

Fairness, Alignment and Audit: A Triptych for Responsible Modern AI

A THESIS
SUBMITTED FOR THE DEGREE OF
Master of Technology (Research)
IN THE
Faculty of Engineering

BY
Nirjhar Das



Computer Science and Automation
Indian Institute of Science
Bangalore – 560 012 (INDIA)

October, 2026

Declaration of Originality

I, **Nirjhar Das**, with SR No. **24761** hereby declare that the material presented in the thesis titled

Fairness, Alignment and Audit: A Triptych for Responsible Modern AI

represents original work carried out by me in the **Department of Computer Science and Automation** at **Indian Institute of Science** during the years **2024-26**.

With my signature, I certify that:

- I have not manipulated any of the data or results.
- I have not committed any plagiarism of intellectual property. I have clearly indicated and referenced the contributions of others.
- I have explicitly acknowledged all collaborative research and discussions.
- I have understood that any false claim will result in severe disciplinary action.
- I have understood that the work may be screened for any form of academic misconduct.

Date: 29th April, 2026

Student Signature

In my capacity as supervisor of the above-mentioned work, I certify that the above statements are true to the best of my knowledge, and I have carried out due diligence to ensure the originality of the report.

Advisor Name: Prof. Siddharth Barman

Advisor Signature

© Nirjhar Das
October, 2026
All rights reserved

DEDICATED TO

Maa, Baba & Dimitri

Acknowledgements

This section cannot but start with my deep gratitude to my advisor Prof. Siddharth Barman—he has not only been a stellar guide and generously supported me, but has also helped me develop my taste in research. I am grateful to him for the independence he bestowed upon me to pursue research of my liking. Apart from being a source of constant inspiration, he has also broadened my horizons both in research and in life. I am also deeply indebted to him for the invaluable career advice I have received from him, which has helped me never lose sight of what is important.

Much of this thesis and my MTech journey would not have been possible without the help of my senior collaborators Kirankumar Shiragur, Sayak Ray Chowdhury and Amit Deshpande, to whom I am immensely thankful. In Kiran, I have found not only an expert collaborator with deep technical insights, but also a friend, philosopher and guide (and a great table tennis coach). Brainstorming with him has taught me how to reason from first principles. I have learnt much from his comic punchlines that disguise deep truths and valuable advice. Working with Sayak has been a delightful experience in terms of learning and executing research projects. The trust he put in me at a time when I had very little experience, helped me become more confident. I hope someday I will be able to pass it on. I am grateful to Amit for lending his profound expertise in an area that I hardly knew when we started working together. His humility and persistence are some of his many qualities that I wish to emulate. I also owe many thanks to Gaurav Sinha for hosting my stints at Microsoft Research which have been one of the major turning points of my life both in terms of professional and personal growth. Being an amazingly kind and humble person, he has always given me freedom in research, and at times went above and beyond to ensure beneficial outcomes for me. I only hope that someday I am able to imbibe not only his inspiring habit of deep thinking, but also his kindness and benevolence. I would also like to express my gratitude towards the faculties at IISc, especially Prof. Siddharth Barman, Prof. Arindam Khan, Prof. Anand Louis and Prof. Manjunath Krishnapur, from whom I have learned not just course materials but ways of thinking.

I am extremely grateful to my friend-cum-collaborators Ayush Sawarni, Shivam Gupta,

Acknowledgements

Mohit Sharma, Arka Ray, Ritabrata Barat, Sukruta Midigeshi, Souradip Chakraborty and Praharsh Nanavati. Each one of them, with their idiosyncrasies, has made the projects so much more exciting. Technical discussions with Ayush have always enhanced my understanding, given his amazing clarity and simplicity of presentation. Perhaps the only other discussion with Ayush that I have enjoyed more is our heated debates on society and politics, where, unlike our technical discussions, we never agreed in the end—I am thankful to him. Shivam’s zeal and quick execution of ideas never let me slack off, and I will always cherish his unmatched giving nature. I am grateful to Mohit not only for his topical expertise but also his elite knowledge of good restaurants. My heartfelt thanks go to Arka for informal (with the exception of his short TA stint) but insightful introductions to many areas of theory CS, which have helped me immensely to grow as a researcher. I sincerely thank Ritabrata for the technical discussions that inevitably deepened my understanding of the problems, and for making brainstorming fun with his good (with low probability) jokes. Most of all, I am grateful to Arka and Ritabrata for our late night tea sessions where much tea was spilled—they made my time at IISc thoroughly enjoyable.

I deeply thank all my friends at CSA, IISc—Sreenivas (KVN), Mrinalkanti, Agrim, Gayathri, Debajyoti, Aditya, Satya, Aditi, Atasi, Suraj, Kirtan, Sreeramji, Ajay, Parag and Rameesh—who through game nights, bantering over snacks, restaurant hopping and impromptu discussions on theory CS, have made the journey utterly delightful. Special mention to Debajyoti and Satya (along with Ritabrata and Praharsh) for keeping the lab environment fun and engaging. I am also thankful for the (implicit) table tennis lessons I received (by being beaten badly in the game every time) from Debajyoti, Arka, Sreeramji and KVN. While I am grateful to Mrinal for sharing his deep technical insights on a wide gamut of topics, perhaps the real reason for my gratitude is the espresso machine in the lab he sponsored. I also owe many thanks to the CSA office staff, especially Kushael, Padmavathi and Meenakshi, for their indispensable support in administrative tasks.

Finally, I would like to express my utmost gratitude towards my parents, and my partner Dimitri, without whom I would not be who and where I am today. They have unwaveringly supported me in material and emotional ways, and have frequently made quiet sacrifices for my benefit. Without their profound love and care, this thesis would have been impossible, and I hope it makes them at least a fraction as proud as I feel to have them in my life.

Abstract

In this thesis, we investigate three different building blocks of modern artificial intelligence (AI), namely—nearest neighbor search, large language models, and classification in algorithmic decision-making—through the lens of responsible AI. As AI models get deployed in all aspects of society, fairness, alignment and audit become crucial ingredients for designing responsible AI and for regulating it. Through this thesis, we study principled formulations and algorithmic questions in these areas: fairness or diversity in nearest neighbor search, alignment of large language models, and auditing fairness of classifiers in algorithmic decision-making.

In the first part of the thesis, we study fairness or diversity in the context of nearest neighbor search. Nearest Neighbor Search (NNS) is a fundamental problem in data structures with wide-ranging applications, such as web search, recommendation systems, and, more recently, retrieval-augmented generations (RAG). In such recent applications, in addition to the relevance (similarity) of the returned neighbors, diversity among the neighbors is a central requirement. In this part of the thesis, we develop principled welfare-based formulations in NNS for realizing diversity across attributes. Our formulations are based on welfare functions—from mathematical economics—that satisfy central diversity (fairness) and relevance (economic efficiency) axioms. With a particular focus on Nash social welfare, we note that our welfare-based formulations provide objective functions that adaptively balance relevance and diversity in a query-dependent manner. Notably, such a balance was not present in the prior constraint-based approach, which forced a fixed level of diversity and optimized for relevance. In addition, our formulation provides a parametric way to control the trade-off between relevance and diversity, providing practitioners with flexibility to tailor search results to task-specific requirements. We develop efficient nearest neighbor algorithms with provable guarantees for the welfare-based objectives. Notably, our algorithm can be applied on top of any standard ANN method (i.e., use standard ANN method as a subroutine) to efficiently find neighbors that approximately maximize our welfare-based objectives. Experimental results demonstrate that our approach is practical and substantially improves diversity while maintaining high relevance of the retrieved neighbors.

In the second part, we investigate the problem of preference-based alignment of large-language models (LLMs). LLMs aligned using Reinforcement Learning from Human Feedback (RLHF) have shown remarkable generation abilities in numerous tasks. However, collecting high-quality human preferences creates costly bottlenecks in practical deployments, and hence, training data are often budgeted. In these scenarios, it is crucial to collect training data (e.g., contexts, a pair of generations for each context, and a preference indicating which generation is better) carefully, yet most of the existing methods sample contexts uniformly at random from a given collection. Given this, under the Bradley-Terry-Luce preference model and with a small budget of training data, we show that uniform sampling of contexts could lead to a policy (i.e., an aligned model) that suffers a constant sub-optimality gap from the optimal policy. This highlights the need for an adaptive context sampling strategy for effective alignment under a small sample budget. To address this, we reformulate RLHF within the contextual preference bandit framework, treating generations as actions, and give a nearly complete characterization of the sub-optimality gap in terms of both lower and upper bounds. First, when the action set is a d -dimensional hypercube and the number of samples is T , we show an $\Omega(d/\sqrt{T})$ lower bound. Next, we propose an algorithm, *Active Preference Optimization* (APO), that iteratively collects preferences for the most uncertain contexts. We show that the sub-optimality gap of the policy learned via APO matches the lower bound up to a log factor and a non-linearity constant. Finally, we perform experiments on practical datasets to validate APO’s efficacy over existing methods, establishing it as a sample-efficient and cost-effective solution for LLM alignment.

In the third and final part of the thesis, we study the problem of auditing the fairness of a given classifier under partial feedback, where true labels are available only for positively classified individuals, (e.g., loan repayment outcomes are observed only for approved applicants). We introduce a novel cost model for acquiring additional labeled data, designed to more accurately reflect real-world costs such as credit assessment, loan processing, and potential defaults. Our goal is to find optimal fairness audit algorithms that are more cost-effective than random exploration and natural baselines. In our work, we consider two audit settings: a black-box model with no assumptions on the data distribution, and a mixture model, where features and true labels follow a mixture of exponential family distributions. In the black-box setting, we propose a near-optimal auditing algorithm under mild assumptions and show that a natural baseline can be strictly suboptimal. In the mixture model setting, we design a novel algorithm that achieves significantly lower audit cost than the black-box case. Our approach leverages prior work on learning from truncated samples and maximum-a-posteriori oracles, and extends known results on spherical Gaussian mixtures to handle exponential family mixtures, which may be of independent interest. Moreover, our algorithms apply to popular fairness metrics

Abstract

including demographic parity, equal opportunity, and equalized odds. Empirically, we demonstrate strong performance of our algorithms on real-world fair classification datasets like Adult Income and Law School, consistently outperforming baselines by around 50% in audit cost.

Publications based on this Thesis

1. **Welfarist Formulations for Diverse Similarity Search**

Joint work with Siddharth Barman, Shivam Gupta, and Kirankumar Shiragur.

To appear in The Fourteenth International Conference on Learning Representations (ICLR 2026)

2. **Active Preference Optimization for Sample Efficient RLHF**

Joint work with Souradip Chakraborty, Aldo Pacchiano, and Sayak Ray Chowdhury.

In Machine Learning and Knowledge Discovery in Databases. Research Track (ECML PKDD 2025)

3. **Cost Efficient Fairness Audit under Partial Feedback**

Joint work with Mohit Sharma, Praharsh Nanavati, Kirankumar Shiragur, and Amit Deshpande

Accepted at The Fortieth Annual Conference on Neural Information Processing Systems. (NeurIPS 2026)

Contents

Acknowledgements	i
Abstract	iii
Publications based on this Thesis	vi
Contents	vii
1 Introduction	1
1.1 Diversity in NNS: A Welfarist Perspective	2
1.2 Active Learning for Sample Efficient RLHF	7
1.3 A Framework for Auditing Fairness under Partial Feedback	9
1.4 Overview of the Thesis	11
2 Diversity in Similarity Search: A Welfarist Perspective	13
2.1 Problem Formulation and Main Results	13
2.1.1 Our Results	13
2.2 NaNNS in the Single-Attribute Setting	17
2.2.1 Proofs of Theorem 2.1 and Corollary 2.1	18
2.3 NaNNS in the Multi-Attribute Setting	23
2.3.1 Proof of Theorem 2.2	23
2.3.2 Algorithm for the Multi-Attribute Setting	25
2.3.3 Proof of Theorem 2.3	26
2.4 Experimental Evaluations	27
2.4.1 Metrics for Measuring Relevance and Diversity	28
2.4.2 Experimental Setup and Datasets	29
2.4.3 Algorithms	33
2.4.4 Results: Balancing Relevance and Diversity	34

CONTENTS

2.4.5	p -FetchUnion-ANN: A Faster Heuristic for the Single Attribute Setting	37
2.5	Proofs of Examples 2.1 and 2.2	37
2.6	Extensions for p -NNS	39
2.7	Further Experimental Evaluation and Analysis	48
2.7.1	Balancing Relevance and Diversity: Single-attribute Setting	48
2.7.1.1	Performance of Nash-ANN	48
2.7.1.2	Performance of p -mean-ANN	49
2.7.1.3	Approximation Ratio Versus Inverse Simpson Index	51
2.7.1.4	Approximation Ratio Versus Distinct Attribute Count	52
2.7.1.5	Recall Versus Entropy	53
2.7.2	Balancing Relevance and Diversity: Multi-attribute Setting	55
2.7.3	More Experiments for p -FetchUnion-ANN	58
2.8	Conclusion	61
3	Active Learning for Sample Efficient RLHF	65
3.1	Problem Setup	65
3.2	Lower Bounds	66
3.2.1	Auxiliary Lemmas for Proof of Theorem 3.2	70
3.3	Our Approach: Active Preference Optimization	71
3.3.1	Suboptimality Gap of APO	72
3.4	Generalization Beyond BTL Model	75
3.4.1	Algorithm	76
3.4.2	Suboptimality Gap Guarantee of APO-Gen	77
3.5	Experiments	78
3.5.1	Practical Version of APO for RLHF	79
3.5.2	Results on Controlled Sentiment Generation Task	79
3.5.3	Results on Single-turn Dialogue task	81
3.5.4	Hyperparameter Details	82
3.6	Proof of Theorem 3.3	82
3.6.1	Auxiliary Lemmas for Proof of Theorem 3.3	84
3.7	Proof of Theorem 3.4	86
3.7.1	Auxiliary Lemmas for Proof of Theorem 3.4	87
3.8	Some Useful Results	88
3.9	Conclusion	91

4 Auditing Fairness under Partial Feedback	92
4.1 Problem Formulation	92
4.2 Our Results	94
4.3 Fairness Audit under Blackbox Model	97
4.3.1 RS-Audit: Rejection Sampling based Audit	98
4.3.2 Proof of Theorem 4.2	100
4.3.3 Lower Bound for the Blackbox Model: Proof of Theorem 4.3	102
4.4 Fairness Audit under Mixture Model	107
4.4.1 Proof of Theorem 4.4	110
4.5 Experiments & Results	113
4.6 Proof of Theorem 4.1 and Other Auxiliary Lemmas	114
4.6.1 Proof of Theorem 4.1	115
4.6.2 Auxiliary Lemmas	117
4.7 Missing Proofs from Section 4.4	119
4.7.1 Approximate Maximum-a-posteriori Oracle for Exponential Family	119
4.7.2 Parameter Estimation from Truncated Samples	122
4.7.3 Some Technical Lemmas	123
4.8 Discussion on Exponential Family	125
4.9 Concentration Lemmas	125
4.9.1 Concentration of the Negative Binomial Distribution	125
4.9.2 Supporting Lemmas	127
4.10 Further Experiments	128
4.10.1 Experiments for the Blackbox Model	129
4.10.1.1 Experimental Setup	130
4.10.1.2 Results	131
4.10.2 Experiments for the Mixture Model	133
4.10.2.1 Instance Construction	133
4.10.2.2 Results	135
4.11 Conclusion and Future Work	135
Bibliography	137

Chapter 1

Introduction

With artificial intelligence (AI) and algorithmic decision-making becoming pervasive in every aspect of society, ensuring that such systems behave responsibly is increasingly critical. In this thesis, we investigate three building blocks of modern AI—nearest neighbor search (NNS), classification, and large language models (LLMs)—with a special focus on fairness and alignment. Taking a step towards responsible AI algorithms, we develop principled frameworks and efficient algorithms that address diversity (fairness) in NNS, better alignment of LLMs under small data budget, and auditing fairness of a given classifier. While fairness has received steady attention both from computer science and economics communities, alignment and audit are relatively recent concerns arising out of large-scale deployment of AI models in the society. Therefore, in this thesis we develop a rigorous theoretical understanding of these aspects of responsible AI, in the hope of providing deeper insights in practice.

In questions related to fairness, the goal is to distribute or allocate resources for a set of agents in a way that is fair to all the agents involved. Various axiomatic definitions of fairness have been studied in classical settings like resource allocation and participatory budgeting, however, fairness in modern AI continues to be attained by adding ad-hoc constraints to the original optimization problem. In the first part of the thesis, we address this gap by studying fairness or diversity through an axiomatic lens in the context of nearest neighbor search, a fundamental problem in data structures with ubiquitous applications ranging from web search to retrieval-augmented generation in recent times.

In the second part, we investigate the problem of aligning AI models. The problem of aligning AI models has received much attention in recent times as a consequence of the widespread use of large language models (LLMs) in various facets of our life. In the alignment problem, the goal is to incorporate human values and safety guardrails in the AI models so that they do not adversely affect the human user. A popular framework for alignment of AI models is *reinforce-*

ment learning from human feedback (RLHF), where a human labeler marks which of the two outputs of the model is better for a given input. Using this human feedback, a reward model is learned which assigns a reward to an input-output pair. Finally, this learned reward model is used to fine-tune the AI model via reinforcement learning. While this method of alignment is quite successful in practice, less is understood theoretically (though there has been a steady stream of recent works). In our work, we address the sample-complexity of RLHF in the active learning framework—where the algorithm has a small budget for collection of human feedback data and hence, needs to carefully choose the comparisons on which feedback is to be collected.

In the final part of the thesis, we focus on the question of auditing classifiers for their claimed fairness. As fairness concerns become more pressing with classifiers being deployed to decide the outcome for individuals in scenarios like job hiring and criminal recidivism, it becomes important to equip regulators with tools for testing the fairness of such algorithms. Through our work, we develop an auditing framework for settings like bank loans and criminal recidivism, where the true label (e.g. loan default) is observable only for individuals who are assigned positive classification (e.g. loan disbursed) by the classifier.

Overall, this thesis aims to advance the theoretical foundations of responsible AI and its regulation by addressing key challenges in fairness, alignment, and audit, while grounding the analysis in practically relevant models and constraints.

In the following sections, we provide detailed exposition on each of the problems studied in this thesis.

1.1 Diversity in NNS: A Welfarist Perspective

Nearest Neighbor Search (NNS) is a fundamental problem in computer science with wide-ranging applications in diverse domains, including computer vision [93], data mining [19], information retrieval [63], classification [30], and recommendation systems [25]. The relevance of NNS has grown further in recent years with the advent of retrieval-augmented generation (RAG); see, e.g., [64], [97], and references therein. Formally, given a set of vectors $P \subset \mathbb{R}^d$, in ambient dimension d , and a query vector $q \in \mathbb{R}^d$, the objective in NNS is to find a subset S of k (input) vectors from P that are most similar to q under a similarity function $\sigma : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}_+$. That is, NNS corresponds to the optimization problem $\operatorname{argmax}_{S \subseteq P: |S|=k} \sum_{v \in S} \sigma(q, v)$. Note that, while most prior works in neighbor search express the problem in terms of minimizing distances, we work with the symmetric version of maximizing similarity.¹

In practice, the input vectors are high dimensional; in many of the above-mentioned applications the ambient dimension d is close to a thousand. Furthermore, most applications

¹This enables us to directly apply welfare functions.

involve a large number of input vectors. This scale makes exact NNS computationally expensive, since applications require, for real-time queries q , NNS solutions in time (sub)linear in the number of input vectors $|P|$. To address this challenge, the widely studied framework of Approximate Nearest Neighbor (ANN) search relaxes the requirement of exactness and instead seeks neighbors whose similarities are approximately close to the optimal ones.

ANN search has received substantial attention over the past three decades. Early techniques relied on space-partitioning methods, including Locality-Sensitive Hashing (LSH) [39, 4], k -d trees [6], and cover trees [15]. More recent industry-scale systems adopt clustering-based [41, 10] and graph-based [62, 31, 88, 87] approaches, along with other practically-efficient methods [89, 84].

While relevance—measured in terms of a similarity function $\sigma(\cdot, \cdot)$ —is a primary objective in NNS, prior work has shown that *diversity* in the retrieved set of vectors is equally important for user experience, fairness, and reducing redundancy [20]. For instance, in 2019 Google announced a policy update to limit the number of results from a single domain, thereby reducing redundancy [56]. Similarly, Microsoft recently introduced diversity constraints in ad recommendation systems to ensure that advertisements from a single seller do not dominate the results [3]. Such an adjustment was crucial for improving user experience and promoting fairness for advertisers. These examples highlight how diversity, in addition to enhancing fairness and reducing redundancy, directly contributes to improved search quality for end users.

A natural way to formalize diversity in these settings is to associate each input vector with one or more *attributes*. Diversity can then be measured with respect to these attributes, complementing the similarity-based relevance. Building on this idea, the current work develops a principled framework for diversity in neighbor search by drawing on the theory of collective welfare from mathematical economics [69]. This perspective enables the design of performance metrics (i.e., optimization criteria) that balance similarity-based relevance and attribute-based diversity in a theoretically grounded manner.

This formulation is based on the perspective that *algorithms can be viewed as economic policies*. Indeed, analogous to economic policies, numerous deployed algorithms induce utility (monetary or otherwise) among the participating agents. For instance, an ANN algorithm—deployed to select display advertisements for search queries—impacts the exposure and, hence, the sales of the participating advertisers. Notably, there are numerous other application domains wherein the outputs of the underlying algorithms impact the utilities of individuals; see [5] and [43] for multiple examples. Hence, in contexts where fairness (diversity) and welfare are important considerations, it is pertinent to evaluate algorithms analogous to how one evaluates economic policies that induce welfare.

In mathematical economics, welfare functions, $f : \mathbb{R}^c \mapsto \mathbb{R}$, provide a principled approach to aggregate the utilities of $c \in \mathbb{Z}_+$ agents into a single measure. Specifically, if an algorithm induces utilities $u_1, u_2, \dots, u_c$ among a population of c agents, then the collective welfare is $f(u_1, u_2, \dots, u_c)$. A utilitarian way of aggregation is by considering the arithmetic mean (average) of the utilities u_ℓ s. However, note that the arithmetic mean is not an ideal criterion if we are required to be fair among the c agents: the utilitarian welfare (arithmetic mean) can be high even if the utility of only one agent, say u_1 , is large and all the remaining utilities, $u_2, \dots, u_c$, are zero. The theory of collective welfare develops meaningful alternatives to the arithmetic mean by identifying welfare functions, f s, that satisfy fairness and efficiency axioms.

Among such alternatives, Nash social welfare (NSW) is an exemplar that upholds multiple fairness axioms, including symmetry, independence of unconcerned agents, scale invariance, and the Pigou-Dalton transfer principle [69]. Nash social welfare is obtained by setting the function f as the geometric mean, $\text{NSW}(u_1, \dots, u_c) := (\prod_{\ell=1}^c u_\ell)^{1/c}$. The fact that NSW strikes a balance between fairness and economic efficiency is supported by the observation that it sits between egalitarian and utilitarian welfare: the geometric mean is at least as large as the minimum value, $\min_{1 \leq \ell \leq c} u_\ell$, and it is also at most the arithmetic mean $\frac{1}{c} \sum_{\ell=1}^c u_\ell$ (the AM-GM inequality).

The overarching goal of this work is to realize diversity (fairness) across attributes in nearest neighbor search while maintaining relevance of the returned k vectors. Our modeling insight here is to equate attributes with agents and apply Nash social welfare.

In particular, consider a setting where we have $c \in \mathbb{Z}_+$ different attributes (across the input vectors), and let S be any subset of k vectors (neighbors) among the input set P . In our model, each included vector $v \in S$, with attribute $\ell \in [c]$, contributes to the utility u_ℓ (see Section 2.1.1), and the Nash social welfare (NSW) induced by S is the geometric mean of these utilities, $u_1, u_2, \dots, u_c$. Our objective is to find a size- k subset, $S^* \subseteq P$, of input vectors with as large NSW as possible.

The following two instantiations highlight the applicability of our model in NNS settings: In a display-advertising context with c sellers, each selected advertisement $v \in S$ of a seller $\ell \in [c]$ contributes to ℓ 's exposure (utility) u_ℓ . Similarly, in an apparel-search setup with c colors in total, each displayed product $v \in S$ with color $\ell \in [c]$ contributes to the utility u_ℓ .

