usability. Here, SR has some advantages. Its ability to include application-specific features makes it more flexible, and it has a well-developed inference procedures like construction of confidence intervals and goodness-of-fit tests. However, we have evaluated SR only in two domains, so any applications in domains with different characteristics (including number and types of explanatory and sensitive features, and dataset sizes) require additional validation.

In Section 3.1.2, we mentioned challenges that arise in conceptualization stages of disaggregated evaluation. Once the disaggregated results are produced, a complementary set of challenges arises in how to interpret them. We have conspicuously omitted analysis of regression coefficients, because in our preliminary experiments, we found that lasso coefficients exhibit too much variance for reliable inference. Instead, we suggest to use goodness-of-fit tests and we have demonstrated several ways how. We acknowledge that we have just taken some initial steps in this area, and there are many opportunities to apply more sophisticated statistical techniques. Our exploration also completely leaves out important sociotechnical questions about how to draw actionable conclusions, and how to best communicate the results to relevant stake-holders, both

of which are key in translating fairness assessments into a reduction in fairness-related harms.

## 3.2 Identifying and Mitigating Clinical Natural Language Inference Artifacts

This work was done in collaboration with my co-author, Rachel Rudinger; see [14].

### 3.2.1 Introduction

In the clinical domain, the ability to conduct natural language inference (NLI) on unstructured, domain-specific texts such as patient notes, pathology reports, and scientific papers, plays a critical role in the development of predictive models and clinical decision support (CDS) systems.

Considerable progress in domain-agnostic NLI has been facilitated by the development of large-scale, crowdworker-constructed datasets, including the Stanford Natural Language Inference corpus (SNLI), and the Multi-Genre Natural Language Inference (MultiNLI) corpus [9, 10]. MedNLI is a similarly-motivated, healthcare-specific dataset created by a small team of physician-annotators in lieu of crowdworkers, due to the extensive domain expertise required [7].

Poliak et al. [11], Gururangan et al. [8], Tsuchiya [108], and McCoy et al. [109] empirically demonstrate that SNLI and MultiNLI contain lexical and syntactic annotation artifacts that are disproportionately associated with specific classes, allowing a hypothesis-only classifier to significantly outperform a majority-class baseline model. The presence of such artifacts is hypothesized to be partially attributable to the priming effect of the example hypotheses provided to crowdworkers at annotation-time. Romanov and Shivade [7] note that a hypothesis-only baseline is able to outperform a majority class baseline in MedNLI, but they do not identify specific artifacts.

We confirm the presence of annotation artifacts in MedNLI and proceed to identify their lexical and semantic characteristics. We then conduct adversarial filtering to partition MedNLI into *easy* and *difficult* subsets [12]. We find that performance of off-the-shelf `fastText`-based



hypothesis-only and hypothesis-plus-premise classifiers is lower on the *difficult* subset than on the *full* and *easy* subsets [110]. We provide partition information for downstream use, and conclude by advocating alternative dataset construction strategies for knowledge-intensive domains.<sup>1</sup>

### 3.2.2 The MedNLI Dataset

MedNLI is domain-specific evaluation dataset inspired by general-purpose NLI datasets, including SNLI and MultiNLI [7, 9, 10]. Much like its predecessors, MedNLI consists of premise-hypothesis pairs, in which the premises are drawn from the `Past Medical History` sections of a randomly selected subset of de-identified clinical notes contained in MIMIC-III [111, 112]. MIMIC-III was created from the records of adult and neonatal intensive care unit (ICU) patients. As such, complex and clinically severe cases are disproportionately represented, relative to their frequency of occurrence in the general population.

Physician-annotators were asked to write a *definitely true*, *maybe true*, and *definitely false* set of hypotheses for each premise, corresponding to *entailment*, *neutral* and *contradiction* labels, respectively. The resulting dataset has cardinality:  $n_{\text{train}} = 11232$ ;  $n_{\text{dev}} = 1395$ ;  $n_{\text{test}} = 1422$ .

### 3.2.3 MedNLI Contains Artifacts

To determine whether MedNLI contains annotation artifacts that may artificially inflate the performance of models trained on this dataset, we train a simple, premise-unaware, `fastText` classifier to predict the label of each premise-hypothesis pair, and compare the performance of this classifier to a majority-class baseline, in which all training examples are mapped to the most commonly occurring class label [8, 11, 110]. Note that since annotators were asked to create an

---

<sup>1</sup>See [https://github.com/crherlihy/clinical\\_nli\\_artifacts](https://github.com/crherlihy/clinical_nli_artifacts) for code and partition ids.

entailed, contradictory, and neutral hypothesis for each premise, MedNLI is class-balanced. Thus, in this setting, a majority class baseline is equivalent to choosing a label uniformly at random for each training example.

The micro F1-score achieved by the `fastText` classifier significantly exceeds that of the majority class baseline, confirming the findings of Romanov and Shivade [7], who report a micro-F1 score of 61.9 but do not identify or analyze artifacts:

	<b>dev</b>	<b>test</b>
majority class	33.3	33.3
<code>fastText</code>	<b>64.8</b>	<b>62.6</b>

Table 3.3: Performance (micro F1-score) of the `fastText` hypothesis-only classifier.

As the confusion matrix for the test set shown in Table 3.4 indicates, the `fastText` model is most likely to misclassify entailment as neutral, and neutral and contradiction as entailment. Per-class precision and recall on the test set are highest for *contradiction* (73.2; 72.8) and lowest for *entailment* (56.7; 53.8).

	entailment	neutral	contradiction
entailment	<b>255</b>	151	68
neutral	126	<b>290</b>	58
contradiction	69	60	<b>345</b>

Table 3.4: Confusion matrix for `fastText` classifier.

### 3.2.4 Characteristics of Clinical Artifacts

In this section, we conduct class-specific lexical analysis to identify the clinical and domain-agnostic characteristics of annotation artifacts associated with each set of hypotheses in MedNLI.

### 3.2.4.1 Preprocessing

We cast each hypothesis string in the MedNLI training dataset to lowercase. We then use a `scispaCy` model pre-trained on the `en_core_sci_lg` corpus for tokenization and clinical named entity recognition (CNER) [113]. One challenge associated with clinical text, and scientific text more generally, is that semantically meaningful entities often consist of spans rather than single tokens. To mitigate this issue during lexical analysis, we map each multi-token entity to a single-token representation, where sub-tokens are separated by underscores.

### 3.2.4.2 Lexical Artifacts

Following Gururangan et al. [8], to identify tokens that occur disproportionately in hypotheses associated with a specific class, we compute token-class pointwise mutual information (PMI) with add-50 smoothing applied to raw counts, and a filter to exclude tokens appearing less than five times in the overall training dataset. Table 3.5 reports the top 15 tokens for each class.

$$\text{PMI}(\text{token}, \text{class}) = \log_2 \frac{p(\text{token}, \text{class})}{p(\text{token}, \cdot)p(\cdot, \text{class})} \quad (3.8)$$

**Entailment** Entailment hypotheses are characterized by tokens about: (1) patient status and response to treatment (e.g., *responsive*; *failed*; *longer* as in *no longer intubated*); (2) medications and procedures which are common among ICU patients (e.g., *broad\_spectrum*; *antibiotics*; *pressors*; *steroid\_medication*; *underwent*; *removal*); (3) generalized versions of specific words in the premise (e.g., *comorbidities*; *multiple\_medical\_problems*), which Gururangan et al. [8] also observe in SNLI; and (4) modifiers related to duration, frequency, or probability (e.g., *frequent*,

entailment	%	neutral	%	contradiction	%
just	0.25%	cardiogenic_shock	0.33%	no_history_of_cancer	0.27%
high_risk	0.26%	pelvic_pain	0.30%	no_treatment	0.27%
pressors	0.25%	joint_pain	0.30%	normal_breathing	0.27%
possible	0.26%	brain_injury	0.32%	no_history_of_falls	0.27%
elevated_blood_pressure	0.26%	delerium	0.30%	normal_heart_rhythm	0.28%
responsive	0.25%	intracranial_pressure	0.30%	health	0.26%
comorbidities	0.26%	smoking	0.42%	normal_head_ct	0.26%
spectrum	0.27%	obesity	0.41%	normal_vision	0.26%
steroid_medication	0.25%	tia	0.32%	normal_aortic_valve	0.27%
longer	0.26%	acquired	0.31%	bradycardic	0.26%
history_of_cancer	0.26%	head_injury	0.31%	normal_blood_sugars	0.27%
broad	0.26%	twins	0.30%	normal_creatinine	0.28%
frequent	0.25%	fertility	0.30%	cancer_history	0.26%
failed	0.26%	statin	0.30%	cardiac	0.33%
medical	0.29%	acute_stroke	0.30%	normal_chest	0.28%

Table 3.5: Top 15 tokens by PMI(token, class); % of *class* training examples containing the token.

*possible, high\_risk*).

**Neutral** Neutral hypotheses feature tokens related to: (1) chronic and acute clinical conditions (e.g., *obesity; joint\_pain; brain\_injury*); (2) clinically relevant behaviors (e.g., *smoking; alcoholic; drug\_overdose*); and (3) gender and reproductive status (e.g., *fertility; pre\_menopausal*). Notably, the most discriminative conditions tend to be commonly occurring within the general population and generically stated, rather than rare and specific. This presumably contributes to the relative difficulty that the hypothesis-only `fastText` model has distinguishing between the entailment and neutral classes.

**Contradiction** Contradiction hypotheses are characterized by tokens that convey normalcy and good health. Lexically, such sentiment manifests as: (1) explicit negation of clinical severity, medical history, or in-patient status (e.g., *denies\_pain; no\_treatment; discharged\_home*), or (2) affirmation of clinically unremarkable findings (e.g., *normal\_heart\_rhythm; normal\_blood\_sugars*),

which would generally be rare among ICU patients. This suggests a heuristic of inserting negation token(s) to contradict the premise, which Gururangan et al. [8] also observe in SNLI.

### 3.2.4.3 Syntactic Artifacts

**Hypothesis Length** In contrast to Gururangan et al. [8]’s finding that entailed hypotheses in SNLI tend to be shorter while neutral hypotheses tend to be longer, hypothesis sentence length does not appear to play a discriminatory role in MedNLI, regardless of whether we consider merged- or separated-token representations of multi-word entities, as illustrated by Table 3.6:

	<b>entailment</b>		<b>neutral</b>		<b>contradiction</b>	
	mean	median	mean	median	mean	median
<b>separate</b>	5.6	5.0	5.2	5.0	5.6	5.0
<b>merged</b>	5.3	5.0	4.9	5.0	5.3	5.0

Table 3.6: Average and median hypothesis length by class and entity representation.

### 3.2.5 Physician-Annotator Heuristics

In this section, we re-introduce premises to our analysis to evaluate a set of hypotheses regarding latent, class-specific annotator heuristics. If annotators *do* employ class-specific heuristics, we should expect the semantic contents,  $\varphi$ , of a given hypothesis,  $h \in \mathcal{H}$ , to be influenced not only by the semantic contents of its associated premise,  $p \in \mathcal{P}$ , but also by the target class,  $c \in \mathcal{C}$ .

To investigate, we identify a set of heuristics parameterized by  $\varphi(p)$  and  $c$ , and characterized by the presence of a set of heuristic-specific Medical Subject Headings (MeSH) linked entities in the premise and hypothesis of each heuristic-satisfying example. These heuristics are described below; specific MeSH features are detailed in the Appendix.

**Hypernym Heuristic** This heuristic applies when the premise contains clinical condition(s), medication(s), finding(s), procedure(s) or event(s), the target class is *entailment*, and the generated hypothesis contains term(s) that can be interpreted as super-types for a subset of elements in the premise (e.g., clindamycin < : antibiotic).

**Probable Cause Heuristic** This heuristic applies when the premise contains clinical condition(s), the target class is *neutral*, and the generated hypothesis provides a plausible, often subjective or behavioral, causal explanation for the condition, finding, or event described in the premise (e.g., associating altered mental status with drug overdose).

**Everything Is Fine Heuristic** This heuristic applies when the premise contains condition(s) or finding(s), the target class is *contradiction*, and the generated hypothesis negates the premise or asserts unremarkable finding(s). This can take two forms: repetition of premise content plus negation, or inclusion of modifiers that convey good health.

**Analysis:** We conduct a  $\chi^2$  test for each heuristic to determine whether we are able to reject the null hypothesis that pattern-satisfying premise-hypothesis pairs are uniformly distributed over classes.

heuristic	$\chi^2$	p-value	top class
hypernym	59.15	1.4e−13‡	entail (45.2%)
probable cause	111.05	7.7e−25‡	neutral (57.8%)
everything fine	874.71	1.1e−190‡	contradict (83.8%)

Table 3.7: Results of  $\chi^2$  test statistic by heuristic, computed using the combined MedNLI dataset (‡  $p < 0.001$ , †  $p < 0.01$ , \*  $p < 0.5$ ). Top class presented with % of heuristic-satisfying pairs.

The results support our hypotheses regarding each of the three heuristics. Notably, the

percentage of heuristic-satisfying pairs accounted for by the top class is lowest for the HYPERNYM hypothesis, which we attribute to the high degree of semantic overlap between entailed and neutral hypotheses.

### 3.2.6 Adversarial Filtering

To mitigate the effect of clinical annotation artifacts, we employ `AFLite`, an adversarial filtering algorithm introduced by Sakaguchi et al. [12] and analyzed by Bras et al. [114], to create *easy* and *difficult* partitions of MedNLI.

`AFLite` requires distributed representations of the full dataset as input, and proceeds in an iterative fashion. At each iteration, an ensemble of  $n$  linear classifiers are trained and evaluated on different random subsets of the data. A score is then computed for each premise-hypothesis instance, reflecting the number of times the instance is correctly labeled by a classifier, divided by the number of times the instance appears in any classifier’s evaluation set. The top- $k$  instances with scores above a threshold,  $\tau$ , are filtered out and added to the *easy* partition; the remaining instances are retained. This process continues until the size of the filtered subset is  $< k$ , or the number of retained instances is  $< m$ ; retained instances constitute the *difficult* partition.

To represent the full dataset, we use `fastTextMIMIC-III` embeddings, which have been pretrained on deidentified patient notes from MIMIC-III [7, 111]. We represent each example as the average of its component token vectors.

We proportionally adjust a subset of the hyperparameters used by Sakaguchi et al. [12] to account for the fact that MedNLI contains far fewer examples than WINOGRANDE<sup>2</sup>: specifically,

---

<sup>2</sup>MedNLI’s training dataset contains 14049 examples when the training, dev, and test sets are combined, while WINOGRANDE contains 47K after excluding the 6K used for fine-tuning.

we set the training size for each ensemble,  $m$ , to 5620, which represents  $\approx \frac{2}{5}$  of the MedNLI combined dataset. The remaining hyperparameters are unchanged: the ensemble consists of  $n = 64$  logistic regression models, the filtering cutoff,  $k = 500$ , and the filtering threshold  $\tau = 0.75$ .

We apply `AFLite` to two different versions of MedNLI: (1)  $\mathcal{X}_{h,m}$ : hypothesis-only, multi-token entities merged, and (2)  $\mathcal{X}_{ph,m}$ : premise and hypothesis concatenated, multi-token entities merged. `AFLite` maps each version to an *easy* and *difficult* partition, which can in turn be split into training, dev, and test subsets. We report results for the `fastText` classifier trained on the original, hypothesis-only (hypothesis + premise) MedNLI training set, and evaluated on the *full*, *easy* and *difficult* dev and test subsets of  $\mathcal{X}_{h,m}$  ( $\mathcal{X}_{ph,m}$ ), and observe that performance decreases on the *difficult* partition:

	model	eval dataset	full	easy ( $\Delta$ )	difficult ( $\Delta$ )
<b>no premise</b>	majority class	dev	0.33	0.34 (+0.01)	0.35 (+0.02)
<b>no premise</b>	majority class	test	0.33	0.35 (+0.02)	0.37 (+0.04)
<b>no premise</b>	<code>fastText</code>	dev	0.65	0.67 (+0.02)	0.46 (-0.19)
<b>no premise</b>	<code>fastText</code>	test	0.63	0.65 (+0.02)	0.4 (-0.23)
<b>with premise</b>	majority class	dev	0.33	0.45 (+0.12)	0.36 (+0.03)
<b>with premise</b>	majority class	test	0.33	0.48 (+0.15)	0.37 (+0.04)
<b>with premise</b>	<code>fastText</code>	dev	0.53	0.6 (+0.07)	0.43 (-0.1)
<b>with premise</b>	<code>fastText</code>	test	0.51	0.55 (+0.04)	0.4 (-0.11)

Table 3.8: Performance (micro F1-score) for the majority class baseline and `fastText` classifiers, with and without premise, by partition (e.g., *full*, *easy*, *difficult*).



## 3.2.7 Discussion

### 3.2.7.1 MedNLI is Not Immune from Artifacts

In Chapter 3.2, we demonstrate that MedNLI suffers from the same challenge associated with annotation artifacts that its domain-agnostic predecessors have encountered: namely, NLI models trained on {Med, S, Multi}NLI can perform well even without access to the training examples’ premises, indicating that they often exploit shallow heuristics, with negative implications for out-of-sample generalization.

Interestingly, many of the high-level lexical characteristics identified in MedNLI can be considered domain-specific variants of the more generic, class-specific patterns identified in SNLI. This observation suggests that a set of abstract design patterns for inference example generation exists across domains, and may be reinforced by the prompts provided to annotators. Creative or randomized priming, such as Sakaguchi et al. [12]’s use of anchor words from WikiHow articles, may help to decrease reliance on such design patterns, but it appears unlikely that they can be systematically sidestepped without introducing new, “corrective” artifacts.

### 3.2.7.2 A Prescription for Dataset Construction

To mitigate the risk of performance overestimation associated with annotation artifacts, Zellers et al. [115] advocate adversarial dataset construction, such that benchmarks will co-evolve with language models. This may be difficult to scale in knowledge-intensive domains, as expert validation of adversarially generated benchmarks is typically required. Additionally, in high-stakes domains such as medicine, information-rich inferences should be preferred over correct but trivial

inferences that time-constrained expert annotators may be rationally incentivized to produce, because entropy-reducing inferences are more useful for downstream tasks.

We advocate the adoption of a mechanism design perspective, so as to develop modified annotation tasks that reduce the cognitive load placed on expert annotators while incentivizing the production of domain-specific NLI datasets with high downstream utility [116, 117]. An additional option is to narrow the generative scope by defining a set of inferences deemed to be useful for a specific task. Annotators can then map (premise, relation) tuples to relation-satisfying, potentially fuzzy subsets of this pool of useful inferences, or return partial functions when more information is needed.

### 3.2.8 Ethical Considerations

When working with clinical data, two key ethical objectives include: (1) the preservation of patient privacy, and (2) the development of language and predictive models that benefit patients and providers to the extent possible, without causing undue harm. With respect to the former, MedNLI’s premises are sampled from de-identified clinical notes contained in MIMIC-III [111, 112], and the hypotheses generated by annotators do not refer to specific patients, providers, or locations by name. MedNLI requires users to complete Health Insurance Portability and Accountability Act (HIPAA) training and sign a data use agreement prior to being granted access, which we have complied with.

Per MedNLI’s data use agreement requirements, we do not attempt to identify any patient, provider, or institution mentioned in the de-identified corpus. Additionally, while we provide AFLite *easy* and *difficult* partition information for community use in the form of split-example

ids and a checksum, we do not share the premise or hypothesis text associated with any example. Interested readers are encouraged to complete the necessary training and obtain credentials so that they can access the complete dataset [7, 112].

With respect to benefiting patients, the discussion of natural language artifacts we have presented is intended to encourage clinical researchers who rely on (or construct) expert-annotated clinical corpora to train domain-specific language models, or consume such models to perform downstream tasks, to be aware of the presence of annotation artifacts, and adjust their assessments of model performance accordingly. It is our hope that these findings can be used to inform error analysis and improve predictive models that inform patient care.

## Chapter 4: Balancing Exploration & Exploitation in LLM-informed Decision-making

### 4.1 Overcoming Miscalibrated Conversational Priors in LLM-based Chatbots

This work was done in collaboration with Jennifer Neville, Tobias Schnabel, and Adith Swaminathan; see [19].

#### 4.1.1 Introduction

In contrast to their task- or domain-specific predecessors, modern conversational agents have employed large language models (LLMs) to achieve high proficiency levels (i.e., at or exceeding that of humans) in challenging, open-domain settings [118]. The implicit objective for the agent in such settings is to respond to a user in a way that maximizes the user’s utility given their conversation goal(s).

However, humans are often unable or rationally unwilling to fully verbalize (i.e., explicitly state) their goals and preferences for various reasons (e.g., efficiency) and may instead rely on their conversational partner(s) to fill in the gaps [15]. This leads users to issue *under-specified* queries in which the LLM-based chatbot observes only a subset of the preferences and constraints required to provide a high-quality answer – see Figure 4.1 for an example.

**User:** Is {my outfit} {appropriate} for {the event} I'm attending {tonight}?

**Copilot:** I'm happy to help you with your outfit choice for tonight. 😊

- **To answer your question, I need to know what kind of event you are attending tonight.** There are many different types of events, such as conferences, concerts, parties, weddings, etc. Each event may have a different dress code or style that is appropriate for the occasion.
- 📁 For example, if you are going to a **conference**, you may want to wear something professional and formal, such as a suit, a dress, or a blouse and skirt. You can also accessorize with a watch, a necklace, or a scarf.
- 🎵 If you are going to a **concert**, you may want to wear something casual and comfortable, such as jeans, a t-shirt, or a hoodie. You can also show your support for the artist or band by wearing their merchandise or colors
- 🎉 If you are going to a **party**, you may want to wear something fun and festive, such as a cocktail dress, a jumpsuit, or a shirt and pants. You can also add some flair with a hat, a belt, or a pair of earrings.
- 💍 If you are going to a **wedding**, you may want to wear something elegant and respectful, such as a gown, a suit, or a dress and jacket. You can also match the color scheme or theme of the wedding with your outfit.

These are just some examples of different types of events and what you could wear to them. Of course, **there may be variations depending on the specific event details, such as the location, the time, the season, the culture, etc.** Therefore, it is always good to check the invitation or the event website for more information.

**User:** <...reading... ⌚ > // 😞 hm, why didn't you just ask? ❌

Figure 4.1: An example failure where a user's query is under-specified (blue text). Current LLM-based chatbots produce long responses to hedge against uncertainty (purple text). Clarifying the user's context can avert this failure.

Empirically, we observe that under-specification is common: we classified a random subsample of the queries in the OpenAssistant dataset [119] and found that more than 23% of queries posed to LLM-based chatbots today are severely under-specified (see Figure 4.2 and Section 4.1.3.1 for details).

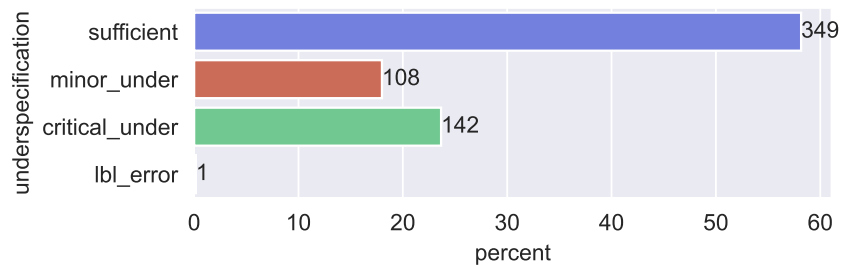


Figure 4.2: Real-world users asked severely under-specified queries more than 23% of the time in the OpenAssistant dataset ( $n = 600$ ).

In this chapter, we explore the relationship between query under-specification, LLM response behavior, and user satisfaction. We begin by proposing a taxonomy of LLM response strategy types (see Table 4.2) to characterize the behavior of SoTA models in the face of query under-specification—i.e., their “conversational priors”— with respect to utility and cognitive cost [120].

Figure 4.3 provides a demonstrative example. Note that each response strategy (a) can be characterized by syntactic and semantic features (i.e., length, presence or absence of conditional statements/questions, etc.) and (b) will give rise to a joint distribution over cost and utility that impose different trade-offs depending on the user’s true but latent preferences.

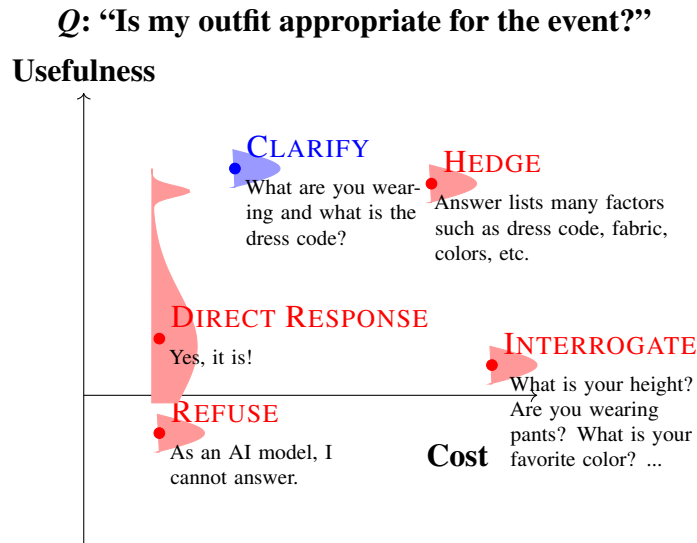


Figure 4.3: For a user query such as  $q$ : “Is my outfit appropriate for the event I’m attending tonight?”, an LLM-based chatbot can choose different response strategies. These strategies produce responses that differ in their cognitive costs (x-axis) while providing final answers with different, user-specific levels of usefulness (y-axis). A good chatbot should respond so as to maximize overall utility—i.e., by providing useful and low-cost answers for the user.

We use this taxonomy and a combination of synthetic and real-world queries to empirically demonstrate that: (a) SoTA LLMs are predisposed to respond directly or hedge in lieu of asking a small number of clarifying questions when queries are under-specified; and (b) such miscalibration can lead to unsatisfactory and/or sub-optimal performance on downstream tasks (as illustrated in Figure 4.1 and Section 4.1.3.2).

To address the miscalibration of LLMs outlined above, we formalize user-chatbot interactions as a partially observable decision process (PODP), where a user with a partially observable goal engages in a turn-by-turn conversation with a chatbot. In this PODP, the chatbot’s policy  $\pi$  is

a fixed mapping from conversation prefixes (which can span multiple turns) to natural language responses. Then, for any given conversation and user goal, the chatbot seeks to provide a natural language response that maximizes utility according to a fixed but unknown user utility function. Note that utility is computed with respect to the user’s *latent* goal, which may be *fully* or *partially* observable via their query.

Intuitively, when the goal is partially observable and the user is amenable to answering a small number of clarifying questions, a policy that produces a natural language response containing questions at timestep  $t_0$  and incorporates the information gained to produce higher-quality responses at future timestep(s) will yield higher expected *cumulative* utility, relative to a myopic policy that tends to respond directly or hedge at  $t_0$ . We build upon this insight to propose two interventions (Sections 4.1.4.1 and 4.1.4.2) to make LLM-based chatbots produce better-calibrated responses in the face of query under-specification. Both of the interventions require only API access to frozen, black-box LLMs.

Our first intervention (Section 4.1.4.1) is inspired by prior research on the generation of clarification questions [18, 121], and uses a static, “clarification-aware” prompt to nudge LLMs to clarify when appropriate rather than reverting to default response behavior. Our second intervention (Section 4.1.4.2) leverages historical conversation logs to learn a meta-policy—i.e., a mapping from conversation prefixes to a finite set of prompts. Then during a PODP episode, the chatbot first invokes this meta-policy, and then calls the LLM with the resulting prompt to produce a contextually appropriate PODP action. We expect the two proposed interventions to be effective in different data regimes — if high-quality logged data is readily available, the approach in Section 4.1.4.2 is a practical alternative to resource-intensive approaches such as fine-tuning LLMs on the collected data. Conversely, if we do not have access to sufficient high-quality data,

we may prefer the data-agnostic approach of Section 4.1.4.1.

In Section 4.1.6, we highlight that our proposed interventions can be further improved—for instance, by incorporating reasoning related to “good” clarification questions to ask (currently left up to the LLM), and the propensity of users to answer with relevant information. Empirically, we evaluate both interventions on recommendation tasks featuring a synthetic user model. We find that each intervention achieves higher expected utility relative to baseline when queries are under-specified, and converges to baseline as query specification increases.

## 4.1.2 Problem Formulation

In the PODP setting that we consider (see Figure 4.4), let  $\theta \in \Theta$  represent a user’s latent, conversation-level *goal*. Each PODP *episode*—i.e., user-chatbot conversation—begins with the user expressing their goal,  $\theta$ , in a potentially lossy manner via a natural language query,  $q \in \mathcal{Q}$ .

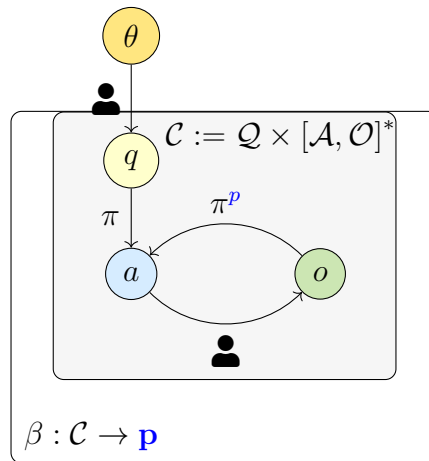


Figure 4.4: PODP plate diagram illustrating user-chatbot interactions, prompt-induced policies ( $\pi^p$ ), and the meta-policy mapping from conversations to prompts ( $\beta$ ).

Per Definition 4.1.1, we consider a query  $q$  to be *under-specified* if there is an information gap between the user’s goal and stated query:



**Definition 4.1.1** (Under-specification). Query under-specification is the partial observability of user’s goal given a query, i.e.,  $\mathbb{P}(\Theta \mid \mathcal{Q})$  is unknown and not deterministic.

Table 4.1 lists some examples of under-specified queries in the OpenAssistant dataset (see Section 4.1.3.1 for details).

Once initiated, a conversation dialogue is assumed to proceed iteratively until terminated by the user (see Figure 4.4). In this context, the chatbot’s natural language responses constitute the *action space* of the PODP and are denoted by  $a \in \mathcal{A}$ , while the user’s follow-up utterances constitute *observations* denoted by  $o \in \mathcal{O}$ . We denote the multi-turn, variable-length *conversation history* between the user and chatbot by  $C := q \times [a, o]^*$ . We use  $\mathcal{C} := \mathcal{Q} \times [\mathcal{A}, \mathcal{O}]^*$  to refer to the space of conversation histories. Then, for any chat conversation  $C$  with user goal  $\theta$ , the task of the chatbot system is to produce actions with maximum utility according to a fixed but unknown *user utility function*,  $\mathcal{U} : \Theta \times \mathcal{C} \mapsto \mathbb{R}$ .