Prior work [3] imposed constraints for achieving diversity in NNS. These constraints enforced that, for each $\ell \in [c]$ and among the k returned vectors, at most k' many can have attribute ℓ . Such hard constraints rely on a fixed ad hoc quota parameter k' and may fail to adapt to the intent expressed in the query. In contrast, our NSW-based approach balances relevance and diversity in a query-dependent manner. For example, in the apparel-search setup, if the

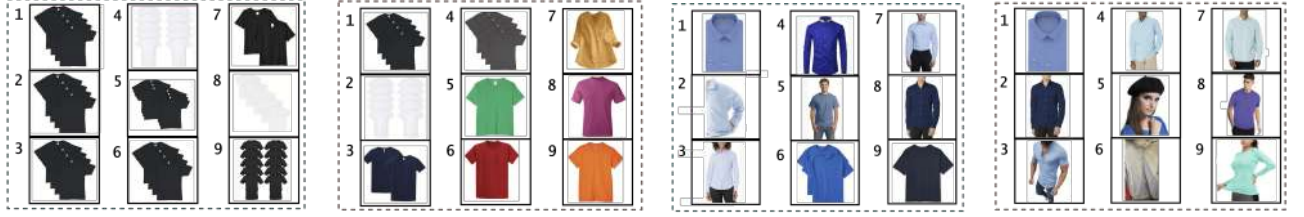


Figure 1.1: Neighbor search results ($k = 9$) on the Amazon dataset. From left: **First** and **Second** images - ANN and Nash-based results for query “shirts”, respectively. **Third** and **Fourth** images - ANN and Nash-based results for query “blue shirt”, respectively. Note that the Nash-based method selects diverse colors for the query “shirts” but conforms to the blue color for the query “blue shirt”.

search query is “blue shirt,” then a constraint on the color attribute ‘blue’ (i.e., when ℓ stands for ‘blue’) would limit the relevance by excluding valid vectors. NSW, however, for the “blue shirt” query, is free to select all the k vectors with attribute ‘blue’ upholding relevance; see Figure 1.1 for supporting empirical results. On the other hand, if the apparel-search query is just “shirts,” then NSW criterion is inclined to select vectors with different color attributes. These features of NSW are substantiated by the stylized instances given in Examples 2.1 and 2.2 (Section 2.1.1).

We reiterate that our formulation does not require a quota parameter k' to force diversity. For NSW, diversity (fairness) across attributes is obtained via normative properties of Nash social welfare. Hence, with axiomatic support, NSW stands as a meaningful criterion in neighbor search, as it is in the context of economic and allocation policies.

Our welfarist formulation extends further to control the trade-off between relevance and diversity. Specifically, we also consider p -mean welfare. Formally, for exponent parameter $p \in (-\infty, 1]$, the p th mean $M_p(\cdot)$, of c utilities $u_1, u_2, \dots, u_c \in \mathbb{R}_+$, is defined as $M_p(u_1, \dots, u_c) := (\frac{1}{c} \sum_{\ell=1}^c u_\ell^p)^{1/p}$. The p -mean welfare, $M_p(\cdot)$, captures a range of objectives with different values of p : it corresponds to the utilitarian welfare (arithmetic mean) when $p=1$, the NSW (geometric mean) with $p \rightarrow 0$, and the egalitarian welfare when $p \rightarrow -\infty$. Notably, setting $p=1$, we get back the standard nearest neighbor objective, i.e., maximizing $M_1(\cdot)$ corresponds to finding the k nearest neighbors and this objective is not concerned with diversity across attributes. At the other extreme, $p \rightarrow -\infty$ aims to find as attribute-diverse a set of k vectors as possible (while paying scarce attention to relevance).

We study, both theoretically and experimentally, two diversity settings: (i) single-attribute setting and (ii) multi-attribute setting. In the single-attribute setting, each input vector $v \in P$ is associated with exactly one attribute $\ell \in [c]$ – this captures, for instance, the display-advertisement setup, wherein each advertisement v belongs to exactly one seller ℓ . In the more

general multi-attribute setting, each input vector $v \in P$ can have more than one attribute; in apparel-search, for instance, the products can be associated with multiple attributes, such as color, brand, and price.

We note that the constraint-based formulation for diversity considered in [3] primarily addresses single-attribute setting. In fact, generalizing such constraints to the multi-attribute context leads to a formulation wherein it is NP-hard even to determine whether there exist k vectors that satisfy the constraints, i.e., it would be computationally hard to find any size- k constraint-feasible subset S , let alone an optimal one.¹

By contrast, our NSW formulation does not run into such a feasibility barrier. Here, for any candidate subset S of k vectors, each included vector $v \in S$ contributes to the utility u_ℓ of every attribute ℓ associated with v . As before, the NSW induced by S is the geometric mean of the induced utilities, $u_1, u_2, \dots, u_c$, and the objective is to find a subset of k vectors with as large NSW as possible.

In summary, we make the following contributions through this work.

- We view the NSW formulation for diversity, in both single-attribute and multi-attribute settings, as a key contribution of this thesis. Another relevant contribution of this work is the generalization to p -mean welfare, which provides a systematic way to trade off relevance and diversity.
- We also develop efficient algorithms, with provable guarantees, for the NSW and p -mean welfare formulations. For the single-attribute setting, we develop an efficient greedy algorithm for finding k vectors that optimize the Nash social welfare among the c attributes (Theorem 2.1). In addition, this algorithm can be provably combined with any sublinear ANN method (as a subroutine) to find near-optimal solutions for the Nash objective in sublinear time (Corollary 2.1).
- For the multi-attribute setting, we first show that finding the set of k vectors that maximize the Nash social welfare is NP-hard (Theorem 2.2). We complement this hardness result, by developing a polynomial-time approximation algorithm that achieves an approximation ratio of $(1 - 1/e) \approx 0.63$ for maximizing the logarithm of the Nash social welfare (Theorem 2.3).
- We complement our theoretical results with experiments on both real-world and semi-synthetic datasets. These experiments demonstrate that the NSW objective effectively captures the trade-off between diversity and relevance in a query-dependent manner. We

¹This hardness result follows via a reduction from the Maximum Independent Set problem.

further analyze the behavior of the p -mean welfare objective across different values of $p \in (-\infty, 1]$, observing that it interpolates smoothly between prioritizing for diversity, when p is small, and focusing on relevance, when p is large. Finally, we benchmark the solution quality and running times of various algorithms for solving the NSW and p -mean formulations proposed in this work.

1.2 Active Learning for Sample Efficient RLHF

Reinforcement Learning from Human Feedback (RLHF) has proven highly effective in aligning Large Language Models (LLMs) with human preferences [22, 72, 35]. This approach involves collecting extensive data, each comprising a context (e.g., a movie review), a pair of generations (e.g., completions of the review), and a preference indicating which generation is better than the other. First, a reward model is trained to classify preferred generations, and subsequently, a language model policy is trained using RL (e.g., Proximal Policy Optimization [82]) to output high-reward generations while minimizing divergence from a reference policy.

Given the practical success, recent theoretical advances have been made in training reward models as well as aligning policies from pairwise comparisons [79, 21, 101]. In these settings, the learner doesn’t have any control over selecting the contexts, and the aim is to minimize cumulative loss or regret due to not knowing the ground-truth reward or the optimal policy in advance.

However, in the case of aligning LLMs using RLHF, the learner has control over both the contexts and actions, i.e., the contexts and generations for which preference data needs to be collected, yet most of the existing RLHF algorithms pick contexts uniformly at random from a given pool [86, 72]. This is followed by first generating a pair of responses for each sampled prompt based on a supervised fine-tuned (SFT) policy and then sending all the pairs to a human labeler to collect preferences.

The success of RLHF hinges on the quality of human preferences. This could create costly bottlenecks in practical deployments since high-quality preferences are expensive to collect as this demands a certain level of expertise from labelers. Hence, there is often a budget on the number of contexts and associated generation pairs that could be sent to expert labelers for comparison. While uniform sampling of contexts as a simple approach has been proven effective for aligning LLMs so far, one is bound to ask whether this is a good enough strategy, especially given a fixed budget for labeling. Or do we need potentially more involved sampling strategies to deliver better model alignment under budget constraints? Such algorithms need to be sample efficient as high-quality samples are expensive to obtain, while they should not compromise on the performance of the aligned policy. This work takes a step in developing

theory and algorithms for RLHF under a small sample budget.

We first show that the naive way of collecting preferences by choosing contexts uniformly at random can lead to wastage of samples under the Bradley-Terry-Luce (BTL) preference model characterized by a finite dimensional parameter [17, 58].

Result 1 (Constant sub-optimality under uniform context sampling) *There exists an instance of the alignment problem for which an algorithm that (i) collects preferences by sampling contexts uniformly at random, (ii) learns a reward model by maximizing likelihood of the preferred generations, and (iii) trains a greedy policy w.r.t. the learnt reward model suffers an $\Omega(1)$ sub-optimality gap¹ with high probability when the sample budget is smaller than number of contexts.*

To the best of our knowledge, this is the first provable negative result for the alignment performance of RLHF algorithms that sample contexts uniformly under a small sample budget. This necessitates the design of RLHF algorithms that adaptively sample contexts, with an aim to improve alignment of the learned policy with human preferences. Our next result benchmarks the alignment performance of any RLHF algorithm under the BTL preference model when there is no restriction on how contexts can be sampled.

Result 2 (Lower bound on sub-optimality for any sampling strategy) *For any RLHF algorithm there exists an instance of the alignment problem for which the policy that the algorithm outputs after collecting T samples (contexts, generations and preferences) would suffer a sub-optimality gap $\Omega(d/\sqrt{T})$, where d is the dimension of the parameter characterizing the BTL model.*

This is the first theoretical lower bound on the alignment performance of RLHF algorithms, which has been crucially missing in the literature. This result effectively eliminates the possibility of achieving better than sub-linear convergence under a finite sample budget. Next, we propose an algorithm—*Active Preference Optimization (APO)*—that achieves this sub-optimality gap by iteratively sampling the most uncertain contexts and collecting preferences for their generation pairs.

Result 3 (Upper bound on sub-optimality for APO) *The sub-optimality gap of the policy learned via APO after collecting T samples scales as $\tilde{O}(d\sqrt{\kappa/T})$ with high probability, where d is the parameter dimension and κ is a problem-dependent nonlinearity constant.*

¹Measured by the maximum difference between latent rewards of the optimal policy and the trained policy over the context set.

Next, we generalize our result from parameterized BTL model to non-parametric preference models with function approximation. We propose an analogue of APO albeit with non-trivial modifications, namely **APO-Gen**, that achieves a similar sub-optimality gap (see Subsection 3.4 and Appendix 3.7). This is the first known upper bound on the alignment performance of an active context selection strategy under generic preference models, which recovers the result for the parameterized BTL model as a special case.

Result 4 (Upper bound on sub-optimality under general preferences) *The sub-optimality gap of the policy learned via **APO-Gen** after collecting T samples scales as $\tilde{O}(\sqrt{d_\varepsilon \log(\mathcal{N}T)/T})$ with high probability, where $\mathcal{N}$ and d_ε measure the complexity of the underlying preference model.*

Empirical evidence. For practical purposes, we propose a batch version of APO to make it computationally more efficient (see Section 3.5). We experiment with GPT-2 on IMDB sentiment dataset [60] and demonstrate significant improvement in LLM alignment over uniform context sampling and prior baselines [67, 70]. We show similar improvement in the performance of the aligned policy on Anthropic-HH dataset [9] with Gemma-2b. Our work contributes towards a sample-efficient and practical solution to preference data collection for RLHF.

1.3 A Framework for Auditing Fairness under Partial Feedback

Machine learning models can amplify and exacerbate biases present in their training data. To address this, fairness audits and interventions have emerged as key topics in the fair machine learning literature [66, 11, 99]. Consequently, different fairness metrics [28, 36, 33], fairness interventions [14, 16], and audit frameworks have been proposed and adopted [80, 38].

This work addresses the problem of auditing machine learning models for fairness, an issue of growing significance in decision-making systems [18, 51, 68]. Previous work on fairness audit covers a range of individual and group fairness metrics [98, 44] and construction of statistical tests for auditing [90, 83]. In the simplest fairness audit setting, a decision-maker employs a classifier f that takes as input a feature vector x and a sensitive attribute or group label a and makes a binary prediction $f(x, a)$. An auditor aims to estimate the fairness of f for a standard fairness metric such as Equal Opportunity (i.e., equal True Positive Rates for different groups) using the available data.

As a motivating real-world example, consider loan disbursement using a classifier f for approval, where the class label y indicating the loan repayment status is available only for applicants who are offered loans (i.e., when $f(x, a) = 1$). Thus, the historical data is biased

or incomplete, whereas any new data exploration can be costly (e.g., random disbursal may have higher defaults). Previous work under this incomplete data model can be broadly categorized into two settings. First, semi-supervised fair learning from incomplete labeled data and complete unlabeled data, without actively acquiring additional labels [50, 42, 46, 94]. Second, online fair classification with *partial feedback*, where labels are revealed only for positively classified data points [12, 45]. Complementary to these works, we study the fairness audit problem in the partial feedback setting under a novel cost model that incorporates both the access to historically biased or incomplete labeled data and the cost of acquiring additional labels on complete but unlabeled data.

In the partial feedback setting, due to unavailability of labels for negatively classified individuals ($f(x, a) = 0$), any fairness audit algorithm must acquire these labels to complete the data and accurately estimate outcome-based fairness metrics such as equal opportunity and equalized odds on the true data distribution. Notably, in scenarios such as bank loan applications, granting a loan to a previously negatively classified individual may incur a cost if the individual fails to repay. This cost is not modeled in prior fairness audit work primarily focusing on hypothesis testing for a given unfairness tolerance [44, 98].

To address these questions in the literature, we make the following contributions through this work.

- We initiate the study of fairness audit under partial feedback, an important setting that captures several real-world scenarios. We introduce a novel cost model for data exploration to capture a central concern, namely, the audit or the self-assessment incurs a cost when the labels requested on additional unlabeled data turn out to be negative (e.g., loan default, unfavorable outcomes in randomized control trials).
- For our cost model, we provide a complete characterization of the sample and cost complexity required for auditing fairness in the non-parametric setting by establishing a lower bound (Theorem 4.3) and proposing an audit algorithm that matches this bound up to logarithmic factors (Theorem 4.2). The proposed algorithm provably improves over the audit cost of a natural baseline algorithm (Theorem 4.1, Remark 7).
- To further investigate auditing with partial feedback, we study a structured setting where the group-wise distribution of features and labels follows a mixture of exponential families, a common analytical assumption in fair learning [27, 59]. In this setting, we design a new algorithm that achieves significantly lower audit cost under mild assumptions (Theorem 4.4).

- Our algorithms and their cost guarantees are agnostic to the hypothesis class and work for any classifier whose fairness is to be audited. We empirically study them on standard real-world datasets in fair classification (e.g., Adult Income, Law School) for logistic regression classifiers with significantly improved cost of audit or self-assessment.

To the best of our knowledge, this work is the first to address the important problem of fairness audit under the partial feedback setting capturing several real-world scenarios. While the problems of learning fair classifiers in partial feedback, and of fairness audit with full data have been studied (see next section), through this work, we aim to provide a principled framework for fairness audit that accounts for the cost of additional data in the partial feedback setting.

Our results in the blackbox setting require several novel techniques both in terms of algorithm design and analysis. To establish the cost lower bound (Theorem 4.3), we formalize the sequential interaction of the auditor with the online partial feedback data and derive a KL-divergence decomposition lemma (Lemma 4.1, Section 4.3.3). Furthermore, we carefully design two (family of) instances that are close in KL-divergence but only one of them is fair to conclude the lower bound. For the design of algorithm, we rely on the design of a stopping time and analyse it via a novel use of the concentration of the Negative Binomial distribution instead of the standard Chernoff-Hoeffding type bounds.

In the mixture model setting, we provide a novel perspective by viewing the classifier f as inducing a truncation on the input distribution by positively classifying only a subset of samples. This viewpoint allows us to leverage tools from learning with truncated samples [26, 54] to address the fairness audit problem under partial feedback. We believe this novel connection can be of wider interest, for example in understanding the truncation arising from popular fair classification methods. Furthermore, we generalize prior work on maximum a-posteriori (MAP) oracles for mixtures of spherical Gaussians [61] to handle mixtures of exponential families, which may be of independent interest.

1.4 Overview of the Thesis

This thesis focuses on investigating the aforementioned problems, with each chapter dedicated to studying a specific problem. The contributions of each chapter are outlined below:

- In Chapter 2, we investigate the problem of diversity in nearest neighbor search, introducing a welfarist objective that balances diversity and relevance in a query-dependent way, and developing efficient algorithms for it. We also provide strong empirical validation

demonstrating the effectiveness of our formulation over several real-world and synthetic datasets.

- In Chapter 3, we focus on aligning LLMs via RLHF by posing it as a contextual dueling bandit problem. Within this framework, we study the problem of optimally choosing which context and a pair of actions to query the stochastic comparison oracle with, when the number of queries allowed is limited. We show that uniformly random query selection can be arbitrarily bad, following which we establish a minimax lower bound. We also design an algorithm whose performance nearly matches the lower bound. Finally, we provide empirical validation of our algorithm.
- In Chapter 4, we study the problem of auditing fairness under partial feedback. Posing the auditing as a hypothesis test, we propose a novel cost model, and develop algorithms in distribution-free and structured settings. We also establish a minimax lower bound in the distribution-free setting. Our experiments demonstrate the superior performance of our algorithms in real-world datasets.

Chapter 2

Diversity in Similarity Search: A Welfarist Perspective

2.1 Problem Formulation and Main Results

We are interested in neighbor search algorithms that not only achieve a high relevance, but also find a diverse set of vectors for each query. To quantify diversity we work with a model wherein each input vector $v \in P$ is assigned one or more attributes from the set $[c] = \{1, 2, \dots, c\}$. In particular, write $\text{atb}(v) \subseteq [c]$ to denote the attributes assigned to vector $v \in P$. Also, let $D_\ell \subseteq P$ denote the subset of vectors that are assigned attribute $\ell \in [c]$, i.e., $D_\ell := \{v \in P \mid \ell \in \text{atb}(v)\}$.

2.1.1 Our Results

An insight of this work is to equate these c attributes with c distinct agents. Here, the output of a neighbor search algorithm—i.e., the selected subset $S \subseteq P$ —induces utility among these agents. With this perspective, we define the Nash Nearest Neighbor Search problem (NaNNS) below. This novel formulation for diversity is a key contribution of this work. For any query $q \in \mathbb{R}^d$ and subset $S \subseteq P$, we define utility $u_\ell(S) := \sum_{v \in S \cap D_\ell} \sigma(q, v)$, for each $\ell \in [c]$. That is, $u_\ell(S)$ is equal to the cumulative similarity between q and the vectors in S that belong to D_ℓ . Equivalently, $u_\ell(S)$ is the cumulative similarity of the vectors in S that have attribute ℓ .¹

We employ Nash social welfare to identify size- k subsets S that are both relevant (with respect to similarity) and support diversity among the c attribute classes. The Nash social welfare among c agents is defined as the geometric mean of the agents' utilities. Specifically, in the above-mentioned utility model and with a smoothening parameter $\eta > 0$, the Nash social

¹Note that in the above-mentioned display-advertising example, $u_\ell(\cdot)$ is the cumulative similarity between the (search) query and the selected advertisements that are from seller ℓ .

welfare (NSW) induced by any subset $S \subseteq P$ among the c attributes is defined as

$$\text{NSW}(S) := \left(\prod_{\ell=1}^c (u_{\ell}(S) + \eta) \right)^{1/c}. \quad (2.1)$$

Throughout, $\eta > 0$ will be a fixed smoothing constant that ensures that NSW remains nonzero.

Definition 2.1 (NaNNS) *Nash nearest neighbor search (NaNNS) corresponds to the following optimization problem $\arg\max_{S \subseteq P: |S|=k} \text{NSW}(S)$, or, equivalently,*

$$\arg\max_{S \subseteq P: |S|=k} \log \text{NSW}(S) \quad (2.2)$$

Here, we have $\log \text{NSW}(S) = \frac{1}{c} \sum_{\ell \in [c]} \log(u_{\ell}(S) + \eta)$.

To further appreciate the welfarist approach, note that one recovers the standard nearest neighbor problem, NNS, in the single-attribute setting, if—instead of the geometric mean—we maximize the arithmetic mean. That is, maximizing the utilitarian social welfare gives us $\max_{S \subseteq P: |S|=k} \sum_{\ell=1}^c u_{\ell}(S) = \max_{S \subseteq P: |S|=k} \sum_{v \in S} \sigma(q, v)$.

As stated in the introduction, depending on the query and the problem instance, solutions obtained via NaNNS can adjust between the ones obtained through standard NNS and those obtained via hard constraints. This feature is illustrated in the following stylized examples; see Appendix 2.5 for the associated proofs.

The first example shows that if all vectors have same similarity, then an optimal solution, S^* , for NaNNS is completely diverse, i.e., all the vectors in S^* have different attributes.

Example 2.1 (Complete Diversity via NaNNS) *Consider an instance in which, for a given query $q \in \mathbb{R}^d$, all vectors in P are equally similar with the query: $\sigma(q, v) = 1$ for all $v \in P$. Also, let $|\text{atb}(v)| = 1$ for all $v \in P$, and write $S^* \in \arg\max_{S \subseteq P: |S|=k} \text{NSW}(S)$. If $c \geq k$, then here it holds that $|S^* \cap D_{\ell}| \leq 1$ for all $\ell \in [c]$.*

The second example shows that if the vectors of only one attribute have high similarity with the given query, then a Nash optimal solution S^* contains only vectors with that attribute.

Example 2.2 (Complete Relevance via NaNNS) *Consider an instance in which, for a given query $q \in \mathbb{R}^d$ and a particular $\ell^* \in [c]$, only vectors $v \in D_{\ell^*}$ have similarity $\sigma(q, v) = 1$ and all other vectors $v' \in P \setminus D_{\ell^*}$ have similarity $\sigma(q, v') = 0$. Also, suppose that $|\text{atb}(v)| = 1$ for each $v \in P$, along with $|D_{\ell^*}| \geq k$. Then, for a Nash optimal solution $S^* \in \arg\max_{S \subseteq P, |S|=k} \text{NSW}(S)$, it holds that $|S^* \cap D_{\ell^*}| = k$. That is, for all other $\ell \in [c] \setminus \{\ell^*\}$ we have $|S^* \cap D_{\ell}| = 0$.*

With the above-mentioned utility model for the c attributes, we also identify an extended formulation based on generalized p -means. Specifically, for exponent parameter $p \in (-\infty, 1]$, the p th mean $M_p(\cdot)$, of c nonnegative numbers $w_1, w_2, \dots, w_c \in \mathbb{R}_+$, is defined as

$$M_p(w_1, \dots, w_c) := \left(\frac{1}{c} \sum_{\ell=1}^c w_\ell^p \right)^{1/p} \quad (2.3)$$

Note that $M_1(w_1, \dots, w_c)$ is the arithmetic mean $\frac{1}{c} \sum_{\ell=1}^c w_\ell$. Here, when $p \rightarrow 0$, we obtain the geometric mean (Nash social welfare): $M_0(w_1, \dots, w_c) = (\prod_{\ell=1}^c w_\ell)^{1/c}$. Further, $p \rightarrow -\infty$ gives us egalitarian welfare, $M_{-\infty}(w_1, \dots, w_c) = \min_{1 \leq \ell \leq c} w_\ell$.

Analogous to NaNNS, we consider a fixed smoothing constant $\eta > 0$, and, for each subset $S \subseteq P$, define $M_p(S) := M_p(u_1(S) + \eta, \dots, u_c(S) + \eta)$.

With these constructs in hand and generalizing both NNS and NaNNS, we have the p -mean nearest neighbor search (p -NNS) problem defined as follows.