Although the reward function of the PODP,  $\mathcal{U}$ , is unknown we can observe samples from it. For example, many LLM-based chatbots allow users to rate their conversations; these ratings can be directly interpreted as  $\mathcal{U}(\Theta, C)$ . Recent work [122] infers  $\mathcal{U}$  across a user population using a small sample of rated conversations. In general,  $\mathcal{U}$  can rely on a mix of implicit factors, such as response length, and explicit factors, such as thumbs up/down or user ratings of paired responses. Moreover, in Figure 4.3 we saw that the cognitive cost imposed on the user can be another component influencing  $\mathcal{U}$ ; in our experiments, we use response length— $\text{len}(a)$ —as a simple proxy for a user’s cognitive cost from action  $a$ .

We define the *policy*  $\pi$  of a chatbot interacting with a user as a stationary (but not necessarily Markovian) mapping from conversation histories to natural language responses,  $\pi : \mathcal{C} \rightarrow \mathcal{A}$

(Figure 4.4). An optimal chatbot policy is one that maximizes expected utility:

$$\pi^* \approx \arg \max_{\pi} \mathbb{E}_{\{\theta, q\}} \mathbb{E}_{a \sim \pi} [\mathcal{U}(\theta, q, [a, o]^*, a)]. \quad (4.1)$$

In Equation 4.1, note that the policy influences the responses  $a$  in all turns of the conversation, and that  $\mathbb{P}(\Theta, \mathcal{Q})$  is sampled from the user population.

#### 4.1.2.1 Policies Induced By Prompting LLMs

System messages (also known as *prompts*) are often used to “steer” an LLM and induce specific behaviors (e.g.,  $\mathbf{p}$  = “Behave as a helpful assistant”). For LLMs that do not support a separate system message  $\mathbf{p}$ , the prompt and conversation transcript can be concatenated together into the LLM’s input context  $:= \mathbf{p} \circ C$ . Otherwise, PODP policies can be induced by using a prompt  $\mathbf{p}$  and LLM input context  $:= C$ . Such PODP policies are denoted as  $\pi^{\mathbf{p}}$ . If we restrict our attention to the chatbot policies we can access via prompting, we can rephrase the *policy* optimization objective (i.e., Equation 4.1) in terms of *prompt* optimization:

$$\mathbf{p}^* \approx \arg \max_{\mathbf{p}} \mathbb{E}_{\{\theta, q\}} \mathbb{E}_{a \sim \pi^{\mathbf{p}}} [\mathcal{U}(\theta, q, [a, o]^*, a)]. \quad (4.2)$$

When we implement a policy by querying a blackbox LLM API with context  $:= C$  (i.e.  $\mathbf{p}$  is empty), we refer to the induced PODP policy as the RLHF policy,  $\pi^{\text{RLHF}}$ . We can expect good PODP performance out-of-the-box from an LLM only if its RLHF-finetuning guarantees that  $\pi^{\text{RLHF}} \approx \pi^*$  (which is unverifiable).

### 4.1.2.2 Query Under-specification Causes Sub-optimal Interactions

Modern LLMs are typically fine-tuned via RLHF, where the training objective [123] corresponds to:

$$\pi^{\text{RLHF}} \approx \arg \max_{\pi} \mathbb{E}_{\{\theta, q\} \sim \text{lab}} \mathbb{E}_{a \sim \pi} [\mathcal{U}(\theta, q, a)]. \quad (4.3)$$

The combination of query under-specification and RLHF fine-tuning impacts policy learning (i.e., via Equation 4.3) in two ways: (1) distribution shifts between the preferences of annotators and those of end-users may skew the learned policy; and (2) RLHF’s emphasis on annotation of, and optimization over, *single-turn* interactions produces myopic policies that greedily maximize single-turn utility.

With respect to (1), annotators may not be able to reliably infer users’ *true* preferences (i.e.,  $\theta$ ) when evaluating possible responses to user queries—i.e.,  $\mathbb{P}_{\text{lab}}(\Theta | \mathcal{Q}) \neq \mathbb{P}(\Theta | \mathcal{Q})$ . Additionally, the utility function may also shift. For example, Singhal et al. [124] observe that RLHF annotators may prefer longer, more detailed responses relative to end-users.

With respect to (2), the focus on single-turn interactions means annotators are less likely to be exposed to conversations where a chatbot asks the user clarification questions to better understand and respond to the user’s query, because such conversations will, by definition, require multiple turns. In the single-turn setting, annotators may also perceive responses that attempt to answer users’ queries (albeit incorrectly or verbosely) as more *helpful* than responses containing clarification questions. Policy learning with such preferences may thus underestimate the value of uncertainty-reducing behaviors such as clarification, and the resulting policy may be sub-optimal for *multi-turn* conversational outcomes in PODPs. We empirically show that these challenges

render  $\pi^{\text{RLHF}}$  sub-optimal compared to  $\pi^*$ .

### 4.1.2.3 Meta-Policies

When prompting LLMs to produce chatbot responses, we are not limited to using a fixed prompt for all conversation turns. Instead, we can define a meta-policy,  $\beta : \mathcal{C} \mapsto \mathbf{p}$  as a mapping from conversation prefixes to prompts. A PODP agent acting during an episode can first invoke the meta-policy  $\beta$ , and then query the LLM with prompt  $\mathbf{p} := \beta(C)$  to produce its action. For PODP policies implemented through a composition of a meta-policy with an LLM, the original problem of finding a good  $\pi^*$  is replaced with finding a good *meta-policy*  $\beta^*$ :

$$\beta^* \approx \arg \max_{\beta} \mathbb{E}_{\{\theta, q\}} \mathbb{E}_{a \sim \pi_{\mathbf{p}}} [\mathcal{U}(\theta, q, [a, o]^*, a) \mid \mathbf{p} = \beta(q, [a, o]^*)].$$

Note that learning a meta-policy  $\beta$  is a *different* decision-making problem than the PODP decision-making problem (i.e., an action space of prompts instead of chatbot responses).

### 4.1.2.4 Characterizing and Inducing Chatbot Response Behaviors

To empirically evaluate  $\pi^{\text{RLHF}}$  and to design prompt-based interventions, we introduce a taxonomy (detailed in Section 4.1.3.2) that can be used to (1) characterize LLM response behavior; and (2) constrict the meta-policy’s action space. Regarding (1), we refer to the distribution of response strategies of  $\pi^{\text{RLHF}}$  as the LLM’s “conversational prior” (e.g., see Figure 4.7 for GPT-4’s conversational prior).

To build intuition for how this taxonomy may serve both purposes, note that  $\pi^{\text{RLHF}}$  can be

viewed as a hierarchical probabilistic process (Figure 4.5) in which the chatbot first samples a latent response strategy,  $\tau \sim \mathcal{T}$ , and then generates a natural language response conditioned on the response strategy,  $a \mid \tau$ . Then, if  $\pi^{\text{RLHF}}$  is found to be miscalibrated in its distribution over  $\mathcal{T}$ , we can *intervene* via prompts to promote desired response behavior(s).

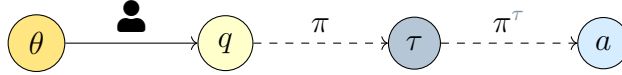


Figure 4.5:  $\pi^{\text{RLHF}}$  as an hierarchical probabilistic process.

We specifically consider a set of response strategies,  $\mathcal{T} = \{\text{REFUSE}, \text{RESPOND}, \text{HEDGE}, \text{CLARIFY}, \text{INTERROGATE}\}$ . To motivate this choice, recall that in the PODP, the chatbot cannot observe the user’s intent,  $\theta$ , and must instead act based on the *belief state*—i.e.,  $\mathbb{P}(\theta \mid q, [a, o]^*)$ . In this context, possible response strategies lie along a spectrum characterized by the relative *absence* or *presence* of (belief)-uncertainty-reducing behaviors (see Figure 4.6).

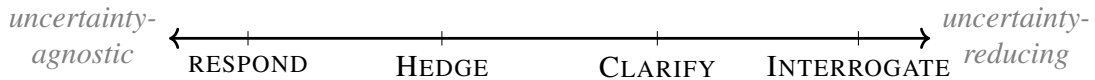


Figure 4.6: Spectrum characterizing the response strategies that a LLM-based chatbot can take. RLHF fine-tuning encourages RESPOND and HEDGE, whereas CLARIFY may be more appropriate when queries are under-specified.

On the *uncertainty-agnostic* end of this spectrum, the chatbot may rely on its inductive prior to *respond directly*—i.e., despite uncertainty about the user’s preferences. Responding directly relies on assumptions and/or potentially spurious semantic correlations between the preferences the user *does* express and those that the LLM-based chatbot must infer. On the *uncertainty-reducing* end, a chatbot may ask an unbounded number of questions before responding (*Interrogate*). This can allow the system to best approximate a user’s fully specified intent but is completely irrational

for the user to engage with. As Figure 4.3 shows, any deviations from the *Respond* response strategy must be done in a thoughtful manner, lest the user have a worse cost-utility benefit even as the system reduces uncertainty in its beliefs.

In a PODP, it is critical to balance information-seeking (exploration) against utility maximization (exploitation). In Section 4.1.3, we demonstrate that  $\pi^{\text{RLHF}}$  places too much weight on response strategies that myopically maximize one-step utility (i.e., RESPOND and HEDGE). In Section 4.1.4.1, we demonstrate that a simple prompt is able to shift the distribution over response strategies toward CLARIFY when queries are under-specified, and thereby improve the PODP policy.

### 4.1.3 Motivating Experiments

Here, we establish that: (1) query under-specification is common in real-world human-chatbot conversations; and (2)  $\pi^{\text{RLHF}}$  can be sub-optimal when queries are under-specified.

#### 4.1.3.1 Query Underspecification is Common

We annotated the OpenAssistant dataset [119] to explore how often users issue under-specified queries to open-domain LLM-based chatbots. We restrict our study to queries in English with at least 3 words ( $\approx 40\%$  of over 10,000 conversations) and subsample 600 queries uniformly at random. We created an LLM-based classifier to map each query to a predicted under-specification label, whose accuracy we also validate on a synthetic corpus (see Appendix E.1.1).

Class labels include:

- **CRITICAL UNDER:** One or more important factors upon which an answer to this query

might depend are not specified or are unknown; it is difficult to provide a high-quality response without knowing these factors.

- **MINOR UNDER:** Less important factors that the query might depend on are not specified or are unknown; however, it is possible to provide a high-quality response even without knowing these factors.
- **SUFFICIENT:** All important factors upon which an answer to this query might depend are sufficiently specified.

Figure 4.2 summarizes the results of this experiment, which shows that query under-specification is prevalent. A few examples of critically under-specified queries are listed in Table 4.1. Note also that many OpenAssistant users have experience with prompting, and we conjecture a higher prevalence of under-specified queries from novice user populations.

<b>Critically Under-specified Queries (Abridged)</b>
Suggest me places near 72nd St where I can park my car.
What are some up and coming and high quality youtube channels in science and technology that I have probably not heard of?
A friend of mine barely talks to me anymore and I don't know why.

Table 4.1: Examples from the OpenAssistant dataset tagged by our classifier as critically underspecified (details in Appendix E.1.1).

#### 4.1.3.2 LLM Policies Can Be Sub-optimal When Queries are Under-specified

When queries are under-specified,  $\pi^{\text{RLHF}}$  has difficulties optimally trading-off information seeking with greedy, utility-maximizing response tendencies. To study this, we define seven broad categories for query responses in Table 4.2. We use these definitions with an LLM-based classifier, which we validate in Appendix E.1.2. Let  $\tau$  be the predicted response type of response  $a$ . Our

experiments show that both for real-world and synthetic queries, the current SoTA LLM, GPT-4, prefers to either directly respond or hedge, instead of clarifying via a short question.

Response type $\tau$	Response characteristics
REFUSE	Contains an explicit or implicit refusal to answer.
DIRECT RESPONSE	No questions or hedging; addresses query.
HEDGE	Many answers, conditioned on uncertain factors.
CLARIFY	Limited/prioritized set of questions (i.e., $\leq 3$ ).
INTERROGATE	Large/exhaustive number of questions (i.e., $> 3$ ).
MISSING	The response is empty/blank.
MISCELLANEOUS	Describes or follows query instructions.

Table 4.2: For the motivating experiments in Section 4.1.3, we categorize LLM responses into seven response types.

**Synthetic query corpus:** The goal for the synthetic corpus is to have a full-information setting where we can explicitly control the degree of under-specification and measure the utility of any given response. We generate queries for three different recommendation domains (movies, gifts, plants) that each have four constraint dimensions  $\theta_i$  that can be active (set to a specific value, e.g.,  $\theta_{age} = \text{“25-35 years”}$ ), or inactive, (e.g.,  $\theta_{age} = \emptyset$ ). We base this setup on Radlinski et al. [125], who studied users’ preferences for movies expressed in a conversational recommendation setting. The user goal is then to get a recommendation that satisfies *all* of these constraints. Constraint values and the number of active dimensions are sampled via uniform sampling. After determining the ground truth user goal  $\theta$ , we generate a potentially under-specified user query by sampling a subset of active constraint dimensions to reveal. With a slight abuse of notation, let  $q$  be the vector of revealed active constraints. We categorize the resulting queries as:

$$q \mapsto \begin{cases} \text{CRITICAL UNDER} & |q| \leq 1, \\ \text{SUFFICIENT} & |q| = |\theta|, \\ \text{MINOR UNDER} & \text{otherwise.} \end{cases} \quad (4.4)$$



Details can be found in Appendix E.3.1.

**Sub-optimality of LLM in single-step interaction:** For each query, we use GPT-4 with the default system message to generate a natural language response  $a \sim \pi^{\text{RLHF}}$  and assign it a response type label  $\tau$  from Table 4.2 using our LLM-based classifier. Figure 4.7 shows the distribution over response strategies (by corpus and under-specification severity) for  $\pi^{\text{RLHF}}$ .

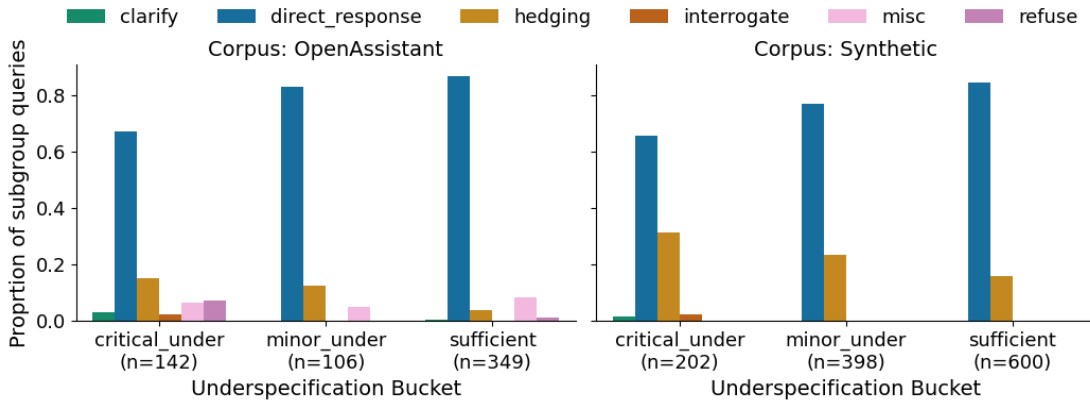


Figure 4.7: Even under severe levels of under-specification, GPT-4 prefers to directly answer a user’s query.

We observe that for both synthetic and real-world queries, using the uncertainty-agnostic DIRECT RESPONSE strategy is preferred by a large margin across *all* under specification buckets. While there is evidence that uncertainty-aware response strategies (i.e., HEDGE, CLARIFY, and INTERROGATE) are increasingly used when under-specification rises, the sheer magnitudes still express a clear bias for  $\pi^{\text{RLHF}}$  to respond or hedge—rather than clarify—in the face of under-specification. This indicates that there is headroom to improve utility even over SoTA LLMs.

**Sub-optimality of LLM in multi-step interactions:** Intuitively, a policy asking a few relevant questions in the beginning should be able to outperform  $\pi^{\text{RLHF}}$  in many cases since  $\pi^{\text{RLHF}}$  often defaults to DIRECT RESPONSE. The following two-step recommendation task shows this. We compare  $\pi^{\text{RLHF}}$  with two simple static policies described in Table 4.3. We use modified system

messages to encourage different behavior for the first response, and follow  $\pi^{\text{RLHF}}$  as the default policy after (for full prompts, see Appendix E.2).

Policy $\pi^{\text{P}}$	System prompt $\text{p}_0$
$\pi^{\text{RLHF}}$	Default LLM system message (unmodified).
$\pi^{\text{Clarify}}$	Ask about $\leq 3$ of <i>most relevant</i> factors.
$\pi^{\text{Hedge}}$	Condition on option(s) for each uncertain factor.

Table 4.3: We evaluated three different policies that encourage different initial response strategies to show the possible room for improvement in multi-step interactions.

We use the queries and ground truth user goals from the synthetic query corpus outlined in Section 4.1.3.2, but focus on the movie domain only, following Cheng et al. [126]. Each episode begins at  $t = 0$  (denoted  $t_0$ ) with the user issuing query  $q$  to ask for movie recommendations that satisfy their true preferences  $\theta$ . When the LLM-based chatbot provides recommendations (i.e., chooses action types DIRECT RESPONSE or HEDGE), we terminate the episode and compute the utility of the recommendation. If the chatbot asks questions, we use another LLM as a user simulator, requiring the latter to divulge information in a templated format about constraints  $\theta_i$  *only* if explicitly asked (see Appendix E.3.4).

**Item Utilities.** We begin by measuring the utility of items recommend by each  $\pi$ , operationalized as the fraction of constraints (out of 4) that an item satisfies, averaged across all items recommended to the user. Instead of comparing individual policies, we compare response types  $\tau$  to eliminate cases where setting the system message  $\text{p}$  did not induce the desired response type. Figure 4.8 shows how multi-step episode utilities develop when we group by the type of the first system response,  $\tau_0$ .

CLARIFY does not generate any utility at time  $t_0$ , since no recommendations have been made, but does much better in the second time step  $t = 1$ , especially for critically under-specified

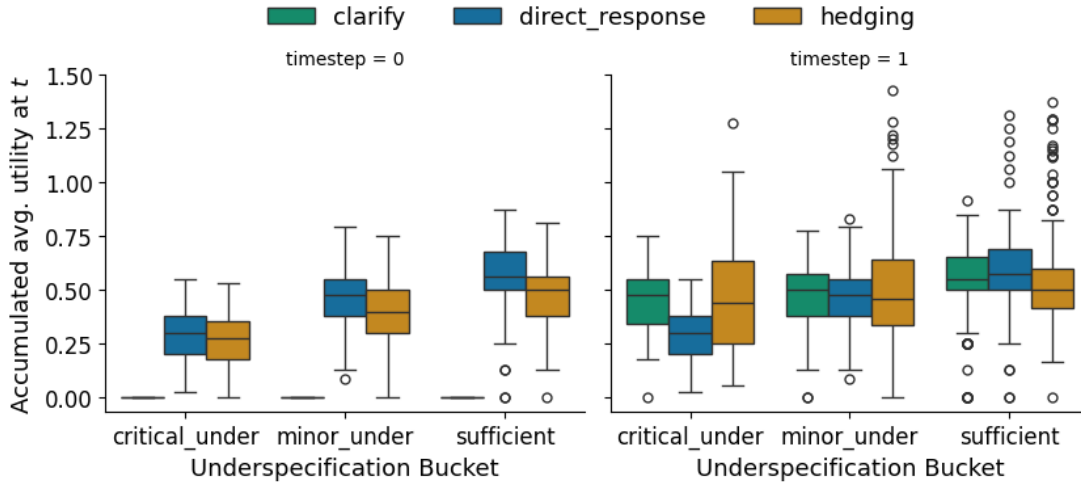


Figure 4.8: Distribution of accumulated item utilities  $\mathcal{U}$  at timesteps  $t = 0, 1$ ; grouped by under-specification levels.

queries. When we HEDGE in the beginning, we do get utility at  $t_0$ , but generate less than when we directly reply, since utility is averaged over *all* (possibly irrelevant) recommendations. These findings suggest that there is headroom for improvement over  $\pi^{\text{RLHF}}$  in multi-step interactions.

**Costs.** We now consider how the *cost* of capturing this headroom—i.e., moving from an under-specified query to a more fully specified version—varies over the uncertainty-aware strategies that we consider—i.e., HEDGE and CLARIFY. To proxy for the cognitive burden associated with reading and answering clarification questions or parsing the many cases or conditions mentioned in hedging responses, we define a cost function,  $c : \mathcal{A} \rightarrow \mathbb{R}_{\geq 0} := \text{len}(a)$  (measured by counting all unigrams in  $a$ ).

Figure 4.9 illustrates the benefit of CLARIFY – it carries a relatively low cost in terms of output length. Interestingly, the DIRECT RESPONSE action produces the longest answer among all other response types when queries are critically under-specified. Inspecting the produced responses, we see that DIRECT RESPONSE produces long answers by adding explanations or extended lists of recommendations. When queries are sufficiently specified, HEDGE leads to

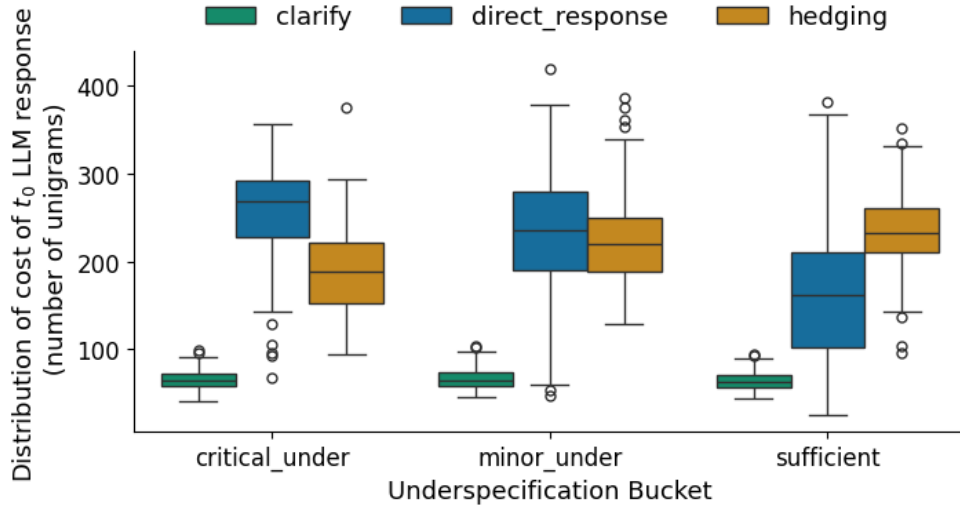


Figure 4.9: Distribution of response cost at  $t = 0$  for each response strategy  $\tau_0$ ; grouped by under-specification levels.

the highest cost answers, as it still enumerates over many answer options. Overall, we see that CLARIFY obtains the lowest average cost across *all* under-specification buckets, suggesting that a policy could achieve higher utility with lower costs by considering the CLARIFY action more often.

#### 4.1.4 Algorithmic Approach

In this section, we outline two algorithmic interventions to improve upon  $\pi^{\text{RLHF}}$  in PODPs. The first intervention uses a fixed prompt to an LLM-based policy  $\pi^{\text{P}}$  that nudges the LLM to prefer cost-aware uncertainty-reducing response strategies like clarifications when appropriate. We saw in Section 4.1.3 that this *data-agnostic* approach can be substantially better than  $\pi^{\text{RLHF}}$  when queries are under-specified and users patiently respond to all clarifications. However, real-world users may have varying propensities to engage with clarifying questions. So, we devise a second intervention in Section 4.1.4.2 that uses historical conversational logs to fit an appropriate *meta-policy*  $\beta$  that can be more optimal for the PODP.

#### 4.1.4.1 Data-Agnostic Interventions

We saw in Section 4.1.3 that LLM-based chatbots have sufficient capabilities at *detecting* under-specified queries (Section 4.1.3.1) and generating CLARIFY responses if prompted explicitly (Section 4.1.3.2). However, they do *not* appear to sufficiently condition on their latent under-specification judgments when generating responses in the absence of intervention (i.e., when relying on the baseline system message in  $\pi^{\text{RLHF}}$ ). Thus, we consider two approaches that explicitly emphasize the possibility of under-specification and the benefits of clarification when appropriate and allow graceful recovery of default system behavior when warranted—e.g., when queries are well-specified.

**Approach 1: Chain of Thought (CoT):** We evaluate a chain-of-thought [127] intervention in the form of a modified system message that encourages the LLM-based chatbot to “ask yourself whether you have sufficient information to provide a good answer, and then respond accordingly” when responding to queries (see Appendix E.2).

**Approach 2: Clarify When Appropriate (Clarify-Flex):** We also evaluate a more flexible, context-aware relaxation of the “always clarify” system message that we experimented with in Section 4.1.3.2. This modified system message instructs the LLM-based chatbot to ask clarifying questions about important factors only *if* they have not been specified, and to respond directly otherwise (see Appendix E.2).

**Key Findings and Limitations:** In order to compare our data-agnostic interventions to  $\pi^{\text{RLHF}}$ , we conduct a slightly modified version of the two-step recommendation experiment presented

in Section 4.1.3.2. Here, we consider  $\mathbf{p}_0$  values  $\in \{\text{BASELINE}, \text{COT}, \text{CLARIFYFLEX}\}$ , and sequential combinations  $\in \{(\mathbf{p}_0, \mathbf{p}_1) \mid \mathbf{p}_1 = \mathbf{p}_0 \vee \mathbf{p}_1 = \text{BASELINE}\}$ .

We begin by using our LLM-based  $\tau$ -classifier to map each intervention to a distribution over response strategies, so as to assess the extent to which highlighting uncertainty and encouraging contextual awareness at response generation time induces changes in response behavior relative to baseline. As Figure 4.10 illustrates, while the COT intervention behaves quite similarly to the BASELINE, CLARIFYFLEX meaningfully diverges, favoring *interrogation* when queries are critically under-specified, then shifting toward *clarify*, and finally toward *direct response* (i.e., converging with BASELINE) as the degree of specification increases.

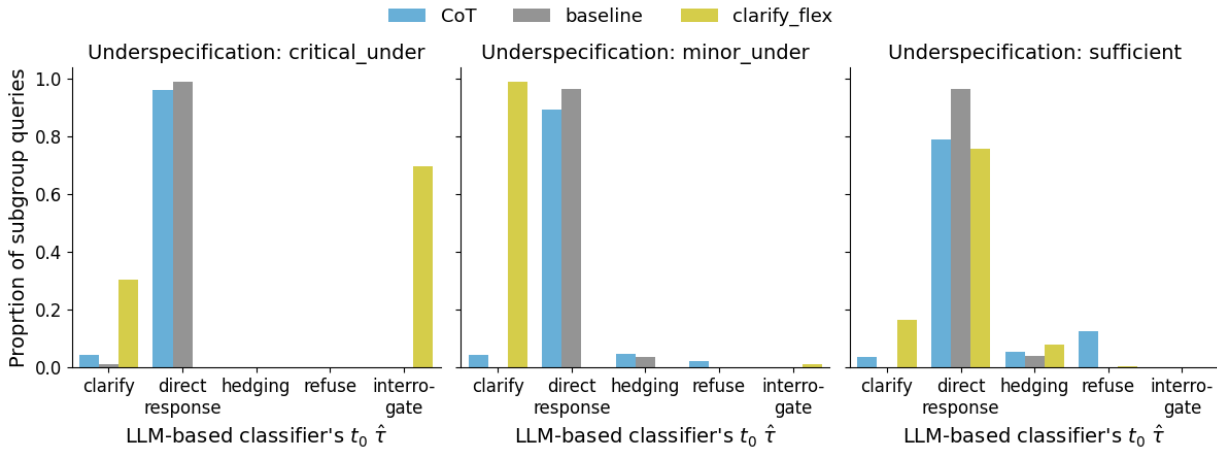


Figure 4.10: Distribution of the response strategies  $\hat{\tau}_0$  induced by the three prompts =  $\{\text{BASELINE}, \text{COT}, \text{CLARIFYFLEX}\}$ ; grouped by underspecification levels.

Next, we examine the distribution over the average utility of recommended items for each sequential combination of  $(\mathbf{p}_0, \mathbf{p}_1)$ . As Figure 4.11 illustrates, (CLARIFYFLEX, BASELINE) is the best-performing sequential combination when queries are critically under-specified, with relative advantage diminishing as specification increases. When queries are sufficiently specified, (CLARIFYFLEX, BASELINE) and (COT, BASELINE) obtain slightly higher median  $\bar{U}$  than

(BASELINE, BASELINE), but we generally see convergence due to the fact that both baseline and interventions tend toward direct response in this setting.

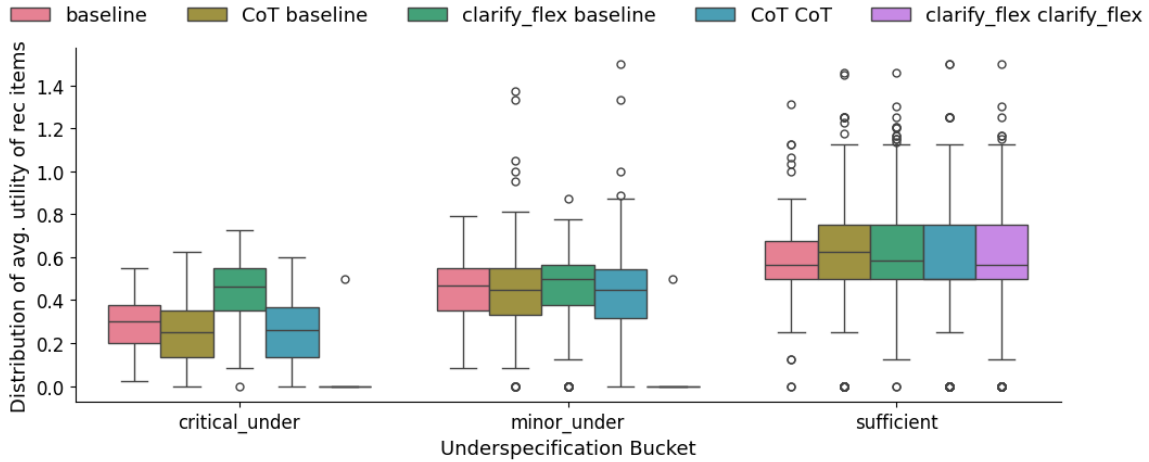


Figure 4.11: Distribution of  $\bar{U}$  for each  $(\mathbf{p}_0, \mathbf{p}_1)$  sequence; grouped by underspecification levels.

From this analysis, we conclude that among the data-agnostic interventions we consider, CLARIFYFLEX is best able to improve upon the baseline  $\pi^{\text{RLHF}}$  when queries are critically underspecified, while maintaining the flexibility to converge to *direct response* as specification increases. In summary, through Figures 4.10 and 4.11, we see that in a synthetic user model (that provides templated answers to clarification questions), it is possible to improve upon the performance of the baseline LLM—i.e., CLARIFYFLEX performs better than  $\pi^{\text{RLHF}}$  when evaluated in the PODP.

#### 4.1.4.2 Data-Based Intervention

Here, we introduce an intervention that leverages collected conversation logs to learn *when* and *how* to improve upon  $\pi^{\text{RLHF}}$ —i.e., by redistributing probability mass away from uncertainty-agnostic *direct response* and cost-agnostic *hedging* toward cost- and context-aware response strategies such as *clarify* when appropriate—in a way that is more tunable and adaptive to different

user populations than the data-agnostic interventions we consider in Section 4.1.4.1.

We begin by considering meta-policies  $\beta$  as described in Section 4.1.2.3. Remember that learning a mapping  $\beta : \mathcal{C} \mapsto \mathbf{p}$  is a *different* decision-making problem than the original PODP policy. As described in Section 4.1.2.4, we will use the taxonomy we developed in Table 4.2 to reduce the action space of the meta-policies. Given a  $\mathcal{T}$  with corresponding prompts  $\mathbf{p}_\tau : \tau \in \mathcal{T}$ , we consider the restricted set of meta-policies  $\beta : \mathcal{C} \mapsto \mathcal{T}$ . A PODP agent using  $\beta$  will, at each timestep, first calculate  $\hat{\tau} = \beta(\mathcal{C})$ , look up the corresponding prompt  $\mathbf{p}_{\hat{\tau}}$  and finally query the LLM with  $(\mathbf{p}_{\hat{\tau}}, \mathcal{C})$  to produce an action in the PODP.

Conceptually, if we had the ability to simulate the PODP environment, then we could learn a meta-policy  $\beta$  through online Reinforcement Learning (RL): i.e., sample prompts at each turn in the conversation from the current  $\beta$ , observe the resulting conversation-level outcomes, and update the parameters of  $\beta$  using e.g., PPO. However, we typically cannot simulate user-chatbot conversations with high fidelity, and running online RL with users directly can be very sample inefficient and result in a poor user experience.

Instead, we use an offline approach inspired by Asymmetric Imitation Learning [128]. We assume access to a dataset  $\mathcal{D}$  containing logs of user-chatbot dialogues along with conversation-level utility ratings,  $\mathcal{D} = \{(C_1, U_1) \dots (C_n, U_n)\}$ . Such a dataset can be collected, for example, from an already deployed chatbot. Notice that the data contains signals about the true  $\theta_i$  (i.e.  $U_i := \mathcal{U}(\theta_i, C_i)$ ) beyond what can be inferred from  $C_i$ , but the learner  $\beta$  does not have access to  $\theta_i$ . Hence, imitating optimal actions in  $\mathcal{D}$  reduces to asymmetric imitation learning.

We use the  $\tau$ -classifier developed in Section 4.1.3.2 to annotate all of the chatbot responses in  $\mathcal{D}$  with their response strategy  $\hat{\tau}$ . Consequently, we can estimate a Q-value function  $Q(C, \hat{\tau})$  on



the annotated data as:

$$\hat{Q} = \arg \min_Q \sum_{i \in \mathcal{D}} \sum_{a_j \in C_i} (Q(C_i[: a_j], \hat{\tau}(a_j)) - U_i)^2,$$

where  $C[: a]$  denotes the conversation prefix up to the chatbot response indicated by  $a$ . The Q-value function  $\hat{Q}(C, \tau)$  estimates the eventual utility the learner will receive if we take action  $\tau$  upon observing conversation  $C$  and then follow the baseline system (i.e.,  $\pi^{\text{RLHF}}$ ) at all future timesteps.

When new conversations arrive, we evaluate the predicted  $Q$  values for each  $\tau \in \mathcal{T}$  and choose the argmax:

$$\beta(C) = \arg \max_{\tau \in \mathcal{T}} \hat{Q}(C, \tau). \quad (4.5)$$

We empirically evaluate this Q-value estimation approach in the synthetic recommendation experiment. We operationalize reward as the average utility (i.e., alignment between an item’s features and the user’s true preferences) over the set of recommended items. In the synthetic setup, we can generate responses (and eventual conversation rewards) for all possible  $\tau \in \mathcal{T}$  for each query seen in the dataset  $\mathcal{D}$ . So we compute  $Q^*$  for all queries seen in  $\mathcal{D}$ . However we need to estimate  $Q$  for new queries as they arrive so as to implement Equation 4.5.

We construct a regressor to estimate  $Q^*$  as follows: we use a pre-trained SentenceTransformer model [129] to encode a stratified sample of our synthetic corpus (we stratify by the degree of under-specification so that the resulting distribution over labels mimics the OpenAssistant results we report in Figure 4.2).

Then, for new conversation histories, e.g.  $q$ , we encode it using the same embedding

model and retrieve its  $k$ -nearest neighbors, with  $k = 5$ . We then retrieve each neighbor’s  $Q^*$  and corresponding  $\tau$ . We can then predict the Q-value of each candidate  $\tau$  as the average of the  $Q^*(\tau)$  values contributed by neighbors. This is akin to an asymmetric imitation learning baseline [130]. We greedily choose the  $\text{argmax } \tau$  at  $t_0$ , simulate user answers to LLM responses containing questions as in Section 4.1.3.2, follow  $\pi^{\text{RLHF}}$  at  $t_1$ , and report the resulting episode-level rewards (i.e., average utility over items in the rec set). We present empirical results for this approach in Figure 4.12, and observe that our learned meta-policy achieves higher reward relative to baseline. The empirical results demonstrate that both strategies we evaluate—i.e., *designing good prompts* (Section 4.1.4.1), and *learning meta-policies* (Section 4.1.4.2) can be better than  $\pi^{\text{RLHF}}$ . We observe in Figure 4.12 that the meta-policy is slightly preferred over CLARIFYFLEX, however this ordering may not be universal: when historical data is not representative of future conversations, we may prefer CLARIFYFLEX over learning a meta-policy.

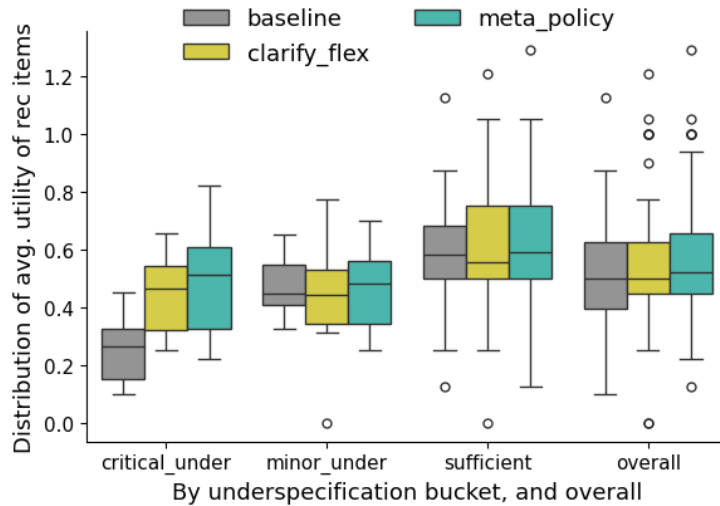


Figure 4.12: Our learned meta-policy outperforms baseline across all underspecification buckets, especially when queries are critically under-specified. And it converges to baseline when queries are sufficiently specified.

### 4.1.5 Related Work

Even though LLMs are powerful conversationalists and recommenders [131], they have many failure modes [132] such as generating hallucinations or failing to complete more complex reasoning tasks [133] (Section 8). Regarding LLM-powered conversations that require stronger collaboration between two parties, Lin et al. [134] introduce the concept of “decision-oriented dialogues” and show that current LLMs still are far from human performance. In this chapter, we investigate a specific cause (query under-specification) and show how we can improve LLMs for them.

We conjecture that query under-specification is an artifact introduced or amplified during post-training and alignment workflows such as reinforcement learning from human feedback (RLHF) [123]. In RLHF, LLMs are fine-tuned to output results that align with the preferences of annotators. Status quo approaches focus on pairwise comparisons of single-step responses to a given input query. As such, well-specified and/or simpler queries that admit multiple possible, high-quality responses *without* the need for clarification questions may be over-represented during fine-tuning. Additionally, when annotators *do* encounter under-specified queries, their preferences about how to handle ambiguity may differ in meaningful ways from those of end-users, skewing the learned policy. For example, Singhal et al. [124] observe that annotators tend to prefer longer responses—which help to “cover all bases” when queries are under-specified—relative to end-users, who must bear the cognitive cost of LLM verbosity. Annotators may also provide feedback they feel is “expected” of them that diverges from their true conversational preferences (due to the Hawthorne effect; see [135]).

Query under-specification has been studied and addressed in information retrieval [136, 137].

There are two broad approaches: algorithmic or user-centric techniques. Algorithmic approaches include query expansion [137], query reformulation [136] etc. User-centric approaches focus on asking good clarifications [18, 121]. Hybrid approaches are possible: for instance, Diao et al. [138] use active learning to determine what questions to ask in an LLM’s context window so as to improve its reasoning. We take a user-centric approach of seeking clarification, and rely on a suitably prompted LLM (rather than a separate active learning policy) to discover appropriate questions to ask.

We showed that LLMs are misaligned when queries are under-specified. Others have shown misalignment for other reasons (e.g. toxicity [139]) and studied better ways to align LLMs. There are two kinds of approaches to align LLMs better: fine-tuning (e.g., DPO [140], KTO [141], RLHF [123], etc.) and prompt injection (e.g., Constitutional A, I [139], meta-prompting [142]). We take the latter approach and extend the meta-prompting of Qin and Eisner [142] to work not only with soft-prompts but with natural language prompts and black-box LLMs.

Our proposed interventions rely on asking users clarification questions. User studies conducted with search engines [143] and pre-LLM conversation systems [144] demonstrated that users *do* engage with clarifying questions in those contexts. Conducting user studies in LLM-based chatbots to assess users’ propensity to answer questions is an exciting avenue for future work.

We frame the conversation between a user and chatbot as a PODP, which is mathematically equivalent to a partially observable Markov Decision Process (POMDP) [145]. Others have framed the interactions as multiple rounds of bandit interactions [146], but as we argued before, single-turn utility maximization is too myopic for multi-turn conversational outcomes. Thus, we adapt solution concepts from POMDP like Q-learning [147], information-gathering [148] for use with LLM-induced policies.

### 4.1.6 Limitations

While our empirical results demonstrate that both of our proposed interventions improve expected conversation-level utility when queries are under-specified, it is worth noting some limitations associated with the way we have modeled user-chatbot interactions. First, we note that our model relies on the assumption that users are both *willing* and *able* to answer clarification questions when asked—that is, that they will (1) “tolerate” the questions with high probability (i.e., will not defect by exiting the conversation), and (2) truthfully reveal their preferences. In practice, the propensity and ability to answer will vary among users and over query intent domains (e.g., due to personal preferences, epistemic uncertainty regarding a specific topic, etc.).

In our empirical results, the optimistic nature of these assumptions is offset by the conservative nature of the information gain we consider: oftentimes, LLM questions will ask for more granularity about already-revealed  $\theta$ s, and while real users would often be able to provide such detail, our lossy, parameterized approximation cannot. As such, any improvement in expected reward associated with sequential response strategies that incorporate uncertainty reduction at  $t_0$  may be underestimated. We have focused on undiscounted expected utility maximization, but the incorporation of a discount rate would be one way to incorporate heterogeneity with respect to question tolerance. Human validation of our proposed interventions will also be critical: while the interventions are well-motivated from an information-theoretic perspective, for some users, the marginal improvement in expected utility may not outweigh the cognitive cost associated with having to answer questions.

Additionally, we note that while we have relied on helper LLMs to classify queries and responses (i.e., with respect to under-specification, and response strategy), human validation

of these classifiers is an important next step. We have considered a relatively restricted intent domain, but in more general settings, reasonable annotators may disagree about whether a query is under-specified when they do not have access to ground-truth  $\theta$ . Relatedly, we have focused on a recommendation setting (i.e., movie recs) that admits objective computation of utility; extension of our approach to intents characterized by more subjective evaluation criteria may require alternative approaches to modeling utility.

In the data-based intervention outlined in Section 4.1.4.2, we have assumed that historical conversation logs are representative of the user population and joint distribution over users and queries seen in the online setting. This assumption may be violated in practice, with potentially negative consequences for meta-policy performance. Our estimates regarding the prevalence of query underspecification may also contain artifacts—e.g., due to small sample size, and non-stationarity of the user population.

Finally, we have made assumptions regarding the prompt-based steerability of LLMs, along with the ability of LLMs to select “good” clarification questions when prompted to clarify. Empirical validation of these assumptions on a broad set of LLMs, along with studying the generation and selection of marginal information-gain maximizing questions, are important directions for future work.

#### 4.1.7 Summary of Contributions

This chapter explores how user underspecification affects the behavior of LLM-based chatbots that are fine-tuned with human feedback. We show that chatbots have difficulty handling vague user requests and explain how this issue stems from the annotation process of LLMs. Our

study of a public chat logs dataset confirms that this problem is common – over 25% of the queries are highly underspecified. We formulate the problem of underspecification as a partially observable decision process (PODP) and generate synthetic data from a recommendation scenario with hidden item values for experimental evaluation. Our experiments show that pre-trained LLMs perform poorly on underspecified user queries and propose a method to adjust LLMs through prompting (with learned control messages). We demonstrate that our lightweight learning method can effectively leverage previous conversation data to improve the response behavior of LLM-based chatbots for recommendation tasks.

## Chapter 5: Conclusion & Future Work

### 5.1 Summary of Contributions

In the preceding chapters, we have considered several instances of algorithmic decision-making from the perspective of a decision-maker who must: (1) contend with partial observability; (2) incorporate problem structure; (3) strike an appropriate balance between uncertainty-reducing (or fairness-promoting) exploration and reward-maximizing exploitation; and (4) determine the extent to which model outputs and behaviors are generalizable and desirable—versus upwardly biased or myopic—for the purpose of identifying subgroups or domain-specific reasoning tasks at risk, and proposing corrective interventions. We summarize our core contributions below.

In Chapter 2, we consider two sequential decision-making tasks framed as restless multi-armed bandit (RMAB) problems, in which we must select a strict subset of individuals (i.e., arms) to receive a health-promoting intervention at each timestep. In Chapter 2.2, we empirically demonstrate that asymptotically optimal, Whittle-index-based solutions to this problem produce bimodal distributions, in which a small subset of arms repeatedly receive the intervention while the majority of arms are ignored. To improve algorithmic acceptability while preserving utility, we introduce PROBFair, a probabilistically fair policy that maximizes total expected constrained reward and satisfies the budget constraint while ensuring a configurable, strictly positive lower bound on the probability of being pulled at each timestep.



Next, in Chapter 2.3, we begin by observing that while prior work in the RMAB space assumes that arms only benefit if they receive the resource directly, many real-world allocation tasks *occur within communities* and can be characterized by positive externalities that allow arms to derive partial benefit when their neighbor(s) receive the resource. We introduce *networked restless bandits*, a novel multi-armed bandit setting in which arms are both restless *and* embedded within a directed graph. We then present GRETA, a graph-aware, Whittle index-based heuristic algorithm used to efficiently construct a constrained reward-maximizing action vector at each timestep.

In Chapter 3, we shift our focus from algorithm development to model evaluation. The specific evaluation tasks we consider are motivated by prosocial concerns such as algorithmic fairness and robustness (i.e., with respect to out-of-sample observations in domains such as health-care where model predictions may inform high-stakes real-world decisions). In Chapter 3.1, we consider disaggregated evaluation. In this setting, the objective is to estimate model performance metrics for intersectional subgroups of interest in order to identify subgroups that may be at risk of experiencing algorithmic harms. We propose a structured regression-based approach that can be used to accurately estimate model performance, even for intersectional subgroups with small sample sizes.

Next, in Chapter 3.2, we focus on model evaluation issues that may arise as unintended consequences of the training data generation process. We specifically consider MedNLI, a physician-annotator-constructed clinical natural language inference dataset, and demonstrate how the priming effects of annotation examples and/or annotators' use of domain-specific heuristics to expedite the generation of logically sound (premise, hypothesis) pairs may lead to upwardly biased performance estimates and poor generalizability for models trained on this dataset.

Finally, in Chapter 4, we consider LLM-based conversational recommendation. This setting is unique because it combines aspects covered in earlier chapters: first, the LLM-based chatbot must “decide” which action (i.e., natural language response) to produce at each turn of the conversation. In practice, this form of “decision-making” is operationalized by querying a black-box LLM that has been trained on web-scale data and fine-tuned via single-turn reinforcement learning from human feedback (RLHF). When we evaluate the default response behavior of state-of-the-art LLM-based chatbots, we observe a myopic bias toward direct response or hedging in the face of query underspecification. This insight informs our prompt- and meta-policy-based interventions, which improve expected conversation-level utility relative to baseline by encouraging context-aware clarification questions when appropriate, and allowing for subsequent exploitation of the knowledge gained.

## 5.2 Future Research Directions

Looking toward the future, as LLMs are incorporated into a broader set of decision-making, feature engineering, knowledge discovery, and information retrieval contexts, it will become increasingly important to: (1) characterize the fine-tuning-induced “default” behaviors of LLMs across a wide range of goal-directed use cases; (2) identify areas where LLMs’ inductive priors (and the outcomes induced by their default behaviors) meaningfully diverge from the partially observable and often heterogeneous preferences of human stakeholders; and (3) develop corrective interventions that incorporate external, domain-specific knowledge and principles, encourage grounded uncertainty quantification and calibration, and preserve rationality of participation by preventing mode collapse and amplification of existing harms.

## Appendix A: Probabilistically Fair Allocation in the Restless Bandit Setting

### A.1 Notation

In Table [A.1](#), we present an overview of the notation used in Chapter [2.2](#).  $[N]$  denotes the set  $\{1, 2, \dots, N\}$ .

Table A.1: Notation used in Chapter 2.2, with notes on interpretation.

<b>MDP Variables</b> Here, timestep $t \in [T] = \{1, 2, \dots, T\}$ (subscript) and arm index $i \in [N] = \{1, 2, \dots, N\}$ (superscript) are implied.		
State space	$s \in \mathcal{S} = \{0, 1\}$	$s = \begin{cases} 1 & \text{arm is in the 'good' state.} \\ 0 & \text{else} \end{cases}$
Belief space	$b \in \mathcal{B} = [0, 1]$ $b_{t+1} = \begin{cases} s_{t+1} & \text{if known} \\ b_t P_{1,1}^0 + (1 - b_t) P_{0,1}^0 & \text{else} \end{cases}$	If an arm's true state is unknown, the recursively computed belief state approximates it.
Action space	$a \in \mathcal{A} = \{0, 1\}$	$a = \begin{cases} 1 & \text{pull arm (i.e., provide intervention)} \\ 0 & \text{else, don't pull} \end{cases}$
<b>MDP Functions</b>		
Transition function	$P: \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ $s_t, a_t, s_{t+1} \mapsto \mathbb{P}(s_{t+1} \mid s_t, a_t)$	The probability of an arm going from state $s_t$ to $s_{t+1}$ , given action $a_t$ . Equivalent (matrix) notation: $P_{s_t, s_{t+1}}^{a_t}$ .
Reward function	$r: \mathcal{S} \text{ or } \mathcal{B} \rightarrow \mathbb{R}$	$r(b)$ is used in computing the Whittle index.
Policy function	$\pi: \mathcal{S} \rightarrow \mathcal{A}$	A policy for actions. The set of optimal policies is $\pi^* \in \Pi^*$ .
<b>RMAB Variables</b>		
Timestep	$\{t \in \mathbb{N} \mid t \leq T\}$	This timestep is implicit in the MDP.
Arm index	$i \in \{1, 2, \dots, N\}$	Each arm can represent a patient. $k$ arms can be pulled at any timestep $t$ .
<b>Objective Functions</b> The objective is to find a policy $\pi^* = \max_{\pi} \mathbb{E}_{\pi}[R(\cdot)]$ .		
Discounted reward function	$R_{\beta}^{\pi}: \mathcal{S}^N \rightarrow \mathbb{R}$ $s_0^1, s_0^2, \dots, s_0^N \mapsto \sum_{i \in [N]} \sum_{t \in [T]} \beta^{t-1} r(s_t^i)$	$\beta \in [0, 1]$ is some <i>discount parameter</i> .
<b>Fairness-motivated Constraint Functions</b>		
Integer periodicity	$\bigwedge_{j=0}^{\lceil \frac{T}{\nu} \rceil} \sum_{t=j\nu+1}^{(j+1)\nu} a_t^i \geq 1$	A form of time-indexed fairness. Guarantees arm $i$ is pulled at least once within each period of $\nu$ timesteps.
Minimum selection fraction	$\bigwedge_{i \in [N]} \frac{1}{T} \sum_{t=1}^T a_t^i \geq \psi$	A form of time-indexed fairness. Arm $i$ should be pulled at least some minimum fraction $\psi \in (0, 1)$ of timesteps.
Probabilistic	$\bigwedge_{i \in [N]} \bigwedge_{t \in [T]} \mathbb{P}(a_t^i = 1 \mid i, t) \in [\ell, u]$	Pull each arm with probability $p_i \in [\ell, u]$ , where $\ell \in (0, \frac{k}{N}]$ and $u \in [\frac{k}{N}, 1]$ .

## A.2 Empirical Inequity in the Distribution of Actions under Whittle Index Policies

Here, we present numerical results confirming Prins et al. [24]’s findings that THRESHOLD WHITTLE (TW) tends to allocate pulls according to a bimodal distribution: a small subset of arms are pulled frequently, while others are largely ignored.

**Experimental Setup:** For each iteration, we generate  $N = 2$  forward threshold-optimal arms and run TW for a  $T = 365$  horizon simulation, where the budget constraint  $k = 1$ . We run 1,000 such iterations.

**Results:** In 515 out of 1,000 (51.5%) simulations, the arms’ Whittle indices never overlap, meaning that for *any* combination of initial states, state transitions, and pulls, TW would pull one arm for all timesteps  $t \in T$  and completely ignore the second arm. We visualize one such case in Figure A.1.

Recall that TW precomputes the infimum subsidy  $m$  per arm and belief combination. Since belief is a function of last known state  $s \in \{0, 1\}$  and time-since-seen  $u \in [T]$  (using the notation of Mate et al. [20]), we plot the infimum subsidy of each arm-state combination with time-since-seen,  $u$ , on the  $x$ -axis. There exists a horizontal line that divides the two arms, so arm  $i = 2$  will be pulled for every timestep and arm  $i = 1$  will never be pulled.

In order to modify the Whittle index to guarantee time-indexed fairness constraint satisfaction, one would need to ensure that no such horizontal line exists. Additionally, if we consider a specific form of time-indexed fairness known as an *integer periodicity* constraint, which allows a decision-maker to guarantee that arm  $i$  is pulled at least once within each period of  $\nu$  days, the

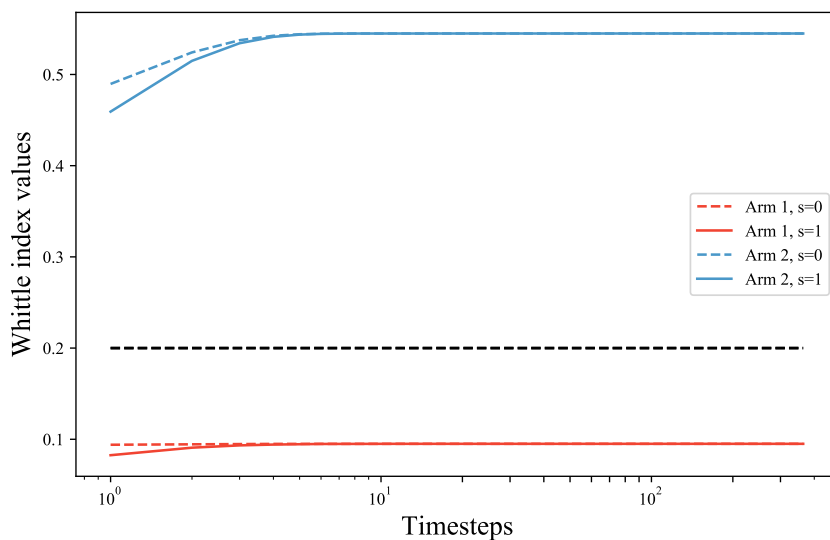


Figure A.1: The Whittle index values for Arm 1 and 2 can be separated by a horizontal line, meaning that (WLOG) Arm 2 will always be chosen over Arm 1 because its index value dominates.

lines associated with the arms in Figure A.1 must cross before  $\nu$  timesteps elapse to guarantee fairness constraint satisfaction.

Another perspective we can take is to ask: what's the smallest interval  $\nu_i$  for each arm  $i$  we could have specified such that THRESHOLD WHITTLE would have satisfied the integer periodicity constraint? Note that this is retrospective, as there is no way to enforce this constraint at planning time. We visualize the minimum such  $\nu_i$  in Figure A.2. On the far right, we see the 515 cases where (WLOG) the second arm is never pulled—that is, the minimum  $\nu_i$  such that THRESHOLD WHITTLE satisfies the hard integer periodicity constraint must be *larger* than the horizon,  $T = 365$ . There is one case where arm  $i = 2$  is pulled exactly once. In a majority of the remaining simulations, TW pulls each arm with approximately equal frequency.

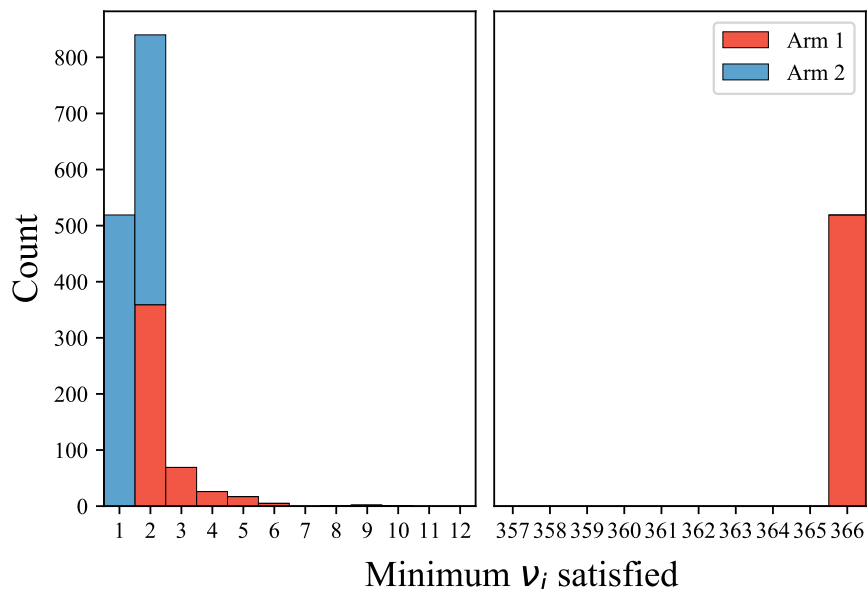


Figure A.2: The smallest interval  $\nu_i$  such that TW satisfies an integer periodicity definition of time-indexed fairness, given  $N = 2$  random arms. In over 50% of iterations, no such fairness constraint satisfaction is possible (i.e.,  $\exists i$  s.t.  $\nu_i > T$ ).

### A.3 Intractability of Alternative Approaches

In this section, we motivate the algorithmic design choices we have made when constructing PROBFair by discussing the feasibility of an alternative math programming-based approach.

#### A.3.1 Why not use a math-programming approach?

Our constrained maximization problem can be readily formulated as an integer program (IP) with a totally unimodular (TU) constraint matrix. However, this approach is intractable because the objective function coefficients of this IP cannot be efficiently enumerated. To support this intractability claim, we begin by presenting an integer program (IP) that maximizes total expected reward under both budget *and* time-indexed fairness constraints, for problem instances with feasible hyperparameters. We then prove that any problem instance with feasible hyperparameters yields a totally unimodular (TU) constraint matrix, which ensures that the linear program (LP) relaxation of our IP will yield an integral solution. We proceed to demonstrate that tractability issues arise because we incur an exponential dependency on the time horizon,  $T$ , when we construct the IP's objective function coefficients. We conclude by comparing PROBFair to the IP for small values of  $N$  and  $T$ .

##### A.3.1.1 Integer Program Formulation

To leverage a math programming approach for our constrained reward maximization task, we seek to construct an integer program (IP) whose solution is the policy  $\vec{x} \in \{0, 1\}^{N|A|^T}$ . We require this policy to be reward-maximizing, subject to the guaranteed satisfaction of both budget and time-



indexed fairness constraints. To begin, let each decision variable  $x_{i,a,t} \in \{0, 1\}$  represent whether or not we take action  $a \in \mathcal{A} = \{0, 1\}$  for arm  $i \in [N]$  at time  $t \in [T]$ . Then, let each objective function coefficient  $c_{i,a,t}$  represent the expected reward associated with an arm-action-timestep combination.

To formalize the objective function, recall that the agent seeks to maximize total expected reward,  $\mathbb{E}_\pi[R(\cdot)]$ . For clarity of exposition, we specifically consider the linear global reward function  $R(r(s)) = \sum_{i=1}^N \sum_{t=0}^T s_t^i$ . Note that this implies the discount rate,  $\beta = 1$ ; however, the approach outlined here can be extended in a straightforward manner for  $\beta \in (0, 1)$ . In order to compute the expected reward associated with taking action  $a$  for arm  $i$  at time  $t$ , we must consider: (1) what state is the arm currently in (i.e., what is the realized value of  $s_t^i \in \{0, 1\}$ )? (2) when the arm transitions from  $s_t$  to  $s_{t+1}$  by virtue of taking action  $a$ , what reward,  $r(\cdot)$ , should we expect to earn?