Definition 2.2 (p -NNS) *For any exponent parameters $p \in (-\infty, 1]$, the p -mean nearest neighbor search (p -NNS) corresponds to the following optimization problem*

$$\max_{S \subseteq P: |S|=k} M_p(S) = \max_{S \subseteq P: |S|=k} \left(\frac{1}{c} \sum_{\ell=1}^c (u_\ell(S) + \eta)^p \right)^{1/p} \quad (2.4)$$

Diversity in Single- and Multi-Attribute Settings. The current work addresses two diversity settings: the single-attribute setup and, the more general, the multi-attribute one. The single-attribute setting refers to the case wherein $|\text{atb}(v)| = 1$ for each input vector $v \in P$ and, hence, the attribute classes D_ℓ s are pairwise disjoint. In the more general multi-attribute setting, we have $|\text{atb}(v)| \geq 1$; here, the sets D_ℓ -s intersect.¹ Notably, the NaNNS seamlessly applies to both these settings.

Algorithmic Results for Single-Attribute NaNNS and p -NNS. In addition to introducing the NaNNS and p -NNS formulations for capturing diversity, we develop algorithmic results for these problems, thereby demonstrating the practicality of our approach in neighbor search. In particular, in the single-attribute setting, we show that both NaNNS and p -NNS admit efficient algorithms.

Theorem 2.1 *In the single-attribute setting, given any query $q \in \mathbb{R}^d$ and an (exact) oracle **ENN** for k most similar vectors from any set, Algorithm 1 (**Nash-ANN**) returns an optimal solution for*

¹For a motivating instantiation for multi-attributes, note that, in the apparel-search context, it is possible for a product (input vector) v to have multiple attributes based on v 's seller and its color(s).

NaNNS, i.e., it returns a size- k subset $\text{ALG} \subseteq P$ that satisfies $\text{ALG} \in \arg\max_{S \subseteq P: |S|=k} \text{NSW}(S)$. Furthermore, the algorithm runs in time $O(kc) + \sum_{\ell=1}^c \text{ENN}(D_\ell, q)$, where $\text{ENN}(D_\ell, q)$ is the time required by the exact oracle to find k most similar vectors to q in D_ℓ .

Further, to establish the practicality of our formulations, we present an approximate algorithm for NaNNS that leverages any standard ANN algorithm as an oracle (subroutine), i.e., works with any α -approximate ANN oracle ($\alpha \in (0, 1)$) which returns a subset S containing k vectors satisfying $\sigma(q, v_{(i)}) \geq \alpha \sigma(q, v_{(i)}^*)$, for all $i \in [k]$, where $v_{(i)}$ and $v_{(i)}^*$ are the i -th most similar vectors to q in S and P , respectively. Formally,

Corollary 2.1 *In the single-attribute setting, given any query $q \in \mathbb{R}^d$ and an α -approximate oracle ANN for k most similar vectors from any set, Algorithm 1 (**Nash-ANN**) returns an α -approximate solution for NaNNS, i.e., it returns a size- k subset $\text{ALG} \subseteq P$ with $\text{NSW}(\text{ALG}) \geq \alpha \max_{S \subseteq P: |S|=k} \text{NSW}(S)$. The algorithm runs in time $O(kc) + \sum_{\ell=1}^c \text{ANN}(D_\ell, q)$, where $\text{ANN}(D_\ell, q)$ is the time required by the oracle to find k similar vectors to q in D_ℓ .*

Furthermore, both Theorem 2.1 and Corollary 2.1 generalize to p -NNS problem with a slight modification in Algorithm 1. Specifically, there exists exact, efficient algorithm (Algorithm 3) for the p -NNS problem (Theorem 2.4 and Corollary 2.2). Appendix 2.6 details this algorithm and the results for p -NNS in the single-attribute setting.

Algorithmic Results for Multi-Attribute NaNNS. Next, we address the multi-attribute setting. While the optimization problem (2.2) in the single attribute setting can be solved efficiently, the problem is NP-hard in the multi-attribute setup (see Section 2.3.1 for the proof).

Theorem 2.2 *In the multi-attribute setting, with parameter $\eta = 1$, NaNNS is NP-hard.*

Complementing this hardness result, we show that, considering the logarithm of the objective, NaNNS in the multi-attribute setting admits a polynomial-time $(1 - \frac{1}{e})$ -approximation algorithm. This result is established in Section 2.3.3.

Theorem 2.3 *In the multi-attribute setting with parameter $\eta = 1$, there exists a polynomial-time algorithm (Algorithm 2) that, given any query $q \in \mathbb{R}^d$, finds a size- k subset $\text{ALG} \subseteq P$ with $\log \text{NSW}(\text{ALG}) \geq (1 - \frac{1}{e}) \log \text{NSW}(\text{OPT})$; here, OPT denotes an optimal solution for the optimization problem (2.2).*

Experimental Validation of our Formulation and Algorithms. We complement our theoretical results with several experiments on real-world datasets. Our findings highlight that the Nash-based formulation strikes a balance between diversity and relevance. Specifically, we find that, across datasets and in both single- and multi-attribute settings, the Nash formulation maintains relevance and consistently achieves high diversity. By contrast, the hard-constrained formulation from [3] is highly sensitive to the choice of the constraint parameter k' , and in some cases incurs a substantial drop in relevance. Section 2.4 details our experimental evaluations.

2.2 NaNNS in the Single-Attribute Setting

Algorithm 1: Nash-ANN: Algorithm for NaNNS in the single-attribute setting

Input: Query $q \in \mathbb{R}^d$ and, for each attribute $\ell \in [c]$, the set of input vectors $D_\ell \subset \mathbb{R}^d$

- 1 For each $\ell \in [c]$, fetch $\widehat{D}_\ell$, the k (exact or approximate) nearest neighbors of $q \in \mathbb{R}^d$ from D_ℓ .
- 2 For every $\ell \in [c]$ and each index $i \in [k]$, let $v_{(i)}^\ell$ denote the i th most similar vector to q in $\widehat{D}_\ell$.
- 3 Initialize subset $\text{ALG} \leftarrow \emptyset$, along with count $k_\ell \leftarrow 0$ and utility $w_\ell \leftarrow 0$, for each $\ell \in [c]$.
- 4 **while** $|\text{ALG}| < k$ **do**
- 5 $a \leftarrow \underset{\ell \in [c]}{\operatorname{argmax}} \left(\log(w_\ell + \eta + \sigma(q, v_{(k_\ell+1)}^\ell)) - \log(w_\ell + \eta) \right)$. $\triangleright$ **Ties broken arbitrarily**
- 6 $\text{ALG} \leftarrow \text{ALG} \cup \{v_{(k_a+1)}^a\}$. $w_a \leftarrow w_a + \sigma(q, v_{(k_a+1)}^a)$. $k_a \leftarrow k_a + 1$.
- 7 **return** ALG

In this section, we first provide our exact, efficient algorithm (Algorithm 1) for NaNNS in the single-attribute setting and then present the proof of optimality of the algorithm (i.e., proof of Theorem 2.1 and Corollary 2.1).

The algorithm has two parts: a prefetching step and a greedy, iterative selection. In the prefetching step, for each attribute $\ell \in [c]$, we populate k vectors from within D_ℓ ¹ that are most similar to the given query $q \in \mathbb{R}^d$. Such a size- k subset, for each $\ell \in [c]$, can be obtained by executing any nearest neighbor search algorithm within D_ℓ and with respect to query q . Alternatively, we can execute any standard ANN algorithm as a subroutine and find sufficiently good approximations for the k nearest neighbors (of q) within each D_ℓ .

Write $\widehat{D}_\ell \subseteq D_\ell$ to denote the k —exact or approximate—nearest neighbors of $q \in \mathbb{R}^d$ in D_ℓ . We note that our algorithm is robust to the choice of the search algorithm (subroutine) used for

¹Recall that in the single-attribute setting, the input vectors P are partitioned into subsets $D_1, \dots, D_c$, where D_ℓ denotes the subset of input vectors with attribute $\ell \in [c]$.

finding $\widehat{D}_\ell$ s: If $\widehat{D}_\ell$ s are exact nearest neighbors, then Algorithm 1 optimally solves NaNNS in the single-attribute setting (Theorem 2.1). Otherwise, if $\widehat{D}_\ell$ s are obtained via an ANN algorithm with approximation guarantee $\alpha \in (0, 1)$, then Algorithm 1 achieves an approximation ratio of α (Corollary 2.1).

The algorithm then considers the vectors with each $\widehat{D}_\ell$ in decreasing order of their similarity with q . Confining to this order, the algorithm populates the k desired vectors iteratively. In each iteration, the algorithm greedily selects a new vector based on maximizing the marginal increase in $\log \text{NSW}(\cdot)$; see Lines 5 and 6 in Algorithm 1. Theorem 2.1 and Corollary 2.1 (stated previously) provide our main results for Algorithm 1. Proofs of Theorem 2.1 and Corollary 2.1 are presented below.

2.2.1 Proofs of Theorem 2.1 and Corollary 2.1

Now we state the proof of optimality of Algorithm 1. To obtain the proof, we make use of two lemmas stated below. We defer their proofs till after the proof of Theorem 2.1 and Corollary 2.1.

As in Algorithm 1, write $\widehat{D}_\ell$ to denote the k nearest neighbors of the given query q in the set D_ℓ . Recall that in the single-attribute setting the sets D_ℓ s are disjoint across $\ell \in [c]$. Also, $v_{(j)}^\ell \in \widehat{D}_\ell$ denotes the j^{th} most similar vector to q in $\widehat{D}_\ell$, for each index $j \in [k]$. For a given attribute $\ell \in [c]$, we define the logarithm of cumulative similarity upto the i^{th} most similar vector as

$$F_\ell(i) := \log \left(\sum_{j=1}^i \sigma(q, v_{(j)}^\ell) + \eta \right) \quad (2.5)$$

Note that $F_\ell(i)$ is also equal to the logarithm of the cumulative similarity of the i most similar (to q) vectors in D_ℓ when the neighbor search oracle is exact. The lemma below shows that $F_\ell(\cdot)$ satisfies a useful decreasing marginals property.

Lemma 2.1 (Decreasing Marginals) *For all attributes $\ell \in [c]$ and indices $i', i \in [k]$, with $i' < i$, it holds that*

$$F_\ell(i') - F_\ell(i' - 1) \geq F_\ell(i) - F_\ell(i - 1) .$$

The following lemma asserts the Nash optimality of the subset returned by Algorithm 1, ALG, within a relevant class of solutions.

Lemma 2.2 *In the single-attribute setting, let ALG be the subset of vectors returned by Algorithm 1 and S be any subset of input vectors with the property that $|S \cap D_\ell| = |\text{ALG} \cap D_\ell|$, for*

each $\ell \in [c]$. Then, $\text{NSW}(\text{ALG}) \geq \text{NSW}(S)$.

Now we are ready to state the proof of Theorem 2.1.

Theorem 2.1 *In the single-attribute setting, given any query $q \in \mathbb{R}^d$ and an (exact) oracle ENN for k most similar vectors from any set, Algorithm 1 (**Nash-ANN**) returns an optimal solution for NaNNS, i.e., it returns a size- k subset $\text{ALG} \subseteq P$ that satisfies $\text{ALG} \in \arg\max_{S \subseteq P: |S|=k} \text{NSW}(S)$. Furthermore, the algorithm runs in time $O(kc) + \sum_{\ell=1}^c \text{ENN}(D_\ell, q)$, where $\text{ENN}(D_\ell, q)$ is the time required by the exact oracle to find k most similar vectors to q in D_ℓ .*

Proof: The runtime of Algorithm 1 can be established by noting that Line 1 requires $\sum_{\ell=1}^c \text{ENN}(D_\ell, q)$ time to populate the subsets $\widehat{D}_{\ell s}$, and the while-loop (Lines 4-6) iterates k times and each iteration (specifically, Line 5) runs in $O(c)$ time. Hence, as stated, the time complexity of the algorithm is $O(kc) + \sum_{\ell=1}^c \text{ENN}(D_\ell, q)$.

Next, we prove the optimality of the returned set ALG . Let $\text{OPT} \in \arg\max_{S \subseteq P: |S|=k} \text{NSW}(S)$ be an optimal solution with attribute counts $|\text{OPT} \cap D_\ell|$ as close to $|\text{ALG} \cap D_\ell|$ as possible. That is, among the optimal solutions, it is one that minimizes $\sum_{\ell=1}^c |k_\ell^* - k_\ell|$, where $k_\ell^* = |\text{OPT} \cap D_\ell|$ and $k_\ell = |\text{ALG} \cap D_\ell|$, for each $\ell \in [c]$. We will prove that OPT satisfies $k_\ell^* = k_\ell$ for each $\ell \in [c]$. This guarantee, along with Lemma 2.2, implies that, as desired, ALG is a Nash optimal solution.

Assume, towards a contradiction, that $k_\ell^* \neq k_\ell$ for some $\ell \in [c]$. Since $|\text{OPT}| = |\text{ALG}| = k$, there exist attributes $x, y \in [c]$ with the property that $k_x^* < k_x$ and $k_y^* > k_y$. For a given attribute $\ell \in [c]$, write the logarithm of cumulative similarity upto the i^{th} most similar vector as $F_\ell(i) := \log(\sum_{j=1}^i \sigma(q, v_{(j)}^\ell) + \eta)$, where $v_{(j)}^\ell$ is defined in Line 2 of Algorithm 1.

Next, note that for any attribute $\ell \in [c]$, if Algorithm 1, at any point during its execution, has included k'_ℓ vectors of attribute ℓ in ALG , then at that point the maintained utility $w_\ell = \sum_{j=1}^{k'_\ell} \sigma(q, v_{(j)}^\ell)$. Hence, at the beginning of any iteration of the algorithm, if the k'_ℓ denotes the number of selected vectors of each attribute $\ell \in [c]$, then the marginals considered in Line 5 are $F_\ell(k'_\ell + 1) - F_\ell(k'_\ell)$. These observations and the selection criterion in Line 5 of the algorithm give us the following inequality for the counts $k_x = |\text{ALG} \cap D_x|$ and $k_y = |\text{ALG} \cap D_y|$ of the returned solution ALG :

$$F_x(k_x) - F_x(k_x - 1) \geq F_y(k_y + 1) - F_y(k_y) \quad (2.6)$$

Specifically, equation (2.6) follows by considering the iteration in which k_x^{th} (last) vector of attribute x was selected by the algorithm. Before that iteration the algorithm had selected $(k_x - 1)$ vectors of attribute x , and let k'_y denote the number of vectors with attribute y that

have been selected till that point. Note that $k'_y \leq k_y$. The fact that the k_x^{th} vector was (greedily) selected in Line 5, instead of including an additional vector of attribute y , gives $F_x(k_x) - F_x(k_x - 1) \geq F_y(k'_y + 1) - F_y(k'_y) \geq F_y(k_y + 1) - F_y(k_y)$; here, the last inequality follows from Lemma 2.1. Therefore we have,

$$F_x(k_x^* + 1) - F_x(k_x^*) \stackrel{(i)}{\geq} F_x(k_x) - F_x(k_x - 1) \stackrel{(ii)}{\geq} F_y(k_y + 1) - F_y(k_y) \stackrel{(iii)}{\geq} F_y(k_y^*) - F_y(k_y^* - 1) \quad (2.7)$$

Here, inequality (i) follows from $k_x^* < k_x$ and Lemma 2.1, inequality (ii) is due to equation (2.6), and inequality (iii) is via $k_y^* > k_y$ and Lemma 2.1.

Next, observe that the definition of $\widehat{D}_\ell$ ensures that $v_{(i)}^\ell$ is, in fact, the i^{th} most similar (to q) vector among the ones that have attribute ℓ , i.e., i^{th} most similar in all of D_ℓ . Since OPT is an optimal solution, the $k_\ell^* = |\text{OPT} \cap D_\ell|$ vectors of attribute ℓ in OPT are the most similar k_ℓ^* vectors from D_ℓ . That is, $\text{OPT} \cap D_\ell = \{v_{(1)}^\ell, \dots, v_{(k_\ell^*)}^\ell\}$, for each $\ell \in [c]$. This observation and the definition of $F_\ell(\cdot)$ imply that the logarithm of OPT's NSW satisfies $\log \text{NSW}(\text{OPT}) = \frac{1}{c} \sum_{\ell=1}^c F_\ell(k_\ell^*)$. Now, consider a subset of vectors S obtained from OPT by including vector $v_{(k_x^*+1)}^x$ and removing $v_{(k_y^*)}^y$, i.e., $S = \left(\text{OPT} \cup \left\{ v_{(k_x^*+1)}^x \right\} \right) \setminus \left\{ v_{(k_y^*)}^y \right\}$. Note that

$$\log \text{NSW}(S) - \log \text{NSW}(\text{OPT}) = \frac{1}{c} \left(F_x(k_x^* + 1) - F_x(k_x^*) \right) + \frac{1}{c} \left(F_y(k_y^* - 1) - F_y(k_y^*) \right) \geq 0,$$

where the last inequality is via equation (2.7). Hence, $\text{NSW}(S) \geq \text{NSW}(\text{OPT})$. Given that OPT is a Nash optimal solution, the last inequality must hold with an equality, $\text{NSW}(S) = \text{NSW}(\text{OPT})$, i.e., S is an optimal solution as well. This, however, contradicts the choice of OPT as an optimal solution that minimizes $\sum_{\ell=1}^c |k_\ell^* - k_\ell|$; note that $\sum_{\ell=1}^c |\widehat{k}_\ell - k_\ell| < \sum_{\ell=1}^c |k_\ell^* - k_\ell|$, where $\widehat{k}_\ell := |S \cap D_\ell|$.

Therefore, by way of contradiction, we obtain that $|\text{OPT} \cap D_\ell| = |\text{ALG} \cap D_\ell|$ for each $\ell \in [c]$. As mentioned previously, this guarantee along with Lemma 2.2 imply that ALG is a Nash optimal solution. This completes the proof of the theorem. $\square$

Corollary 2.1 *In the single-attribute setting, given any query $q \in \mathbb{R}^d$ and an α -approximate oracle ANN for k most similar vectors from any set, Algorithm 1 (Nash-ANN) returns an α -approximate solution for NaNNS, i.e., it returns a size- k subset $\text{ALG} \subseteq P$ with $\text{NSW}(\text{ALG}) \geq \alpha \max_{S \subseteq P: |S|=k} \text{NSW}(S)$. The algorithm runs in time $O(kc) + \sum_{\ell=1}^c \text{ANN}(D_\ell, q)$, where $\text{ANN}(D_\ell, q)$ is the time required by the oracle to find k similar vectors to q in D_ℓ .*

Proof: The running time of the algorithm follows via an argument similar to the one used in the proof of Theorem 2.1. Therefore, we only argue correctness here.

For every $\ell \in [c]$, let the α -approximate oracle return $\widehat{D}_\ell$. Recall that $v_{(i)}^\ell$, $i \in [k]$, denotes the i^{th} most similar point to q in the set $\widehat{D}_\ell$. Further, for every $\ell \in [c]$, let D_ℓ^* be the set of k most similar points to q within D_ℓ and define $v_{(i)}^{*\ell}$, $i \in [k]$, to be the i^{th} most similar point to q in D_ℓ^* . Recall that by the guarantee of the α -approximate NNS oracle, we have $\sigma(q, v_{(i)}^\ell) \geq \alpha \cdot \sigma(q, v_{(i)}^{*\ell})$ for all $i \in [k]$. Let OPT be an optimal solution to the NaNNS problem containing k_ℓ^* most similar points of attribute ℓ for every $\ell \in [c]$.

Finally, let $\widehat{\text{OPT}}$ be the optimal solution to the NaNNS problem when the set of vectors to search over is $P = \cup_{\ell \in [c]} \widehat{D}_\ell$.

By an argument similar to the proof of Theorem 2.1, we have $\text{NSW}(\text{ALG}) = \text{NSW}(\widehat{\text{OPT}})$. Therefore

$$\begin{aligned}
\text{NSW}(\text{ALG}) &= \text{NSW}(\widehat{\text{OPT}}) \\
&\geq \left(\prod_{\ell \in [c]} \left(\sum_{i=1}^{k_\ell^*} \sigma(q, v_{(i)}^\ell) + \eta \right) \right)^{\frac{1}{c}} \quad (\cup_{\ell \in [c]: k_\ell^* \geq 1} \{v_{(1)}^\ell, \dots, v_{(k_\ell^*)}^\ell\} \text{ is a feasible solution}) \\
&\geq \left(\prod_{\ell \in [c]} \left(\sum_{i=1}^{k_\ell^*} \alpha \sigma(q, v_{(i)}^{*\ell}) + \eta \right) \right)^{\frac{1}{c}} \\
&\quad \text{(by } \alpha\text{-approximate guarantee of the oracle; } k_\ell^* \leq k) \\
&\geq \left(\prod_{\ell \in [c]} \alpha \left(\sum_{i=1}^{k_\ell^*} \sigma(q, v_{(i)}^{*\ell}) + \eta \right) \right)^{\frac{1}{c}} \quad (\alpha \in (0, 1)) \\
&= \alpha \text{NSW}(\text{OPT}) \quad \text{(definition of OPT)}
\end{aligned}$$

Hence, the corollary stands proved. $\square$

We complete this section by stating the proofs of Lemmas 2.1 and 2.2.

Proof: [Proof of Lemma 2.1] Note that $\exp(F_\ell(i)) = \sum_{j=1}^i \sigma(q, v_{(j)}^\ell) + \eta = \exp(F_\ell(i-1)) + \sigma(q, v_{(i)}^\ell)$. Therefore, we have

$$\exp(F_\ell(i) - F_\ell(i-1)) = \frac{\exp(F_\ell(i))}{\exp(F_\ell(i-1))} = \frac{\exp(F_\ell(i-1)) + \sigma(q, v_{(i)}^\ell)}{\exp(F_\ell(i-1))} = 1 + \frac{\sigma(q, v_{(i)}^\ell)}{\exp(F_\ell(i-1))} \quad (2.8)$$

Similarly, we have $\exp(F_\ell(i') - F_\ell(i'-1)) = 1 + \frac{\sigma(q, v_{(i')}^\ell)}{\exp(F_\ell(i'-1))}$.

In addition, the indexing of the vectors $v_{(j)}^\ell$ ensures that $\sigma(q, v_{(j')}^\ell) \geq \sigma(q, v_{(j)}^\ell)$ for $j' < j$. Moreover, $\exp(F_\ell(i))$ is non-decreasing since it is the cumulative sum of non-negative similarities

upto i^{th} vector $v_{(i)}^\ell$. Hence, $\exp(F_\ell(i)) \geq \exp(F_\ell(i'))$ for $i' < i$. Combining these inequalities, we obtain

$$\frac{\sigma(q, v_{(i')}^\ell)}{\exp(F_\ell(i' - 1))} \geq \frac{\sigma(q, v_{(i)}^\ell)}{\exp(F_\ell(i - 1))}.$$

That is, $1 + \frac{\sigma(q, v_{(i')}^\ell)}{\exp(F_\ell(i' - 1))} \geq 1 + \frac{\sigma(q, v_{(i)}^\ell)}{\exp(F_\ell(i - 1))}$. Hence, equation (2.8) gives us

$$\exp(F_\ell(i') - F_\ell(i' - 1)) \geq \exp(F_\ell(i) - F_\ell(i - 1)) \quad (2.9)$$

Since $\exp(\cdot)$ is an increasing function, inequality (2.9) implies

$$F_\ell(i') - F_\ell(i' - 1) \geq F_\ell(i) - F_\ell(i - 1) .$$

The lemma stands proved. $\square$

Proof: [Proof of Lemma 2.2] Assume, towards a contradiction, that there exists a subset of input vectors S that satisfies $|S \cap D_\ell| = |\text{ALG} \cap D_\ell|$, for each $\ell \in [c]$, and still induces NSW strictly greater than that of ALG. This strict inequality implies that there exists an attribute $a \in [c]$ with the property that the utility $u_a(S) > u_a(\text{ALG})$.

$$\sum_{t \in S \cap D_a} \sigma(q, t) > \sum_{v \in \text{ALG} \cap D_a} \sigma(q, v) \quad (2.10)$$

On the other hand, note that the construction of Algorithm 1 and the definition of $\hat{D}_a$ ensure that the vectors in $\text{ALG} \cap D_a$ are, in fact, the most similar to q among all the vectors in D_a . This observation and the fact that $|S \cap D_a| = |\text{ALG} \cap D_a|$ gives us $\sum_{v \in \text{ALG} \cap D_a} \sigma(q, v) \geq \sum_{t \in S \cap D_a} \sigma(q, t)$. This equation, however, contradicts the strict inequality (2.10).