Because we define  $r(s) = s$ , (2) can be reframed as: what is the probability  $p(s_{t+1} = 1 | s_t^i, a_t^i)$  that action  $a$  causes a transition from  $s_t$  to the adherent state? Because each arm's state at time  $t$  is stochastic, depending not only on the sequence of actions taken in previous timesteps, but the associated set of stochastic transitions informed by the arm's underlying MDP, each coefficient of

our objective function must be computed as the expectation over the possible values of  $s_t \in \mathcal{S}$ :

$$\vec{c} = \mathbb{E}_s[p(s_{t+1} = 1|x_{i,a,t}, s_t)] \quad \forall i, a, t \in [N] \times \mathcal{A} \times [T] \quad (\text{A.1})$$

$$= \frac{1}{2^t} \sum_{s \in \mathcal{S}} p(s_t = s) \sum_{s' \in \mathcal{S}} p(s_{t+1} = s' | x_{i,a,t}, s_t = s) r(s') \quad \forall i, a, t \in [N] \times \mathcal{A} \times [T] \quad (\text{A.2})$$

$$= \frac{1}{2^t} \sum_{s \in \mathcal{S}} p(s_t = s) p(r(s_{t+1}) = 1 | x_{i,a,t}, s_t = s) \quad \forall i, a, t \in [N] \times \mathcal{A} \times [T] \quad (\text{A.3})$$

$$= \frac{1}{2^t} \sum_{s \in \mathcal{S}} p(s_t = s) p(s_{t+1} = 1 | x_{i,a,t}, s_t = s) \quad \forall i, a, t \in [N] \times \mathcal{A} \times [T] \quad (\text{A.4})$$

Within the context of this IP, the time-indexed fairness constraint we introduce in Section 2.2.2 can be more specifically defined as either an *integer periodicity* or *minimum selection fraction* constraint. We formalize each of these below:

The **integer periodicity constraint** allows a decision-maker to guarantee that arm  $i$  is pulled at least once within each period of  $\nu$  days. We define this constraint as a function  $g$ , over the vector of actions,  $\vec{a}^i$  associated with arm  $i$ , and user-defined interval length  $\nu \in [1, T]$ :

$$g(\vec{a}^i) = \sum_{t=j\nu+1}^{(j+1)\nu} a_t^i \geq 1 \quad (\text{A.5})$$

$$\forall j \in \left\{ 0, 1, 2, \dots, \left\lceil \frac{T}{\nu} \right\rceil \right\}; \forall i \in \{1, 2, \dots, N\}$$

The **minimum selection fraction constraint** introduced by Li et al. [28] forces the agent to pull arm  $i$  at least a minimum fraction,  $\psi \in (0, 1)$ , of the total number of steps, but is agnostic to how these pulls are distributed over time. We define this constraint,  $g'$ , as a function over the

vector of actions,  $\vec{a}^i$  associated with arm  $i$  and user-defined  $\psi$ :

$$g'(\vec{a}^i) = \frac{1}{T} \sum_{t=1}^T a_t^i \geq \psi \quad \forall i \in \{1, 2, \dots, N\} \quad (\text{A.6})$$

The resulting integer program is given by:

$$\begin{aligned} \max \quad & c^T x && (\text{A.7}) \\ \text{s.t.} \quad & \sum_{a=1}^{|A|} x_{i,a,t} = 1 && \forall i \in [N], t \in [T] \quad (\text{a}) \text{ Select exactly one action per} \\ & && \text{arm } i \text{ at each } t \\ \text{if int. per:} \quad & \sum_{t \in I_j} x_{i,1,t} \geq 1 && \forall j \in \left\{0, 1, \dots, \frac{T-\nu}{\nu}\right\} \quad (\text{b.i}) \text{ Pull each arm } i \text{ at least once} \\ & && \text{during each interval of length } \nu \\ \text{if min. sel:} \quad & \frac{1}{T} \sum_{t=1}^T x_{i,1,t} \geq \psi && \psi \in (0, 1), \forall i \in N \quad (\text{b.ii}) \text{ Pull each arm } i \text{ at least a} \\ & && \text{minimum fraction } \psi \text{ of } T \text{ rounds} \\ & \sum_{i=1}^N x_{i,1,t} = k && \forall t \in [T] \quad (\text{c}) \text{ Pull exactly } k \text{ arms at each } t \\ & x_{i,a,t} \in \{0, 1\} && \forall i \in [N], a \in \mathcal{A}, t \in [T] \quad (\text{d}) \text{ Each arm-action-timestep choice} \\ & && \text{is a binary decision variable} \end{aligned}$$

### A.3.1.2 LP Relaxation and Integrality of Solution

We now prove that the IP we have formulated in Section A.3.1.1 has an attractive property: namely, any feasible problem instance will produce a totally unimodular constraint matrix. Our proof leverages a theorem introduced by [Ghouila-Houri](#) and restated below for convenience, which can be used to determine whether a matrix,  $A \in \mathbb{R}^{m \times n}$  is totally unimodular:

**Lemma A.3.1.** (Ghouila-Houri [149]) A matrix  $A \in \mathbb{Z}^{m \times n}$  is totally unimodular (TU) if and only if for every subset of the rows  $R \subseteq [m]$ , there is a partition  $R = R_1 \cup R_2$  such that for every  $j \in [n]$ ,

$$\sum_{i \in R_1} A_{ij} - \sum_{i \in R_2} A_{ij} \in \{-1, 0, 1\} \quad (\text{A.8})$$

**Theorem A.3.2.** Within the context of the integer program outlined in Appendix A.3.1.1, any feasible problem instance will produce a constraint matrix that is totally unimodular (TU).

*Proof.* To begin, we establish the dimensions of any such constraint matrix  $\mathbf{A}$  and note the maximum possible column-wise sum that each of its component submatrices may contribute. Note that the minimum selection fraction constraint (b.ii in Equation A.7), which requires the agent to pull each arm  $i$  at least a minimum fraction,  $\psi \in (0, 1)$ , of  $T$  rounds, can be thought of as a special case of the integer periodicity constraint, (b.i in Equation A.7), where  $\nu = T$  and each arm must be pulled at least  $\lceil T\psi \rceil$  times. As such, we assume that at most one of the time-indexed fairness constraints can be selected, and focus on the more general of the two, which is the integer periodicity constraint. For notational convenience, we refer to constraints by their alphabetic identifiers. Let (b) represent the integer periodicity constraint, and define a function  $\varphi : r \in \mathbf{R} \subseteq \mathbf{A} \mapsto e \in \{a, b, c\}$  that maps each row to its corresponding constraint type.

First, recall that each  $x_{i,a,t}$  represents a single binary decision variable, and corresponds to a column in  $\mathbf{A}$ . There are  $N \times |\mathcal{A}| \times T$  such columns. Next, note that constraint (a) enforces the requirement that we select *exactly* one action per arm per timestep. Formally,  $\forall i, t \in N \times T$ ,  $\exists! a \in \mathcal{A}$  s.t.  $x_{i,a,t} = 1$ . Correspondingly,  $\forall a' \in \mathcal{A} \setminus a$ ,  $x_{i,a',t} = 0$ . The column vectors of the associated sub-matrix,  $\mathbf{A}_a \in \mathbb{Z}^{NT \times N|\mathcal{A}|T}$ , are indexed by disjoint  $(i, a, t) \in N \times |\mathcal{A}| \times T$ ; thus,

each column vector contains a single non-zero entry and for  $\mathbf{R}_a \subseteq \mathbf{A}_a$ , taking the column-wise sum will yield a vector  $\vec{v} \in \mathbb{Z}^{N|\mathcal{A}|T}$  with every entry equal to 1.

In a similar vein, equity constraint (b) enforces the requirement that we must pull each arm,  $i$  at least once during each interval  $I_j$  of length  $\nu_i$ . Within the associated sub-matrix,  $\mathbb{A}_b \in \mathbb{Z}^{N\lceil \frac{T}{\nu_i} \rceil \times N|\mathcal{A}|T}$ , each column that corresponds to a passive action (e.g.,  $x_{i,a=0,t}$ ) will have *only* zero-valued entries, since passive action decision variables are not impacted by constraint (b). Conversely, each column that corresponds to an active action (e.g.,  $x_{i,a=1,t}$ ) will have a single non-zero entry. Each active action column corresponding to a specific arm-timestep can be mapped to exactly one interval. Thus, for  $\mathbf{R}_b \subseteq \mathbf{A}_b$ , taking the column-wise sum will yield a vector  $\vec{v} \in \mathbb{Z}^{N|\mathcal{A}|T}$  with every entry taking a value  $\in \{0, 1\}$ .

The budget constraint (c) enforces the requirement that we must pull exactly  $k$  of the  $N$  arms at each timestep. Much like equity constraint (b), only columns corresponding to active actions are impacted. Thus, within the associated sub-matrix,  $\mathbf{A}_c \in \mathbb{Z}^{T \times N|\mathcal{A}|T}$ , each column that corresponds to a passive action (e.g.,  $x_{i,a=0,t}$ ) will have *only* zero-valued entries, while each column that corresponds to an active action (e.g.,  $x_{i,a=1,t}$ ) can be mapped to a single timestep, and will have a single non-zero entry. Thus, for  $\mathbf{R}_c \subseteq \mathbf{A}_c$ , taking the column-wise sum also yield a vector  $\vec{v} \in \mathbb{Z}^{N|\mathcal{A}|T}$  with every entry taking a value  $\in \{0, 1\}$ .

The complete constraint matrix  $\mathbf{A}$  thus contains  $NT + N\lceil \frac{T}{\nu_i} \rceil + T$  rows. Three possible cases arise when we consider every subset of these rows: (1)  $\mathbf{R} \subsetneq \mathbf{A} = \emptyset$ ; (2)  $\mathbf{R} \subsetneq \mathbf{A}$ ;  $\mathbf{R} \cap \mathbf{A} \neq \emptyset$ ; (3)  $\mathbf{R} \subseteq \mathbf{A}$ .

*Case 1.*  $\mathbf{R} \subsetneq \mathbf{A} = \emptyset$ . To satisfy Lemma A.3.1, partition  $\mathbf{R}$  such that  $\mathbf{R} = \mathbf{R}_1 \cup \mathbf{R}_2 = \emptyset \cup \emptyset$ . Then,

for every  $j \in [n]$ ,

$$\sum_{i \in \mathbf{R}_1} \mathbf{A}_{ij} - \sum_{i \in \mathbf{R}_2} \mathbf{A}_{ij} = 0 - 0; 0 \in \{-1, 0, 1\} \square$$

*Case 2.*  $\mathbf{R} \subsetneq \mathbf{A}; \mathbf{R} \cap \mathbf{A} \neq \emptyset$ . If we consider  $\cup_{r \in \mathbf{R}} \varphi(r)$ , there are  $\sum_{k=1}^3 \binom{3}{k}$  possible sets of observed constraint types:  $\{a\} \vee \{b\} \vee \{c\} \vee \{a, b\} \vee \{a, c\} \vee \{b, c\} \vee \{a, b, c\}$ .

1. If  $|\cup_{r \in \mathbf{R}} \varphi(r)| = 1$ , then any partition of  $\mathbf{R}$  will satisfy Lemma A.3.1. Without loss of generality, let each row  $r \in \mathbf{R}$  belong to  $\mathbf{R}_1$  and  $\mathbf{R}_2 = \emptyset$ .

i. If  $\cup_{r \in \mathbf{R}} \varphi(r) = \{a\}$ , taking the column-wise sum of  $\mathbf{R}_1$  will yield a vector  $\vec{v} \in \mathbb{Z}^{N|\mathcal{A}|T}$  with every entry  $\in \{1\}$  if  $\mathbf{R} \subseteq \mathbf{A}_a$ , and  $\in \{0, 1\}$  otherwise. Thus,  $\forall j \in [N|\mathcal{A}|T]$ ,

$$\sum_{i \in \mathbf{R}_1} \mathbf{A}_{ij} - \sum_{i \in \mathbf{R}_2} \mathbf{A}_{ij} = 0 - 0 \vee 1 - 0; \{0, 1\} \subsetneq \{-1, 0, 1\} \square$$

ii. If  $\cup_{r \in \mathbf{R}} \varphi(r) = \{b\} \vee \{c\}$ , taking the column-wise sum of  $\mathbf{R}_1$  will yield a vector  $\vec{v} \in \mathbb{Z}^{N|\mathcal{A}|T}$  with every entry corresponding to a passive action column  $\in \{0\}$  and every entry corresponding to an active action column  $\in \{1\}$  if  $\mathbf{R} \subseteq \mathbf{A}_{b \vee c}$ , and  $\in \{0, 1\}$  otherwise. Thus,  $\forall j \in [N|\mathcal{A}|T]$ ,

$$\sum_{i \in \mathbf{R}_1} \mathbf{A}_{ij} - \sum_{i \in \mathbf{R}_2} \mathbf{A}_{ij} = 0 - 0 \vee 1 - 0; \{0, 1\} \subsetneq \{-1, 0, 1\} \square$$

2. If  $|\cup_{r \in \mathbf{R}} \varphi(r)| = 2$ , without loss of generality, partition as follows: sort the elements  $\in \cup_{r \in \mathbf{R}} \varphi(r)$  lexicographically, and let  $\mathbf{R}_1 = \{r | \varphi(r) = \min \cup_{r \in \mathbf{R}} \varphi(r)\}$  and  $\mathbf{R}_2 = \mathbf{R} \setminus \mathbf{R}_1$ . Per Case 2.1 (i) and (ii), taking the column-wise sums of  $\mathbf{R}_1$  and  $\mathbf{R}_2$  will yield two vectors,

$\vec{v}_1, \vec{v}_2 \in \mathbb{Z}^{N|\mathcal{A}|T}$ , each of which will contain only entries  $\in \{0, 1\}$ . Thus,  $\forall j \in [N|\mathcal{A}|T]$ ,

$$\sum_{i \in \mathbf{R}_1} \mathbf{A}_{ij} - \sum_{i \in \mathbf{R}_2} \mathbf{A}_{ij} = 0 - 0 \vee 0 - 1 \vee 1 - 0 \vee 1 - 1; \{-1, 0, 1\} \subseteq \{-1, 0, 1\} \square$$

3. If  $|\cup_{r \in \mathbf{R}} \varphi(r)| = 3$ , partition according to Algorithm 8:

---

**Algorithm 8** Partition for Case 2.3

---

```

1: procedure PARTITION( $N, \mathcal{A}, T, \mathbf{R}$ )
2:    $C_{R_1}; C_{R_2} \leftarrow \{0\}^{N|\mathcal{A}|T}$  ▷ Initialize two sets of counters
3:    $\mathbf{R}_1; \mathbf{R}_2 \leftarrow \emptyset$ 
4:   for element  $e \in \{a, b, c\}$  do
5:      $\mathbf{R}_e \leftarrow \{r | \varphi(r) = e\}$  ▷ Use  $\varphi$  to partition the rows of  $\mathbf{R}$  by constraint type
6:   for  $r \in \mathbf{R}_a$  do
7:      $\mathbf{R}_1 \leftarrow \mathbf{R}_1 \cup r$  ▷ Let  $r \in \mathbf{R}_1$ 
8:     for  $i \in 0 : N|\mathcal{A}|T$  do
9:       if  $r[i] > 0$  then ▷ For each non-zero entry in  $r$ 
10:         $C_{R_1}[i] \leftarrow C_{R_1}[i] + 1$  ▷ Increment corresponding  $C_{R_1}$  counter
11:   for  $r \in \mathbf{R}_b$  do
12:     flag  $\leftarrow$  false
13:     for  $i \in 0 : N|\mathcal{A}|T$  do
14:       if  $r[i] > 0 \wedge C_{R_1}[i] > 0$  then ▷ If any non-zero element  $r_i$  has  $C_{R_1} > 0$ 
15:         flag  $\leftarrow$  true ▷ Set flag to true
16:     if flag then
17:        $\mathbf{R}_2 \leftarrow \mathbf{R}_2 \cup r$  ▷ Let  $r \in \mathbf{R}_2$ 
18:       for  $i \in 0 : N|\mathcal{A}|T$  do
19:         if  $r[i] > 0$  then ▷ For each non-zero entry in  $r$ 
20:            $C_{R_2}[i] \leftarrow C_{R_2}[i] + 1$  ▷ Increment corresponding  $C_{R_2}$  counter
21:     else
22:        $\mathbf{R}_1 \leftarrow \mathbf{R}_1 \cup r$  ▷ Let  $r \in \mathbf{R}_1$ 
23:       for  $i \in 0 : N|\mathcal{A}|T$  do
24:         if  $r[i] > 0$  then ▷ For each non-zero entry in  $r$ 
25:            $C_{R_1}[i] \leftarrow C_{R_1}[i] + 1$  ▷ Increment corresponding  $C_{R_1}$  counter
26:   for  $r \in \mathbf{R}_c$  do
27:      $\mathbf{R}_2 \leftarrow \mathbf{R}_2 \cup r$  ▷ Let  $r \in \mathbf{R}_2$ 
28:     for  $i \in 0 : N|\mathcal{A}|T$  do
29:       if  $r[i] > 0$  then ▷ For each non-zero entry in  $r$ 
30:          $C_{R_2}[i] \leftarrow C_{R_2}[i] + 1$  ▷ Increment corresponding  $C_{R_2}$  counter
return  $\mathbf{R}_1; \mathbf{R}_2$ 

```

---

Taking the column-wise sums of the resulting  $\mathbf{R}_1$  and  $\mathbf{R}_2$  will yield two vectors  $\vec{v}_1, \vec{v}_2 \in \mathbb{Z}^{N|\mathcal{A}|T}$ , which can contain entries  $\in \{0, 1\}$  and  $\{0, 1, 2\}$ , respectively. Note that since  $\vec{v}_2$  is constructed by taking only rows with constraint types  $\in \{b, c\}$ , only entries corresponding to

active action columns can take values  $> 1$ . Moreover,  $\forall j \in [N|\mathcal{A}|T], \sum_{i \in \mathbf{R}_2} \mathbf{A}_{ij} = 2 \rightarrow \sum_{i \in \mathbf{R}_1} \mathbf{A}_{ij} = 1$ . Thus,  $\forall j \in [N|\mathcal{A}|T]$ ,

$$\sum_{i \in \mathbf{R}_1} \mathbf{A}_{ij} - \sum_{i \in \mathbf{R}_2} \mathbf{A}_{ij} = 0 - 0 \vee 0 - 1 \vee 1 - 0 \vee 1 - 1 \vee 1 - 2; \{-1, 0, 1\} \subseteq \{-1, 0, 1\} \square$$

*Case 3.*  $R \subseteq A$ . Since  $|\cup_{r \in \mathbf{R}} \varphi(r)| = 3$ , proceed as outlined in Case 2.3. Only a slight modification is required: since  $\mathbf{R}$  is now equal to  $\mathbf{A}$ , taking the column-wise sums of the resulting  $\mathbf{R}_1$  and  $\mathbf{R}_2$  will yield two vectors  $\vec{v}_1, \vec{v}_2 \in \mathbb{Z}^{N|\mathcal{A}|T}$ , which can contain entries  $\in \{1\}$  and  $\{0, 2\}$ , respectively. Thus,  $\forall j \in [N|\mathcal{A}|T]$ ,

$$\sum_{i \in \mathbf{R}_1} \mathbf{A}_{ij} - \sum_{i \in \mathbf{R}_2} \mathbf{A}_{ij} = 1 - 0 \vee 1 - 2; \{-1, 1\} \subsetneq \{-1, 0, 1\}$$

□

### A.3.1.3 Enumeration of Objective Function Coefficients

The key challenge we encounter when we seek to enumerate the IP outlined in Section A.3.1.1 is that exact computation of the objective function coefficients,  $\vec{c} \in \mathbb{R}^{N|\mathcal{A}|T}$  is intractable. Each arm contributes  $|\mathcal{A}| \times T$  coefficients, and while calculation is trivially parallelizable over arms, we must consider a probability tree like the one in Figure A.4 for each arm.

The number of decision variables required to enumerate each arm's game tree is of order  $O(|\mathcal{A}||\mathcal{S}|^T)$  and there are  $N$  such trees, so even a linear program (LP) relaxation is not tractable for larger values of  $T$  and  $N$ , which motivates us to propose PROBFair (Section 2.2.4) as an efficient alternative.



*Example 3.* Suppose we wish to find the coefficient  $c'$  corresponding to  $x_{i,a=0,t=2}$ . From Equation A.4, we have  $c' = \frac{1}{2^2} \sum_{s \in \mathcal{S}} p(s_2 = s)p(s_3 = 1 | x_{i,a=0,t=2}, s_2 = s)$ . Equivalently, we sum the weight of each path from the root node to the highlighted end nodes in Figure A.4 and normalize by  $\frac{1}{2^2}$ :

$$\begin{aligned}
 c' = \frac{1}{4} & \left( P_{s_{0,0}}^0 P_{0,0}^0 P_{0,1}^0 + P_{s_{0,0}}^0 P_{0,0}^1 P_{0,1}^0 + P_{s_{0,0}}^0 P_{0,1}^0 P_{1,1}^0 + P_{s_{0,0}}^0 P_{0,1}^1 P_{1,1}^0 \right. \\
 & + P_{s_{0,0}}^1 P_{0,0}^0 P_{0,1}^0 + P_{s_{0,0}}^1 P_{0,0}^1 P_{0,1}^0 + P_{s_{0,0}}^1 P_{0,1}^0 P_{1,1}^0 + P_{s_{0,0}}^1 P_{0,1}^1 P_{1,1}^0 \\
 & \left. + \dots \right) \quad \text{For each of the } (|\mathcal{A}||\mathcal{S}|)^t = 16 \text{ paths to a blue node in Figure A.4.}
 \end{aligned} \tag{A.9}$$

### A.3.1.4 Comparison of PROBFAIR with the True Optimal Policy

In Section 2.2.5, we normalize intervention benefit with THRESHOLD WHITTLE, which is asymptotically optimal for forward threshold-optimal transition matrices under a budget constraint  $k$  [20]. However, with the integer program (IP) we formulate in Section A.3.1.1, we can find the *optimal* policy for any set of transition matrices under budget *and* fairness constraints as long as  $N$  and  $T$  are small.

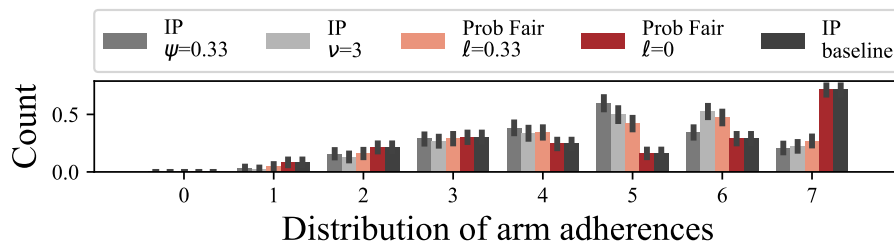


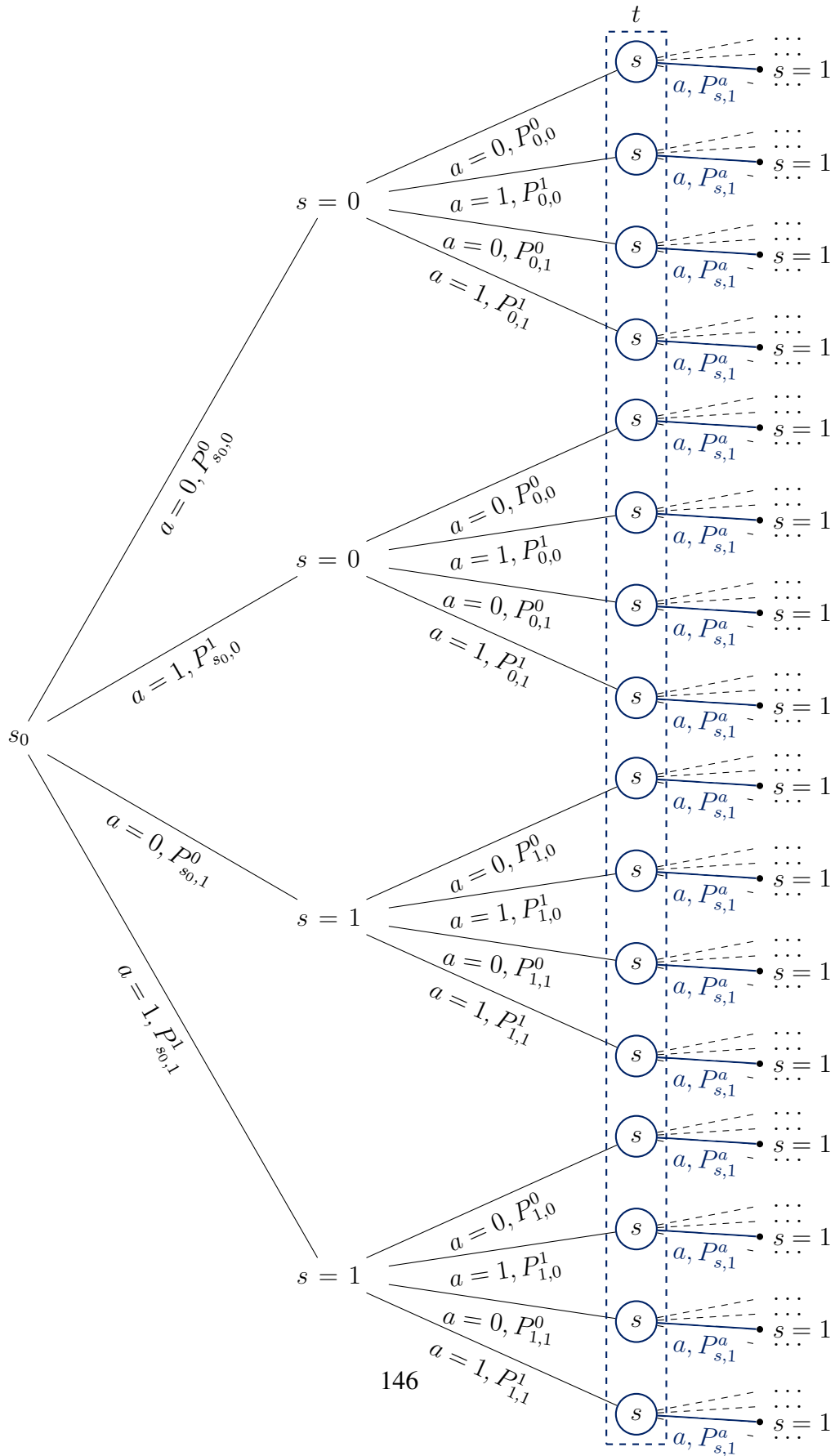
Figure A.3: Adherences of PROBFAIR, compared to the IP formulation

We generate  $N = 2$  random arms such that the structural constraints outlined in Section 2.2.2 are satisfied. We set  $k = 1$  and  $T = 6$ . Though the variance in reward is large due to the small

$T$ , Figure A.3 shows that PROBFair obtains 100% of the intervention benefit when no fairness constraints are applied. Similarly, PROBFair with  $\ell = 0.33$  obtains the same adherence behavior as the IP policy with under hard fairness constraint  $\nu = 3$  or minimum selection fraction constraint  $\psi = 0.33$ . (within 95% confidence interval shown). All results shown are bootstrapped over 500 iterations.

**Minimum Selection Fraction Constraints.** As we discuss in Appendix A.2, the optimal policy is often to pull the same  $k$  arms at every timestep and ignore all other arms. Under minimum selection fraction constraints (Equation A.6), each arm must be pulled at least a minimum fraction  $\psi$  of  $T$  rounds, with no conditions on when these pulls should take place. We confirm with the optimal IP implementation our intuition that these additional pulls are allocated at the beginning or end of the simulation. That is, the optimal policy under minimum selection fraction constraints is to take advantage of the finite time horizon, which is not suitable for the applications we consider.

Figure A.4: Illustration of the probability tree for finding the coefficient corresponding to  $x_{i,a=0,t=2}$ .



## A.4 PROBFair: a Probabilistically Fair Policy

Our main contribution is the novel probabilistic policy PROBFair (see Section 2.2.4.1). Here, in Section A.4.1, we present complete proofs to Theorems 2.2.2-2.2.5. Then, in Section A.4.2, we provide additional details about how we sample from our probabilistic policy to select discrete actions at each timestep.

### A.4.1 Proofs

In this section, we provide proofs for the theorems introduced in Section 2.2.4.1. When relevant, we begin by restating the theorem for convenience.

**Theorem 2.2.2.** *For every arm  $i \in [N]$ ,  $f_i(p_i)$  is either **concave** or **strictly convex** in all of  $p_i \in [0, 1]$ .*

*Proof.* For notational convenience, let:

$$c_1 = P_{0,1}^0;$$

$$c_2 = P_{0,1}^1 - P_{0,1}^0;$$

$$c_3 = 1 - P_{1,1}^0 + P_{0,1}^0;$$

$$c_4 = P_{1,1}^0 - P_{1,1}^1 - P_{0,1}^0 + P_{0,1}^1.$$

Then,  $f_i(p_i) = \frac{c_1 + c_2 p_i}{c_3 + c_4 p_i}$ . We observe that  $\forall i \in [N]$ ,  $f_i(p_i)$  is a valid probability since the term  $1 - (1 - p_i)P_{1,1}^0 - p_i P_{1,1}^1$  in the denominator is at least  $1 - (1 - p_i) - p_i = 0$  for all  $p_i \in [0, 1]$ .

Then, there are three cases which describe the possible shapes of  $f_i(p_i)$ :

*Case 1.*  $c_4 = 0$ . Here,  $f_i(p_i)$  is **linear** and hence, **concave**.

*Case 2.*  $c_4 \neq 0; c_2 = 0$ . Here,  $f_i''(p_i) = \frac{2c_1c_4^2}{(c_3+c_4p_i)^3} \geq 0$ , so  $f_i(p_i)$  is linear (hence **concave**) if  $c_1 = 0$  or **strictly convex** (if  $c_1 > 0$ ) in the domain  $p_i \in [0, 1]$ .

*Case 3.*  $c_4 \neq 0; c_2 \neq 0$ . Here,

$$f_i(p_i) = \frac{\frac{c_2}{c_4} \left( \frac{c_1c_4}{c_2} + c_4p_i \right)}{c_3 + c_4p_i} = \frac{c_2}{c_4} + \frac{\left( c_1 - \frac{c_2c_3}{c_4} \right)}{c_3 + c_4p_i}. \quad (\text{A.10})$$

Thus,

$$f_i''(p_i) = \frac{2c_4^2 \left( c_1 - \frac{c_2c_3}{c_4} \right)}{(c_3 + c_4p_i)^3} \quad (\text{A.11})$$

The sign of  $f_i''(p_i)$  is the same as the sign of  $d = c_1 - \frac{c_2c_3}{c_4}$ . It follows that  $f_i(p_i)$  is **strictly convex** if  $c_1 > \frac{c_2c_3}{c_4}$ , and **concave** otherwise for  $p_i \in [0, 1]$ .

□

**Theorem 2.2.3.** For each arm  $i \in [N]$ , the structural constraints introduced in Section 2.1 ensure that  $f_i(p_i)$  is monotonically non-decreasing in  $p_i$  over the interval  $[0, 1]$ .