Therefore, by way of contradiction, we obtain that there does not exist a subset S such that $|S \cap D_\ell| = |\text{ALG} \cap D_\ell|$, for each $\ell \in [c]$, and $\text{NSW}(\text{ALG}) < \text{NSW}(S)$. The lemma stands proved. $\square$

Remark 1 Corollary 2.1 assumes that the α -approximate nearest neighbor oracle **ANN**, when run on a set of vectors $P' \subseteq \mathbb{R}^d$, $|P'| = n$, and query vector $q \in \mathbb{R}^d$, returns an ordered set $L \subseteq P'$, $|L| = k$ such that the i -th vector $\hat{v}_{(i)}$ in L satisfies $\sigma(q, \hat{v}_{(i)}) \geq \alpha \cdot \sigma(q, v_{(i)}^*)$ for all $i \in [k]$; here $v_{(i)}^*$ is the i -th most similar vector in P' to query q . Moreover, recall that the similarity function $\sigma(\cdot, \cdot)$ is non-negative, which is crucial for the proof of Corollary 2.1.

2.3 NaNNS in the Multi-Attribute Setting

In this section, we first establish the NP-Hardness of the NaNNS problem in the multi-attribute setting by stating the proof of Theorem 2.2. Thereafter, we describe an efficient algorithm that obtains a constant approximation in terms of the logarithm of the Nash Social Welfare objective, and prove the approximation ratio (Theorem 2.3).

2.3.1 Proof of Theorem 2.2

Recall that in the multi-attribute setting, input vectors $v \in P$ are associated with one or more attributes, $|\text{atb}(v)| \geq 1$.

Theorem 2.2 *In the multi-attribute setting, with parameter $\eta = 1$, NaNNS is NP-hard.*

Proof: Consider the decision version of the optimization problem: given a threshold $W \in \mathbb{Q}$, decide whether there exists a size- k subset $S \subseteq P$ such that $\log \text{NSW}(S) \geq W$. We will refer to this problem as NaNNS. Note that the input in an NaNNS instance consists of: a set of n vectors $P \subset \mathbb{R}^d$, a similarity function $\sigma : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}_+$, an integer $k \in \mathbb{N}$, the sets $D_\ell = \{p \in P : \ell \in \text{atb}(p)\}$ for every attribute $\ell \in [c]$, a query point $q \in \mathbb{R}^d$, and threshold $W \in \mathbb{Q}$. We will show that NaNNS is NP-complete by reducing EXACT REGULAR SET PACKING (ERSP) to it. ERSP is known to be NP-complete [32] and is also W[1]-hard with respect to solution size [7].

In ERSP, we are given a universe of n elements, $\mathcal{U} = \{1, 2, \dots, n\}$, an integer $k \in \mathbb{N}$, and a collection of m subsets $\mathcal{S} = \{S_1, \dots, S_m\}$, with each subset $S_i \subseteq \mathcal{U}$ of cardinality τ (i.e., $|S_i| = \tau$ for each $i \in [m]$). The objective here is to decide whether there exists a size- k sub-collection $I \subseteq \mathcal{S}$ such that for all distinct $S, S' \in I$ we have $S \cap S' = \emptyset$.

For the reduction, we start with the given instance of ERSP and construct an instance of NaNNS: Consider $\mathcal{U}$ as the set of attributes, i.e., set $c = n$. In addition, we set the input vectors $P = \{\frac{1}{\tau} \mathbf{1}_S \mid S \in \mathcal{S}\}$; here, $\mathbf{1}_S \in \mathbb{R}^n$ is the characteristic vector (in $\mathbb{R}^n$) of the subset S , i.e., for each $i \in [n]$, the i -th coordinate of $\mathbf{1}_S$ is $\mathbb{1}\{i \in S\}$. Note that the set of vectors P is of cardinality m .

Furthermore, we set the query vector $q = \mathbf{1}$ as the all-ones vector in $\mathbb{R}^n$. In this NaNNS instance, each input vector $\frac{1}{\tau} \mathbf{1}_S$ is assigned attribute $\ell \in [n]$ iff element $\ell \in S$. That is, $D_\ell = \{\frac{1}{\tau} \mathbf{1}_S \mid S \in \mathcal{S} \text{ and } \ell \in S\}$. The number of neighbors to be found in the constructed NaNNS is equal to k , which is the count in the given the ERSP instance. Also, the similarity function $\sigma : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ is taken to be the standard dot-product. Finally, we set the threshold $W = \frac{\tau k \log 2}{c}$.

Note that the reduction takes time polynomial in n and m . In addition, for each input vector $v \in P$ it holds that $v = \frac{1}{\tau} \cdot \mathbf{1}_S$ for some $S \in \mathcal{S}$ and, hence, $\sigma(q, v) = \langle \frac{1}{\tau} \mathbf{1}_S, \mathbf{1} \rangle = 1$.

Now we establish the correctness of the reduction.

Forward direction “ $\Rightarrow$ ”: Suppose the given **ERSP** instance admits a (size- k) solution $I^* \subset \mathcal{S}$. Consider the subset of vectors $N^* := \{\frac{1}{\tau} \mathbf{1}_S | S \in I^*\}$. Indeed, $N^* \subseteq P$ and $|N^*| = k$, hence N^* is a feasible set of the **NaNNS** problem. Now, since I^* is a solution to the **ERSP** instance, for distinct $S, S' \in I^*$ we have $S \cap S' = \emptyset$. In particular, if for an element $\ell \in [c]$, we have $\ell \in S$ for some $S \in I^*$, then $\ell \notin S'$ for all $S' \in I^* \setminus \{S\}$. Therefore, $|N^* \cap D_\ell| \leq 1$ for all $\ell \in [c]$, which in turn implies that $u_\ell(N^*)$ is either 1 or 0 for every $\ell \in [c]$. Finally, note that each vector $v \in P$ belongs to exactly τ attributes, i.e., $|\text{atb}(v)| = \tau$. Hence,

$$\log \text{NSW}(N^*) = \frac{1}{c} \sum_{\ell=1}^c \log(1 + u_\ell(N^*)) = \frac{1}{c} \sum_{v \in N^*} \sum_{\ell \in \text{atb}(v)} \log(1 + 1) = \frac{\tau k \log 2}{c}.$$

Therefore, if the given **ERSP** instance admits a solution (exact packing), then the constructed **NaNNS** instance has k neighbors with sufficiently high $\log \text{NSW}$.

Reverse direction “ $\Leftarrow$ ”: Suppose $N^* \subseteq P$ is a solution in the constructed **NaNNS** instance with $|N^*| = k$ and $\log \text{NSW}(N^*) \geq W$. Define $I^* := \{S \mid \frac{1}{\tau} \cdot \mathbf{1}_S \in N^*\}$ and note that $|I^*| = k$. We will show that I^* is a solution for the given **ERSP** instance, i.e., it consists of disjoint subsets.

Towards this, first note that N^* induces social welfare:

$$\sum_{\ell \in [c]} u_\ell(N^*) = \sum_{\ell \in [c]} \sum_{v \in N^* \cap D_\ell} \sigma(q, v) = \sum_{v \in N^*} \sum_{\ell \in \text{atb}(v)} \sigma(q, v) = \tau k \quad (2.11)$$

Furthermore, any attribute $\ell \in [c]$ has a non-zero utility under N^* iff $\ell \in S$ for some subset $S \in I^*$. Hence, $\mathcal{A} := \cup_{S \in I^*} S$ corresponds to the set of attributes with non-zero utility under N^* . We have $1 \leq |\mathcal{A}| \leq \tau k$. Next, using the fact that $\log \text{NSW}(N^*) \geq W = \frac{\tau k \log 2}{c}$ we obtain

$$\begin{aligned} \frac{\tau k \log 2}{c} &\leq \log \text{NSW}(N^*) = \frac{1}{c} \sum_{\ell \in [c]} \log(1 + u_\ell(N^*)) \\ &= \frac{1}{c} \sum_{\ell \in \mathcal{A}} \log(1 + u_\ell(N^*)) \\ &= \frac{|\mathcal{A}|}{c} \frac{1}{|\mathcal{A}|} \sum_{\ell \in \mathcal{A}} \log(1 + u_\ell(N^*)) \\ &\leq \frac{|\mathcal{A}|}{c} \log \left(\frac{1}{|\mathcal{A}|} \sum_{\ell \in \mathcal{A}} (1 + u_\ell(N^*)) \right) \quad (\text{concavity of log}) \end{aligned}$$

$$\begin{aligned}
&= \frac{|\mathcal{A}|}{c} \log \left(1 + \frac{\sum_{\ell \in \mathcal{A}} u_{\ell}(N^*)}{|\mathcal{A}|} \right) \\
&= \frac{|\mathcal{A}|}{c} \log \left(1 + \frac{\tau k}{|\mathcal{A}|} \right) \quad (\text{via (2.11)}) \\
&\leq \frac{\tau k \log 2}{c}.
\end{aligned}$$

Here, the last inequality follows from Lemma 2.3 (stated and proved below). Hence, all the inequalities in the derivation above must hold with equality. In particular, we must have $|\mathcal{A}| = \tau k$ by the quality condition of Lemma 2.3. Therefore, for distinct sets $S, S' \in I^*$ it holds that $S \cap S' = \emptyset$. Hence, as desired, I^* is a solution of the ERSP instance.

This completes the correctness of the reduction, and the theorem stands proved. $\square$

Lemma 2.3 *For any $a > 0$ and for all $x \in (0, a]$ it holds that $x \log(1 + \frac{a}{x}) \leq a \log 2$. Furthermore, the equality holds iff $x = a$.*

Proof: Write $f(x) := x \log(1 + \frac{a}{x})$. At the end points of the domain $(0, a]$, the function $f(\cdot)$ satisfies: $f(a) = a \log(2)$ and

$$\lim_{x \rightarrow 0^+} f(x) = \lim_{x \rightarrow 0^+} x \log(a + x) - x \log x = \lim_{x \rightarrow 0^+} x \log(a + x) - \lim_{x \rightarrow 0^+} x \log(x) = 0 - 0 = 0.$$

Note that $f'(x) = \log(1 + \frac{a}{x}) - \frac{a}{a+x}$. We will show that $f'(x) > 0$ for all $x \in (0, a]$ which will conclude the proof.

Case 1: $x \in (0, \frac{a}{2}]$. We have $\log(1 + \frac{a}{x}) \geq \log(1 + \frac{a}{a/2}) = \log(3) > 1$. On the other hand, $\frac{a}{a+x} \leq 1$.

Case 2: $x \in (\frac{a}{2}, a]$. In this case, $\log(1 + \frac{a}{x}) \geq \log(1 + \frac{a}{a}) = \log(2) > 0.693$. However, $\frac{a}{a+x} < \frac{a}{a+\frac{a}{2}} = \frac{2}{3} \leq 0.667$.

Therefore, $f'(x) = \log(1 + \frac{a}{x}) - \frac{a}{a+x} > 0$ for all $x \in (0, a]$, which completes the proof. $\square$

2.3.2 Algorithm for the Multi-Attribute Setting

This section details Algorithm 2, based on which we obtain Theorem 2.3. The algorithm greedily selects the k neighbors for the given query q . Specifically, the algorithm iterates k times, and in each iteration, it selects a new vector (from the given set P) whose inclusion in the current solution ALG yields that maximum increase in log NSW (Line 3). After k iterations, the algorithm returns the k selected vectors ALG.

Algorithm 2: MultiNashANN: Approximation algorithm in the multi-attribute setting

Input: Query $q \in \mathbb{R}^d$, and the set of input vectors $P \subset \mathbb{R}^d$

- 1 Initialize $\text{ALG} = \emptyset$.
- 2 **for** $i = 1$ **to** k **do**
- 3 Set $\hat{v} = \operatorname{argmax}_{v \in P \setminus \text{ALG}} (\log \text{NSW}(\text{ALG} \cup \{v\}) - \log \text{NSW}(\text{ALG}))$.
- 4 Update $\text{ALG} \leftarrow \text{ALG} \cup \{\hat{v}\}$.
- 5 **return** ALG

2.3.3 Proof of Theorem 2.3

We now establish Theorem 2.3, which provides the approximation ratio achieved by Algorithm 2.

Theorem 2.3 *In the multi-attribute setting with parameter $\eta = 1$, there exists a polynomial-time algorithm (Algorithm 2) that, given any query $q \in \mathbb{R}^d$, finds a size- k subset $\text{ALG} \subseteq P$ with $\log \text{NSW}(\text{ALG}) \geq (1 - \frac{1}{e}) \log \text{NSW}(\text{OPT})$; here, OPT denotes an optimal solution for the optimization problem (2.2).*

Proof: For each subset $S \subseteq P$, write function $f(S) := \log \text{NSW}(S)$. Since parameter $\eta = 1$, we have $f(\emptyset) = 0$ and the function is nonnegative. Moreover, we will show that this set function $f : 2^P \rightarrow \mathbb{R}_+$ is monotone and submodular. Given that Algorithm 2 follows the marginal-gain greedy criterion in each iteration, it achieves a $(1 - \frac{1}{e})$ -approximation for the submodular maximization problem (2.2).

To establish the monotonicity of f , consider any pair of subsets $S \subseteq T \subseteq P$. Here, for each $\ell \in [c]$, we have $D_\ell \cap S \subseteq D_\ell \cap T$. Hence, $u_\ell(S) \leq u_\ell(T)$. Further, since $\log$ is an increasing function, it holds that $\log(u_\ell(S) + 1) \leq \log(u_\ell(T) + 1)$, for each $\ell \in [c]$. Hence, $f(S) \leq f(T)$, and we obtain that f is monotone.

For submodularity, let $S \subseteq T \subseteq P$ be any two subsets and let $w \in P \setminus T$. Write $S + w$ and $T + w$ to denote the sets $S \cup \{w\}$ and $T \cup \{w\}$, respectively. Here, we have

$$\begin{aligned}
& f(S + w) - f(S) - f(T + w) + f(T) \\
&= \frac{1}{c} \sum_{\ell \in [c]} \log \left(\frac{1 + \sum_{v \in D_\ell \cap (S+w)} \sigma(q, v)}{1 + \sum_{v \in D_\ell \cap S} \sigma(q, v)} \cdot \frac{1 + \sum_{v \in D_\ell \cap T} \sigma(q, v)}{1 + \sum_{v \in D_\ell \cap (T+w)} \sigma(q, v)} \right) \\
&= \frac{1}{c} \sum_{\ell \in \text{atb}(w)} \log \left(\left(1 + \frac{\sigma(q, w)}{1 + \sum_{v \in D_\ell \cap S} \sigma(q, v)} \right) \cdot \left(1 + \frac{\sigma(q, w)}{1 + \sum_{v \in D_\ell \cap T} \sigma(q, v)} \right)^{-1} \right)
\end{aligned}$$

$$\begin{aligned}
&= \frac{1}{c} \sum_{\ell \in \text{atb}(w)} \log \left(\left(1 + \frac{\sigma(q, w)}{1 + u_\ell(S)} \right) \cdot \left(1 + \frac{\sigma(q, w)}{1 + u_\ell(T)} \right)^{-1} \right) \\
&\geq 0 \qquad \qquad \qquad (u_\ell(S) \leq u_\ell(T) \text{ for } S \subseteq T)
\end{aligned}$$

Therefore, upon rearranging, we obtain $f(S + w) - f(S) \geq f(T + w) - f(T)$, i.e., f is submodular. Hence, Algorithm 2, which follows marginals-gain greedy selection, achieves a $(1 - \frac{1}{e})$ -approximation [71] for the optimization problem (2.2). $\square$

Remark 2 *The approximation guarantee in Theorem 2.3 is in terms of the log of Nash Social Welfare, which is the objective stated in the definition of NaNNS (Definition 2.1). Note that $x \mapsto \log(1 + x)$ is increasing and non-negative for $x \geq 0$. Thus, any $O^* \in \arg\max_{L \subseteq P, |L|=k} \text{NSW}(L)$ is also a solution of $\max_{L \subseteq P, |L|=k} \log \text{NSW}(L)$ and vice versa. Hence, Theorem 2.3 implies that the solution ALG returned by Algorithm 2 satisfies $\text{NSW}(\text{ALG}) \geq \text{NSW}(O^*)^{(1-1/e)}$. Achieving a multiplicative approximation of the form $\text{NSW}(\text{ALG}) \geq c \cdot \text{NSW}(O^*)$, for constant c , is an interesting future direction.*

2.4 Experimental Evaluations

In this section, we validate the welfare-based formulations and the performance of our proposed algorithms against existing methods on a variety of real and semi-synthetic datasets. We perform three sets of experiments:

- In the first set of experiments (Figure 2.3), we compare **Nash-ANN** (Algorithm 1) with prior work on hard-constraint based diversity [3]. Here, we show that **Nash-ANN** strikes a balance between relevance and diversity both in the single- and multi-attribute settings.
- In the second set of experiments (Figure 2.4), we study the effect of varying the exponent parameter p in the p -NNS objective on relevance and diversity, in both single- and multi-attribute settings.
- In the final set of experiments (Tables 2.2 and 2.3), we compare our algorithm, **Nash-ANN** (with provable guarantees), and a heuristic we propose to improve the runtime of **Nash-ANN**. The heuristic directly utilizes a standard ANN algorithm to first fetch a sufficiently large candidate set of vectors (irrespective of their attributes). Then, it applies the greedy procedure for Nash social (or p -mean) welfare maximization (similar to Lines 4-6 in Algorithm 1) only within this set.

In what follows, we provide the details of the experimental set-ups, the baseline algorithms and the results of the experiments.

Additional plots for the experiments appear in Appendices 2.7.1 and 2.7.2.

2.4.1 Metrics for Measuring Relevance and Diversity

Relevance Metrics: In our experiments, we capture the relevance of a solution to the query through two metrics detailed below.

1. **Approximation Ratio:** For a given query q , let S be the set of k vectors returned by an NNS algorithm that we wish to study, and let O be the k most similar vectors to q in P . Then the approximation ratio of the algorithm is defined as the ratio $\frac{\sum_{v \in S} \sigma(q, v)}{\sum_{v \in O} \sigma(q, v)}$. Therefore, a higher approximation ratio indicates a more relevant solution. Note that the highest possible value of this metric is 1.
2. **Recall:** For a given query q , let S be the set of k vectors returned by an NNS algorithm that we wish to study and let O be the k most similar vectors to q in P . The recall of the algorithm is defined as the quantity $\frac{|S \cap O|}{|O|}$. Therefore, higher the recall, more relevant the solution, and the maximum possible value of recall is 1.

Remark 3 *Although recall is a popular metric in the context of the standard NNS problem, it is important to note that it is a fragile metric when the objective is to retrieve a relevant and diverse set of vectors for a given query. This can be illustrated with the following stylized example in the single-attribute setting. Suppose for a given query q , all the vectors in the similarity-wise optimal set O have similarity 1 and share the same attribute $\ell^* \in [c]$, i.e., for each $u \in O$ we have $\sigma(q, u) = 1$ and $\text{atb}(u) = \ell^*$. That is, the set O of the k most similar vectors to q are not at all diverse. However, it is possible to have another set S of k vectors each with a distinct attribute and $\sigma(q, v) = 0.99$ for each $v \in S$. Such a set provides a highly relevant set of vectors that are also completely diverse. However, for the set S , the recall is actually 0 (since $S \cap O = \emptyset$), but the approximation ratio is 0.99. Hence, in the context of diverse nearest neighbor search problem, approximation ratio may be a more meaningful relevance metric than recall.*

Diversity Metrics: To measure the diversity of the solutions obtained by various algorithms, we consider the following metrics.

- **Entropy:** Let $S \subseteq P$ be a size- k subset computed by an algorithm. Then the entropy of the set S in the single-attribute setting is given by the quantity $\sum_{\ell \in [c]: p_\ell > 0} -p_\ell \log(p_\ell)$

Table 2.1: Summary of considered datasets. For synthetic attributes, we use two strategies: clustering-based (suffixed by **Clus**) and distribution-based (suffixed by **Prob**), see Section 2.4.2 for details.

Dataset	# Input Vectors	# Query Vectors	Dimension	Attributes
Amazon	92,092	8,956	768	product color
ArXiv	200,000	50,000	1536	year, paper category
Sift1m	1,000,000	10,000	128	synthetic
Deep1b	9,990,000	10,000	96	synthetic

where $p_\ell = \frac{|S \cap D_\ell|}{|S|}$. Note that a higher entropy value indicates greater diversity. Moreover, it is not hard to see that the highest possible value of entropy is $\log(k)$ (achieved when S contains at most 1 vector from each attribute).

- **Inverse Simpson Index:** For a given set $S \subseteq P$ in the single-attribute setting, the inverse Simpson index is defined as $\frac{1}{\sum_{\ell=1}^c p_\ell^2}$ where p_ℓ is the same as in the definition of entropy above. A higher value of this metric indicates greater diversity.
- **Distinct Attribute Count:** In the single-attribute setting, the distinct attribute count of a set $S \subseteq P$ is the number of different attributes that have at least one vector in S , i.e., the count is equal to $|\{\ell \in [c] \mid |S \cap D_\ell| > 0\}|$.

Note that the diversity metrics defined above are for the single-attribute setting. In the multi-attribute setting, in our experiments, we focus on settings where the attribute set $[c]$ is partitioned into m sets $\{C_i\}_{i=1}^m$ (i.e., $[c] = \sqcup_{i=1}^m C_i$) and every input vector $v \in P$ is associated with exactly one attribute from each C_i . In particular, $|\text{atb}(v)| = m$ and $|\text{atb}(v) \cap C_i| = 1$ for each $1 \leq i \leq m$. We call each C_i an attribute class. To measure diversity in the multi-attribute setting, we consider the aforementioned diversity metrics like entropy and inverse Simpson index restricted to an attribute class C_i . For instance, the entropy a set $S \subseteq P$ restricted to a particular C_i is given by $\sum_{\ell \in C_i: p_\ell > 0} -p_\ell \log(p_\ell)$, where $p_\ell = \frac{|S \cap D_\ell|}{|S|}$. Similarly, the inverse Simpson index of a set $S \subseteq P$ restricted to C_i is given by $\frac{1}{\sum_{\ell \in C_i} p_\ell^2}$.

2.4.2 Experimental Setup and Datasets

Hardware Details. All the experiments were performed in memory on an Intel(R) Xeon(R) Silver 4314 CPU (64 cores, 2.40GHz) with 128 GB RAM. We set the number of threads to 32.

Datasets. We report results on both semi-synthetic and real-world datasets consistent with prior works [3]. These are summarized in Table 2.1 and detailed below.

1. **Amazon Products Dataset (Amazon):** The dataset, also known as the Shopping Queries Image Dataset (SQID) [34], includes vector embeddings of about 190,000 product images and about 9,000 text queries by users. The embeddings of both product images and query texts are obtained via OpenAI’s CLIP model [74], which maps both images and texts to a shared vector space. Given this dataset, our task is to retrieve relevant and diverse product images for a given text query. SQID also contains metadata for every product image, such as product image url, product id, product description, product title, and product color. The dataset is publicly available on Hugging Face platform.¹

For this dataset, we choose the set of all possible product colors in the dataset as our set of attributes $[c]$. We noted that for a lot of products, the color of the product in its image did not match the product color in the associated metadata. Hence, to associate a clean attribute (color) to each vector (product) in the dataset, we use the associated metadata as follows: we assign to the vector the majority color among the colors listed in the product color, product description, and title of the product. In case of a tie, we assign a separate attribute (color) called ‘color_mix’. Further, we remove from consideration product images whose metadata does not contain any valid color names. The processed dataset contains 92,092 vector embeddings of product images and constitutes our set P . Note that the dataset exhibits a skewed color distribution, shown in Figure 2.1, with dominant colors such as black and white. No processing is applied to the query set which contains 8,956 vectors. The vector embeddings of both images and queries are $d = 768$ dimensional. We use $\sigma(u, v) = 1 + \frac{u^\top v}{\|u\| \cdot \|v\|}$ as the similarity function between two vectors u and v . Since the CLIP model was trained using the cosine similarity metric in the loss function (see [34], Section 4.2), this similarity function is a natural choice for the Amazon dataset.