*Proof.* For notational convenience, let:

$$c_1 = P_{0,1}^0;$$

$$c_2 = P_{0,1}^1 - P_{0,1}^0;$$

$$c_3 = 1 - P_{1,1}^0 + P_{0,1}^0;$$

$$c_4 = P_{1,1}^0 - P_{1,1}^1 - P_{0,1}^0 + P_{0,1}^1.$$

$$\text{Then } f_i(p_i) = \frac{c_1+c_2p_i}{c_3+c_4p_i} \text{ and } f_i'(p_i) = \frac{c_2c_3-c_1c_4}{(c_3+c_4p_i)^2}.$$

Observe  $c_2c_3 - c_1c_4 \geq 0$  implies  $f'_i(p_i) \geq 0$ .

$$c_2c_3 - c_1c_4 \geq 0$$

$$c_2c_3 \geq c_1c_4$$

$$(P_{0,1}^1 - P_{0,1}^0)(1 - P_{1,1}^0 + P_{0,1}^0) \geq P_{0,1}^0(P_{1,1}^0 - P_{1,1}^1 - P_{0,1}^0 + P_{0,1}^1)$$

$$(1 - P_{1,1}^0)P_{0,1}^1 \geq (1 - P_{1,1}^1)P_{0,1}^0$$

Per our structural constraints,  $P_{1,1}^0 < P_{1,1}^1$  and  $P_{0,1}^1 > P_{0,1}^0$ . □

**Lemma 2.2.4.** ***P2** has an optimal solution in which  $p_i \in (\ell, u)$  for at most one  $i \in \mathcal{Y}$ .*

*Proof.* Note by compactness that **P2** has not just a supremum, but an actual maximum solution. Suppose for contradiction there is some optimal solution  $\vec{p}$  with distinct indices  $i, j \in \mathcal{Y}$  such that  $p_i, p_j \in (\ell, u)$ . Now, suppose we perturb by an infinitesimal  $\epsilon$  (of arbitrary sign but tiny positive absolute value) such that  $p_i := p_i + \epsilon$  and  $p_j := p_j - \epsilon$ . This satisfies all our constraints for small-enough  $|\epsilon|$ . The change in the objective  $\sum_{i \in \mathcal{Y}} f_i(p_i)$  is now  $\epsilon \cdot (f'_i(p_i) - f'_j(p_j)) + O(\epsilon^2)$ ; hence, if  $f'_i(p_i) - f'_j(p_j)$  is nonzero, then we can take a tiny  $\epsilon$  of the appropriate sign to increase the objective, a contradiction. Therefore,  $f'_i(p_i) - f'_j(p_j) = 0$ , and so, we now focus on lower-order terms: the change in the objective  $\sum_{i \in \mathcal{Y}} f_i(p_i)$  is now  $(\epsilon^2/2) \cdot (f''_i(p_i) + f''_j(p_j)) + O(\epsilon^3)$ . However, since  $f_i$  and  $f_j$  are strictly convex, we have that  $f''_i(p_i) + f''_j(p_j) > 0$ , and hence the objective increases regardless of the sign of (the tiny)  $\epsilon$ , again a contradiction. Thus we have our structural result. □

**Theorem 2.2.5.** *Alg. 2 yields the mapping from arms in  $\mathcal{Y}$  to subsets in  $\{\mathcal{Y}_1, \mathcal{Y}_2, \mathcal{Y}_3\}$  which*

maximizes  $\sum_{i \in \mathcal{Y}} f_i(p_i)$  s.t.  $\sum_{i \in \mathcal{Y}} p_i = k - z$ . (See Appendix A.4.1 for the complete proof).

We begin by introducing Lemma A.4.1, which we use in our proof of Theorem 2.2.5:

**Lemma A.4.1.** For a given  $\gamma, p' \in (\ell, u]$ .

*Proof.* To begin, observe that  $z - k = \gamma\ell + (|\mathcal{Y}| - 1 - \gamma)u + p'$ . Then, to prove the lower bound, observe that:

$$\begin{aligned} \gamma &= \left\lfloor \frac{|\mathcal{Y}|u - (k - z)}{u - \ell} \right\rfloor \\ \gamma &> \frac{|\mathcal{Y}|u - (k - z)}{u - \ell} - 1 \\ \rightarrow \gamma &> \frac{|\mathcal{Y}|u - (\gamma\ell + (|\mathcal{Y}| - 1 - \gamma)u + p')}{u - \ell} - 1 \\ \gamma &> \frac{|\mathcal{Y}|u - \gamma\ell - |\mathcal{Y}|u + u + \gamma u - p' - u + \ell}{u - \ell} \\ 0 &> \ell - p' \implies p' > \ell \end{aligned}$$

To prove the upper bound, observe that:

$$\begin{aligned} \gamma &= \left\lfloor \frac{|\mathcal{Y}|u - (k - z)}{u - \ell} \right\rfloor \\ \gamma &\leq \frac{|\mathcal{Y}|u - (k - z)}{u - \ell} \\ \rightarrow \gamma &\leq \frac{|\mathcal{Y}|u - (\gamma\ell + (|\mathcal{Y}| - 1 - \gamma)u + p')}{u - \ell} \\ \gamma &\leq \frac{\gamma(u - \ell) + u - p'}{u - \ell} \\ 0 &\leq u - p' \implies p' \leq u \end{aligned}$$

Thus,  $\ell < p' \leq u$ . □

Now, we proceed with our proof of **Theorem 2.2.5**.

*Proof.* Recall **P2**: maximize  $\sum_{i \in \mathcal{Y}} f_i(p_i)$  such that  $p_i \in [\ell, u]$  for all  $i \in \mathcal{Y}$  and  $\sum_{i \in \mathcal{Y}} p_i = k - z$ .

By **Lemma 2.2.4**, there exists *at most one arm* with optimal value  $p_i^* \in (\ell, u)$ .

First, we discuss an edge case. If  $k - z = |\mathcal{Y}|\ell$ , Line 2 of Algorithm 2 assigns  $\gamma = |\mathcal{Y}|$ , so

Line 11 assigns

$$\pi_{\mathcal{Y}} := i \mapsto \left\{ \begin{array}{l} \ell, \quad \text{for } i \in \mathcal{Y}_1 = \{\mathcal{Y}\} \\ p', \quad \text{for } i \in \mathcal{Y}_2 = \emptyset \\ u, \quad \text{for } i \in \mathcal{Y}_3 = \emptyset \end{array} \right\} \quad (\text{A.12})$$

Thus Algorithm 2 returns the only valid solution to **P2** in this case, which is to set  $p_i = \ell$  for all arms  $i \in \mathcal{Y}$ .

For all other cases  $k - z > |\mathcal{Y}|\ell$ , we introduce the following notation: let  $\mathcal{Y}_1$  be the set of arms for which  $p_i = \ell$ ,  $\mathcal{Y}_2$  be a set containing exactly one arm (WLOG  $j$ ) where  $p_j = p' \in (\ell, u]$ , and  $\mathcal{Y}_3$  be the remaining set of arms for which  $p_i = u$ , with  $\bigcap_{x=1}^3 \mathcal{Y}_x = \emptyset$ . Then by **Lemma A.4.1**,

$$\gamma = |\mathcal{Y}_1| = \left\lfloor \frac{|\mathcal{Y}|u - (k - z)}{u - \ell} \right\rfloor \text{ and } p' = k - z - \gamma\ell - (|\mathcal{Y}| - 1 - \gamma)u \in (\ell, u].$$



**P2** is then equivalent to finding a partition  $\mathcal{Y} \rightarrow \mathcal{Y}_1 \cup \mathcal{Y}_2 \cup \mathcal{Y}_3$  which maximizes the following:

$$\begin{aligned}
& \arg \max_{\{\mathcal{Y}_1, \mathcal{Y}_2, \mathcal{Y}_3\}} \sum_{i \in \mathcal{Y}_1} f_i(\ell) + f_j(p') + \sum_{i'' \in \mathcal{Y}_3} f_{i''}(u) \\
& \text{s.t. } |\mathcal{Y}_1| = \gamma, \mathcal{Y}_2 = \{j\}, \\
& \quad \bigcap_{x=1}^3 \mathcal{Y}_x = \emptyset, \text{ and } \bigcup_{x=1}^3 \mathcal{Y}_x = \mathcal{Y}
\end{aligned} \tag{A.13}$$

Subtracting the constant  $\sum_{i \in \mathcal{Y}} f_i(\ell)$  and simplifying yields:

$$\begin{aligned}
& \arg \max_{\{\mathcal{Y}_1, \mathcal{Y}_2, \mathcal{Y}_3\}} f_j(p') - f_j(\ell) + \sum_{i'' \in \mathcal{Y}_3} f_{i''}(u) - f_{i''}(\ell) \\
& \text{s.t. } |\mathcal{Y}_1| = \gamma, \mathcal{Y}_2 = \{j\}, \\
& \quad \bigcap_{x=1}^3 \mathcal{Y}_x = \emptyset, \text{ and } \bigcup_{x=1}^3 \mathcal{Y}_x = \mathcal{Y}
\end{aligned} \tag{A.14}$$

Suppose we sort arms in ascending order by  $f_i(u) - f_i(\ell)$ . Let us create the set  $\mathcal{Y}'_3$  from the last  $|\mathcal{Y}| - \gamma - 1$  arms.

By monotonicity, for all  $i \in \mathcal{Y}'_3$  and  $j \notin \mathcal{Y}'_3$ ,

$$f_j(p') - f_j(\ell) \leq f_i(u) - f_i(\ell) \tag{A.15}$$

Thus, setting  $\mathcal{Y}_3^* = \mathcal{Y}'_3$  reduces the optimization problem in Eq. [A.13](#) to finding a partition

over the remaining sets  $\mathcal{Y}_1$  and  $\mathcal{Y}_2$ .

$$\begin{aligned}
& \arg \max_{\{\mathcal{Y}_1, \mathcal{Y}_2\}} f_j(p') - f_j(\ell) \\
& \text{s.t. } |\mathcal{Y}_1| = \gamma, \mathcal{Y}_2 = \{j\}, \\
& \bigcap_{x=1}^2 \mathcal{Y}_x = \emptyset, \text{ and } \bigcup_{x=1}^2 \mathcal{Y}_x = \mathcal{Y} \setminus \mathcal{Y}_3^*
\end{aligned} \tag{A.16}$$

Finally, we solve Equation A.16 by finding the arm  $j$  with maximal value  $f_j(p') - f_j(\ell)$ .

Then  $\mathcal{Y}_2^* = j$ ,  $\mathcal{Y}_1^* = \mathcal{Y} \setminus (\mathcal{Y}_2 \cup \mathcal{Y}_3^*)$ , and we are done.  $\square$

**Corollary 2.2.6.** *Alg. 2 has time complexity  $O(|\mathcal{Y}| \log |\mathcal{Y}|)$ .*

At worst, Algorithm 2 requires two sorts: once on Line 5, and a second time on Line 8, for a total computational cost of  $O(2|\mathcal{Y}| \log |\mathcal{Y}|)$ . In total, the computational cost of Algorithm 1 is at worst  $O\left(\frac{kN}{\epsilon^3}\right)$  when all  $N$  arms are in  $\mathcal{X}$ .

## A.4.2 Dependent Rounding Sampling Approach

Here we provide pseudocode for the sampling algorithm introduced in Section 2.2.4.2, along with its associated SIMPLIFY subroutine [29].

---

**Algorithm 9** Sampling Subroutine (adapted from Srinivasan [29])

---

```
1: procedure SIMPLIFY( $\alpha \in [0, 1], \beta \in [0, 1]$ )
2:   if  $\alpha = \beta = 0$  then
3:      $p_i, p_j \leftarrow [0, 0]$ 
4:   else if  $\alpha = \beta = 1$  then
5:      $p_i, p_j \leftarrow [1, 1]$ 
6:   else if  $\alpha + \beta = 1$  then
7:      $\text{flag} \leftarrow X \sim B(n = 1, p = \alpha)$ 
8:      $p_i, p_j \leftarrow [1, 0]$  if  $\text{flag}$  else  $[0, 1]$ 
9:   else if  $0 < \alpha + \beta < 1$  then
10:     $\text{flag} \leftarrow X \sim B\left(n = 1, p = \frac{\alpha}{\alpha + \beta}\right)$ 
11:     $p_i, p_j \leftarrow [\alpha + \beta, 0]$  if  $\text{flag}$  else  $[0, \alpha + \beta]$ 
12:   else if  $1 < \alpha + \beta < 2$  then
13:     $\text{flag} \leftarrow X \sim B\left(n = 1, p = \frac{1 - \beta}{2 - \alpha - \beta}\right)$ 
14:     $p_i, p_j \leftarrow [1, \alpha + \beta - 1]$  if  $\text{flag}$  else  $[\alpha + \beta - 1, 1]$ 
15:   return  $p_i, p_j$ 
```

---

---

**Algorithm 10** Sampling Algorithm (adapted from Srinivasan [29])

---

```
1: procedure SAMPLE( $G = (V, E)$ )
2:    $H \leftarrow G \setminus \{v \mid \exists e \in G \text{ s.t. } e_{\text{dst}} = v\}$   $\triangleright$  Select subgraph containing nodes without a parent
3:   if  $|H| = 1$  then return  $G$   $\triangleright z_v \in \{0, 1\} \forall v \in G; \sum_v z_v = k$ 
4:   else if  $|H| \geq 2$  then
5:      $A \subsetneq G \in \binom{H}{\lfloor \frac{|H|}{2} \rfloor}$ 
6:      $B \leftarrow G \setminus A$ 
7:      $\text{pairs} \leftarrow \{(a_i, b_i) \in A \times B \mid i \in \mathcal{I}\}$ 
8:      $H' \leftarrow (V = \emptyset, E = \emptyset)$ 
9:     for  $(v_i, v_j) \in \text{pairs}$  do
10:       $H' \leftarrow H' \cup \{v_i, v_j\}$ 
11:       $H' \leftarrow H' \cup \{v'; e_{v', v_\alpha} \mid \alpha \in \{i, j\}\}$ 
12:       $X_i, X_j \leftarrow \text{SIMPLIFY}(p_{v_i}, p_{v_j})$   $\triangleright$  Defined in Algorithm 9
13:       $z_{v_i} \leftarrow X_i$   $\triangleright$  If  $X_i$  was fixed,  $z_{v_i} \in \{0, 1\}$ 
14:       $z_{v_j} \leftarrow X_j$   $\triangleright$  If  $X_j$  was fixed,  $z_{v_j} \in \{0, 1\}$ 
15:      if  $z_{v_i} \in \{0, 1\}$  then
16:         $p_{v'} \leftarrow X_j$ 
17:      else  $p_{v'} \leftarrow X_i$ 
18:      $F \leftarrow G \cup H'$   $\triangleright \forall v \in G \cap H', \text{ update attribute values per } H'$ 
19:     return SAMPLE(F)
```

---

### A.4.3 Additional Experimental Details

In this section, we discuss additional details of our empirical study in Section 2.2.5. We provide a description and pseudocode of the heuristic policies (A.4.3.1), discuss our choice of fairness metric (A.4.3.2), and provide details of our Synthetic dataset (A.4.3.3).

All results presented in this chapter are bootstrapped over 100 simulation iterations, with a time horizon  $T = 180$ , cohort size  $N = 100$ , and budget  $k = 20$ , unless otherwise noted. We utilize seeds to ensure reproducible variation for each randomized parameter, including actualized transitions in each simulation. We have run simulations on an Intel(R) Core i7 CPU with 16Gb of RAM. Simulations are configurable via configuration files; runs are trivially parallelizable via these configuration files.

#### A.4.3.1 Heuristic Algorithms

In Section 2.2.5, three heuristics based on the THRESHOLD WHITTLE algorithm are introduced:  $H_{\text{FIRST}}$ ,  $H_{\text{LAST}}$ , and  $H_{\text{RAND}}$ . Here, we go into more detail and provide pseudocode.

**Definition A.4.2.** Within the context of Algorithm 11, we define a *constrained pull* to be one that is executed to satisfy an integer periodicity constraint. Only arms that have not yet been pulled the required number of times within the  $\nu$ -length interval are available; other arms are excluded from consideration, unless *all* arms have already satisfied their constraints. In this case, all arms are available to be pulled.

If a pull is not constrained, we say it is *unconstrained* or *residual*.

The  $H_{\text{FIRST}}$  heuristic requires that all constrained pulls must occur at the start of the interval.

This implies that the first  $N/k$  timesteps in each interval are dedicated to pulling all  $N$  arms.

The  $H_{\text{LAST}}$  heuristic requires that all constrained pulls must occur at the end of the interval. Unlike the  $H_{\text{FIRST}}$  heuristic, not all arms will necessarily be pulled in the last  $N/k$  timesteps, as some arms will have already satisfied their constraint earlier in the interval via unconstrained pull(s). These leftover constrained pulls function as unconstrained pulls, per Definition A.4.2.

The  $H_{\text{RAND}}$  heuristic chooses random positions within the interval for constrained pulls to occur. Similarly to the  $H_{\text{LAST}}$  heuristic, some of the later constrained pulls may become unconstrained pulls if all arms have already satisfied their constraint earlier in the interval.

---

**Algorithm 11** Periodicity Constraint-Enforcing Heuristic Based on TW

---

```

1: procedure SIMULATION( $A, T, \nu, k$ )
2:   for interval  $\in [0, T]$  with step size  $\nu$  do
3:      $C_{\text{interval}} \leftarrow \emptyset$  ▷  $C_{\text{interval}} := \text{arm(s) with constraint satisfied during the interval}$ 
4:
5:     for  $a \in A$  do
6:        $a.\text{last observed state} \leftarrow 1$ 
7:        $a.\text{time since pull} \leftarrow 1$ 
8:
9:     for  $t \in T$  do
10:       $i \leftarrow \text{GetInterval}(t)$ 
11:
12:      if  $t$  is a constrained pull  $\wedge C_i \subsetneq A$  then ▷ Consider arms with constraint not yet satisfied in interval
13:         $A' \leftarrow \{a | a \in A \setminus C_i\}$ 
14:      else if  $t$  is a residual pull  $\vee C_i = A$  then ▷ Consider all arms
15:         $A' \leftarrow A$ 
16:
17:         $A'_k \leftarrow \text{SelectTopK}(A', k, t)$  ▷ Select  $k$  arms with highest Whittle index
18:         $C_i \leftarrow C_i \cup A'_k$ 
19:
20:      for  $a \in A$  do
21:         $s_{t+1}(a) \leftarrow \text{UpdateState}(a)$  ▷ Update each arm's state using belief
return

```

---

### A.4.3.2 Fairness Metric Choices

It is not immediately obvious which evaluation metric(s) best indicate whether we have improved distributive fairness. While constraint satisfaction itself is a logical candidate, it is

Boolean-valued at the arm level, and thus does not reflect *to what extent* a policy fairly allocates pulls. Even if we were to report population-level constraint satisfaction (i.e., by noting the proportion of arms for which a given fairness constraint is satisfied, either over the course of a single simulation, or in expectation over a set of simulation iterations), this would be tautologically biased in favor of PROBFair and the THRESHOLD WHITTLE-based heuristics, which explicitly encode constraint satisfaction. This observation motivates us to consider proxy metrics, including the price of fairness (PoF), the Herfindahl–Hirschman Index (HHI), and the earth mover’s distance (EMD).

Price of Fairness. Consider *price of fairness*, defined formally as:

$$\text{POF}_{\text{TW}}(\text{ALG}) := \frac{\mathbb{E}_{\text{TW}}[R(\cdot)] - \mathbb{E}_{\text{ALG}}[R(\cdot)]}{\mathbb{E}_{\text{TW}}[R(\cdot)]} \quad (\text{A.17})$$

Price of fairness is the relative loss in total expected reward associated with following a distributive fairness-enforcing policy, as compared to THRESHOLD WHITTLE [50]. A small loss ( $\sim 0\%$ ) indicates that fairness has a small impact on total expected reward; conversely, a large loss means total expected reward is sacrificed in order to satisfy the fairness constraints.

**Lemma A.4.3.** *Price of fairness is inversely proportional to intervention benefit.*

*Proof.* The statement in Lemma A.4.3 is equivalent to the statement “Given  $y, z > 0$ , there exists  $\alpha \in \mathbb{R}$  such that  $\frac{x-z}{y-z} = \alpha \frac{z-x}{z}$  for all  $x > 0$ ”. Here  $x = \mathbb{E}_{\text{ALG}}[R(\cdot)]$ ,  $y = \mathbb{E}_{\text{NoAct}}[R(\cdot)]$ , and  $z = \mathbb{E}_{\text{TW}}[R(\cdot)]$ . Consider  $\alpha = \frac{-z}{y-z}$ . Then  $\alpha \frac{z-x}{z} = \frac{x-z}{y-z}$ . Thus, for any algorithm ALG,  $\text{POF}_{\text{TW}}(\text{ALG}) \propto \text{IB}_{\text{NoAct}, \text{TW}}(\text{ALG})^{-1}$ .  $\square$

Herfindahl–Hirschman Index (HHI). The Herfindahl–Hirschman Index (HHI) [150], is a statistical measure of concentration useful for measuring the extent to which a small set of arms receive a large proportion of attention due to an unequal distribution of scarce pulls [151]. It is defined as:

$$\text{HHI}(\text{ALG}) := \sum_{i=1}^N \left( \frac{1}{kT} \sum_{t=1}^T a_t^i \right)^2 \quad (\text{A.18})$$

HHI ranges from  $1/N$  to 1; higher values indicate that pulls are concentrated on a small subset of arms. However, HHI is an imperfect evaluation metric for addressing our prioritarian concern for arms that would be *deprived* of algorithmic attention (i.e., fail to receive any pulls) under THRESHOLD WHITTLE (see Appendix A.2). Since entries are squared, reducing  $u$  offers a more direct path to lowering HHI than increasing  $\ell$ . However, reducing  $u$  will not accomplish our stated goal of guaranteeing each arm a strictly positive lower bound on the probability that it will receive a pull at any given timestep.

Earth Mover’s Distance The earth mover’s distance (EMD), or Wasserstein metric, is a measure of distance between two distributions. Specifically, we measure the distance of an algorithm’s distribution of cumulative pull allocations to a fair reference distribution, ROUND-ROBIN. Though differences in distances are meaningful, EMD does not directly map to our fairness desiderata. That is, a given level of fairness enforcement (e.g., as characterized by the hyperparameters  $\ell$  or  $\nu$ ) is not associated with a specific range of EMD values. Hence, our discussion of (normalized) earth mover’s distances in Section 2.2.5 focuses on relative differences between policies.

### A.4.3.3 Synthetic Dataset

*Conjecture A.4.4.* The set of forward (reverse) threshold-optimal arms are a subset of the set of concave (strictly convex) arms for the local reward function we consider,  $r(s) = s$ .

Mate et al. [34] provide conditions for threshold optimality. First, the arm must satisfy the structural constraints (Section 2.1) and the *indexability* condition  $P_{1,1}^0 - P_{0,1}^0 + P_{1,1}^1 - P_{0,1}^1 \leq 1$ . Then, the following inequalities determine forward (reverse) threshold optimality:

$$\begin{cases} P_{1,1}^0 - P_{0,1}^0 \geq P_{1,1}^1 - P_{0,1}^1 & \text{forward threshold-optimal} \\ P_{1,1}^0 - P_{0,1}^0 \leq P_{1,1}^1 - P_{0,1}^1 & \text{reverse threshold-optimal} \end{cases} \quad (\text{A.19})$$

We conjecture that these conditions necessarily imply the conditions for concavity, repeated here for convenience:

$$\begin{cases} P_{0,1}^0 \leq \frac{(P_{0,1}^1 - P_{0,1}^0)(1 - P_{1,1}^0 + P_{0,1}^0)}{P_{1,1}^0 - P_{1,1}^1 - P_{0,1}^0 + P_{0,1}^1} & \text{concave} \\ P_{0,1}^0 > \frac{(P_{0,1}^1 - P_{0,1}^0)(1 - P_{1,1}^0 + P_{0,1}^0)}{P_{1,1}^0 - P_{1,1}^1 - P_{0,1}^0 + P_{0,1}^1} & \text{strictly convex} \end{cases} \quad (\text{A.20})$$



## Appendix B: Networked Restless Bandits with Positive Externalities

### B.1 Heuristic Policy

In this section, we provide additional details related to the heuristic policy we introduce in Section 2.3.3, including pseudocode for subroutines and complete proofs for theorems.

#### B.1.1 Pseudocode

##### B.1.1.1 Construct $G'$

Recall from Section 2.3.3.1 that GRETA takes as input a pre-constructed augmented graph,  $G' = (V', E')$ , where  $V'$  and  $E'$  represent augmented vertex and edge sets, respectively (Algorithm 4, line 1). Algorithm 12 provides pseudocode for the construction of this augmented graph:

---

**Algorithm 12** Construct  $G'$ 

---

```
1: procedure CONSTRUCT( $V, E$ )
2:    $V' \leftarrow V \cup \{-1\}$ 
3:    $E' \leftarrow E \cup \{(u, -1) | u \in V\}$ 
4:    $G' = (V', E')$ 
5:   return  $V', E', G'$ 
```

---

### B.1.1.2 PULLONLY subroutine

Here we present pseudocode for the PULLONLY subroutine called in line 6 of GRETA (Algorithm 4).

---

#### Algorithm 13 Cumulative subsidy of top-2 arms (pulls)

Note: all sorts are descending; arrays are zero-indexed.

---

```

1: procedure PULLONLY( $E', b \in \mathbb{N}, W_2$ )
2:    $V'_2 \leftarrow \{u \mid (u, v) \in E' \wedge v = -1\}$ 
3:    $b \leftarrow \min(|V'_2|, b)$ 
4:    $\text{pull\_vals} \leftarrow \text{SORT}(g : u \in V'_2 \mapsto (u, W_2^u))$ 
5:    $\hat{a}_2 : u \in \pi_\ell(\text{pull\_vals}[ :b]) \mapsto 2$ 
6:    $\nu_2 \leftarrow \sum_{i=0}^b \pi_r(\text{pull\_vals}_i)$ 
7:
    $\triangleright$  Return top- $b$  arm-actions and their cumulative subsidy
8:   return  $\hat{a}_2, \nu_2$ 

```

---

### B.1.1.3 MODACTSB subroutine

Here we present pseudocode for the MODACTSB subroutine called in line 9 or 12 (depending on the IF/ELSE in line 8) of GRETA (Algorithm 4), as well as in line 18 of the MP subroutine (Algorithm 6).

---

#### Algorithm 14 Update $\vec{a}_t$ and budget per max-value action(s)

```

1: procedure MODACTSB( $G', C, \hat{a}^*, \vec{a}_t, B'$ )
    $\triangleright$  For every vertex  $u$  with an updated action:
2:   for  $u \in \text{dom}(\hat{a}^*)$  do
3:     if  $\hat{a}^*(u) = 1$  then
        $\triangleright$  New action is a message; update  $\vec{a}_t, B'$ 
4:        $B' \leftarrow C(1)(1 - \mathbb{1}(\hat{a}^*(u) = 1 \vee u = -1))$ 
5:        $\vec{a}_t^u \leftarrow 1$ 
6:     else if  $\hat{a}^*(u) = 2$  then
        $\triangleright$  New action is a pull; update  $\vec{a}_t, B'$ 
7:        $B' \leftarrow C(2)(1 - \mathbb{1}(\hat{a}^*(u) > 0)) +$ 
8:          $\mathbb{1}(\hat{a}^*(u) = 1)(C(2) - C(1))$ 
9:        $\vec{a}_t^u \leftarrow 2$ 
        $\triangleright$  If  $C(1) = 0$ , message every  $v \in \mathcal{N}'_{\text{out}}(u)$ 
10:      if  $\psi = 0$  then
11:        for  $v \in \mathcal{N}'_{\text{out}}(u)$  do
12:           $\vec{a}_t^v \leftarrow \max(\vec{a}_t^v, 1)$ 
13:   return  $\vec{a}_t, B'$ 

```

---

### B.1.1.4 UPDATEG subroutine

---

**Algorithm 15** Update  $E'$  and  $G'$ 


---

```

1: procedure UPDATEGRAPH( $V', E', \hat{a}^*, E'_{\circ}$ )
2:   for  $u \in \text{dom}(\hat{a}^*)$  do
3:     if  $\hat{a}^*(u) = 2$  then
4:        $E'_{\circ} \leftarrow E'_{\circ} \cup \{e_{s,t} \mid e_{s,t} \in E' \wedge$ 
5:          $(t = u \vee (s = u \wedge t = -1))\}$ 
6:    $E' \leftarrow E' \setminus E'_{\circ}$ 
7:   return  $E', G' = (V', E')$ 

```

---

## B.1.2 Proofs

### B.1.2.1 Bounding expected reward

Here, we begin with Remark B.1.1, and proceed to prove Theorem 2.3.1 (restated for convenience below).

*Remark B.1.1.* For any restless arm  $i \in [n]$  whose transition matrix entries satisfy the structural constraints introduced in Section 2.3.2.3, action space  $\mathcal{A} := \{0 : \text{no-act}, 1 : \text{message}, 2 : \text{pull}\}$ , and non-decreasing local reward function,  $r : \mathcal{S} \rightarrow \mathbb{R}$ , we have:

$$\forall (a, a') \in \mathcal{A} \times \mathcal{A}, a < a' \rightarrow \mathbb{E}[r_t^i | s_t^i, a_t] < \mathbb{E}[r_t^i | s_t^i, a'_t]$$

**Theorem 2.3.1.** For a given set of  $[n]$  restless or collapsing arms with transition matrices satisfying the structural constraints outlined in Section 2.3.2.3, corresponding directed graph,  $G = (V, E)$ , budget  $B \in \mathbb{R}_{\geq 0}$ , non-decreasing local reward function,  $r : \mathcal{S} \rightarrow \mathbb{R}$ , cumulative reward function,  $R$ , and cost vector  $\vec{c} = [0, \psi, 1]$  such that  $\psi \in [0, 1)$ , we have:  $\mathbb{E}_{\text{TW}}[R] \leq \mathbb{E}_{\text{GH}, \psi > 0}[R] \leq$

$\mathbb{E}_{\text{GH},\psi=0}[R]$

*Proof.* The *first inequality*,  $\mathbb{E}_{\text{TW}}[R] \leq \mathbb{E}_{\text{GH},\psi>0}[R]$ , follows directly from how GRETA constructs each  $\vec{a}_t$ . GRETA’s PULLONLY subroutine (Algorithm 13) computes candidate action vector  $\hat{a}_2$ , which represents the actions we would take when following graph-agnostic TW for each budget chunk,  $b \leftarrow \min(B', 2)$ .

Since we do not execute  $\vec{a}_t$  until we are finished constructing it—i.e., until we have run out of budget or edges, per the WHILE-loop termination logic in line 8 of Algorithm 4, the arms’ states don’t change from while loop iteration  $i$  to  $i + 1$ . Thus, per the inequality on line 11, GRETA will return the same mapping from arms to actions as that returned by graph-agnostic TW with  $k = \lfloor B \rfloor$  if  $\bigwedge_i \mathbb{1}(\nu_2^{i,(\cdot)} \geq \nu_{(1,2)}^{i,(\cdot)})$ , and a mapping containing pulls and at least one message otherwise —i.e., when it is strictly advantageous to do so.

The *second inequality*,  $\mathbb{E}_{\text{GH},\psi>0}[R] \leq \mathbb{E}_{\text{GH},\psi=0}[R]$ , follows directly from Remark B.1.1 and the fact that when  $\psi = 0$ , we can message at least as many arms as we can when  $\psi > 0$ . Let  $\mathcal{M}_t^u$  represent the set of *potential* messages that is “unlocked” when a given node  $u$  receives a pull at time  $t$ . The cardinality of  $\mathcal{M}_t^u$  can be bound as follows:

$$|\mathcal{M}_t^u| = \begin{cases} |\mathcal{N}'_{\text{out}}(u)|, & \text{if } \psi = 0 \\ \min(\lfloor b/\psi \rfloor, |\mathcal{N}'_{\text{out}}(u)|), & \psi > 0 \end{cases} \quad (\text{B.1})$$

This is because when  $\psi = 0$ , for each node  $u$  that we pull at time  $t$ , we message *every* one-hop outdegree neighbor  $v \in \mathcal{N}'_{\text{out}}(u)$  that has not already received a pull or message at time  $t$  (see Algorithm 14, lines 10-12). In contrast, when  $\psi > 0$ , we must take the budget constraint into account.

The original claim follows from the transitive property.  $\square$

### B.1.2.2 Computational complexity

Here, we prove that GRETA is efficiently computable in time polynomial in its inputs. We begin by introducing several lemmas related to the runtime of GRETA's subroutines, and then proceed to prove Theorem 2.3.2 (restated for convenience below, immediately preceding its corresponding proof).

*Remark B.1.2.* The CONSTRUCT subroutine (Algorithm 12) used to construct the augmented graph,  $G'$ , has computational complexity  $O(|V|)$ , since we insert the dummy vertex,  $-1$ , and a directed edge  $(u, -1)$  for each  $u \in V$ .

**Lemma B.1.3.** *Using graph-agnostic THRESHOLD WHITTLE to pre-compute the Whittle index for each vertex-active action combination  $(v, \alpha) \in V' \times \mathcal{A} \setminus \{0\}$  has time complexity  $O(|V| |\mathcal{A}| |\mathcal{S}|^2 T)$ .*

*Proof.* Per Mate et al. [20], THRESHOLD WHITTLE has time complexity  $O(|\mathcal{S}|^2 T)$  per arm-active action. There are  $N = |V|$  arms and  $|\mathcal{A} \setminus \{0\}|$  active actions. Thus, Algorithm 3 has time complexity  $O(|V| |\mathcal{A}| |\mathcal{S}|^2 T)$ .  $\square$

**Lemma B.1.4.** *The PULLONLY subroutine (Algorithm 13) has time complexity  $O(|V'| \log |V'|)$ .*

*Proof.* Line 2 of Algorithm 13 has cost  $O(|V'|)$ . Lines 3, 5, and 6 have cost  $O(1)$ . The cost of this subroutine is thus dominated by the cost of sorting the Whittle index values (line 4), which is  $O(|V'| \log |V'|)$ .  $\square$

**Lemma B.1.5.** *The EDGEINDICES subroutine (Algorithm 7) has time complexity  $O(|V'| \log |V'|)$ .*

*Proof.* Lines 2 and 4 of Algorithm 7 have time complexity  $O(1)$ . The FOR-loop on lines 5-9 has time complexity  $O(|V'|)$ , since in the worst case,  $|\mathcal{N}'_{\text{out}}(u)| = |V'|$ . Thus, the time complexity of this subroutine is dominated by the  $O(|V'| \log |V'|)$  cost of sorting the Whittle index values (line 3).  $\square$

**Lemma B.1.6.** *For  $\psi \in [0, 1)$ , the MODACTSB subroutine (Algorithm 14) has time complexity  $O(|V'|)$  if  $\psi > 0$  and  $O(|V'|^2)$  otherwise.*

*Proof.* In the FOR-loop contained in lines 2-12 of Algorithm 14, we iterate over each vertex  $u$  with an updated action (i.e.,  $\forall u \in \text{dom } \hat{a}^*$ ), with cost  $O(|V'|)$ . In lines 3-9, we decrement our remaining budget,  $B'$ , and update our candidate action vector; each of these operations have time complexity  $O(1)$ . If  $\psi > 0$ , lines 10-12 are not executed. Conversely, if  $\psi = 0$ , lines 10-12 are executed, and we iterate over  $u$ 's one-hop, outdegree neighbors with cost  $O(|V'|)$ . Thus, the time complexity of this subroutine is  $O(|V'|)$  if  $\psi > 0$  and  $O(|V'|^2)$  otherwise.  $\square$

**Lemma B.1.7.** *The UPDATEGRAPH subroutine (Algorithm 15) has time complexity  $O(|V'| |E'|)$ .*

*Proof.* In the FOR-loop contained in lines 2-5 of Algorithm 15, we iterate over each vertex  $u$  with an updated action (i.e.,  $\forall u \in \text{dom } \hat{a}^*$ ). Lines 4-5 and 6 each have cost  $O(|E'|)$ . Thus, the time complexity of this subroutine is  $O(|V'| |E'|)$ .  $\square$

**Lemma B.1.8.** *Let  $\mathcal{W}_{MP}$  represent the number of WHILE-loop iterations that occur during any given call to the MSGPULL subroutine (i.e., Algorithm 6, lines 7-22). For  $\psi \in [0, 1)$ , we can upper-bound  $\mathcal{W}_{MP}$  as follows:*

$$\begin{cases} O(\min(|E'|, \lfloor \frac{b-2}{\psi} \rfloor)), & \text{if } \psi > 0 \\ O(|V'|), & \text{otherwise} \end{cases}$$

*Proof.* There are two cases:  $\psi > 0$  and  $\psi = 0$ . We consider each below:

*Case 1.*  $\psi > 0$ : For any given arm  $i \in [n]$ , and WHILE-loop iteration  $j > 0$ , let an *action upgrade* represent a modification to  $\hat{a}_{(1,2)}$  such that  $\hat{a}_{(1,2)}^{(i,j)} > \hat{a}_{(1,2)}^{(i,j-1)}$ . The small- $b$  budget is initialized = 2, and is strictly decreasing in the number of action upgrades when  $\psi > 0$ . The worst case from a time complexity perspective will be when our only upgrades are messages, since  $C(\text{pull}) > \psi$  for  $\psi \in [0, 1)$ —i.e.,  $\lfloor \frac{b-2}{\psi} \rfloor$  iterations.

Additionally, at each iteration we either: (a) have at least one cost-feasible action upgrade, which results in the removal of at least one edge from  $E''$  when we call UPDATEG in line 19 (note:  $E''$  is initialized in line 2 as a copy of  $E'$ , so it is always the case that  $|E''| \leq |E'|$ ); or (b) do not have any cost-feasible action upgrades remaining, in which case, the WHILE-loop terminates when we call GETCOST in line 7. Thus, we can conduct at most  $\min(|E'|, \lfloor \frac{b-2}{\psi} \rfloor)$  iterations before breaking.

*Case 2.*  $\psi = 0$ : Each time we select to pull vertex  $u$  in lines 14-22, we message *every* vertex  $v \in \mathcal{N}'_{\text{out}}(u)$  that is currently slated to receive a no-act (i.e., when we call the MODACTSB subroutine on line 18). The next time we encounter  $u$  in the FOR-loop on line 8, the construction of  $\mathcal{N}'_{\text{out}}(u)$  on line 9 will yield  $\emptyset$ , since we include only neighbors for which  $\hat{a}_{(1,2)} = 0$ . This will hold for each vertex  $\in V''$ , so we will break on line 13 after at most  $O(|V'|)$  iterations of the WHILE-loop.

□

**Lemma B.1.9.** *For notational convenience, let  $\xi = \mathbb{1}(\psi > 0) \times \min(|E'|^2, \lfloor \frac{b-2}{\psi} \rfloor |E'|) + \mathbb{1}(\psi = 0) \times |V'| |E'|$ . For  $\psi \in [0, 1)$ , the MSGPULL subroutine (Alg. 6) has time complexity:*

$$\begin{cases} O(\max(\xi|V'|^2 \log |V'|, \xi|V'||E'|^2)), & \text{if } \psi > 0 \\ O(\max(\xi|V'|^2 \log |V'|, \xi|V'||E'|^2, \xi|E'||V'|^2)), & \text{otherwise} \end{cases}$$

*Proof.* Regardless of the value of  $\psi$ , lines 2-6 of Algorithm 6 are dominated by the cost of constructing  $G''$  (line 2), which has cost  $O(|V'| + |E'|)$ . The WHILE-loop termination check we perform in line 7 has time complexity  $O(|E'|)$  since we call GETCOST (Algorithm 5) for each edge  $e \in E''$ .

In the FOR-loop contained in lines 8-10, we loop over each vertex  $v \in V''$  (line 8;  $O(|V'|)$ ), collect the one-hop, outdegree neighbors currently slated to receive a `no-act` (line 9;  $O(|E'|)$ ), and then call EDGEINDICES. Thus, per Lemma B.1.5, the time complexity of lines 8-10 is  $O(\max(|V'|^2 \log |V'|, |V'||E'|))$ .

The cost to sort the edge index values in line 11 is  $O(|E'| \log |E'|)$ ; the termination check on line 12 has cost  $O(1)$ .

In the FOR-loop contained in lines 14-22, we iterate over edge index values (line 14;  $O(|E'|)$ ). Lines 15-16 and 20-21 each have cost  $O(1)$ . Per Lemma B.1.6, the call to MODACTSB (Algorithm 14) in line 18 has cost  $O(|V'|)$  if  $\psi > 0$  and  $O(|V'|^2)$  otherwise. Per Lemma B.1.7, the call to UPDATEG (Algorithm 15) in line 19 has time complexity  $O(|V'||E'|)$ . Thus, for  $\psi \in [0, 1)$ , lines 14-22 have time complexity  $O(|V'||E'|^2)$  if  $\psi > 0$  and  $O(\max(|V'||E'|^2, |V'|^2|E'|))$  otherwise.

Finally, we consider the subroutine in its entirety. For notational convenience, let  $\xi$  represent the outer cost of the WHILE-loop, where the maximum number of iterations is defined per



Lemma B.1.8:

$$\xi = \mathbb{1}(\psi > 0) \times \min(|E'|^2, \lfloor \frac{b=2}{\psi} \rfloor |E'|) + \mathbb{1}(\psi = 0) \times |V'| |E'|$$

Then, for  $\psi \in [0, 1)$ , the time complexity of the MSGPULL is:

$$\begin{cases} O(\max(\xi |V'|^2 \log |V'|, \xi |V'| |E'|^2)), & \text{if } \psi > 0 \\ O(\max(\xi |V'|^2 \log |V'|, \xi |V'| |E'|^2, \xi |V'|^2 |E'|)), & \text{otherwise} \end{cases}$$

□

**Theorem 2.3.2.** For convenience, let:  $\xi = \mathbb{1}(\psi > 0) \times \min(|E'|^2, \lfloor \frac{B}{\psi} \rfloor |E'|) + \mathbb{1}(\psi = 0) \times |V'| |E'|$ .

Then, for  $\psi \in [0, 1)$  and time horizon,  $T$ , the time complexity of GRETA is:

$$\begin{cases} O(\max(\xi^2 |V'|^2 \log |V'|, \xi^2 |V'| |E'|^2) T), & \text{if } \psi > 0 \\ O(\max(\xi^2 |V'|^2 \log |V'|, \xi^2 |V'| |E'|^2, \xi^2 |V'|^2 |E'|) T), \\ \text{otherwise} \end{cases}$$

*Proof.* Per Remark B.1.2 and Lemma B.1.3, the time complexity of constructing GRETA's inputs is dominated by precomputation of the Whittle indices, which has cost  $O(|V| |\mathcal{A}| |\mathcal{S}|^2 T)$ . Lines 2-3 of GRETA (Algorithm 4) have cost  $O(1)$ .

The WHILE-loop termination check we perform in line 4 has time complexity  $O(|E'|)$  since we call GETCOST (Algorithm 5) for each edge  $e \in E'$ . Line 5 has cost  $O(1)$ .

Inside the WHILE-loop (i.e., Algorithm 4, lines 4-13), the call to the MSGPULL subroutine on line 7 contributes the dominating cost. Per Lemma B.1.9, for  $\psi \in [0, 1)$ , this subroutine has

time complexity:

$$\begin{cases} O(\max(\xi|V'|^2 \log |V'|, \xi|V'||E'|^2)), & \text{if } \psi > 0 \\ O(\max(\xi|V'|^2 \log |V'|, \xi|V'||E'|^2, \xi|V'|^2|E'|)), & \text{otherwise} \end{cases}$$

where

$$\xi = \mathbb{1}(\psi > 0) \times \min(|E'|^2, \lfloor \frac{b=2}{\psi} \rfloor |E'|) + \mathbb{1}(\psi = 0) \times |V'||E'|$$

Next, we can bound the number of WHILE-loop iterations that occur during any given call to GRETA in a way that proceeds identically to the bound we establish for the MSGPULL subroutine's WHILE-loop in Lemma B.1.8, with one small modification: we replace  $b$  with the full budget,  $B$ , noting that  $b \leq B$ . This accounts for the presence of the  $\xi^2$  term in Theorem 2.3.2.

Finally, we note that we call GRETA once per timestep  $t$ . Thus, when we consider the time complexity over time horizon,  $T$ , the cost of computing  $\vec{a}_t$   $T$  times dominates the  $O(|V||\mathcal{A}||\mathcal{S}|^2 T)$  cost of precomputing the Whittle indices, and we are done.

□

## B.2 Additional empirical results

Here we provide additional empirical support for Theorem 2.3.1, reproduced below for convenience:

**Theorem 2.3.1.** *For a given set of  $[n]$  restless or collapsing arms with transition matrices satisfying the structural constraints outlined in Section 2.3.2.3, corresponding directed graph,  $G = (V, E)$ ,*

budget  $B \in \mathbb{R}_{\geq 0}$ , non-decreasing local reward function,  $r : \mathcal{S} \rightarrow \mathbb{R}$ , cumulative reward function,  $R$ , and cost vector  $\vec{c} = [0, \psi, 1]$  such that  $\psi \in [0, 1)$ , we have:  $\mathbb{E}_{\text{TW}}[R] \leq \mathbb{E}_{\text{GH}, \psi > 0}[R] \leq \mathbb{E}_{\text{GH}, \psi = 0}[R]$

We note that for any given directed graph,  $G = (V, E)$ , for fixed  $|V|$ , as  $|E|$  tends toward a complete graph (i.e.,  $|E| \rightarrow |V| * (|V| - 1)$ ), as long as we have budget  $B \geq C(2) + C(1)$  (or, equivalently,  $B \geq 1 + \psi$ ), we can pull *any* arm and gain the ability to message *any other* arm. Thus, in the complete graph setting, we can modify the constrained optimization-based approach represented by Equation 2.10 (reproduced for convenience below) to: (1) include the requirement that at least one arm receive a pull (i.e., constraint c); and (2) remove the neighborhood constraint (i.e., constraint a), since it is guaranteed to be satisfied without being explicitly enforced.

$$\begin{aligned}
J(\mathbf{s}) &= \max_{\mathbf{X}} \left\{ \sum_{i=0}^{n-1} r^i(s^i) + \beta \mathbb{E}[J(\mathbf{s}'), \mathbf{X}] \right\} \\
\text{subject to } & \sum_{i=0}^{n-1} \sum_{j=0}^{|\mathcal{A}|-1} x_{i,j} \cdot c_j \leq B \\
(a) \quad & x_{i,1} \leq \bigvee_{i' \in \mathcal{N}_{\text{in}}(i)} x_{i',2} \quad \forall i \in [n] \\
(b) \quad & \sum_{j=0}^{|\mathcal{A}|-1} x_{i,j} = 1 \quad \forall i \in [n] \\
(c) \quad & \sum_{i=0}^{n-1} x_{i,2} \geq 1 \\
(d) \quad & \mathbf{X} \in \{0, 1\}^{n \times |\mathcal{A}|}
\end{aligned}$$

Thus, for fixed set of restless arms with cardinality  $|V|$ , and message cost,  $\psi$ , we can upper-bound  $\mathbb{E}_{\text{GH}, \psi}[R]$  by the expected reward achieved by the modified math program when the

graph in question is complete. Intuitively, this bound will become tighter as the cardinality of  $|E|$  is increased.

To empirically validate this claim, we consider a synthetic cohort of  $n = 100$  restless arms whose transition matrices are randomly generated in such a way so as to satisfy the structural constraints introduced in Section 2.3.2. We let  $T = 120$ ,  $B = 10$ , and  $\psi = 0.5$ . We can then construct graphs using this fixed vertex set but containing edge sets with varying cardinalities, expressed as a percentage of the number of edges the complete graph would contain.

More concretely, we define a set of six edge generation seeds, and use each seed to select subsets of edges uniformly at random, such that the subsets have cardinalities  $\in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$  of the complete graph. For each graph so constructed, we report unnormalized  $\mathbb{E}_\pi[R]$ , along with margins of error for 95% confidence intervals computed over 30 simulation iterations (see subfigures a-f). We observe that:

1. For each seed, and every value of  $|E|$ ,  $\mathbb{E}_{\text{TW},\psi}[R] \leq \mathbb{E}_{\text{GH},\psi}[R] < \mathbb{E}_{\text{MP},\psi}[R]$ .
2. We note that while GRETA's expected reward does not monotonically increase with  $|E|$  in every case, this is to be expected, since we are not guaranteed to get the same subset of edges we had for smaller percentage values as we increase  $|E|$ . This being said, expected reward is generally increasing with  $|E|$ , and as  $|E| \rightarrow |V|(|V| - 1)$ ,  $\mathbb{E}_{\text{GH},\psi}[R] \rightarrow \mathbb{E}_{\text{MP},\psi}[R]$ .

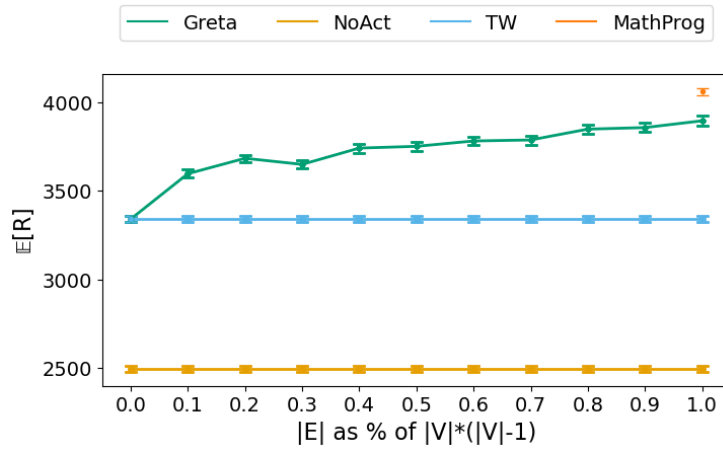


Figure B.1: Edge generation seed 1

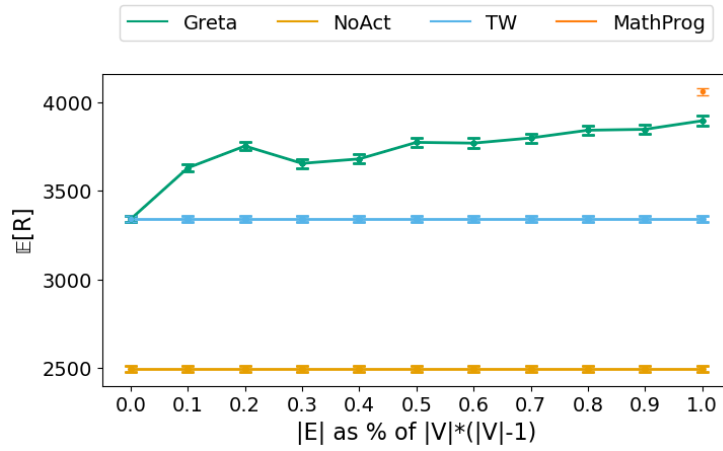


Figure B.2: Edge generation seed 2

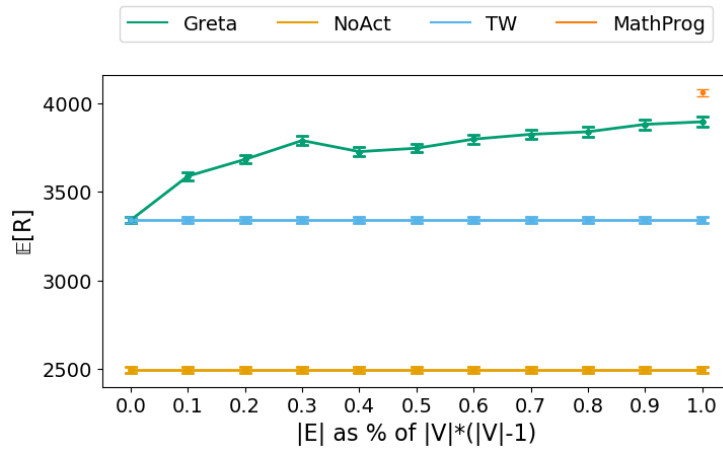


Figure B.3: Edge generation seed 3

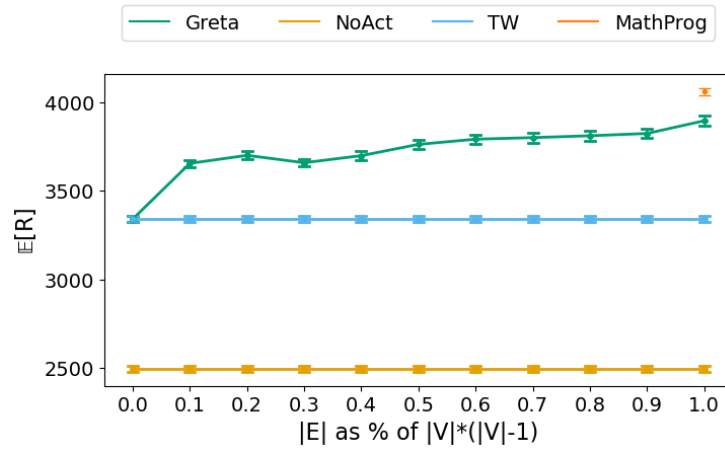


Figure B.4: Edge generation seed 4

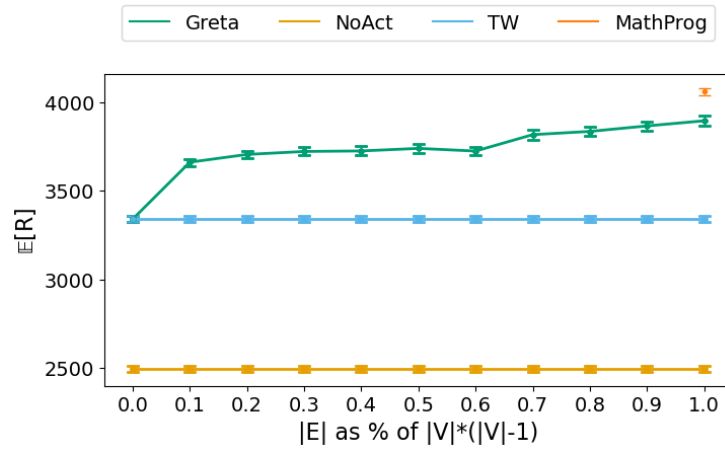


Figure B.5: Edge generation seed 5

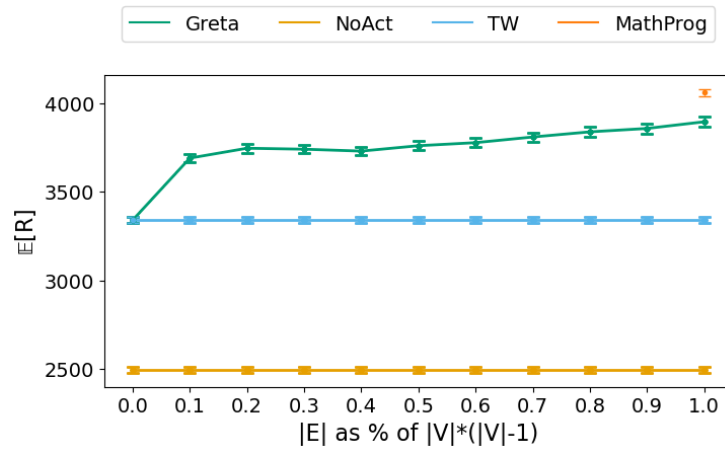


Figure B.6: Edge generation seed 6

### B.3 Additional experimental details

**Source code:** Our supplemental materials contain the source code and instructions needed to reproduce the experimental results we report in Section [2.3.4](#).

**Compute resources:** We ran all of our simulations on a MacBook Pro with a 2 GHz Quad-Core Intel(R) Core i5 CPU and 16 GB of RAM.

## Appendix C: Evaluating Model Performance Across Intersectional Subgroups

### C.1 Confidence intervals for standard estimates

We consider three methods for constructing confidence intervals for standard estimators  $Z_a$  at a given confidence level  $\gamma$  (e.g., 95%), or equivalently, at a significance level  $\alpha = 1 - \gamma$  (e.g., 5%).

Two of the methods are based on normal approximation and take form

$$[Z_a + q_{\alpha/2}\hat{\sigma}_a, Z_a + q_{1-\alpha/2}\hat{\sigma}_a],$$

where  $q_p$  is the  $p$ -th quantile of the standard normal distribution and  $\hat{\sigma}_a^2$  is an estimate of variance of  $Z_a$ . We consider either the pooled estimate of variance derived in Eq. (3.6), or the estimate  $(\hat{\sigma}_a^{\text{boot}})^2$  obtained by bootstrap on  $S_a$ . The third method uses bootstrap percentiles on  $S_a$ .

In Figure C.1, we compare coverage properties of the resulting confidence intervals on diabetes data. Confidence intervals constructed from pooled variance estimates are well-calibrated, with coverage closely matching their confidence level. The other two methods substantially undercover true values.



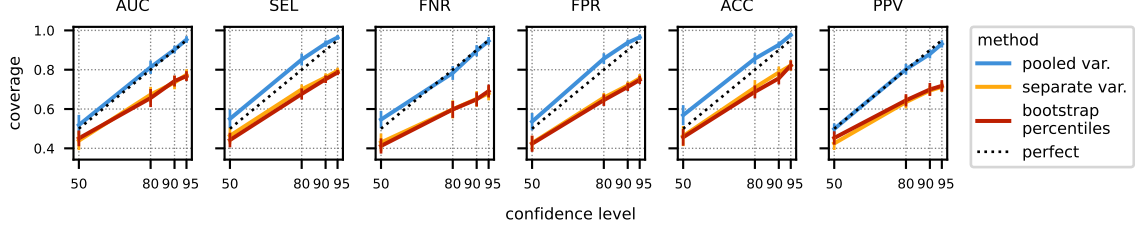


Figure C.1: *Comparison of methods for constructing confidence intervals for the standard estimator.* Showing coverage of confidence intervals constructed for six metrics on diabetes data, averaged over all groups and over 20 draws of evaluation dataset. Confidence intervals constructed from pooled variance are close to the perfect line (corresponding to coverage equal to confidence level). Confidence intervals derived from separately estimated variances undercover true values.

## C.2 Synthetic data generation

To generate synthetic data, we use the ground-truth diabetes dataset as described in Section 3.1.7.1, but with different values  $\hat{Y}$ . Specifically, we consider the performance metric  $m(p) = \mathbb{E}_p[\hat{Y}]$  (this is quite similar to the selection rate or word error rate) and generate  $\hat{Y}$  in such a way that ground-truth metric values  $\mu_a$  have a specific structure. We consider 4 different ground-truth structures, which we refer to as  $\text{model}_{age}$ ,  $\text{model}_{age+rc}$ ,  $\text{model}_{age \cdot rc}$ , and  $\text{model}_{expl}$ , depending which variables they depend on and how, with “+” denoting additive dependence and “.” presence of interactions:

Model name	Ground-truth value of $\mu_a$	Data-generating process
$\text{model}_{age}$	$\mu_a = 0.35 - 0.3 \cdot \phi_{age,40-60}^a$	$\hat{Y} = \text{Bernoulli}(\mu_A)$
$\text{model}_{expl}$	$\mu_a = -0.93 + 0.16 \cdot \mathbb{E}_{\mathcal{D}_a}[X_{number.diagnoses}]$	$\hat{Y} = \mathcal{N}(\mu_A, 0.1)$
$\text{model}_{age+rc}$	$\mu_a = 0.65 - 0.15 \cdot \phi_{age,40-60}^a - 0.45 \cdot \phi_{race,white}^a$	$\hat{Y} = \text{Bernoulli}(\mu_A)$
$\text{model}_{age \cdot rc}$	$\mu_a = 0.32 - 0.27 \cdot \phi_{age,40-60}^a \cdot \phi_{race,white}^a$	$\hat{Y} = \text{Bernoulli}(\mu_A)$

Model coefficients have been chosen to ensure that in all cases  $\mathbb{E}_{\mathcal{D}}[\hat{Y}] \approx 0.27$  and  $(\text{Var}_{\mathcal{D}}[\hat{Y}])^{1/2} \approx$

0.44.

## Appendix D: Identifying and Mitigating Clinical Natural Language Inference Artifacts

### D.1 Hypothesis-only Baseline Analysis

To conduct the analysis presented in Chapter 3.2.3, we take the MedNLI training dataset as input, and exclude the premise text for each training example. We cast the text of each training hypothesis to lowercase, but do not perform any additional preprocessing. We use an off-the-shelf `fastText` classifier, with all model hyperparameters set to their default values with the exception of `wordNgrams`, which we set equal to 2 to allow the model to use bigrams in addition to unigrams [110]. We evaluate the trained classifier on the hypotheses contained in the MedNLI dev and test datasets, and report results for each split.

### D.2 Lexical Artifact Analysis

To perform the analysis presented in Chapter 3.2.4.2, we cast each hypothesis string in the MedNLI training dataset to lowercase. We then use a `scispaCy` model pre-trained on the `en_core_sci_lg` corpus for tokenization and clinical named entity recognition (CNER) [113]. Next, we merge multi-token entities, using underscores as delimiters—e.g., “brain injury” → “brain\_injury”.

When computing token-class pointwise mutual information (PMI), we exclude tokens that appear less than five times in the overall training dataset’s hypotheses. Then, following Gururangan et al. [8], who apply add-100 smoothing to raw counts to highlight particularly discriminative token-class co-occurrence patterns, we apply add-50 smoothing to raw counts. Our approach is similarly motivated; our choice of 50 reflects the smaller state space associated with a focus on the clinical domain.