2. **ArXiv OpenAI Embedding (ArXiv):** This dataset published by Cornell University consists of vector embeddings and metadata for approximately 250,000 machine learning papers from arXiv.org [95]. The vector embeddings of the papers were generated via OpenAI’s `text-embedding-ada-002` model, executed on each paper’s title, authors, year, and abstract. The dataset is publicly available on Kaggle [95].² Note that there are no user queries prespecified in this dataset.

We use this dataset to study both the single- and multi-attribute setting. For the single-attribute setting, we only consider the year in which a paper was last updated as its

¹<https://huggingface.co/datasets/crossingminds/shopping-queries-image-dataset>

²<https://www.kaggle.com/datasets/awester/arxiv-embeddings>

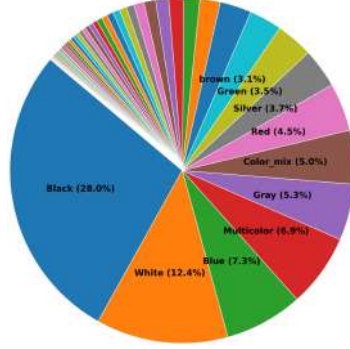


Figure 2.1: Distribution of product colors in the processed (cleaned) Amazon dataset.

attribute. For the multi-attribute setting, the paper’s update year and its arXiv category are the two associated attributes. Therefore, we have $m = 2$ attribute classes: update years and arXiv categories. Specifically, for the experiments we only consider papers with update-year between 2012 and 2025 and belonging to one or more of the following arXiv categories: `cs.ai`, `math.oc`, `cs.lg`, `cs.cv`, `stat.ml`, `cs.ro`, `cs.cl`, `cs.ne`, `cs.ir`, `cs.sy`, `cs.hc`, `cs.cr`, `cs.cy`, `cs.sd`, `eess.as`, and `eess.iv`.

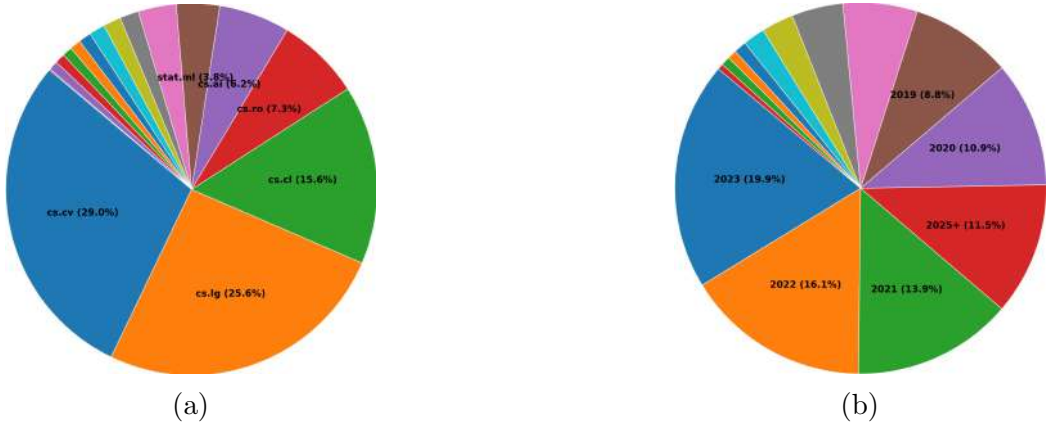


Figure 2.2: Distribution of (a) arXiv categories (b) last update year in the ArXiv dataset.

Since this dataset does not contain predefined queries, we randomly split the dataset in 4 : 1 ratio, and use the larger part as the set P and the smaller part as the queries. Here, by using the vector embeddings of papers themselves as queries, we aim to simulate the task of finding papers similar to a given query paper. The similarity function used for this dataset is the reciprocal of the Euclidean distance, i.e., $\sigma(u, v) = \frac{1}{\|u-v\|+\delta}$, for any two vectors u and v ; here, $\delta > 0$ is a small constant to avoid division by zero in case $\|u - v\| = 0$. Typically, we set $\delta = \eta$, where η is the smoothing parameter used in the definition of $\text{NSW}(\cdot)$. The distribution of the attributes across the vectors is shown in

Figure 2.2.

3. **SIFT Embeddings:** This is a standard benchmarking dataset for approximate nearest neighbor search in the Euclidean distance metric [92]. The dataset consists of SIFT vector embeddings in $d = 128$ dimensional space. In particular, here input set P contains 1,000,000 vectors and we have 10,000 vectors as queries. The embeddings are available at [92].¹ Note that this dataset does not contain any metadata that can be naturally used to assign attributes to the given vectors. Therefore, we utilize the following two methods for synthetically assigning attributes to the input vectors.

- **Clustering-based (Sift1m-(Clus)):** Since attributes such as colors often occupy distinct regions in the embedding space, we apply k-means clustering [57, 65] to identify 20 clusters. Each cluster is then assigned a unique index, and vectors in the same cluster are associated with the index of the cluster, which serves as our synthetic attribute. This simulates a single-attribute setting with $c = 20$.

To simulate the multi-attribute setting, we extend the above method of attribute generation. Given an input vector v of 128 dimensions, we split it into 4 vectors $\{v^i\}_{i=1}^4$, each of 32 dimensions. In particular, the vector v^1 consists of the first 32 components of v , the vector v^2 is obtained from the next 32 components of v , so on. Next, for each $i \in [4]$, we apply k-means clustering on $P^i := \{v^i : v \in P\}$ to identify 20 clusters. We write $C_i = \{c_1^i, \dots, c_{20}^i\}$ to denote these 20 clusters for P^i and if vector v^i belongs to a cluster c_j^i , then v^i is assigned the attribute c_j^i , i.e., we set $\text{atb}(v^i) = c_j^i$. Finally, for a vector $v \in P$, we assign $\text{atb}(v) = \{\text{atb}(v^1), \dots, \text{atb}(v^4)\}$. Note that this method of simulating the multi-attribute setting yields $m = 4$ attribute classes.

- **Probability distribution-based (Sift1m-(Prob)):** As in prior work [3], we also consider a setting wherein the input vectors have randomly assigned attributes. Specifically, we consider the single-attribute setting with $c = 20$. Here, we assign each vector $v \in P$ an attribute as follows: with probability 0.9, select an attribute from $\{1, 2, 3\}$ uniformly at random, otherwise, with the remaining 0.1 probability, select an attribute from $\{4, \dots, 20\}$ uniformly at random. This results in a skewed distribution of vectors among attributes that mimics real-world settings (e.g., market dominance by a few sellers).

¹<https://www.tensorflow.org/datasets/catalog/sift1m>

4. **Deep Descriptor Embeddings:** This is another standard dataset for nearest neighbor search [92].¹ The version of the dataset used in the current work contains 9,990,000 input vectors and 10,000 separate query vectors. Here, the vectors are 96 dimensional and the distances between them are evaluated using the cosine distance.

As in the case of SIFT dataset, input vectors in **Deep1b** do not have predefined attributes. Hence, we use the above-mentioned methods (based on clustering and randomization) to synthetically assign attributes to the vectors. This gives us the clustering-based **Deep1b-(Clus)** and probability distribution-based **Deep1b-(Prob)** versions of the dataset.

Choice of Parameter η : For our methods, we tune and set the smoothing parameter, η , to 0.01 for the **ArXiv**, **Sift1m-(Clus)** and **Sift1m-(Prob)** datasets, and set it to 0.0001 to analyze p -NNS. For other datasets, namely **Amazon**, **Deep1b-(Clus)** and **Deep1b-(Prob)**, we set η to 50.

2.4.3 Algorithms

Next, we describe the algorithms executed in the experiments.

1. **ANN:** This is the standard ANN algorithm that aims to maximize the similarity of the retrieved vectors to the given query without any diversity considerations. In our experiments, we use the graph based DiskANN method of [87] as the standard ANN algorithm. We instantiate DiskANN with candidate list size $L = 2000$ and the maximum graph degree as 128.² Here, we also set the pruning factor at 1.3, which is consistent with the existing recommendation in [3].
2. **Div-ANN:** This refers to the algorithm of [3] that solves the hard- constraint-based formulation for diversity in the single-attribute setting. Recall that [3] aims to maximize the similarity of the retrieved vectors to the given query subject to the constraint that no more than k' vectors in the retrieved set should have the same attribute. Note that the smaller the value of k' , the more diverse the retrieved set of vectors. Moreover, k' has to be provided as an input to this algorithm. In our experiments, we set different values k' , such as $k' \in \{1, 2, 5\}$ when $k = 10$, and $k' \in \{1, 2, 5, 10\}$ when $k = 50$.
3. **Nash-ANN and p -mean-ANN:** **Nash-ANN** refers to Algorithm 1 and **p -mean-ANN** refers to Algorithm 3 (stated in Appendix 2.6). Recall that Algorithm 1 and Algorithm 3 optimally

¹<https://www.tensorflow.org/datasets/catalog/deep1b>

²Both these choices are sufficiently larger than the standard values, $L = 200$ and maximum graph degree 64.

solve the NaNNS and the p -NNS problems, respectively, in the single-attribute setting given access to an exact nearest neighbor search oracle (Theorems 2.1 and 2.4). Further note that the p -mean welfare function, $M_p(\cdot)$, reduces to the Nash social welfare (geometric mean) when the exponent parameter $p \rightarrow 0^+$. For readability and at required places, we will write $p = 0$ to denote **Nash-ANN**. We conduct experiments with varying values of $p \in \{-10, -1, -0.5, 0, 0.5, 1\}$.

4. **Multi Nash-ANN and Multi Div-ANN**: In the multi-attribute setting, there are no prior methods to address diversity. Hence, for comparisons, we first fetch $L = 10000$ candidate vectors from P for each query q , using the standard ANN method, and then apply the following algorithms on the candidate vectors: (i) our algorithm for the NaNNS problem in the multi-attribute setting (Algorithm 2), which we term as **Multi Nash-ANN**, and (ii) an adaptation of the algorithm of [3], referred hereon as **Multi Div-ANN**, which greedily selects the most similar vectors to the query subject to the constraint that there are no more than k' vectors from each attribute.¹ We compare **Multi Nash-ANN** ($p = 0$) against **Multi Div-ANN** under different choices of k' .
5. **Multi p -mean-ANN**: In the multi-attribute setting, we also implement an analogue of **Multi Nash-ANN** that (in lieu of NSW) focuses on the p -mean welfare, M_p . The objective in these experiments is to understand the impact of varying the parameter p and the resulting tradeoff between relevance and diversity. Here, for each given query, we first fetch a set of $L = 10000$ candidate vectors using ANN. Then, we populate a set of k vectors by executing a marginal-gains greedy method over the L candidate vectors. In particular, we iterate k times, and in each iteration, select a new candidate vector that: (i) for $p \in (0, 1]$, yields the maximum increase in $M_p(\cdot)^p$, or (ii) for $p < 0$, leads to the maximum decrease in $M_p(\cdot)^p$.

2.4.4 Results: Balancing Relevance and Diversity

Single-attribute setting. We first compare, in the single-attribute setting, the performance of our algorithm, **Nash-ANN**, with **ANN** and **Div-ANN** (with different values of k'). The results for the **Amazon** and **Deep1b-(Clus)** datasets with $k = 50$ are shown in Figure 2.3 (columns one and two). Here, **ANN** finds the most relevant set of neighbors (approximation ratio close to 1), albeit with the lowest entropy (diversity). Moreover, as can be seen in the plots, the most

¹Note that one vector can have multiple attributes, hence contributing to the constraint of multiple attributes. Therefore, the issue of identifying an appropriate k' is exacerbated on moving to the multi-attribute setting.

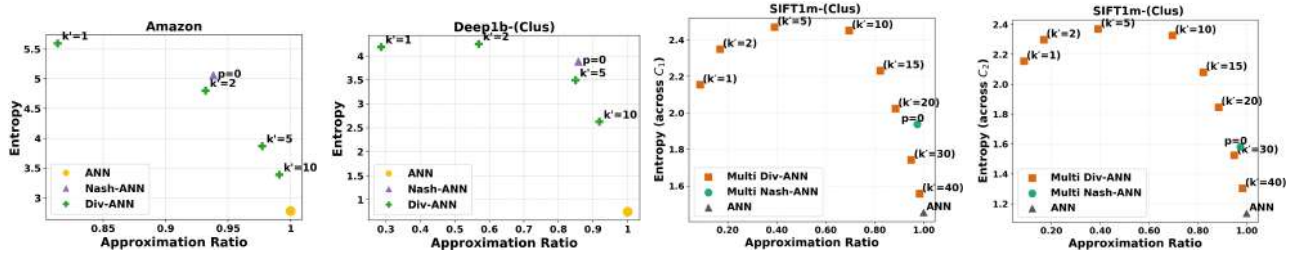


Figure 2.3: **Columns 1 and 2** - Comparison of approximation ratio versus entropy trade-offs between Nash-ANN, and Div-ANN with varying k' , for $k = 50$ on Amazon and Deep1b-(Clus) datasets in the single-attribute setting. **Columns 3 and 4** - Comparison of approximation ratio versus entropy trade-offs (across attribute classes C_1 and C_2) between Multi Nash-ANN, and Multi Div-ANN with varying k' on Sift1m-(Clus) dataset with $k = 50$ in the multi-attribute setting.

diverse (highest entropy) solution is obtained when we set, in Div-ANN, $k' = 1$; this restricts each $\ell \in [c]$ to contribute at most one vector in the output of Div-ANN. Also, note that one can increase the approximation ratio (i.e., increase relevance) of Div-ANN while incurring a loss in entropy (diversity), by increasing the value of the constraint parameter k' . However, selecting a ‘right’ value for k' is non-obvious, since this choice needs to be tailored to the dataset and, even within it, to queries (recall the “blue shirt” query in Figure 1.1).

By contrast, Nash-ANN does not require such ad hoc adjustments and, by design, finds a balance between relevance and diversity. Indeed, as can be seen in Figure 2.3 (columns 1 and 2), Nash-ANN maintains an approximation ratio close to 1 while achieving diversity similar to Div-ANN with $k' = 1$. Moreover, Nash-ANN Pareto dominates Div-ANN with $k' = 2$ for Amazon dataset and $k' = 5$ for Deep1b-(Clus) dataset on the fronts of approximation ratio and entropy. The results for other datasets and metrics follow similar trends and are given in Appendix 2.7.1.

Multi-attribute setting. In the multi-attribute setting, we report results for Multi Nash-ANN and Multi Div-ANN on the Sift1m-(Clus) dataset (Figure 2.3, columns 3 and 4) for $k = 50$ and $c = 80$. These eighty attributes are partitioned into four sets, $\{C_i\}_{i=1}^4$, with each set of size $|C_i| = 20$, i.e., $[c] = \cup_{i=1}^4 C_i$. Further, each input vector v is associated with four attributes ($|\text{atb}(v)| = 4$), one from each C_i ; see Section 2.4.2 for further details. Here, to quantify diversity, we separately consider for each $i \in [4]$, the entropy across attributes within a C_i . In Figure 2.3 (columns 3 and 4), we compare the approximation ratio versus entropy trade-offs of Multi Nash-ANN against Multi Div-ANN with varying k' . Here we show the results for attribute classes C_1 (column 1) and C_2 (column 2) whereas the results for C_3 and C_4 are given in Figure 2.29. We observe that Multi Nash-ANN maintains a high approximation ratio (relevance) while simultaneously achieving significantly higher entropy (higher diversity) than ANN.

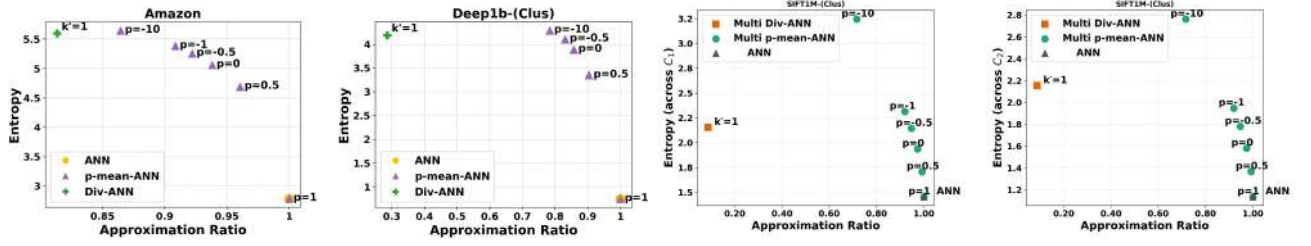


Figure 2.4: **Columns 1 and 2** - Approximation ratio versus entropy trade-offs for p -mean-ANN at various p values, for $k = 50$ on Amazon and Deep1b-(Clus) datasets in the single-attribute setting. **Columns 3 and 4** - Approximation ratio versus entropy trade-offs (across attribute classes C_1 and C_2) for Multi p -mean-ANN with varying p on Sift1m-(Clus) dataset with $k=50$ in the multi-attribute setting.

By contrast, in the constraint-based method Multi Div-ANN, low values of k' lead to a notable drop in the approximation ratio, whereas increasing k' reduces entropy. For example, for k' below 15, one obtains approximation ratio less than 0.8, and to reach an approximation ratio comparable to Multi Nash-ANN, one needs k' as high as 30. Additional results for the ArXiv dataset in the multi-attribute setting are provided in Appendix 2.7.2, and they exhibit trends similar to the ones in Figure 2.3. These findings demonstrate that Multi Nash-ANN achieves a balance between relevance and diversity. In summary

Across datasets, and in both single- and multi-attribute settings, the Nash formulation consistently improves entropy (diversity) over ANN, while maintaining an approximation ratio (relevance) of roughly above 0.9. By contrast, the hard-constrained formulation is highly sensitive to the choice of the constraint parameter k' , and in some cases, incurs a substantial drop in approximation ratio (even lower than 0.2).

Results for p -NNS. Recall that, selecting the exponent parameter $p \in (-\infty, 1]$ enables us to interpolate p -NNS between the standard NNS problem ($p = 1$), NaNNS ($p = 0$), and optimizing solely for diversity ($p \rightarrow -\infty$). We execute p -mean-ANN for $p \in \{-10, -1, -0.5, 0, 0.5, 1\}$ in both single- and multi-attribute settings and show that a trade-off between relevance (approximation ratio) and diversity (entropy) can be achieved by varying p .

For the single-attribute setting, Figure 2.4 (columns 1 and 2), and for the multi-attribute setting, Figure 2.4 (columns 3 and 4) capture this feature on Sift1m-(Clus) dataset with $k = 50$: For lower values of p , we have higher entropy but lower approximation ratio, while $p = 1$ matches ANN. For the multi-attribute setting, we show results for attribute classes C_1 (column 3) and C_2 (column 4) in Figure 2.4, whereas the results for C_3 and C_4 are shown in Figure 2.29. Note that in the multi-attribute setting, Multi p -mean-ANN with $p = -10$ Pareto

dominates **Multi Div-ANN** with $k' = 1$ in terms of approximation ratio and entropy. Moreover, analogous results are obtained for other datasets and metrics; see Appendix 2.7.1 and 2.7.2.

2.4.5 p -FetchUnion-ANN: A Faster Heuristic for the Single Attribute Setting

Further, we empirically study a faster heuristic algorithm for our NSW and p -mean welfare formulations. Specifically, the heuristic—called p -FetchUnion-ANN—first fetches a sufficiently large candidate set of vectors (irrespective of their attributes) using the ANN algorithm. Then, it applies the Nash (or p -mean) selection (similar to Line 5 in Algorithm 1 or Lines 6-8 in Algorithm 3) within this set. That is, instead of starting out with k neighbors for each $\ell \in [c]$ (as in Line 1 of Algorithm 1), the alternative here is to work with sufficiently many neighbors from the set $\cup_{\ell=1}^c D_\ell$.

Table 2.2 shows that this heuristic consistently achieves performance comparable to p -mean-ANN in terms of approximation ratio and diversity in the **Sift1m-(Clus)** dataset. Since p -FetchUnion-ANN retrieves a larger pool of vectors with high similarity, it achieves improved approximation ratio over p -Mean-ANN. However, it comes at the cost of reduced entropy, which can be explained by the fact that in restricting its search to an initially fetched large pool of vectors, p -FetchUnion-ANN may miss a more diverse solution that exists over the entire dataset. Another important aspect of p -FetchUnion-ANN is that, because it retrieves all neighbors from the union at once, the heuristic delivers substantially higher throughput (measured as queries answered per second, QPS) and therefore lower latency, which can be seen in Table 2.3 for the **Sift1m-(Clus)** dataset. In particular, p -FetchUnion-ANN serves almost $10\times$ more queries on **Sift1m-(Clus)** dataset than p -mean-ANN. The latency values exhibit a similar trend with reductions of similar magnitude. In summary, these observations position the heuristic as a notably fast method for NaNNS and p -NNS, particularly when c is large. We provide comparison of p -FetchUnion-ANN with p -mean-ANN on other datasets in Tables 2.4 to 2.9 in Appendix 2.7.3.

2.5 Proofs of Examples 2.1 and 2.2

This section provides the proofs for Examples 2.1 and 2.2. Recall that these stylized examples highlight how NaNNS dynamically recovers complete diversity or complete relevance based on the data.

Example 2.1 (Complete Diversity via NaNNS) *Consider an instance in which, for a given query $q \in \mathbb{R}^d$, all vectors in P are equally similar with the query: $\sigma(q, v) = 1$ for all $v \in P$. Also, let $|\text{atb}(v)| = 1$ for all $v \in P$, and write $S^* \in \arg\max_{S \subseteq P: |S|=k} \text{NSW}(S)$. If $c \geq k$, then*

Table 2.2: Comparison of performance across p values for **Sift1m-(Clus)** at $k = 50$.

Metric	Algorithm	$p = -10$	$p = -1$	$p = -0.5$	$p = 0$	$p = 0.5$	$p = 1$
Approx. Ratio	p -Mean-ANN	0.749±0.051	0.810±0.045	0.812±0.043	0.846±0.036	0.932±0.028	1.000±0.000
	p -FetchUnion-ANN	0.979±0.014	0.980±0.013	0.980±0.013	0.981±0.012	0.983±0.011	1.000±0.000
	ANN			1.000±0.000			
	Div-ANN ($k'=1$)			0.315±0.021			
Entropy	p -Mean-ANN	4.285±0.012	4.293±0.002	4.293±0.001	4.197±0.045	3.506±0.275	0.892±0.663
	p -FetchUnion-ANN	2.235±0.802	2.238±0.802	2.239±0.802	2.239±0.802	2.231±0.800	0.892±0.663
	ANN			0.892±0.663			
	Div-ANN ($k'=1$)			4.289±0.053			

Table 2.3: Comparison of performance on QPS and Latency across p on **Sift1m-(Clus)** dataset for $k = 50$.

Metric	Algorithm	$p = -10$	$p = -1$	$p = -0.5$	$p = 0$	$p = 0.5$	$p = 1$
Query per Second	p -Mean-ANN	120.86	115.78	107.01	135.98	122.59	122.59
	p -FetchUnion-ANN	1324.53	1324.62	1337.28	1442.03	1443.38	1327.03
Latency (μs)	p -Mean-ANN	264566.00	276129.00	298804.00	230318.00	235144.00	260800.00
	p -FetchUnion-ANN	24133.80	24134.00	23907.00	22170.20	22149.30	28990.40
99.9th percentile of Latency	p -Mean-ANN	484601.00	513036.00	478821.00	477925.00	482777.00	479132.00
	p -FetchUnion-ANN	52943.40	53474.70	54283.40	56128.70	53082.20	24088.70

here it holds that $|S^* \cap D_\ell| \leq 1$ for all $\ell \in [c]$.

Proof: Towards a contradiction, suppose there exists $T \in \arg\max_{S \subseteq P: |S|=k} \text{NSW}(S)$ such that $|T \cap D_{\ell^*}| > 1$ for some $\ell^* \in [c]$. Note that according to the setting specified in the example, $u_\ell(T) = |T \cap D_\ell| + \eta$ for all $\ell \in [c]$.

Since $c \geq k$ and $|T \cap D_{\ell^*}| > 1$, there exists $\ell' \in [c]$ such that $|T \cap D_{\ell'}| = 0$. Let $v^* \in T \cap D_{\ell^*}$ and $v' \in D_{\ell'}$ be two vectors. Consider the set $T' = (T \setminus \{v^*\}) \cup \{v'\}$. We have

$$\begin{aligned}
\frac{\text{NSW}(T')}{\text{NSW}(T)} &= \left(\frac{(u_{\ell'}(T') + \eta)}{(u_{\ell'}(T) + \eta)} \cdot \frac{(u_{\ell^*}(T') + \eta)}{(u_{\ell^*}(T) + \eta)} \prod_{\ell \in [c] \setminus \{\ell^*, \ell'\}} \frac{(u_\ell(T') + \eta)}{(u_\ell(T) + \eta)} \right)^{\frac{1}{c}} \\
&= \left(\frac{(1 + \eta)}{\eta} \cdot \frac{(u_{\ell^*}(T) - 1 + \eta)}{(u_{\ell^*}(T) + \eta)} \prod_{\ell \in [c] \setminus \{\ell^*, \ell'\}} \frac{(u_\ell(T) + \eta)}{(u_\ell(T) + \eta)} \right)^{\frac{1}{c}} \\
&= \left(\frac{(1 + \eta)}{\eta} \cdot \frac{(u_{\ell^*}(T) - 1 + \eta)}{(u_{\ell^*}(T) + \eta)} \right)^{\frac{1}{c}}
\end{aligned}$$

$$= \left(\frac{u_{\ell^*}(T) - 1 + \eta u_{\ell^*}(T) + \eta^2}{\eta u_{\ell^*}(T) + \eta^2} \right)^{\frac{1}{c}} > 1 \quad (u_{\ell^*}(T) \geq 2)$$

Therefore, we have $\text{NSW}(T') > \text{NSW}(T)$, which contradicts the optimality of T . Hence, we must have $|T \cap D_\ell| \leq 1$ for all $\ell \in [c]$, which proves the claim. $\square$

Example 2.2 (Complete Relevance via NaNNS) Consider an instance in which, for a given query $q \in \mathbb{R}^d$ and a particular $\ell^* \in [c]$, only vectors $v \in D_{\ell^*}$ have similarity $\sigma(q, v) = 1$ and all other vectors $v' \in P \setminus D_{\ell^*}$ have similarity $\sigma(q, v') = 0$. Also, suppose that $|\text{atb}(v)| = 1$ for each $v \in P$, along with $|D_{\ell^*}| \geq k$. Then, for a Nash optimal solution $S^* \in \arg\max_{S \subseteq P, |S|=k} \text{NSW}(S)$, it holds that $|S^* \cap D_{\ell^*}| = k$. That is, for all other $\ell \in [c] \setminus \{\ell^*\}$ we have $|S^* \cap D_\ell| = 0$.

Proof: Towards a contradiction, suppose there exists $T \in \arg\max_{S \subseteq P, |S|=k} \text{NSW}(S)$ such that $|T \cap D_{\ell^*}| < k$. Therefore, there exists $\ell' \in [c] \setminus \{\ell^*\}$ such that $|T \cap D_{\ell'}| \geq 1$. Let $v^* \in D_{\ell^*} \setminus T$ and let $v' \in T \cap D_{\ell'}$. Note that $u_{\ell'}(T) = 0$ since $\sigma(q, v) = 0$ for all $v \in D_\ell$ for any $\ell \in [c] \setminus \{\ell^*\}$. Moreover, we also have $u_{\ell^*}(T) = |T \cap D_{\ell^*}|$.

Consider the set $T' = (T \setminus \{v'\}) \cup \{v^*\}$. We have,

$$\begin{aligned} \frac{\text{NSW}(T')}{\text{NSW}(T)} &= \left(\frac{(u_{\ell'}(T') + \eta)}{(u_{\ell'}(T) + \eta)} \cdot \frac{(u_{\ell^*}(T') + \eta)}{(u_{\ell^*}(T) + \eta)} \prod_{\ell \in [c] \setminus \{\ell^*, \ell'\}} \frac{(u_\ell(T') + \eta)}{(u_\ell(T) + \eta)} \right)^{\frac{1}{c}} \\ &= \left(\frac{(u_{\ell'}(T) - \sigma(q, v') + \eta)}{(u_{\ell'}(T) + \eta)} \cdot \frac{(u_{\ell^*}(T) + \sigma(q, v^*) + \eta)}{(u_{\ell^*}(T) + \eta)} \prod_{\ell \in [c] \setminus \{\ell^*, \ell'\}} \frac{(u_\ell(T) + \eta)}{(u_\ell(T) + \eta)} \right)^{\frac{1}{c}} \\ &= \left(\frac{(0 - 0 + \eta)}{0 + \eta} \cdot \frac{(|T \cap D_{\ell^*}| + 1 + \eta)}{(|T \cap D_{\ell^*}| + \eta)} \right)^{\frac{1}{c}} \\ &= \left(1 + \frac{1}{|T \cap D_{\ell^*}| + \eta} \right)^{\frac{1}{c}} > 1. \end{aligned}$$

Therefore, we have obtained $\text{NSW}(T') > \text{NSW}(T)$, which contradicts the optimality of T . Hence, it must be the case that $|T \cap D_{\ell^*}| = k$, which proves the claim. $\square$

2.6 Extensions for p -NNS

This section extends our NaNNS results to p -mean nearest neighbor search (p -NNS) in the single-attribute setting. We state our algorithm (Algorithm 3) and present corresponding guarantees (Theorem 2.4 and Corollary 2.2) for finding an optimal solution for the p -NNS problem.

Recall that, for any $p \in (-\infty, 1]$, the p -mean welfare of c agents with utilities $w_1, \dots, w_c$ is given by $M_p(w_1, \dots, w_c) := \left(\frac{1}{c} \sum_{\ell=1}^c w_\ell^p \right)^{\frac{1}{p}}$.

Here, as in Section 2.1.1, the utility is defined as $u_\ell(S) = \sum_{v \in S \cap D_\ell} \sigma(q, v)$, for any subset of vectors S and attribute $\ell \in [c]$. Also, with a fixed smoothing constant $\eta > 0$, we will write $M_p(S) := M_p(u_1(S) + \eta, \dots, u_c(S) + \eta)$, and the p -NNS problem is defined as

$$\max_{S \subseteq P: |S|=k} M_p(S) \quad (2.12)$$

Throughout this section, we will write let $F_\ell(i) := \left(\sum_{j=1}^i \sigma(q, v_{(j)}^\ell) + \eta \right)^p$, for each attribute $\ell \in [c]$; here, as before, $v_{(j)}^\ell$ denotes the j^{th} most similar vector to q in $\hat{D}_\ell$, the set of k most similar vectors to q from D_ℓ .

Lemma 2.4 (Decreasing Marginals for $p > 0$) *Fix any $p \in (0, 1]$ and attribute $\ell \in [c]$. Then, for indices $1 \leq i' < i \leq k$, it holds that*

$$F_\ell(i') - F_\ell(i' - 1) \geq F_\ell(i) - F_\ell(i - 1) .$$

Proof: Write $G(j) := F_\ell(j) - F_\ell(j - 1)$ for all $j \in [k]$, and $f_\ell(i) = \sum_{j=1}^i \sigma(q, v_{(j)}^\ell)$ for $i \in [k]$. We will establish the lemma by showing that $G(j)$ is decreasing in j . Towards this, note that, for indices $j \geq 2$, we have

$$\begin{aligned} & G(j-1) - G(j) \\ &= F_\ell(j-1) - F_\ell(j-2) - F_\ell(j) + F_\ell(j-1) \\ &= 2F_\ell(j-1) - (F_\ell(j) + F_\ell(j-2)) \\ &= 2(f_\ell(j-1) + \eta)^p - ((f_\ell(j) + \eta)^p + (f_\ell(j-2) + \eta)^p) \\ &= 2(f_\ell(j-2) + \eta)^p \left(\left(1 + \frac{\sigma(q, v_{(j-1)}^\ell)}{f_\ell(j-2) + \eta} \right)^p - \frac{1}{2} \left(1 + \left(1 + \frac{\sigma(q, v_{(j-1)}^\ell) + \sigma(q, v_{(j)}^\ell)}{f_\ell(j-2) + \eta} \right)^p \right) \right) \\ &\geq 2(f_\ell(j-2) + \eta)^p \left(\left(1 + \frac{\sigma(q, v_{(j-1)}^\ell)}{f_\ell(j-2) + \eta} \right)^p - \frac{1}{2} \left(1 + \left(1 + \frac{2\sigma(q, v_{(j-1)}^\ell)}{f_\ell(j-2) + \eta} \right)^p \right) \right) \\ &\quad (\sigma(q, v_{(j-1)}^\ell) \geq \sigma(q, v_{(j)}^\ell); x \mapsto x^p \text{ is increasing for } p \in (0, 1] \text{ and } x \geq 0) \\ &\geq 2(f_\ell(j-2) + \eta)^p \left(\left(1 + \frac{\sigma(q, v_{(j-1)}^\ell)}{f_\ell(j-2) + \eta} \right)^p - \left(\frac{1}{2} \cdot 1 + \frac{1}{2} \cdot \left(1 + \frac{2\sigma(q, v_{(j-1)}^\ell)}{f_\ell(j-2) + \eta} \right)^p \right) \right) \\ &\quad (x \mapsto x^p \text{ is concave for } p \in (0, 1] \text{ and } x \geq 0) \\ &= 0. \end{aligned}$$

Therefore, $G(j) \leq G(j-1)$ for each $2 \leq j \leq k$. Equivalently, for indices $1 \leq i' < i \leq k$, it

holds that $G(i') \geq G(i)$. This completes the proof of the lemma. $\square$

Lemma 2.5 (Increasing Marginals for $p < 0$) *Fix any parameter $p \in (-\infty, 0)$ and attribute $\ell \in [c]$. Then, for indices $1 \leq i' < i \leq k$, it holds that*

$$F_\ell(i') - F_\ell(i' - 1) \leq F_\ell(i) - F_\ell(i - 1) .$$

Proof: The proof is similar to that of Lemma 2.4, except that we now seek the reverse inequality. In particular, write $G(j) := F_\ell(j) - F_\ell(j - 1)$ for all $j \in [k]$, and $f_\ell(i) = \sum_{j=1}^i \sigma(q, v_{(j)}^\ell)$ for $i \in [k]$.

To show that $G(j) \geq G(j - 1)$ for all indices $2 \leq j \leq k$, note that

$$\begin{aligned} & G(j - 1) - G(j) \\ &= 2(f_\ell(j - 2) + \eta)^p \left(\left(1 + \frac{\sigma(q, v_{(j-1)}^\ell)}{f_\ell(j - 2) + \eta} \right)^p - \frac{1}{2} \left(1 + \left(1 + \frac{\sigma(q, v_{(j-1)}^\ell) + \sigma(q, v_{(j)}^\ell)}{f_\ell(j - 2) + \eta} \right)^p \right) \right) \\ &\leq 2(f_\ell(j - 2) + \eta)^p \left(\left(1 + \frac{\sigma(q, v_{(j-1)}^\ell)}{f_\ell(j - 2) + \eta} \right)^p - \frac{1}{2} \left(1 + \left(1 + \frac{2\sigma(q, v_{(j-1)}^\ell)}{f_\ell(j - 2) + \eta} \right)^p \right) \right) \\ &\quad (\sigma(q, v_{(j-1)}^\ell) \geq \sigma(q, v_{(j)}^\ell); x \mapsto x^p \text{ is decreasing for } p \in (-\infty, 0) \text{ and } x \geq 0) \\ &\leq 2(f_\ell(j - 2) + \eta)^p \left(\left(1 + \frac{\sigma(q, v_{(j-1)}^\ell)}{f_\ell(j - 2) + \eta} \right)^p - \left(\frac{1}{2} \cdot 1 + \frac{1}{2} \cdot \left(1 + \frac{2\sigma(q, v_{(j-1)}^\ell)}{f_\ell(j - 2) + \eta} \right)^p \right) \right) \\ &\quad (x \mapsto x^p \text{ is convex for } p \in (-\infty, 0) \text{ and } x \geq 0) \\ &= 0 . \end{aligned}$$

Therefore, we have $G(j) \geq G(j - 1)$ for all $2 \leq j \leq k$. That is, $G(i') \leq G(i)$ for indices $1 \leq i' < i \leq k$. Hence, the lemma stands proved. $\square$

Lemma 2.6 *In the single-attribute setting, let ALG be the subset of vectors returned by Algorithm 3 and S be any subset of input vectors with the property that $|S \cap D_\ell| = |\text{ALG} \cap D_\ell|$, for each $\ell \in [c]$. Then, $M_p(\text{ALG}) \geq M_p(S)$.*

Proof: Assume, towards a contradiction, that there exists a subset of input vectors S that satisfies $|S \cap D_\ell| = |\text{ALG} \cap D_\ell|$, for each $\ell \in [c]$, and still induces p-mean welfare strictly greater than that of ALG . This strict inequality combined with the fact that $M_p(w_1, \dots, w_c)$ is an increasing function of w_i s implies that there exists an attribute $a \in [c]$ with the property that

Algorithm 3: p-Mean-ANN: Algorithm for p -NNS in the single-attribute setting

Input: Query $q \in \mathbb{R}^d$ and, for each attribute $\ell \in [c]$, the set of input vectors $D_\ell \subset \mathbb{R}^d$ and parameter $p \in (-\infty, 1] \setminus \{0\}$

- 1 For each $\ell \in [c]$, fetch the k (exact or approximate) nearest neighbors of $q \in \mathbb{R}^d$ from D_ℓ . Write $\widehat{D}_\ell \subseteq D_\ell$ to denote these sets.
- 2 For every $\ell \in [c]$ and each index $i \in [k]$, let $v_{(i)}^\ell$ denote the i th most similar vector to q in $\widehat{D}_\ell$.
- 3 Initialize subset $\text{ALG} = \emptyset$, along with count $k_\ell = 0$ and utility $w_\ell = 0$, for each $\ell \in [c]$.
- 4 **while** $|\text{ALG}| < k$ **do**
- 5 **if** $p \in (0, 1]$ **then**
- 6 $a \leftarrow \underset{\ell \in [c]}{\operatorname{argmax}} \left((w_\ell + \eta + \sigma(q, v_{(k_\ell+1)}^\ell))^p - (w_\ell + \eta)^p \right)$. $\triangleright$ Ties broken arbitrarily
- 7 **else if** $p < 0$ **then**
- 8 $a \leftarrow \underset{\ell \in [c]}{\operatorname{argmin}} \left((w_\ell + \eta + \sigma(q, v_{(k_\ell+1)}^\ell))^p - (w_\ell + \eta)^p \right)$. $\triangleright$ Ties broken arbitrarily
- 9 Update $\text{ALG} \leftarrow \text{ALG} \cup \{v_{(k_a+1)}^a\}$ along with $w_a \leftarrow w_a + \sigma(q, v_{(k_a+1)}^a)$ and $k_a \leftarrow k_a + 1$.
- 10 **return** ALG

the utility $u_a(S) > u_a(\text{ALG})$. That is,

$$\sum_{t \in S \cap D_a} \sigma(q, t) > \sum_{v \in \text{ALG} \cap D_a} \sigma(q, v) \quad (2.13)$$

On the other hand, note that the construction of Algorithm 3 and the definition of $\widehat{D}_a$ ensure that the vectors in $\text{ALG} \cap D_a$ are, in fact, the most similar to q among all the vectors in D_a . This observation and the fact that $|S \cap D_a| = |\text{ALG} \cap D_a|$ gives us $\sum_{v \in \text{ALG} \cap D_a} \sigma(q, v) \geq \sum_{t \in S \cap D_a} \sigma(q, t)$. This equation, however, contradicts the strict inequality (2.13). Therefore, by way of contradiction, we obtain that there does not exist a subset S such that $|S \cap D_\ell| = |\text{ALG} \cap D_\ell|$, for each $\ell \in [c]$, and $M_p(\text{ALG}) < M_p(S)$. The lemma stands proved. $\square$

Theorem 2.4 *In the single-attribute setting, given any query $q \in \mathbb{R}^d$ and an (exact) oracle **ENN** for k most similar vectors from any set, Algorithm 3 (**p-mean-ANN**) returns an optimal solution for p -NNS, i.e., it returns a size- k subset $\text{ALG} \subseteq P$ that satisfies $\text{ALG} \in \operatorname{argmax}_{S \subseteq P: |S|=k} M_p(S)$. Furthermore, the algorithm runs in time $O(kc) + \sum_{\ell=1}^c \text{ENN}(D_\ell, q)$, where $\text{ENN}(D_\ell, q)$ is the time required by the exact oracle to find k most similar vectors to q in D_ℓ .*

Proof: The running time of the algorithm follows via arguments similar to the ones used in the proof of Theorem 2.1.

For the correctness analysis, we divide the proof into two: $p \in (0, 1]$ and $p < 0$.

Case 1: $p \in (0, 1]$. Since $x \mapsto x^p$ is an increasing function for $x \geq 0$, the problem $\max_{S \subseteq P, |S|=k} M_p(S)$ is equivalent to $\max_{S \subseteq P, |S|=k} M_p(S)^p = \max_{S \subseteq P, |S|=k} \frac{1}{c} \sum_{\ell=1}^c (u_\ell(S) + \eta)^p$.

The proof here is similar to that of Theorem 2.1. Let $k_\ell = |\text{ALG} \cap D_\ell|$ for all $\ell \in [c]$. Further, let $\text{OPT} \in \arg\max_{S \subseteq P, |S|=k} \frac{1}{c} \sum_{\ell=1}^c (u_\ell(S) + \eta)^p$ and $k_\ell^* = |\text{OPT} \cap D_\ell|$ for all $\ell \in [c]$, where OPT is chosen such that $\sum_{\ell=1}^c |k_\ell^* - k_\ell|$ is minimized.

We will prove that OPT satisfies $k_\ell^* = k_\ell$ for each $\ell \in [c]$. This guarantee, along with Lemma 2.6, implies that, as desired, ALG is a p -mean welfare optimal solution.

Assume, towards a contradiction, that $k_\ell^* \neq k_\ell$ for some $\ell \in [c]$. Since $|\text{OPT}| = |\text{ALG}| = k$, there exist attributes $x, y \in [c]$ with the property that

$$k_x^* < k_x \quad \text{and} \quad k_y^* > k_y \quad (2.14)$$

Recall that via Lemma 2.4, we have for any pair of indices $1 \leq i' < i \leq k$,

$$F_\ell(i') - F_\ell(i' - 1) \geq F_\ell(i) - F_\ell(i - 1) \quad (2.15)$$

Next, note that for any attribute $\ell \in [c]$, if Algorithm 3, at any point during its execution, has included k'_ℓ vectors of attribute ℓ in ALG , then at that point the maintained utility $w_\ell = \sum_{j=1}^{k'_\ell} \sigma(q, v_{(j)}^\ell)$. Hence, at the beginning of any iteration of the algorithm, if the k'_ℓ denotes the number of selected vectors of each attribute $\ell \in [c]$, then the marginals considered in Line 6 are $F_\ell(k'_\ell + 1) - F_\ell(k'_\ell)$. These observations and the selection criterion in Line 6 of the algorithm give us the following inequality for the counts $k_x = |\text{ALG} \cap D_x|$ and $k_y = |\text{ALG} \cap D_y|$ of the returned solution ALG :

$$F_x(k_x) - F_x(k_x - 1) \geq F_y(k_y + 1) - F_y(k_y) \quad (2.16)$$

Specifically, equation (2.16) follows by considering the iteration in which k_x^{th} (last) vector of attribute x was selected by the algorithm. Before that iteration the algorithm had selected $(k_x - 1)$ vectors of attribute x , and let k'_y denote the number of vectors with attribute y that have been selected till that point. Note that $k'_y \leq k_y$. The fact that the k_x^{th} vector was (greedily) selected in Line 6, instead of including an additional vector of attribute y , gives $F_x(k_x) - F_x(k_x - 1) \geq F_y(k'_y + 1) - F_y(k'_y) \geq F_y(k_y + 1) - F_y(k_y)$; here, the last inequality follows

from equation (2.15). Hence, equation (2.16) holds.

Moreover,

$$\begin{aligned}
F_x(k_x^* + 1) - F_x(k_x^*) &\geq F_x(k_x) - F_x(k_x - 1) && \text{(via eqns. (2.14) and (2.15))} \\
&\geq F_y(k_y + 1) - F_y(k_y) && \text{(via eqn. (2.16))} \\
&\geq F_y(k_y^*) - F_y(k_y^* - 1) && (2.17)
\end{aligned}$$

The last inequality follows from equations (2.14) and (2.15).

Recall that $v_{(i)}^\ell$ denotes the i^{th} most similar (to q) vector in the set $\widehat{D}_\ell$. The definition of $\widehat{D}_\ell$ ensures that $v_{(i)}^\ell$ is, in fact, the i^{th} most similar (to q) vector among the ones that have attribute ℓ , i.e., i^{th} most similar in all of D_ℓ . Since OPT is an optimal solution, the $k_\ell^* = |\text{OPT} \cap D_\ell|$ vectors of attribute ℓ in OPT are the most similar k_ℓ^* vectors from D_ℓ . That is, $\text{OPT} \cap D_\ell = \{v_{(1)}^\ell, \dots, v_{(k_\ell^*)}^\ell\}$, for each $\ell \in [c]$. This observation and the definition of $F_\ell(\cdot)$ imply that the p -th power of OPT's p -mean welfare satisfies

$$M_p(\text{OPT})^p = \frac{1}{c} \sum_{\ell=1}^c F_\ell(k_\ell^*). \quad (2.18)$$

Now, consider a subset of vectors S obtained from OPT by including vector $v_{(k_x^*+1)}^x$ and removing $v_{(k_y^*)}^y$, i.e., $S = \left(\text{OPT} \cup \{v_{(k_x^*+1)}^x\}\right) \setminus \{v_{(k_y^*)}^y\}$. Note that

$$\begin{aligned}
M_p(S)^p - M_p(\text{OPT})^p &= \frac{1}{c} \left(F_x(k_x^* + 1) - F_x(k_x^*) \right) + \frac{1}{c} \left(F_y(k_y^* - 1) - F_y(k_y^*) \right) \\
&\geq 0 && \text{(via eqn. (2.17))}
\end{aligned}$$

Hence, $M_p(S) \geq M_p(\text{OPT})$. Given that OPT is a p -mean welfare optimal solution, the last inequality must hold with equality, $M_p(S) = M_p(\text{OPT})$, i.e., S is an optimal solution as well. This, however, contradicts the choice of OPT as an optimal solution that minimizes $\sum_{\ell=1}^c |k_\ell^* - k_\ell|$ – note that $\sum_{\ell=1}^c |\widehat{k}_\ell - k_\ell| < \sum_{\ell=1}^c |k_\ell^* - k_\ell|$, where $\widehat{k}_\ell := |S \cap D_\ell|$.

Therefore, by way of contradiction, we obtain that $|\text{OPT} \cap D_\ell| = |\text{ALG} \cap D_\ell|$ for each $\ell \in [c]$. As mentioned previously, this guarantee, along with Lemma 2.6, implies that ALG is a p -mean welfare optimal solution. This completes the proof of the theorem for the case $p \in (0, 1]$.

Case 2: $p < 0$. The arguments here are similar to the ones used in the previous case. However, since the exponent parameter p is negative, the key inequalities are reversed. For completeness, we present the proof below.

Note that, with the $p < 0$, the map $x \mapsto x^p$ is a decreasing function for $x \geq 0$. Hence, the optimization problem $\max_{S \subseteq P, |S|=k} M_p(S)$ is equivalent to

$$\min_{S \subseteq P, |S|=k} M_p(S)^p = \min_{S \subseteq P, |S|=k} \frac{1}{c} \sum_{\ell=1}^c (u_\ell(S) + \eta)^p .$$

Let $k_\ell = |\text{ALG} \cap D_\ell|$ for all $\ell \in [c]$. Further, let $\text{OPT} \in \operatorname{argmin}_{S \subseteq P, |S|=k} \frac{1}{c} \sum_{\ell=1}^c (u_\ell(S) + \eta)^p$ and $k_\ell^* = |\text{OPT} \cap D_\ell|$ for all $\ell \in [c]$, where OPT is chosen such that $\sum_{\ell=1}^c |k_\ell^* - k_\ell|$ is minimized.

We will prove that OPT satisfies $k_\ell^* = k_\ell$ for each $\ell \in [c]$. This guarantee, along with Lemma 2.6, implies that, as desired, ALG is a p-mean welfare optimal solution.