### D.3 Semantic Analysis of Heuristics

To perform the statistical analysis presented in Chapter 3.2.5, we take the premise-hypothesis pairs from the MedNLI training, dev, and test splits, and combine them to produce a single corpus. We use a `scispaCy` model pre-trained on the `en_core_sci_lg` corpus for tokenization and entity linking [113], and link against the Medical Subject Headings (MeSH) knowledge base. We take the top-ranked knowledge base entry for each linked entity. Linking against MeSH provides a unique concept id, canonical name, alias(es), a definition, and one or more MeSH tree numbers for each recovered entity. Tree numbers convey semantic type information by embedding each concept into the broader MeSH hierarchy <sup>1</sup>. We operationalize each of our heuristics with a set of MeSH-informed semantic properties, which are defined as follows:

1. **Hypernym Heuristic:** a premise-hypothesis pair satisfies this heuristic if specific clinical concept(s) appearing in the premise appear in a more general form in the hypothesis. Formally:  $\{(p, h) | \varphi(p) \subsetneq \varphi(h)\}$ . MeSH tree numbers are organized hierarchically, and increase in length with specificity. Thus, when a premise entity and hypothesis entity are

---

<sup>1</sup><https://meshb.nlm.nih.gov/treeView>

left-aligned, the hypothesis entity is a hypernym for the premise entity if the hypothesis entity is a substring of the premise entity. To provide a concrete example: *diabetes mellitus* is an *endocrine system disease*; the associated MeSH tree numbers are C19.246 and C19, respectively.

2. **Probable Cause Heuristic:** a premise-hypothesis pair satisfies this heuristic if: (1) the premise contains one or more MeSH entities belonging to high-level categories C (diseases), D (chemicals and drugs), E (analytical, diagnostic and therapeutic techniques, and equipment) or F (psychiatry and psychology); and (2) the hypothesis contains one or more MeSH entities that can be interpreted as providing a plausible causal or behavioral explanation for the condition, finding, or event described in the premise (e.g., smoking, substance-related disorders, mental disorders, alcoholism, homelessness, obesity).
3. **Everything Is Fine Heuristic:** a premise-hypothesis pair satisfies this heuristic if the hypothesis contains one or more of the same MeSH entities as the premise (excluding the *patient* entity, which appears in almost all notes) and also contains: (1) a negation word or phrase (e.g., *does not have*, *no finding*, *no*, *denies*); or (2) a word or phrase that affirms the patient’s health (e.g., *normal*, *healthy*, *discharged*).

For each heuristic, we subset the complete dataset to find pattern-satisfying premise-heuristic pairs. We use this subset when performing the  $\chi^2$  tests.

## D.4 Adversarial Filtering

When implementing `AFLite`, we follow Sakaguchi et al. [12]. We use a smaller training set size of  $m = 5620$ , but keep the remaining hyperparameters unchanged, such that the ensemble

consists of  $n = 64$  logistic regression models, the filtering cutoff,  $k = 500$ , and the filtering threshold  $\tau = 0.75$ .

## Appendix E: Overcoming Miscalibrated Conversational Priors in LLM-based Chatbots

### E.1 Helper LLM-based Classifiers

In this section, we provide descriptions, system messages, and validation results for each of the helper-LLM-based classifiers that we rely on throughout Chapter 4.1.

#### E.1.1 LLM-based query underspecification classifier

**Task description:** We use a helper LLM to map queries from our synthetic and real-world corpora to a set of class labels that describe the extent to which a given query is (or is not) underspecified, i.e., {CRITICAL UNDER, MINOR UNDER, SUFFICIENT}. We introduce these labels in Section 4.1.3.1, where we discuss them within the context of the labels we (i.e., human annotators) manually assign to a randomly sampled subset of the OpenAssistant corpus, but they also apply when the helper LLM is asked to classify queries. For convenience, we repeat them below:

- **CRITICAL UNDER:** One or more important factors upon which an answer to this query might depend are not specified or are unknown; (annotators agree that) it is difficult to provide a high-quality response without knowing these factors.
- **MINOR UNDER:** Less important factors that the query might depend on are not specified or are

unknown; however, it is possible to provide a high-quality response even without knowing these factors.

- **SUFFICIENT:** All important factors upon which an answer to this query might depend are sufficiently specified.

**Helper LLM prompt:** The prompt that we provide to the helper LLM for this task is shown below; it is also included within the `helper_task_system_messages.json` file contained within our supplemental materials.

```
1 {
2   "classify_queries_multiclass": "For each query in this list <list>{{input
3     .question}}</list>, assign exactly one of the following labels:\n
4     - sufficient: All important factors upon which an answer to this
5       query might depend are sufficiently specified.\n
6     - minor_under: One or more less important factors upon which an
7       answer to this query might depend are not specified or are
8       unknown; however, it is possible to provide a high-quality
9       response even without knowing these factors.\n
10    - critical_under: One or more important factors upon which an answer
      to this query might depend are not specified or are unknown; it
      is difficult to provide a high-quality response without knowing
      these factors.\n
      You MUST assign EXACTLY ONE label from the list above.\n
      Return your answer as a string.\n
      DO NOT answer any questions contained in the query, or include any
      expository text.\n
      The result should be DIRECTLY parsable in Python."
```

**Helper LLM configuration:** We use GPT-4 [118] for all query underspecification classification calls.

**Validation:** We use our synthetic query corpus to validate our use of this LLM-based underspecification classifier. As we describe in Section 4.1.3.2 and detail in Appendix E.3.1, by virtue of how we construct these queries, we control the number of attributes that are revealed. As such, we



have access to ground-truth underspecification labels defined in terms of the number of revealed attributes, referred to (with slight abuse of notation) as  $|q|$  in the mapping shown below. Note that  $|q|$  takes values in  $\{0, \dots, |\theta| - 1\}$  for masked queries, and will be equal to  $|\theta|$  for sufficiently specified queries, where  $|\theta|$  refers to the cardinality of the intent-specific attribute space.

$$q \mapsto \begin{cases} \text{CRITICAL UNDER} & |q| \leq 1, \\ \text{SUFFICIENT} & |q| = |\theta|, \\ \text{MINOR UNDER} & \text{otherwise.} \end{cases}$$

We evaluate our LLM-based query underspecification classifier on our synthetic query corpus, which contains 600 queries split across the following intent domains: movie recommendation, gift recommendation, and plant recommendation. We report performance metrics and confusion matrices over all synthetic queries, and broken down by intent-specific queries below.

	precision	recall	f1-score	support
critical_under	0.583	0.139	0.224	202
minor_under	0.443	0.472	0.457	398
sufficient	0.720	0.873	0.789	600
accuracy			0.617	1200
macro avg	0.582	0.495	0.490	1200
weighted avg	0.605	0.617	0.584	1200

Table E.1: Classifier performance: over all intents

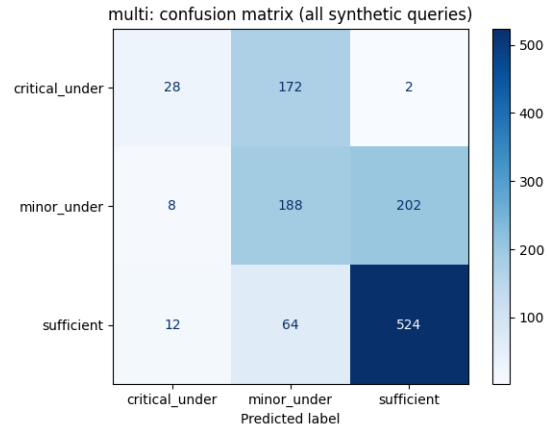


Figure E.1: Confusion matrix: all intents

	precision	recall	f1-score	support
critical_under	0.059	0.010	0.017	99
minor_under	0.284	0.214	0.244	196
sufficient	0.642	0.925	0.758	295
accuracy			0.536	590
macro avg	0.328	0.383	0.340	590
weighted avg	0.425	0.536	0.463	590

Table E.2: Classifier performance: movie recommendation queries

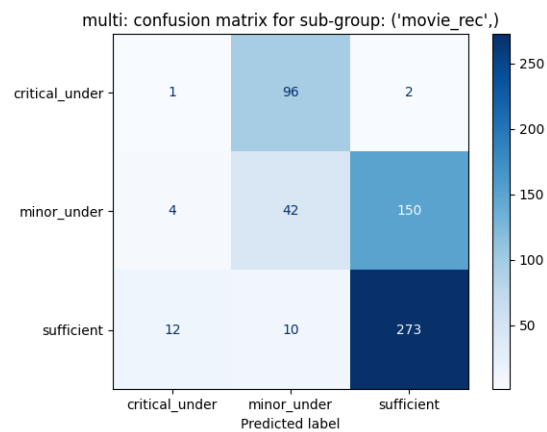


Figure E.2: Confusion matrix: movie recommendation queries

	precision	recall	f1-score	support
critical_under	0.852	0.371	0.517	62
minor_under	0.652	0.861	0.742	122
sufficient	0.928	0.908	0.918	184
accuracy			0.802	368
macro avg	0.811	0.713	0.725	368
weighted avg	0.824	0.802	0.792	368

Table E.3: Classifier performance: gift recommendation queries

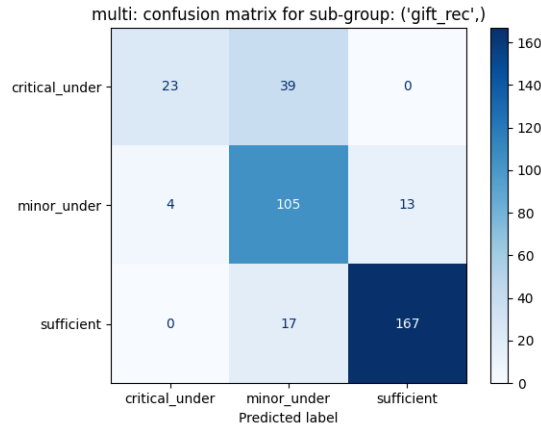


Figure E.3: Confusion matrix: gift recommendation queries

	precision	recall	f1-score	support
critical_under	1.000	0.098	0.178	41
minor_under	0.357	0.512	0.421	80
sufficient	0.683	0.694	0.689	121
accuracy			0.533	242
macro avg	0.680	0.435	0.429	242
weighted avg	0.629	0.533	0.513	242

Table E.4: Classifier performance: plant recommendation queries

## E.1.2 LLM-based Response Strategy classifier

**Task description:** We use a helper-LLM-based  $\tau$  classifier to map chatbot natural language responses to a set of labels intended to characterize a given response’s syntactic and semantic contents. We primarily use this classifier as a way of assessing whether and to what extent the

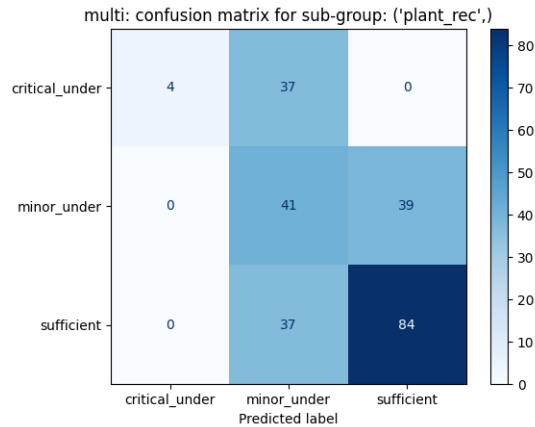


Figure E.4: Confusion matrix: plant recommendation queries

behaviors we seek to induce via modified system messages *actually* produce observable effects in the intended direction(s) and/or converge with the behavior of  $\pi^{\text{RLHF}}$ .

The label set we use for this classifier includes the set of response strategies that we refer to as  $\mathcal{T}$  throughout Chapter 4.1—i.e., {INTERROGATE, CLARIFY, HEDGE}, and also includes additional options—i.e., {DIRECT RESPONSE, REFUSE, MISCELLANEOUS, MISSING}. While we do not explicitly induce this latter set of behaviors, we need the DIRECT RESPONSE option to characterize the baseline system behavior and (more broadly) uncertainty-agnostic LLM responses in general. The REFUSE, MISCELLANEOUS, and MISSING options are needed to characterize the behavior of  $\pi^{\text{RLHF}}$  in open-domain settings such as the OpenAssistant corpus we consider, as well as to handle rare parsing/extraction errors that result in inadvertently blank LLM responses. The defining characteristics of each response strategy are presented/contained within the task system message in the next section.

**Helper LLM prompt:** The prompt that we provide to the helper LLM for this task is shown below; it is also included within the `helper_task_system_messages.json` file contained within our supplemental materials.

---

```

1 {
2   "sm_map_llmr_to_tau": str = "For each (query, response) in this list <list
   >{{input.pair}}</list>, map the response to exactly one of the
   following labels:\n
3
4   - interrogate: The response contains a large number (i.e., more than
   3) of follow-up questions and does NOT contain plausible
   responses conditioned on possible answers to these questions.\n
5   - clarify: The response contains a limited number (i.e., 3 or less)
   of follow-up questions and does NOT contain plausible responses
   conditioned on possible answers to these questions.\n
6   - hedging: The response does not commit to one specific answer but
   instead provides many plausible/possible/qualified answers,
   options, or conditions under which certain answers/options may or
   may not hold. It may also discuss (potentially conflicting)
   different view points without taking a definitive stance.\n
7   - direct_response: The response does NOT contain questions. The
   response does NOT contain multiple plausible answers, with
   corresponding descriptions of conditions or criteria under which
   each response would be suitable.\n
8   - refuse: The response contains an explicit or implicit refusal to
   answer. It may mention criteria which would be needed in order to
   provide an answer, but it does NOT contain plausible responses
   conditioned on these criteria.\n
9   - misc: The response may describe, summarize, or try to explain the
   query, or appear to follow instructions provided in the query (
   rather than answer an information-seeking request or ask
   clarifying questions).\n
10  - missing_response: The response is empty or blank.\n
11
12  You MUST assign exactly one label from the list above.\n
13  Return your answer as a string.\n
14  DO NOT answer any questions contained in the response, or include any
   expository text.\n
15  The result should be DIRECTLY parsable in Python."
16 }

```

---

**Helper LLM configuration:** We use GPT-4 [118] for all query underspecification classification calls.

**Validation:** We manually annotate  $\pi^{\text{RLHF}}$  responses to a subset of the OpenAssistant corpus that we consider, and use these human-annotator assigned ground-truth  $\tau$ s to validate our helper LLM-based  $\tau$  classifier. We note that some of our  $\tau$ s of interest are not sufficiently represented

amongst the  $\pi^{\text{RLHF}}$  responses (i.e., CLARIFY, HEDGE, INTERROGATE). We thus use our system-message-based interventions to induce responses for these strategies and include them (unlabeled) in our manually annotated subset. We report classification performance metrics and a confusion matrix below.

	precision	recall	f1-score	support
clarify	0.763	0.935	0.841	31
direct_response	0.822	0.903	0.861	154
hedging	0.925	0.649	0.763	57
interrogate	1.000	0.680	0.810	25
misc	0.154	0.250	0.190	8
refuse	0.667	0.400	0.500	5
accuracy			0.807	280
macro avg	0.722	0.636	0.661	280
weighted avg	0.831	0.807	0.808	280

Table E.5:  $\tau$ -classifier performance on human-annotated LLM responses to OpenAssistant queries

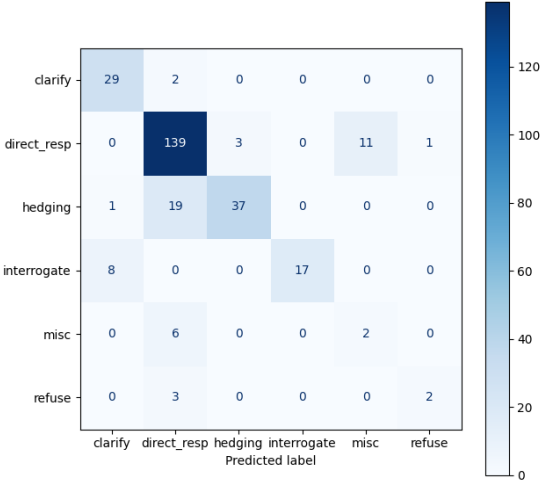


Figure E.5: Confusion matrix: annotated OpenAssistant query responses

## E.2 Response-strategy system messages

In this section, we report the system messages used to operationalize each response strategy  $\tau \in \mathcal{T}$  that we consider in Chapter 4.1, along with the data-agnostic interventions discussed in Section 4.1.4.1.

```
1 {
2   "response strategies":
3   {
4     "baseline": "None",
5     "interrogate": "When you receive a query, always interrogate the user
6       about all factors upon which the answer might depend---but that
7       have not been specified---so that you will be able to produce a
8       good answer.",
9     "clarify": "When you receive a query, always ask the user about up to 3
10      most relevant factors upon which the answer might depend---but
11      that have not been specified---so that you will be able to produce
12      a good answer.",
13     "hedge": "When you receive a query, always identify important factors
14      upon which the answer might depend---but that have not been
15      specified---and then provide a plausible response conditioned on
16      each of these factors.",
17   }
18   "data-agnostic interventions":
19   {
20     "CoT": "When you receive a query, ask yourself whether you have
21       sufficient information to provide a good answer, and then respond
22       accordingly.",
23     "clarify_flex": "When you receive a query, if the query depends on a set
24       of important factors that have not been specified, ask the user about
25       the most relevant factors that have not been specified so that you
26       will be able to produce a good answer; otherwise, respond directly."
27   }
28 }
```

## E.3 Motivating Experiments

### E.3.1 Synthetic Dataset Construction

We follow the synthetic query construction process outlined in Section 4.1.3.2 and formalized in Algorithm 16, along with the parameterized templates shown in Appendix E.3.2 to generate a corpus of sufficiently specified queries. Then, for each sufficiently specified query, we approximate a *partially specified* version by randomly selecting the number of attributes to omit,  $n \in \{1, \dots, |\Theta_i|\}$ . We keep the intent-declaring first sentence unchanged, shuffle the remaining attribute-sentences, draw a subset of sentences to *omit*—i.e., with cardinality  $n$ — and concatenate the remaining sentences.

---

**Algorithm 16** GENERATE SYNTHETIC QUERY

---

**function** GenQuery( $\mathcal{I}, \Theta_{(\cdot)}$ )

```
1:    $i \sim \mathcal{I}$  Draw intent
2:   for each  $\theta \in \Theta_i$  do Loop over intent attributes
3:     selected option(s)  $\leftarrow X' \sim X_\theta$ 
4:     template $_i \leftarrow$  template $_i \cup$  selected option(s)
5:   end for
6:    $q_s \leftarrow$  template $_i$  Sufficient query := filled-in template
7:    $n \sim U(\{1, \dots, |\Theta_i|\})$  Draw # of attrs to mask
8:    $\Theta_i^m \sim U(S_n(\Theta_i))$  Draw n masked attrs
9:    $\Theta_i^r := \Theta_i \setminus \Theta_i^m$  Determine revealed attrs
10:   $q_m \leftarrow$  concat( $\Theta_i^r$ ) Build masked version of  $q_s$ 
11:  return  $q_s, q_m$  Return sufficient & masked queries
```

---

### E.3.2 Synthetic Query Templates and Parameter Options

We use the intent-specific templates and parameter category-option mappings shown below in conjunction with the query construction procedure outlined in Appendix E.3.1 to generate our



synthetic queries:

### Intent-specific query templates:

```
1 {
2   "movie_rec": {
3     "value": "I am looking for a movie recommendation.\n
4     The genre should be \"{param_0}\".\n
5     It should have been released \"{param_1}\".\n
6     The intended audience includes \"{param_2}\".\n
7     The runtime should be \"{param_3}\".\n
8     Please provide movie recommendations that satisfy all of my
9     requirements."
10  },
11  "gift_rec": {
12    "value": "I am looking for a gift recommendation.\n
13    The recipient likes \"{param_0}\".\n
14    The recipient is \"{param_1}\" years old.\n
15    The recipient prefers gifts to be \"{param_2}\" in nature.\n
16    My budget to purchase the recipient a gift is in the \"{param_3}\"
17    range.\n
18    Please provide gift recommendations that satisfy all of my
19    requirements."
20  },
21  "plant_rec": {
22    "value": "I am looking for a house plant recommendation.\n
23    I prefer a plant that \"{param_0}\".\n
24    I'm willing to expend a \"{param_1}\" amount of effort to care for
25    the plant.\n
26    My house gets a \"{param_2}\" amount of natural light.\n
27    I live \"{param_3}\".\n
28    Please provide house plant recommendations that satisfy all of my
29    requirements."
30  }
31 }
```

### Intent-specific query parameter category-option mappings:

```
1 {
2   "movie_rec": {
3     "param_0": {"cat": "genre",
4               "opts": ["Action", "Adventure", "Animation", "Biography",
5                       "Comedy", "Crime", "Documentary", "Drama", "Fantasy",
6                       "Film Noir", "History", "Horror", "Musical", "
7                       Mystery", "Romance", "Sci-Fi", "Sport", "Superhero",
8                       "Thriller", "War", "Western"],
9               "max_sel_allowed": 2,
10              "pref_constraint_type": "set_valued"},
11    "param_1": {"cat": "release date",
```

```

8         "opts": ["in the 1980s", "in the 1990s", "in the 2000s",
9             "in the past few years"],
10        "max_sel_allowed": 1,
11        "pref_constraint_type": "numeric_range"},
12    "param_2": {"cat": "who will be watching",
13        "opts": ["children", "adults and children", "teenagers
14            younger than 17", "adults only"],
15        "max_sel_allowed": 1,
16        "pref_constraint_type": "set_valued"},
17    "param_3": {"cat": "runtime",
18        "opts": ["less than 90 minutes", "90-104 minutes", "105-1
19            19 minutes", "120 minutes or more"],
20        "max_sel_allowed": 1,
21        "pref_constraint_type": "set_valued"}
22    },
23    "gift_rec": {
24        "param_0": {"cat": "recipient interests",
25            "opts": ["outdoors", "crafts", "technology", "books", "
26                active play/sports/fitness", "food/cooking", "music
27                and arts", "apparel/fashion/style"],
28            "max_sel_allowed": 2,
29            "pref_constraint_type": "set_valued"},
30        "param_1": {"cat": "recipient age range",
31            "opts": ["3-5", "6-12", "13-17", "18-40", "41-60", "61+"
32                ],
33            "max_sel_allowed": 1,
34            "pref_constraint_type": "numeric_range"},
35        "param_2": {"cat": "recipient preferred gift type",
36            "opts": ["practical and everyday", "personalized and
37                sentimental", "adventurous and experience-driven", "
38                luxurious and pampering", "high-tech and innovative",
39                "creative and artistic", "sustainable and eco-
40                friendly"],
41            "max_sel_allowed": 1,
42            "pref_constraint_type": "set_valued"},
43        "param_3": {"cat": "giver budget",
44            "opts": ["less than $20", "$20-49", "$50-99", "$100-199"
45                , "$200+"],
46            "max_sel_allowed": 1,
47            "pref_constraint_type": "numeric_range"}
48    },
49    "plant_rec": {
50        "param_0": {"cat": "plant physical attributes",
51            "opts": ["has light green leaves", "has dark green leaves
52                ", "has multi-colored leaves", "yields edible produce
53                ", "produces a single bloom", "produces multiple
54                blooms", "is short in stature", "is tall in stature"]
55            ,
56            "max_sel_allowed": 2,
57            "pref_constraint_type": "set_valued"},
58        "param_1": {"cat": "level of effort",
59            "opts": ["low", "moderate", "high"],

```

```
47         "max_sel_allowed": 1,  
48         "pref_constraint_type": "set_valued"},  
49     "param_2": {"cat": "amount of natural light",  
50               "opts": ["low", "indirect", "moderate", "bright"],  
51               "max_sel_allowed": 1,  
52               "pref_constraint_type": "set_valued"},  
53     "param_3": {"cat": "household members",  
54               "opts": ["by myself", "with young children", "with pets",  
55                       , "with a roommate with seasonal allergies"],  
56               "max_sel_allowed": 1,  
57               "pref_constraint_type": "set_valued"}  
58 }
```

---

### E.3.3 Stable Diffusion Example: Policy-induced responses

Here, we present a query from the OpenAssistant dataset [119] as an example of a situation where CLARIFY may be preferred over DIRECT RESPONSE and HEDGE. Table E.6 presents the policy-induced responses generated by  $\pi^{\text{RLHF}}$ , HEDGE, and CLARIFY.

**Query text:** “Please give me a prompt for stable diffusion to generate a good-looking image.”

Response strategy	Response text
baseline	”Create an image that visually represents the process of stable diffusion, showcasing the gradual and even spread of particles throughout a medium. Use a color gradient to highlight the movement and distribution of particles, and ensure the overall aesthetic is pleasing and balanced.”
hedge	<p>The query is asking for a prompt to generate a good-looking image using stable diffusion. However, the important factors that are not specified include:</p> <ol style="list-style-type: none"> <li>1. The type of image or subject matter desired (e.g., landscape, portrait, abstract, etc.)</li> <li>2. The style or aesthetic the user prefers (e.g., realistic, surreal, impressionistic, etc.)</li> <li>3. The color scheme or palette the user prefers (e.g., warm, cool, monochromatic, etc.)</li> </ol> <p>Given these factors, here are some plausible responses:</p> <ol style="list-style-type: none"> <li>1. If the user wants a landscape image in a realistic style with a warm color scheme, a possible prompt could be: ”Generate a sunset over a serene lake with autumn foliage using stable diffusion.”</li> <li>2. If the user prefers a portrait in a surreal style with a cool color scheme, a possible prompt could be: ”Generate a portrait of a woman with flowing blue hair and eyes that mirror the ocean using stable diffusion.”</li> <li>3. If the user is interested in an abstract image in a monochromatic color scheme, a possible prompt could be: ”Generate an abstract image using stable diffusion that represents the concept of tranquility in shades of blue.”</li> </ol>
clarify	<ol style="list-style-type: none"> <li>1. What type of image are you looking to create? (e.g., abstract, realistic, geometric)</li> <li>2. Are there any specific colors or color schemes you would like to incorporate into the image?</li> <li>3. Do you have any specific themes or elements you want to be included in the image? (e.g., nature, space, patterns)</li> </ol>

Table E.6: Policy-induced responses to the stable diffusion example query.

## E.3.4 Simulating User Responses to LLM Questions

In Section 4.1.3.2, we discuss how we leverage a series of helper LLM calls to construct templated responses to LLM questions based on the overlap (or lack thereof) between the questions and the user’s true preferences,  $\theta$ . Here, we provide the system messages used in these helper calls. The first system message is for the extraction of recommended items and/or questions from LLM responses. The second system message helps us to construct a mapping from the LLM’s questions to the user’s true preferences,  $\theta$ , such that we can determine what subset of previously masked attributes can be “revealed” in the templated user response we construct.

```
1 {"extract_recs_and_questions":
2
3   "For each response in this list <list>{{input.\"" + f' {field_to_use}'
4     + "\"}}</list>, read the response carefully and:\n
5
6     1. Extract the titles of each and every movie recommendation that
7       appears; they may show up as a list of titles.\n
8       Do not extract any additional metadata but DO extract any
9       mentioned titles; represent each title as a string.\n
10
11      Format your answer for task 1 as shown below:\n
12
13      ["rec" for rec in recommended movies] OR [], ONLY if NO movie
14      recommendations appear.
15
16      2. Extract any questions that appear; they may be prefaced with a
17        request to specify preferences, and/or show up as a list of
18        questions.\n
19        DO extract ANY mentioned questions; represent each question as
20        a string.
21
22      Format your answer for task 2 as shown below:\n
23
24      ["question" for question in questions] OR [], ONLY if NO
25      questions appear.\n
26
27      Return your results as a dict:\n
28
29      {"recs": [response to task 1], "questions": [response to task
30        2]} \n
31
32      DO NOT answer any questions contained in the response, or include
```

```

24         ANY expository text.\n
        The result should be DIRECTLY parsable as a valid dict in Python
        .""""
25 }

1 {
2     "map_questions_to_thetas":
3
4         "You will receive a list containing sets of questions.\n
5         Each question is issued by an assistant to a user, in response to a
6         movie recommendation request submitted by the user.\n
7         For each set of questions in this list <list>{{input.questions}}</
8         list>,\n
9         You have the ability to ask the user about their preferences for each
10        of the following movie attributes:
11        [genre, release date, who will be watching, runtime].\n
12        Note that 'who will be watching' is related to the user's preferences
13        for the movie's rating.
14        For each set of questions, map each question to one of these
15        attributes IF asking about this specific attribute would allow
16        you to answer the question.\n
17        If none of the attributes would give you the information you need to
18        answer a given question, map that question to "None".\n
19        Format your response as a list of strings, as shown in the example
20        below:\n
21        ['genre', 'release date', 'None'] \n
22        DO NOT answer any questions contained in the response, or include any
23        expository text.\n
24        The result should be DIRECTLY parsable as a list of strings in Python
25        ."
26 }

```

## Bibliography

- [1] Christos H Papadimitriou and John N Tsitsiklis. The Complexity of Optimal Queueing Network Control. In *Proceedings of IEEE 9th Annual Conference on Structure in Complexity Theory*, pages 318–322. IEEE, 1994.
- [2] Peter Whittle. Restless Bandits: Activity Allocation in a Changing World. *Journal of Applied Probability*, 25(A):287–298, 1988.
- [3] Richard R Weber and Gideon Weiss. On an Index Policy for Restless Bandits. *Journal of Applied Probability*, pages 637–648, 1990.
- [4] Christine Herlihy, Aviva Prins, Aravind Srinivasan, and John Dickerson. Planning to Fairly Allocate: Probabilistic Fairness in the Restless Bandit Setting. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD*, August 2023. URL <https://arxiv.org/abs/2106.07677>.
- [5] Christine Herlihy and John Dickerson. Networked Restless Bandits with Positive Externalities. In *AAAI Conference on Artificial Intelligence*, 2023. URL <https://arxiv.org/abs/2212.05144>.
- [6] Solon Barocas, Anhong Guo, Ece Kamar, Jacquelyn Kronen, Meredith Ringel Morris, Jennifer Wortman Vaughan, W. Duncan Wadsworth, and Hanna Wallach. Designing Disaggregated Evaluations of AI Systems: Choices, Considerations, and Tradeoffs. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society, AIES '21*, page 368–378, New York, NY, USA, May 2021. Association for Computing Machinery. ISBN 9781450384735. doi: 10.1145/3461702.3462610. URL <https://doi.org/10.1145/3461702.3462610>.
- [7] Alexey Romanov and Chaitanya Shivade. Lessons from Natural Language Inference in the Clinical Domain. *CoRR*, abs/1808.06752, 2018. URL <http://arxiv.org/abs/1808.06752>.
- [8] Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2017. URL <https://www.aclweb.org/anthology/N18-2017>.