Assume, towards a contradiction, that $k_\ell^* \neq k_\ell$ for some $\ell \in [c]$. Since $|\text{OPT}| = |\text{ALG}| = k$, there exist attributes $x, y \in [c]$ with the property that

$$k_x^* < k_x \quad \text{and} \quad k_y^* > k_y \quad (2.19)$$

Recall that via Lemma 2.5, we have for any pair of indices $1 \leq i' < i \leq k$,

$$F_\ell(i') - F_\ell(i' - 1) \leq F_\ell(i) - F_\ell(i - 1) \quad (2.20)$$

Next, note that for any attribute $\ell \in [c]$, if Algorithm 3, at any point during its execution, has included k'_ℓ vectors of attribute ℓ in ALG , then at that point the maintained utility $w_\ell = \sum_{j=1}^{k'_\ell} \sigma(q, v_{(j)}^\ell)$. Hence, at the beginning of any iteration of the algorithm, if the k'_ℓ denotes the number of selected vectors of each attribute $\ell \in [c]$, then the marginals considered in Line 8 are $F_\ell(k'_\ell + 1) - F_\ell(k'_\ell)$. These observations and the selection criterion in Line 8 of the algorithm give us the following inequality for the counts $k_x = |\text{ALG} \cap D_x|$ and $k_y = |\text{ALG} \cap D_y|$ of the returned solution ALG :

$$F_x(k_x) - F_x(k_x - 1) \leq F_y(k_y + 1) - F_y(k_y) \quad (2.21)$$

Specifically, equation (2.21) follows by considering the iteration in which k_x^{th} (last) vector of attribute x was selected by the algorithm. Before that iteration the algorithm had selected $(k_x - 1)$ vectors of attribute x , and let k'_y denote the number of vectors with attribute y that have been selected till that point. Note that $k'_y \leq k_y$. The fact that the k_x^{th} vector was (greedily) selected in Line 8, instead of including an additional vector of attribute y , gives $F_x(k_x) - F_x(k_x - 1) \leq F_y(k'_y + 1) - F_y(k'_y) \leq F_y(k_y + 1) - F_y(k_y)$; here, the last inequality follows from equation (2.20). Hence, equation (2.21) holds.

Moreover,

$$\begin{aligned}
F_x(k_x^* + 1) - F_x(k_x^*) &\leq F_x(k_x) - F_x(k_x - 1) && \text{(via eqns. (2.19) and (2.20))} \\
&\leq F_y(k_y + 1) - F_y(k_y) && \text{(via eqn. (2.21))} \\
&\leq F_y(k_y^*) - F_y(k_y^* - 1) && (2.22)
\end{aligned}$$

The last inequality follows from equations (2.19) and (2.20).

Recall that $v_{(i)}^\ell$ denotes the i^{th} most similar (to q) vector in the set $\widehat{D}_\ell$. The definition of $\widehat{D}_\ell$ ensures that $v_{(i)}^\ell$ is, in fact, the i^{th} most similar (to q) vector among the ones that have attribute ℓ , i.e., i^{th} most similar in all of D_ℓ . Since OPT is an optimal solution, the $k_\ell^* = |\text{OPT} \cap D_\ell|$ vectors of attribute ℓ in OPT are the most similar k_ℓ^* vectors from D_ℓ . That is, $\text{OPT} \cap D_\ell = \{v_{(1)}^\ell, \dots, v_{(k_\ell^*)}^\ell\}$, for each $\ell \in [c]$. This observation and the definition of $F_\ell(\cdot)$ imply that the p -th power of OPT's p -mean welfare satisfies

$$M_p(\text{OPT})^p = \frac{1}{c} \sum_{\ell=1}^c F_\ell(k_\ell^*). \quad (2.23)$$

Now, consider a subset of vectors S obtained from OPT by including vector $v_{(k_x^*+1)}^x$ and removing $v_{(k_y^*)}^y$, i.e., $S = \left(\text{OPT} \cup \{v_{(k_x^*+1)}^x\}\right) \setminus \{v_{(k_y^*)}^y\}$. Note that

$$\begin{aligned}
M_p(S)^p - M_p(\text{OPT})^p &= \frac{1}{c} \left(F_x(k_x^* + 1) - F_x(k_x^*) \right) + \frac{1}{c} \left(F_y(k_y^* - 1) - F_y(k_y^*) \right) \\
&\leq 0 && \text{(via eqn. (2.22))}
\end{aligned}$$

Hence, $M_p(S) \geq M_p(\text{OPT})$. Given that OPT is a p -mean welfare optimal solution, the last inequality must hold with equality, $M_p(S) = M_p(\text{OPT})$, i.e., S is an optimal solution as well. This, however, contradicts the choice of OPT as an optimal solution that minimizes $\sum_{\ell=1}^c |k_\ell^* - k_\ell|$ – note that $\sum_{\ell=1}^c |\widehat{k}_\ell - k_\ell| < \sum_{\ell=1}^c |k_\ell^* - k_\ell|$, where $\widehat{k}_\ell := |S \cap D_\ell|$.

Therefore, by way of contradiction, we obtain that $|\text{OPT} \cap D_\ell| = |\text{ALG} \cap D_\ell|$ for each $\ell \in [c]$. As mentioned previously, this guarantee, along with Lemma 2.6, implies that ALG is a p -mean welfare optimal solution. This completes the proof of the theorem for the case $p < 0$.

Combining the two cases, we have that the theorem holds for all $p \in (-\infty, 1] \setminus \{0\}$. $\square$

Corollary 2.2 *In the single-attribute setting, given any query $q \in \mathbb{R}^d$ and an α -approximate oracle ANN for k most similar vectors from any set, Algorithm 3 (**p -mean-ANN**) returns an α -approximate solution for p -NNS, i.e., it returns a size- k subset $\text{ALG} \subseteq P$ with $M_p(\text{ALG}) \geq$*

$\alpha \max_{S \subseteq P: |S|=k} M_p(S)$. The algorithm runs in time $O(kc) + \sum_{\ell=1}^c \text{ANN}(D_\ell, q)$, with $\text{ANN}(D_\ell, q)$ being the time required by the approximate oracle to find k similar vectors to q in D_ℓ .

Proof: The running time of the algorithm follows via an argument similar to one used in the proof of Theorem 2.4. Hence, we only argue correctness here.

For every $\ell \in [c]$, let the α -approximate oracle return $\widehat{D}_\ell$. Recall that $v_{(i)}^\ell$, $i \in [k]$, denotes the i^{th} most similar point to q in the set $\widehat{D}_\ell$. Further, for every $\ell \in [c]$, let D_ℓ^* be the set of k most similar points to q within D_ℓ and, for each $i \in [k]$, define $v_{(i)}^{*\ell}$ to be the i^{th} most similar point to q in D_ℓ^* . Recall that by the guarantee of the α -approximate NNS oracle, we have $\sigma(q, v_{(i)}^\ell) \geq \alpha \sigma(q, v_{(i)}^{*\ell})$ for each $i \in [k]$. Let OPT be an optimal solution to the p -NNS problem. Note that, for each attribute $\ell \in [c]$, the optimal solution OPT contains in it the k_ℓ^* most similar vectors with attribute ℓ .

Finally, let $\widehat{\text{OPT}}$ be the optimal solution to the p -NNS problem when the set of vectors to search over is $P = \cup_{\ell \in [c]} \widehat{D}_\ell$.

By arguments similar to the ones used in the proof of Theorem 2.4, we have $M_p(\text{ALG}) = M_p(\widehat{\text{OPT}})$. Therefore

$$\begin{aligned}
M_p(\text{ALG}) &= M_p(\widehat{\text{OPT}}) \\
&\geq \left(\frac{1}{c} \sum_{\ell \in [c]} \left(\sum_{i=1}^{k_\ell^*} \sigma(q, v_{(i)}^\ell) + \eta \right)^p \right)^{\frac{1}{p}} \\
&\quad (\cup_{\ell \in [c]: k_\ell^* \geq 1} \{v_{(1)}^\ell, \dots, v_{(k_\ell^*)}^\ell\} \text{ is a feasible solution}) \\
&\geq \left(\frac{1}{c} \sum_{\ell \in [c]} \left(\sum_{i=1}^{k_\ell^*} \alpha \sigma(q, v_{(i)}^{*\ell}) + \eta \right)^p \right)^{\frac{1}{p}} \\
&\quad (\text{by } \alpha\text{-approximate guarantee of the oracle, and } M_p \text{ is increasing in its argument; } k_\ell^* \leq k) \\
&\geq \left(\frac{1}{c} \sum_{\ell \in [c]} \alpha^p \left(\sum_{i=1}^{k_\ell^*} \sigma(q, v_{(i)}^{*\ell}) + \eta \right)^p \right)^{\frac{1}{p}} \quad (\alpha \in (0, 1)) \\
&= \alpha M_p(\text{OPT}) \quad (\text{definition of OPT})
\end{aligned}$$

The corollary stands proved. $\square$

2.7 Further Experimental Evaluation and Analysis

In this section, we present additional experimental results to further validate the performance of **Nash-ANN** and compare it with existing methods. The details of the evaluation metrics and the description of the datasets used in our study are already presented in Section 2.4. Here, we report results for the single-attribute setting (Section 2.7.1), where we compare the approximation ratio alongside all diversity metrics for $k = 10$ and $k = 50$. We also include recall values for both $k = 10$ and $k = 50$ (Section 2.7.1.5). The key observation in all these plots is that the NSW objective effectively strikes a balance between relevance and diversity without having to specify any ad hoc constraints. Furthermore, we report experimental results for the multi-attribute setting on both a synthetic dataset (**Sift1m**) and a real-world dataset (**ArXiv**). Finally, we experimentally validate the performance-efficiency trade-offs of a faster heuristic variant of p -**mean-ANN** (detailed in Appendix 2.4.5) that can be used in addition to any existing (standard) ANN algorithm.

2.7.1 Balancing Relevance and Diversity: Single-attribute Setting

In this experiment, we evaluate how well p -**mean-ANN** (and the special case of $p = 0$, **Nash-ANN**) balances relevance and diversity in the single-attribute setting. We begin by examining the tradeoff between approximation ratio and entropy achieved by our algorithms on additional datasets beyond those used in the preceding sections. Moreover, we also report results for other diversity metrics such as the inverse Simpson index (Section 2.7.1.3) and the number of distinct attributes appearing in the k neighbors (Section 2.7.1.4) retrieved by our algorithms. These experiments corroborate the findings in the main paper, namely, **Nash-ANN** and p -**mean-ANN** are able to strike a balance between relevance and diversity whereas **ANN** only optimizes for relevance (hence low diversity) and **Div-ANN** only optimizes for diversity (hence low relevance).

2.7.1.1 Performance of Nash-ANN

We report the results for different datasets in Figures 2.5, 2.6, 2.7, and 2.8. On the **Sift1m-(Clus)** dataset (Figure 2.5), **Nash-ANN** achieves entropy close to that of the most diverse solution (**Div-ANN** with $k' = 1$) in both $k = 10$ and $k = 50$ cases. Moreover, **Nash-ANN** achieves significantly higher approximation ratio than **Div-ANN** in both $k = 10$ and $k = 50$ cases when $k' = 1$. For $k = 10$ case, **Nash-ANN** Pareto dominates **Div-ANN** even with the relaxed constraint of $k' = 5$ for $k = 10$. When the number of required neighbors is increased to $k = 50$, no other method Pareto dominates **Nash-ANN**. Similar observations hold for the **Sift1m-(Prob)** (Figure 2.6) and **Deep1b-(Prob)** (Figure 2.7) datasets. In the results on the **ArXiv** dataset (Figure 2.8) with $k = 10$, we observe that **Div-ANN** already achieves a high approximation

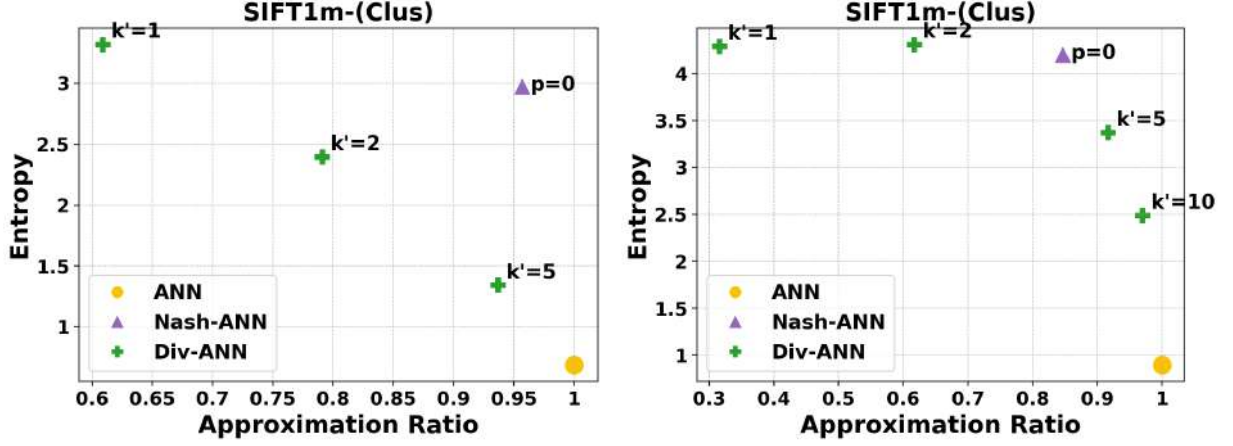


Figure 2.5: The plots show approximation ratio versus entropy trade-offs for various algorithms for $k = 10$ (Left) and $k = 50$ (Right) in single-attribute setting on Sift1m-(Clus) dataset.

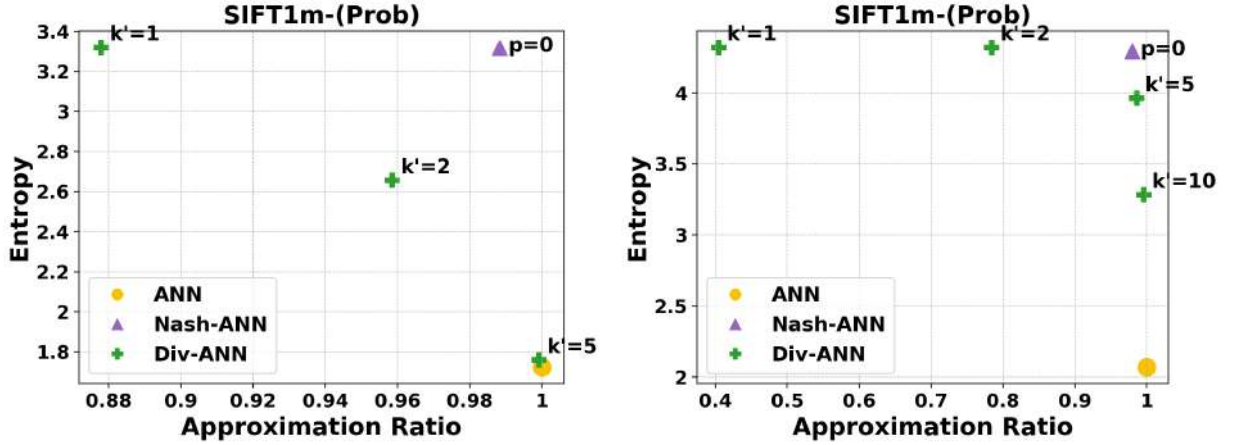


Figure 2.6: The plots show approximation ratio versus entropy trade-offs for various algorithms for $k = 10$ (Left) and $k = 50$ (Right) in single-attribute setting on Sift1m-(Prob) dataset.

ratio. However, Nash-ANN matches the entropy of Div-ANN with $k' = 1$ while improving on the approximation ratio. For $k = 50$, Nash-ANN nearly matches the entropy of Div-ANN with $k' = 1, 2$ whereas it significantly improves on the approximation ratio. In summary, the experimental results clearly demonstrate the ability of Nash-ANN to adapt to the varying nature of queries and consistently strike a balance between relevance and diversity.

2.7.1.2 Performance of p -mean-ANN

In this set of experiments, we study the effect on trade-off between approximation ratio and entropy when the parameter p in the p -NNS objective is varied over a range. Recall that the

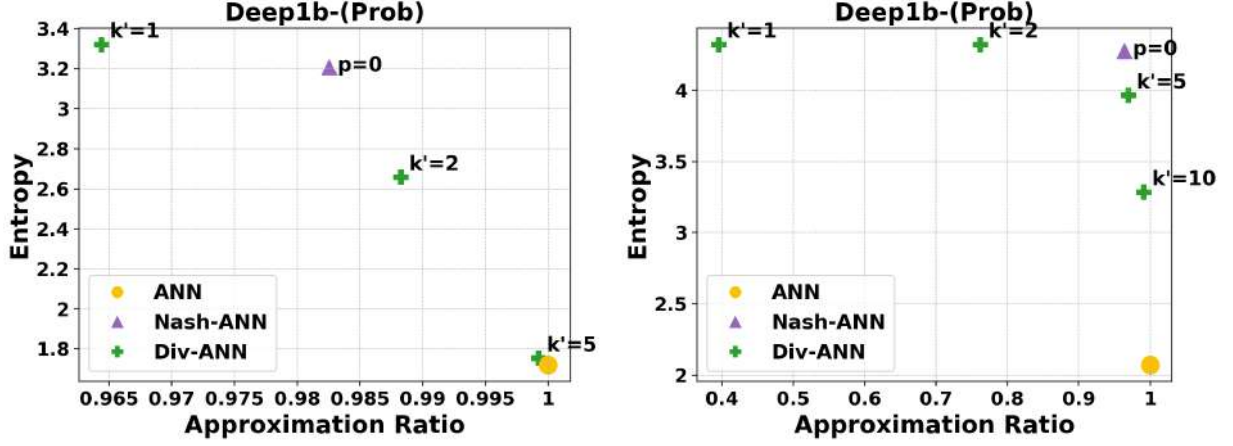


Figure 2.7: The plots show approximation ratio versus entropy trade-offs for various algorithms for $k = 10$ (Left) and $k = 50$ (Right) in single-attribute setting on Deep1b-(Prob) dataset.

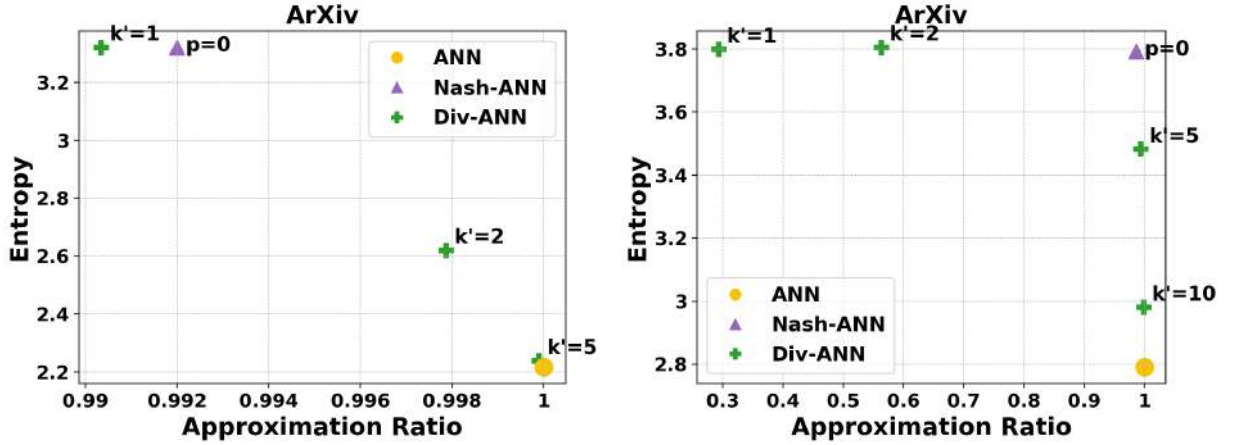


Figure 2.8: The plots show approximation ratio versus entropy trade-offs for various algorithms for $k = 10$ (Left) and $k = 50$ (Right) in single-attribute setting on ArXiv dataset.

p -NNS problem with $p \rightarrow 0$ corresponds to the NaNNS problem, and with $p = 1$, corresponds to the NNS problem. We experiment with values of $p \in \{-10, -1, -0.5, 0, 0.5, 1\}$ by running our algorithm p -mean-ANN (Algorithm 3) on the various datasets. The results are shown in Figures 2.9, 2.10, 2.11, and 2.12. We observe across all datasets for both $k = 10$ and $k = 50$ that, as p decreases from 1, the entropy increases but the approximation ratio decreases. This highlights the key intuition that as p decreases, the behavior changes from utilitarian welfare ($p = 1$ aligns exactly with ANN) to egalitarian welfare (more attribute-diverse). In other words, the parameter p allows us to smoothly interpolate between complete relevance (the standard NNS with $p = 1$) and complete diversity ($p \rightarrow -\infty$).

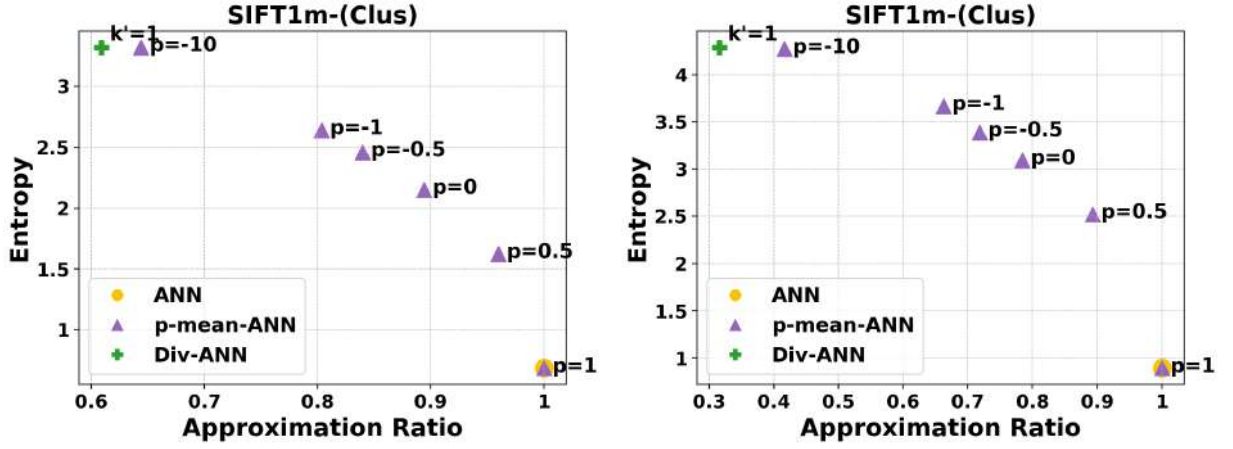


Figure 2.9: The plots report the approximation ratio versus entropy trade-off of p -mean-ANN, as p varies, for $k = 10$ (Left) and $k = 50$ (Right) on Sift1m-(Clus) dataset in the single-attribute setting.

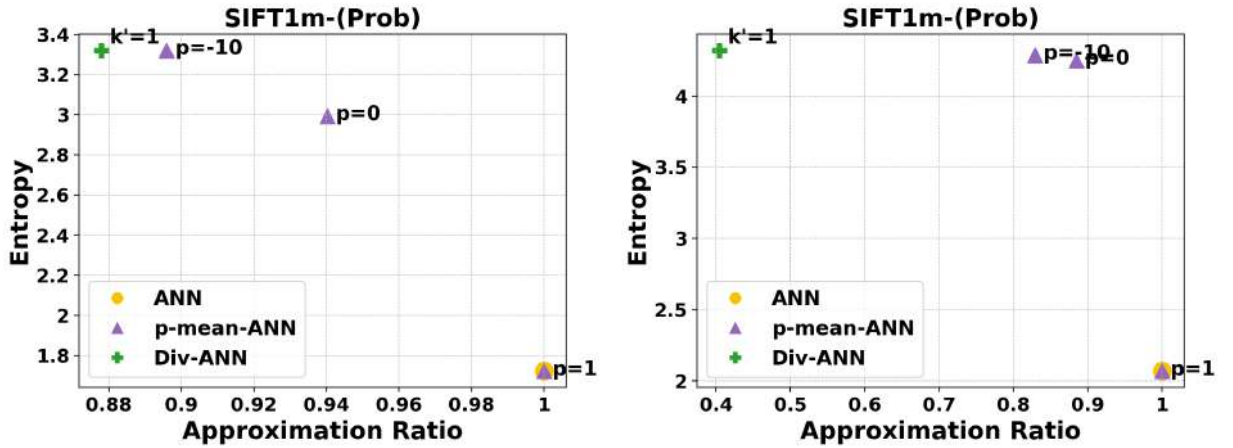


Figure 2.10: The plots report the approximation ratio versus entropy trade-off of p -mean-ANN, as p varies, for $k = 10$ (Left) and $k = 50$ (Right) on Sift1m-(Prob) dataset in single-attribute setting. We omit points corresponding to $p \in \{-1, -0.5, 0.5\}$ since they were extremely close to the points $p = -10$ or $p = 0$.