- [9] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1075. URL <https://www.aclweb.org/anthology/D15-1075>.
- [10] Adina Williams, Nikita Nangia, and Samuel R. Bowman. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. *CoRR*, abs/1704.05426, 2017. URL <http://arxiv.org/abs/1704.05426>.
- [11] Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/S18-2023. URL <https://www.aclweb.org/anthology/S18-2023>.
- [12] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8732–8740, Apr. 2020. doi: 10.1609/aaai.v34i05.6399. URL <https://ojs.aaai.org/index.php/AAAI/article/view/6399>.
- [13] Christine Herlihy, Kimberly Truong, Alexandra Chouldechova, and Miroslav Dudík. A structured regression approach for evaluating model performance across intersectional subgroups. In *ACM Conference on Fairness, Accountability, and Transparency*, ACM FAccT, 2024. URL <https://arxiv.org/abs/2401.14893>.
- [14] Christine Herlihy and Rachel Rudinger. MedNLI Is Not Immune: Natural Language Inference Artifacts in the Clinical Domain. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 1020–1027, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-short.129. URL <https://aclanthology.org/2021.acl-short.129>.
- [15] Steven T Piantadosi, Harry Tily, and Edward Gibson. The communicative function of ambiguity in language. *Cognition*, 122(3):280–291, 2012.
- [16] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, et al. Training language models to follow instructions with human feedback. In *Neural Information Processing Systems*, pages 27730–27744, 2022.
- [17] Sudha Rao and Hal Daumé III. Learning to ask good questions: Ranking clarification questions using neural expected value of perfect information. In *Association for Computational Linguistics*, pages 2737–2746, 2018.
- [18] Bodhisattwa Prasad Majumder, Sudha Rao, Michel Galley, and Julian McAuley. Ask what’s missing and what’s useful: Improving Clarification Question Generation using



- Global Knowledge. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4300–4312, 2021.
- [19] Christine Herlihy, Jennifer Neville, Tobias Schnabel, and Adith Swaminathan. On Overcoming Miscalibrated Conversational Priors in LLM-based Chatbots. In *40th Conference on Uncertainty in Artificial Intelligence, UAI*, 2024.
- [20] Aditya Mate, Jackson Killian, Haifeng Xu, Andrew Perrault, and Milind Tambe. Collapsing Bandits and Their Application to Public Health Interventions. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 15639–15650. Curran Associates, Inc., 2020. URL <https://proceedings.neurips.cc/paper/2020/file/b460cf6b09878b00a3e1ad4c72344ccd-Paper.pdf>.
- [21] Young Hun Jung and Ambuj Tewari. Regret bounds for thompson sampling in episodic restless bandit problems. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL [https://proceedings.neurips.cc/paper\\_files/paper/2019/file/2edfeadfe636973b42d7b6ac315b896c-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2019/file/2edfeadfe636973b42d7b6ac315b896c-Paper.pdf).
- [22] Lauren N. Steimle and Brian T. Denton. *Markov Decision Processes for Screening and Treatment of Chronic Diseases*, pages 189–222. Springer International Publishing, Cham, 2017. ISBN 978-3-319-47766-4. doi: 10.1007/978-3-319-47766-4\_6. URL [https://doi.org/10.1007/978-3-319-47766-4\\_6](https://doi.org/10.1007/978-3-319-47766-4_6).
- [23] Turgay Ayer, Can Zhang, Anthony Bonifonte, Anne C Spaulding, and Jagpreet Chhatwal. Prioritizing Hepatitis C Treatment in US Prisons. *Operations Research*, 67(3):853–873, 2019.
- [24] Aviva Prins, Aditya Mate, Jackson A Killian, Rediet Abebe, and Milind Tambe. *Incorporating Healthcare Motivated Constraints in Restless Bandit Based Resource Allocation*. Workshop on Challenges of Real-World Reinforcement Learning, NeurIPS, 2020. URL [https://teamcore.seas.harvard.edu/files/teamcore/files/human\\_in\\_the\\_loop\\_rmab\\_short.pdf](https://teamcore.seas.harvard.edu/files/teamcore/files/human_in_the_loop_rmab_short.pdf).
- [25] Yundi Qian, Chao Zhang, Bhaskar Krishnamachari, and Milind Tambe. Restless Poachers: Handling Exploration-Exploitation Tradeoffs in Security Domains. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, pages 123–131, 2016.
- [26] Xin Liu, Edwin KP Chong, and Ness B Shroff. A framework for opportunistic scheduling in wireless networks. *Computer networks*, 41(4):451–474, 2003.
- [27] I-H Hou, Vivek Borkar, and PR Kumar. *A Theory of QoS for Wireless*. IEEE, 2009.
- [28] Fengjiao Li, Jia Liu, and Bo Ji. Combinatorial Sleeping Bandits with Fairness Constraints. *CoRR*, abs/1901.04891, 2019. URL <http://arxiv.org/abs/1901.04891>.

- [29] A. Srinivasan. Distributions on Level-Sets with Applications to Approximation Algorithms. In *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science, FOCS '01*, page 588, USA, 2001. IEEE Computer Society. ISBN 0769513905.
- [30] Yuncheol Kang, Vittal Prabhu, Amy Sawyer, and Paul Griffin. Markov Models for Treatment Adherence in Obstructive Sleep Apnea. 05 2013.
- [31] Yuncheol Kang, Amy M Sawyer, Paul M Griffin, and Vittal V Prabhu. Modelling Adherence Behaviour for the Treatment of Obstructive Sleep Apnoea. *European Journal of Operational Research*, 249(3):1005–1013, 2016.
- [32] Keqin Liu and Qing Zhao. Indexability of Restless Bandit Problems and Optimality of Whittle Index for Dynamic Multichannel Access. *IEEE Transactions on Information Theory*, 56(11):5547–5567, Nov 2010. ISSN 1557-9654. doi: 10.1109/tit.2010.2068950. URL <http://dx.doi.org/10.1109/TIT.2010.2068950>.
- [33] José Niño-Mora. A Verification Theorem for Threshold-Indexability of Real-State Discounted Restless Bandits. *Mathematics of Operations Research*, 45(2):465–496, 2020.
- [34] Aditya Mate, Andrew Perrault, and Milind Tambe. Risk-Aware Interventions in Public Health: Planning with Restless Multi-Armed Bandits. In *20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, London, UK, 2021.
- [35] Jeremy Bentham. An Introduction to the Principles of Morals and Legislation. Technical report, McMaster University Archive for the History of Economic Thought, 1781.
- [36] Elliot Marseille and James G. Kahn. Utilitarianism and the Ethical Foundations of Cost-Effectiveness Analysis in Resource Allocation for Global Health. *Philosophy, Ethics, and Humanities in Medicine*, 14(1):1–7, 2019. doi: 10.1186/s13010-019-0074-7.
- [37] John Rawls. *A Theory of Justice*. Belknap Press of Harvard University Press, Cambridge, Massachusetts, 1 edition, 1971. ISBN 0-674-88014-5.
- [38] Leslie Scheunemann and Douglas White. The Ethics and Reality of Rationing in Medicine. *Chest*, 140:1625–32, 12 2011. doi: 10.1378/chest.11-0622.
- [39] Catherine R. Butler, Susan P. Y. Wong, Aaron G. Wightman, and Ann M. O’Hare. US Clinicians’ Experiences and Perspectives on Resource Limitation and Patient Care During the COVID-19 Pandemic. *JAMA Network Open*, 3(11):e2027315–e2027315, 11 2020. ISSN 2574-3805. doi: 10.1001/jamanetworkopen.2020.27315. URL <https://doi.org/10.1001/jamanetworkopen.2020.27315>.
- [40] Alvin Rajkomar, Michaela Hardt, Michael D Howell, Greg Corrado, and Marshall H Chin. Ensuring Fairness in Machine Learning to Advance Health Equity. *Annals of internal medicine*, 169(12):866–872, 2018.
- [41] Christopher J Kelly, Alan Karthikesalingam, Mustafa Suleyman, Greg Corrado, and Dominic King. Key Challenges for Delivering Clinical Impact with Artificial Intelligence. *BMC medicine*, 17(1):195, 2019.

- [42] Maria De-Arteaga, Riccardo Fogliato, and Alexandra Chouldechova. A Case for Humans-in-the-Loop: Decisions in the Presence of Erroneous Algorithmic Scores. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2020.
- [43] Berkeley J Dietvorst, Joseph P Simmons, and Cade Massey. Algorithm aversion: People erroneously avoid algorithms after seeing them err. *Journal of Experimental Psychology: General*, 144(1):114, 2015.
- [44] Matthew Joseph, Michael Kearns, Jamie H Morgenstern, and Aaron Roth. Fairness in Learning: Classic and Contextual Bandits. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL <https://proceedings.neurips.cc/paper/2016/file/eb163727917cbbaleea208541a643e74-Paper.pdf>.
- [45] Yifang Chen, Alex Cuellar, Haipeng Luo, Jignesh Modi, Heramb Nemlekar, and Stefanos Nikolaidis. Fair Contextual Multi-Armed Bandits: Theory and Experiments. volume 124 of *Proceedings of Machine Learning Research*, pages 181–190, Virtual, 03–06 Aug 2020. PMLR. URL <http://proceedings.mlr.press/v124/chen20a.html>.
- [46] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard S. Zemel. Fairness through awareness. *CoRR*, abs/1104.3913, 2011. URL <http://arxiv.org/abs/1104.3913>.
- [47] Dexun Li and Pradeep Varakantham. Towards soft fairness in restless multi-armed bandits. *arXiv preprint arXiv:2207.13343*, 2022.
- [48] Arpita Biswas, Jackson A Killian, Paula Rodriguez Diaz, Susobhan Ghosh, and Milind Tambe. Fairness for workers who pull the arms: An index based policy for allocation of restless bandit tasks. In *22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, London, UK, 2023.
- [49] Yurii Nesterov et al. *Lectures on convex optimization*, volume 137. Springer, 2018.
- [50] Dimitris Bertsimas, Vivek F. Farias, and Nikolaos Trichakis. The Price of Fairness. *Oper. Res.*, 59(1):17–31, jan 2011. ISSN 0030-364X. doi: 10.1287/opre.1100.0865. URL <https://doi.org/10.1287/opre.1100.0865>.
- [51] Yossi Rubner, Carlo Tomasi, and Leonidas Guibas. The earth mover’s distance as a metric for image retrieval. *International Journal of Computer Vision*, 40:99–121, 11 2000. doi: 10.1023/A:1026543900054.
- [52] Naresh M Punjabi. The epidemiology of adult obstructive sleep apnea. *Proceedings of the American Thoracic Society*, 5(2):136–143, 2008.
- [53] Amy M Sawyer, Nalaka S Gooneratne, Carole L Marcus, Dafna Ofer, Kathy C Richards, and Terri E Weaver. A systematic review of CPAP adherence across age groups: clinical and empiric insights for developing CPAP adherence interventions. *Sleep medicine reviews*, 15(6):343–356, 2011.

- [54] Brian W Rotenberg, Dorian Murariu, and Kenny P Pang. Trends in CPAP adherence over twenty years of data collection: a flattened curve. *Journal of Otolaryngology-Head & Neck Surgery*, 45(1):1–9, 2016.
- [55] Kathleen Askland, Lauren Wright, Dariusz R Wozniak, Talia Emmanuel, Jessica Caston, and Ian Smith. Educational, Supportive and Behavioural Interventions to Improve Usage of Continuous Positive Airway Pressure Machines in Adults with Obstructive Sleep Apnoea. *Cochrane Database of Systematic Reviews*, (4), 2020.
- [56] Christine Herlihy, Pranav Goel, and John Dickerson. *Networked Restless Bandits with Positive Externalities*. Disinformation Countermeasures and Machine Learning Workshop, ICML, 2022.
- [57] Kevin D. Glazebrook, David J. Hodge, and Chris Kirkbride. General notions of indexability for queueing control and asset management. *The Annals of Applied Probability*, 21(3): 876 – 907, 2011. doi: 10.1214/10-AAP705. URL <https://doi.org/10.1214/10-AAP705>.
- [58] Jackson A. Killian, Andrew Perrault, and Milind Tambe. Beyond "To Act or Not to Act": Fast Lagrangian Approaches to General Multi-Action Restless Bandits. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '21*, page 710–718, Richland, SC, 2021. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450383073.
- [59] Michal Valko. *Bandits on graphs and structures*. Habilitation à diriger des recherches, École normale supérieure de Cachan - ENS Cachan, June 2016. URL <https://hal.inria.fr/tel-01359757>.
- [60] Shiyin Lu, Yao Hu, and Lijun Zhang. Stochastic bandits with graph feedback in non-stationary environments. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(10):8758–8766, May 2021. URL <https://ojs.aaai.org/index.php/AAAI/article/view/17061>.
- [61] Han-Ching Ou, Christoph Siebenbrunner, Jackson A. Killian, Meredith B. Brooks, David Kempe, Yevgeniy Vorobeychik, and Milind Tambe. Networked restless multi-armed bandits for mobile interventions. *CoRR*, abs/2201.12408, 2022. URL <https://arxiv.org/abs/2201.12408>.
- [62] Edwin B. Fisher, Renée I. Boothroyd, Emily A. Elstad, Laura Hays, Amy Henes, Gary R. Maslow, and Clayton Velicer. Peer support of complex health behaviors in prevention and disease management with special reference to diabetes: systematic reviews. *Clinical Diabetes and Endocrinology*, 3(1):4, May 2017. ISSN 2055-8260. doi: 10.1186/s40842-017-0042-3. URL <https://doi.org/10.1186/s40842-017-0042-3>.
- [63] Alberto Pasanisi, Shuai Fu, and Nicolas Bousquet. Estimating discrete markov models from various incomplete data schemes. *Computational Statistics & Data Analysis*, 56(9):2609–2625, 2012. ISSN 0167-9473. doi: <https://doi.org/10.1016/j.csda.2012>.

- 02.027. URL <https://www.sciencedirect.com/science/article/pii/S0167947312001090>.
- [64] William R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933. ISSN 00063444. URL <http://www.jstor.org/stable/2332286>.
- [65] Ronald Ortner, Daniil Ryabko, Peter Auer, and Rémi Munos. Regret bounds for restless markov bandits. In *International conference on algorithmic learning theory*, pages 214–228. Springer, 2012.
- [66] Young Hun Jung, Marc Abeille, and Ambuj Tewari. Thompson sampling in non-episodic restless bandits. *CoRR*, abs/1910.05654, 2019. URL <http://arxiv.org/abs/1910.05654>.
- [67] Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, and Maosong Sun. Graph neural networks: A review of methods and applications. *CoRR*, abs/1812.08434, 2018. URL <http://arxiv.org/abs/1812.08434>.
- [68] Paul Almasan, José Suárez-Varela, Arnau Badia-Sampera, Krzysztof Rusek, Pere Barlet-Ros, and Albert Cabellos-Aparicio. Deep reinforcement learning meets graph neural networks: An optical network routing use case. *CoRR*, abs/1910.07421, 2019. URL <http://arxiv.org/abs/1910.07421>.
- [69] Paul W. Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137, 1983. ISSN 0378-8733. doi: [https://doi.org/10.1016/0378-8733\(83\)90021-7](https://doi.org/10.1016/0378-8733(83)90021-7). URL <https://www.sciencedirect.com/science/article/pii/0378873383900217>.
- [70] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [71] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA, 2018. ISBN 0262039249.
- [72] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In Sorelle A. Friedler and Christo Wilson, editors, *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, volume 81 of *Proceedings of Machine Learning Research*, pages 77–91. PMLR, 23–24 Feb 2018.
- [73] Ziad Obermeyer, Brian Powers, Christine Vogeli, and Sendhil Mullainathan. Dissecting racial bias in an algorithm used to manage the health of populations. *Science*, 366(6464): 447–453, 2019. doi: 10.1126/science.aax2342. URL <https://www.science.org/doi/abs/10.1126/science.aax2342>.

- [74] Allison Koenecke, Andrew Nam, Emily Lake, Joe Nudell, Minnie Quartey, Zion Mengesha, Connor Touns, John R. Rickford, Dan Jurafsky, and Sharad Goel. Racial disparities in automated speech recognition. *Proceedings of the National Academy of Sciences*, 117(14): 7684–7689, 2020. doi: 10.1073/pnas.1915768117. URL <https://www.pnas.org/doi/abs/10.1073/pnas.1915768117>.
- [75] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias: There’s software used across the country to predict future criminals. and it’s biased against blacks. 2016. <https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing>.
- [76] Latanya Sweeney. Discrimination in online ad delivery: Google ads, black names and white names, racial discrimination, and click advertising. *Queue*, 11(3):10–29, mar 2013. ISSN 1542-7730. doi: 10.1145/2460276.2460278. URL <https://doi.org/10.1145/2460276.2460278>.
- [77] Caner Hazirbas, Joanna Bitton, Brian Dolhansky, Jacqueline Pan, Albert Gordo, and Cristian Canton Ferrer. Towards measuring fairness in AI: The casual conversations dataset. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 4(3):324–332, 2022. doi: 10.1109/TBIOM.2021.3132237.
- [78] Kimberlé Crenshaw. Demarginalizing the intersection of race and sex: A black feminist critique of antidiscrimination doctrine, feminist theory and antiracist politics. *The University of Chicago Legal Forum*, 140:139–167, 1989.
- [79] Andrew C. Miller, Leon A. Gatys, Joseph Futoma, and Emily Fox. Model-based metrics: Sample-efficient estimates of predictive model subpopulation performance. In *Proceedings of the 6th Machine Learning for Healthcare Conference*, pages 308–336. PMLR, 2021. URL <https://proceedings.mlr.press/v149/miller21a.html>.
- [80] Renee Shelby, Shalaleh Rismani, Kathryn Henne, AJung Moon, Negar Rostamzadeh, Paul Nicholas, N’Mah Yilla-Akbari, Jess Gallegos, Andrew Smart, Emilio Garcia, et al. Sociotechnical harms of algorithmic systems: Scoping a taxonomy for harm reduction. In *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, pages 723–741, 2023.
- [81] Solon Barocas, Kate Crawford, Aaron Shapiro, and Hanna Wallach. The problem with bias: Allocative versus representational harms in machine learning. In *9th Annual conference of the special interest group for computing, information and society*, 2017.
- [82] Hilde Weerts, Miroslav Dudík, Richard Edgar, Adrin Jalali, Roman Lutz, and Michael Madaio. Fairlearn: Assessing and improving fairness of AI systems. *Journal of Machine Learning Research*, 24(257):1–8, 2023. URL <http://jmlr.org/papers/v24/23-0389.html>.
- [83] Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *International conference on machine learning*, pages 2564–2572. PMLR, 2018.

- [84] James R Foulds, Rashidul Islam, Kamrun Naher Keya, and Shimei Pan. An intersectional definition of fairness. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, pages 1918–1921. IEEE, 2020.
- [85] James R Foulds, Rashidul Islam, Kamrun Naher Keya, and Shimei Pan. Bayesian modeling of intersectional fairness: The variance of bias. In *Proceedings of the 2020 SIAM International Conference on Data Mining*, pages 424–432. SIAM, 2020.
- [86] Ursula Hébert-Johnson, Michael Kim, Omer Reingold, and Guy Rothblum. Multicalibration: Calibration for the (computationally-identifiable) masses. In *International Conference on Machine Learning*, pages 1939–1948. PMLR, 2018.
- [87] Angelina Wang, Vikram V Ramaswamy, and Olga Russakovsky. Towards intersectionality in machine learning: Including more identities, handling underrepresentation, and performing evaluation. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 336–349, 2022.
- [88] Disi Ji, Padhraic Smyth, and Mark Steyvers. Can I Trust My Fairness Metric? Assessing Fairness with Unlabeled Data and Bayesian Inference, 2020. URL <https://arxiv.org/abs/2010.09851>.
- [89] Disi Ji, Robert L. Logan, Padhraic Smyth, and Mark Steyvers. Active bayesian assessment for black-box classifiers, 2020. URL <https://arxiv.org/abs/2002.06532>.
- [90] Alexandra Chouldechova, Siqi Deng, Yongxin Wang, Wei Xia, and Pietro Perona. Unsupervised and semi-supervised bias benchmarking in face recognition. In *European Conference on Computer Vision*, pages 289–306. Springer, 2022.
- [91] Mathieu Molina and Patrick Loiseau. Bounding and approximating intersectional fairness through marginal fairness. *Advances in Neural Information Processing Systems*, 35:16796–16807, 2022.
- [92] Boris van Breugel, Nabeel Seedat, Fergus Imrie, and Mihaela van der Schaar. Can you rely on your model evaluation? improving model evaluation with synthetic test data. *arXiv preprint arXiv:2310.16524*, 2023.
- [93] Vihari Piratla, Soumen Chakrabarti, and Sunita Sarawagi. Active assessment of prediction services as accuracy surface over attribute combinations. *CoRR*, abs/2108.06514, 2021. URL <https://arxiv.org/abs/2108.06514>.
- [94] W. James and C. Stein. Estimation with quadratic loss. In *Proc. Fourth Berkeley Symposium on Mathematical Statistics and Probability*, pages 361—379, 1961.
- [95] C. Stein. Inadmissibility of the usual estimator for the mean of a multivariate distribution. In *Proc. Third Berkeley Symposium on Mathematical Statistics and Probability*, pages 197–206, 1956.

- [96] Michael Madaio, Lisa Egede, Hariharan Subramonyam, Jennifer Wortman Vaughan, and Hanna Wallach. Assessing the fairness of ai systems: Ai practitioners’ processes, challenges, and needs for support. *Proc. ACM Hum.-Comput. Interact.*, 6(CSCW1), apr 2022. doi: 10.1145/3512899. URL <https://doi.org/10.1145/3512899>.
- [97] Riccardo Fogliato, Alexandra Chouldechova, and Max G’Sell. Fairness evaluation in presence of biased noisy labels. In *International conference on artificial intelligence and statistics*, pages 2325–2336. PMLR, 2020.
- [98] Beata Strack, Jonathan Deshazo, Chris Gennings, Juan Luis Olmo Ortiz, Sebastian Ventura, Krzysztof Cios, and John Clore. Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records. *BioMed research international*, 2014:781670, 04 2014. doi: 10.1155/2014/781670.
- [99] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.
- [100] Cun-Hui Zhang and Stephanie S. Zhang. Confidence intervals for low dimensional parameters in high dimensional linear models. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 76(1):217–242, 2014.
- [101] Sara van de Geer, Peter Bühlmann, Ya’acov Ritov, and Ruben Dezeure. On asymptotically optimal confidence regions and tests for high-dimensional models. *The Annals of Statistics*, 42(3):1166 – 1202, 2014.
- [102] Adel Javanmard and Andrea Montanari. Confidence intervals and hypothesis testing for high-dimensional regression. *Journal of Machine Learning Research*, 15(82):2869–2909, 2014.
- [103] Hanzhong Liu, Xin Xu, and Jingyi Jessica Li. *Statistica Sinica*, 30(3):1333–1355, 2020.
- [104] Jana Janková, Rajen D Shah, Peter Bühlmann, and Richard J Samworth. Goodness-of-fit testing in high dimensional generalized linear models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 82(3):773–795, 2020.
- [105] M. E. Bock. Minimax estimators of the mean of a multivariate normal distribution. *The Annals of Statistics*, 3(1), 1975.
- [106] Sergey Feldman, Maya Gupta, and Bela Frigyik. Multi-task averaging. In *Advances in Neural Information Processing Systems*, volume 25, 2012.
- [107] Triveni Gandhi, Manojit Nandi, Miroslav Dudík, Hanna Wallach, Michael Madaio, Hilde Weerts, Adrin Jalali, and Lisa Ibañez. Fairness in AI Systems: From Social Context to Practice using Fairlearn. Tutorial presented at the 20th annual Scientific Computing with Python Conference (Scipy 2021), Virtual Event, 2021. URL [https://github.com/fairlearn/talks/tree/main/2021\\_scipy\\_tutorial](https://github.com/fairlearn/talks/tree/main/2021_scipy_tutorial).



- [108] Masatoshi Tsuchiya. Performance impact caused by hidden bias of training data for recognizing textual entailment. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA). URL <https://www.aclweb.org/anthology/L18-1239>.
- [109] Tom McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1334. URL <https://www.aclweb.org/anthology/P19-1334>.
- [110] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.
- [111] Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. MIMIC-III, a freely accessible critical care database. *Scientific data*, 3:160035, 2016.
- [112] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. Ch. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000 (June 13). Circulation Electronic Pages: <http://circ.ahajournals.org/content/101/23/e215.full> PMID:1085218; doi: 10.1161/01.CIR.101.23.e215.
- [113] Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 319–327, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-5034. URL <https://www.aclweb.org/anthology/W19-5034>.
- [114] Ronan Le Bras, Swabha Swayamdipta, Chandra Bhagavatula, Rowan Zellers, Matthew E. Peters, Ashish Sabharwal, and Yejin Choi. Adversarial filters of dataset biases, 2020.
- [115] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1472. URL <https://www.aclweb.org/anthology/P19-1472>.
- [116] Chien-Ju Ho, Aleksandrs Slivkins, Siddharth Suri, and Jennifer Wortman Vaughan. Incentivizing high quality crowdwork. In *Proceedings of the 24th International Conference on World Wide Web, WWW '15*, page 419–429, Republic and Canton of Geneva, CHE, 2015. International World Wide Web Conferences Steering Committee. ISBN 9781450334693. doi: 10.1145/2736277.2741102. URL <https://doi.org/10.1145/2736277.2741102>.

- [117] Yang Liu and Yiling Chen. Machine-learning aided peer prediction. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, EC '17, page 63–80, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450345279. doi: 10.1145/3033274.3085126. URL <https://doi.org/10.1145/3033274.3085126>.
- [118] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- [119] Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, et al. OpenAssistant conversations—democratizing large language model alignment. *arXiv preprint arXiv:2304.07327*, 2023.
- [120] Lev Tankelevitch, Viktor Kewenig, Auste Simkute, Ava Elizabeth Scott, Advait Sarkar, Abigail Sellen, and Sean Rintel. The metacognitive demands and opportunities of generative AI. *arXiv preprint arXiv:2312.10893*, 2023.
- [121] Sudha Rao and Hal Daumé III. Learning to ask good questions: Ranking clarification questions using neural expected value of perfect information. In *Association for Computational Linguistics*, pages 2737–2746, 2018.
- [122] Ying-Chun Lin, Jennifer Neville, Jack W. Stokes, Longqi Yang, Tara Safavi, Mengting Wan, Scott Counts, Siddharth Suri, Reid Andersen, Xiaofeng Xu, Deepak Gupta, Sujay Kumar Jauhar, Xia Song, Georg Buscher, Saurabh Tiwary, Brent Hecht, and Jaime Teevan. Interpretable User Satisfaction Estimation for Conversational Systems with Large Language Models, 2024.
- [123] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, et al. Training language models to follow instructions with human feedback. In *Neural Information Processing Systems*, pages 27730–27744, 2022.
- [124] Prasann Singhal, Tanya Goyal, Jiacheng Xu, and Greg Durrett. A long way to go: Investigating length correlations in RLHF. *arXiv preprint arXiv:2310.03716*, 2023.
- [125] Filip Radlinski, Krisztian Balog, Bill Byrne, and Karthik Krishnamoorthi. Coached conversational preference elicitation: A case study in understanding movie preferences. In *SIGdial Meeting on Discourse and Dialogue*, pages 353–360, 2019.
- [126] Ching-An Cheng, Andrey Kolobov, Dipendra Misra, Allen Nie, and Adith Swaminathan. LLF-Bench: Benchmark for interactive learning from language feedback. *arXiv preprint arXiv:2312.06853*, 2023.
- [127] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed H Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, 2022.

- [128] Lerrel Pinto, Marcin Andrychowicz, Peter Welinder, Wojciech Zaremba, and Pieter Abbeel. Asymmetric actor critic for image-based robot learning. In *RSS*, 2018.
- [129] HuggingFace. MPNet-base-v2, 2024. URL <https://huggingface.co/sentence-transformers/all-mpnet-base-v2>.
- [130] Sean R Sinclair, Felipe Vieira Frujeri, Ching-An Cheng, Luke Marshall, Hugo De Oliveira Barbalho, Jingling Li, Jennifer Neville, Ishai Menache, and Adith Swaminathan. Hindsight learning for mdps with exogenous inputs. In *ICML*, 2023.
- [131] Zhankui He, Zhouhang Xie, Rahul Jha, Harald Steck, Dawen Liang, Yesu Feng, Bodhisattwa Prasad Majumder, Nathan Kallus, and Julian McAuley. Large language models as zero-shot conversational recommenders. In *International conference on information and knowledge management*, pages 720–730, 2023.
- [132] Ali Borji. A categorical archive of chatgpt failures. *arXiv preprint arXiv:2302.03494*, 2023.
- [133] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.
- [134] Jessy Lin, Nicholas Tomlin, Jacob Andreas, and Jason Eisner. Decision-oriented dialogue for human-ai collaboration. *arXiv preprint arXiv:2305.20076*, 2023.
- [135] Jim McCambridge, John Witton, and Diana R Elbourne. Systematic review of the Hawthorne effect: new concepts are needed to study research participation effects. *Journal of clinical epidemiology*, 67(3):267–277, 2014.
- [136] Van Dang and Bruce W Croft. Query reformulation using anchor text. In *International Conference on Web Search and Data Mining*, pages 41–50, 2010.
- [137] Hiteshwar Kumar Azad and Akshay Deepak. Query expansion techniques for information retrieval: a survey. *Information Processing & Management*, 56(5):1698–1735, 2019.
- [138] Shizhe Diao, Pengcheng Wang, Yong Lin, and Tong Zhang. Active prompting with chain-of-thought for large language models. *arXiv preprint arXiv:2302.12246*, 2023.
- [139] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.
- [140] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*, 2023.
- [141] Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*, 2024.

- [142] Guanghui Qin and Jason Eisner. Learning how to ask: Querying lms with mixtures of soft prompts. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5203–5212, 2021.
- [143] Hamed Zamani, Bhaskar Mitra, Everest Chen, Gord Lueck, Fernando Diaz, Paul N Bennett, Nick Craswell, and Susan T Dumais. Analyzing and learning from user interactions for search clarification. In *SIGIR*, 2020.
- [144] Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. Towards conversational recommender systems. In *KDD*, 2016.
- [145] Michael L Littman. A tutorial on partially observable markov decision processes. *Journal of Mathematical Psychology*, 53(3):119–125, 2009.
- [146] Jinhang Zuo, Songwen Hu, Tong Yu, Shuai Li, Handong Zhao, and Carlee Joe-Wong. Hierarchical conversational preference elicitation with bandit feedback. In *Conference on Information & Knowledge Management*, pages 2827–2836, 2022.
- [147] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8:279–292, 1992.
- [148] Dorsa Sadigh, S Shankar Sastry, Sanjit A Seshia, and Anca Dragan. Information gathering actions over human internal state. In *Intelligent Robots and Systems*, pages 66–73, 2016.
- [149] Alain Ghouila-Houri. Caractérisation des matrices totalement unimodulaires. *Comptes Rendus Hebdomadaires des Séances de l’Académie des Sciences (Paris)*, 254:1192–1194, 1962.
- [150] Stephen A Rhoades. The Herfindahl-Hirschman Index. *Fed. Res. Bull.*, 79:188, 1993.
- [151] Albert O Hirschman. *National Power and the Structure of Foreign Trade*, volume 105. Univ of California Press, 1980.