2.7.1.3 Approximation Ratio Versus Inverse Simpson Index

We also report results (Figures 2.13, 2.14, 2.15 and 2.16) on approximation ratio versus inverse Simpson index for all the aforementioned datasets, comparing Nash-ANN with Div-ANN with various choices of constraint parameter k' . The trends are similar to those for approximation ratio versus entropy.

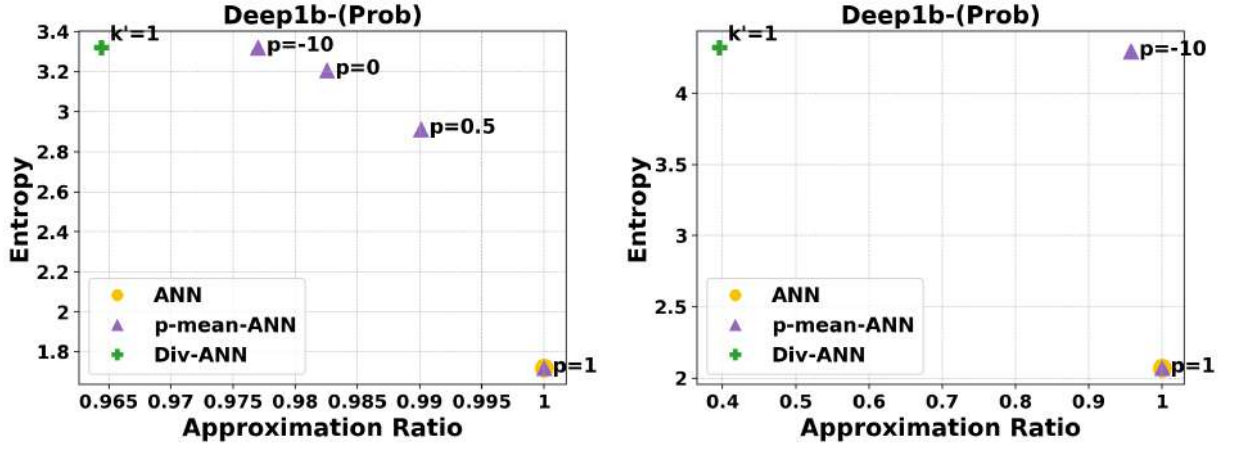


Figure 2.11: The plots report the approximation ratio versus entropy trade-off of p -mean-ANN, as p varies, for $k = 10$ (Left) and $k = 50$ (Right) on Deep1b-(Prob) dataset in single-attribute setting. For $k = 50$, we omit points corresponding to $p \in \{-1, -0.5, 0, 0.5\}$ since they were extremely close to the point $p = -10$. Due to the same reasons, we omit $p \in \{-1, -0.5\}$ for $k = 10$.

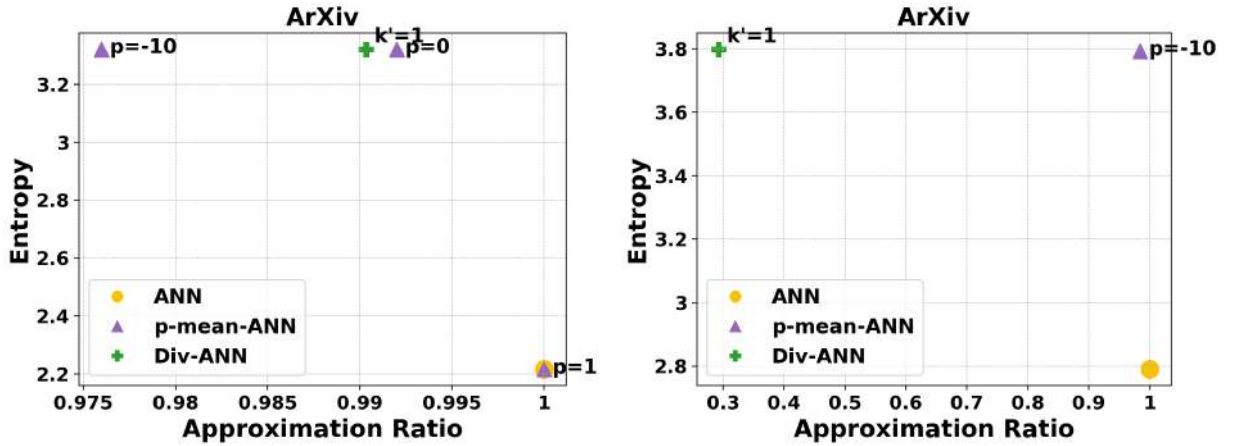


Figure 2.12: The plots report the approximation ratio versus entropy trade-off of p -mean-ANN, as p varies, for $k = 10$ (Left) and $k = 50$ (Right) on ArXiv dataset in single-attribute setting. For $k = 50$, we omit points corresponding to $p \in \{-1, -0.5, 0, 0.5\}$ since they were extremely close to the points $p = -10$. Due to similar reasons we omit $p \in \{-1, -0.5, 0.5\}$ for $k = 10$.

2.7.1.4 Approximation Ratio Versus Distinct Attribute Count

We also report the number of distinct attributes appearing in the set of vectors returned by different algorithms. Note that Div-ANN by design always returns a set where the number of distinct attributes is at least (k/k') . We plot approximation ratio versus number of distinct

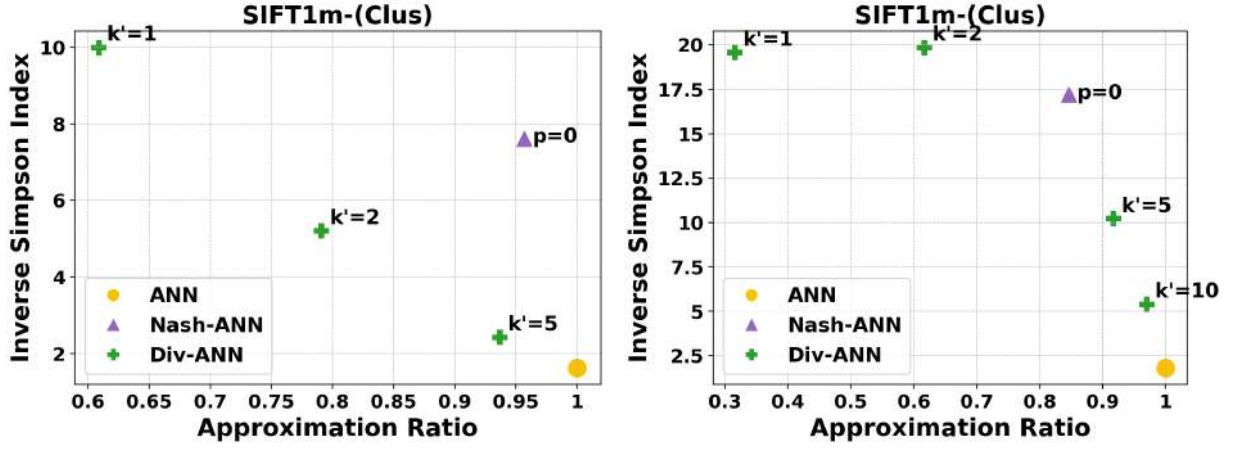


Figure 2.13: The plots show approximation ratio versus inverse Simpson index trade-offs for various algorithms for $k = 10$ (Left) and $k = 50$ (Right) in single-attribute setting on Sift1m-(Clus) dataset.

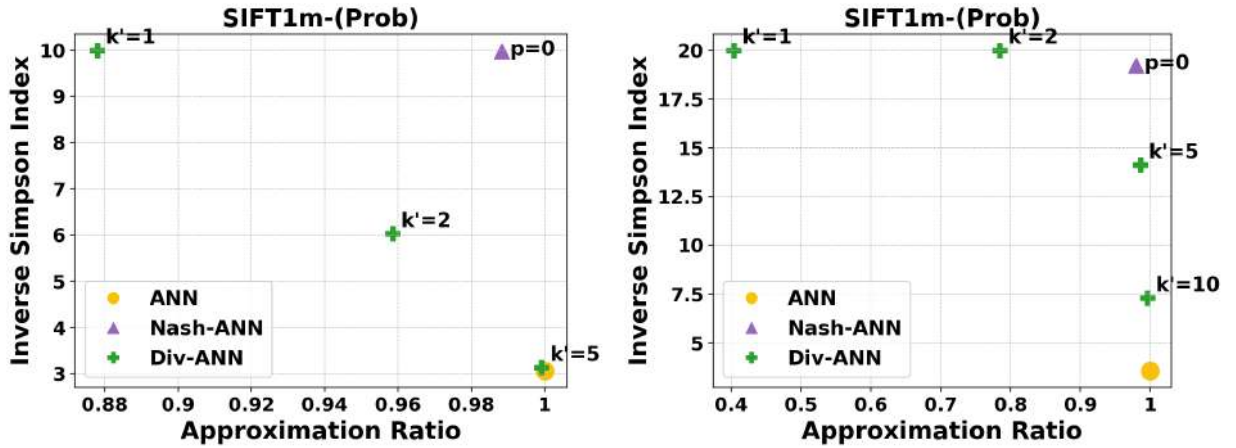


Figure 2.14: The plots show approximation ratio versus inverse Simpson index trade-offs for various algorithms for $k = 10$ (Left) and $k = 50$ (Right) in single-attribute setting on Sift1m-(Prob) dataset.

attributes and the results are shown in Figures 2.17, 2.18, 2.19, and 2.20. The results show that while Div-ANN with $k' = 1$ has high number of distinct attributes (by design), its approximation ratio is quite low. On the other hand, Nash-ANN has almost equal or slightly lower number of distinct attributes but achieves very high approximation ratio.

2.7.1.5 Recall Versus Entropy

We also report results for another popular relevance metric in the nearest neighbor search literature, namely, recall. The results for different datasets are shown in Figures 2.21, 2.22,

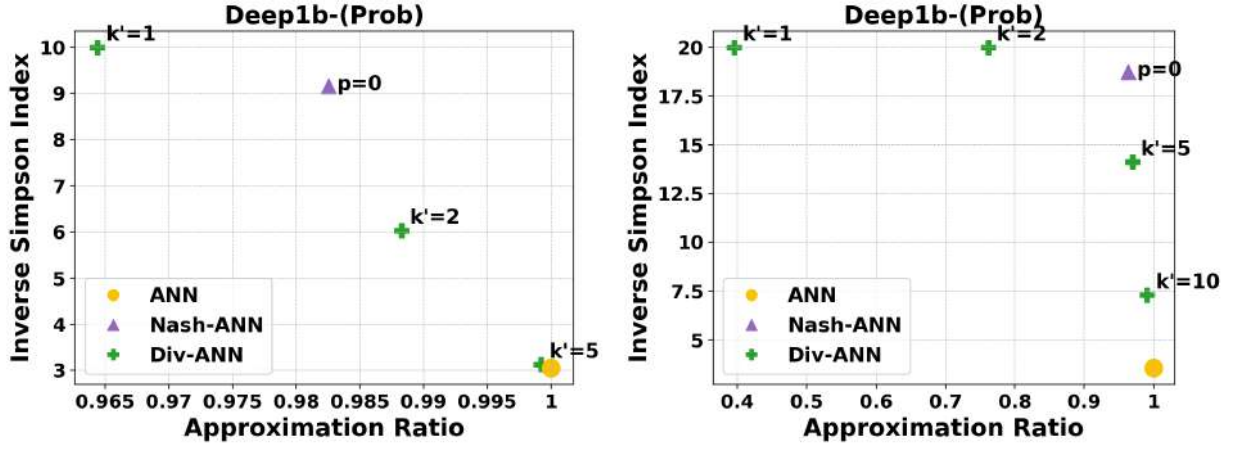


Figure 2.15: The plots show approximation ratio versus inverse Simpson index trade-offs for various algorithms for $k = 10$ (Left) and $k = 50$ (Right) in single-attribute setting on Deep1b-(Prob) dataset.

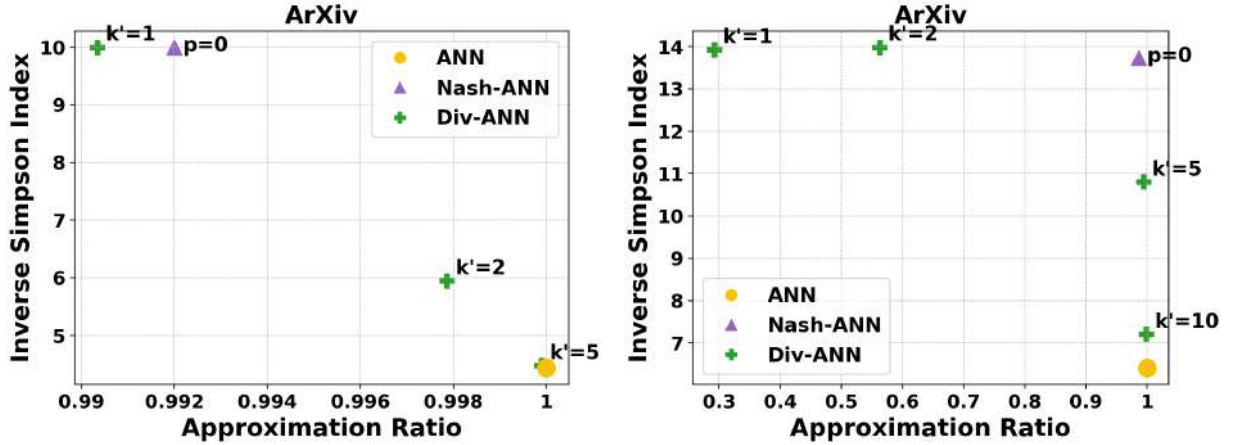


Figure 2.16: The plots show approximation ratio versus inverse Simpson index trade-offs for various algorithms for $k = 10$ (Left) and $k = 50$ (Right) in single-attribute setting on ArXiv dataset.

2.23, 2.24, 2.25, and 2.26.

Note that as discussed earlier (Section 2.4.1, Remark 3), recall can be a fragile metric when the goal is to balance between diversity and relevance. However, we still report recall to be consistent with prior literature and to demonstrate that Nash-ANN does not perform poorly. In fact, it is evident from the plots that Nash-ANN's recall value (relevance) surpasses that of Div-ANN with $k' = 1$ (most attribute diverse solution) while achieving a similar entropy. As already noted, the approximation ratio for Nash-ANN remains sufficiently high, indicating that

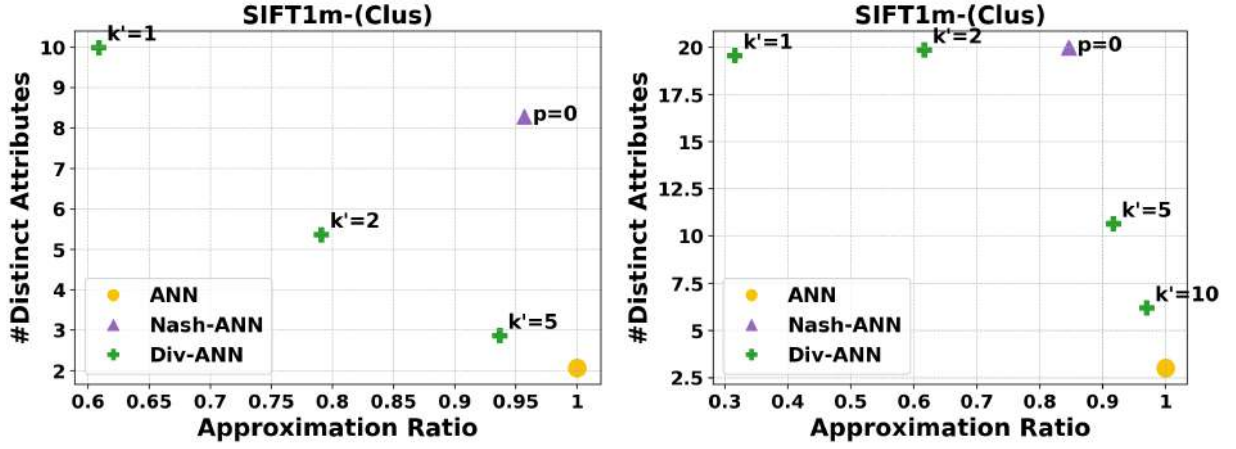


Figure 2.17: The plots show approximation ratio versus distinct counts trade-offs for various algorithms for $k = 10$ (Left) and $k = 50$ (Right) in single-attribute setting on Sift1m-(Clus) dataset

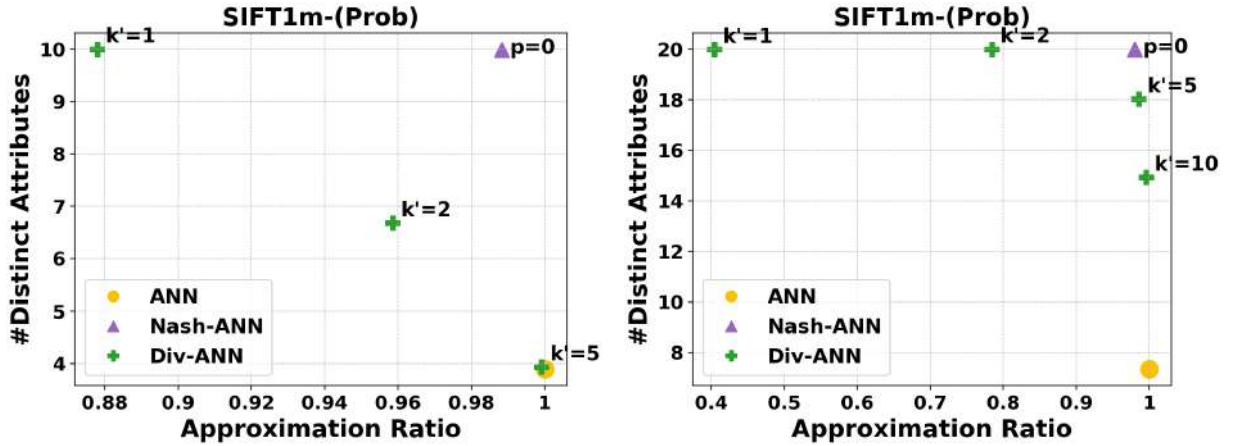


Figure 2.18: The plots show approximation ratio versus distinct counts trade-offs for various algorithms for $k = 10$ (Left) and $k = 50$ (Right) in single-attribute setting on Sift1m-(Prob) dataset.

the retrieved set of neighbors lies within a reasonably good neighborhood of the true nearest neighbors of a given query.

2.7.2 Balancing Relevance and Diversity: Multi-attribute Setting

Recall that our welfarist formulation seamlessly extends to the multi-attribute setting. In Section 2.4, we discussed the performance of Multi Nash-ANN and Multi Div-ANN on Sift1m-(Clus), where each input vector was associated with four attributes. In this section, we repeat the same set of experiments on one of the real-world datasets, namely ArXiv, which naturally contains

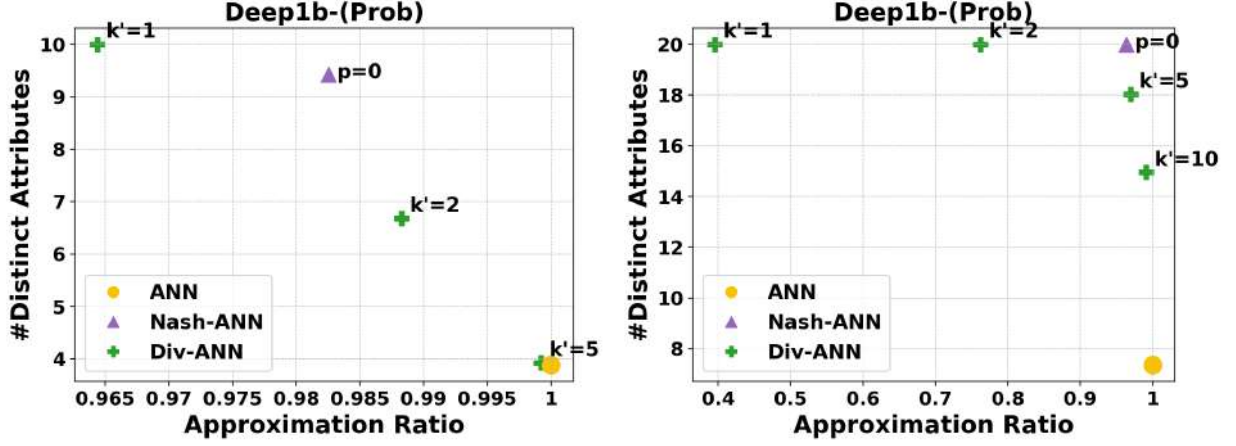


Figure 2.19: The plots show approximation ratio versus distinct counts trade-offs for various algorithms for $k = 10$ (Left) and $k = 50$ (Right) in single-attribute setting on Deep1b-(Prob) dataset.

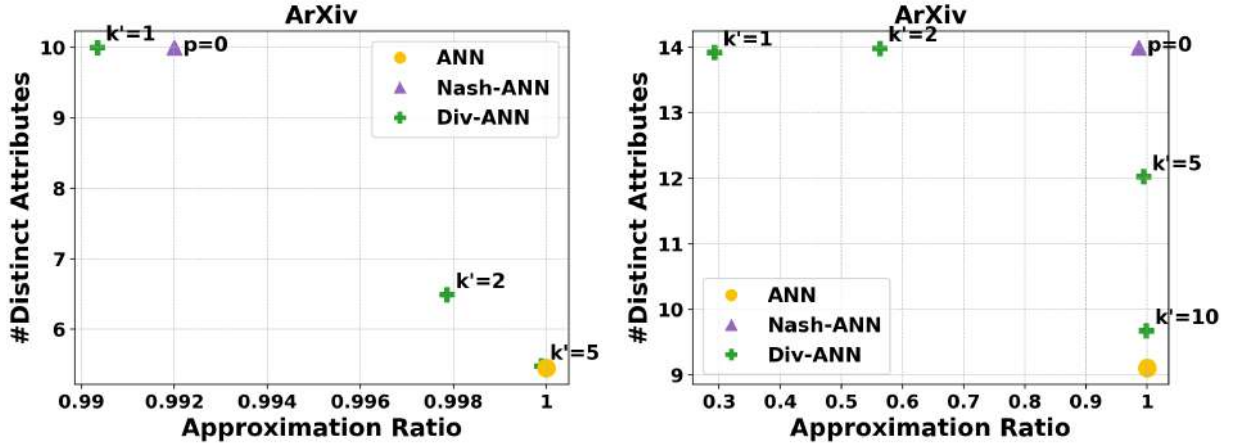


Figure 2.20: The plots show approximation ratio versus distinct counts trade-offs for various algorithms for $k = 10$ (Left) and $k = 50$ (Right) in single-attribute setting on ArXiv dataset.

two attribute classes ($m = 2$; see Section 2.4.1, Diversity Metrics): update year ($|C_1| = 14$) and paper category ($|C_2| = 16$). Therefore, $c = |C_1| + |C_2| = 30$. The results for $k = 50$ are presented in Figure 2.27. Note that, in each plot, we restrict the entropy to one of the attribute classes (C_1 or C_2) so that the diversity within a class can be understood from these plots. The results indicate that Multi Nash-ANN achieves an approximation ratio very close to one while maintaining entropy levels comparable to Multi Div-ANN with $k' = 1$ or 2 for both the attribute classes. In fact, Multi Nash-ANN Pareto dominates Multi Div-ANN with $k' = 5$.

We also study the effect of varying p in p -NNS problem in the multi-attribute setting. The

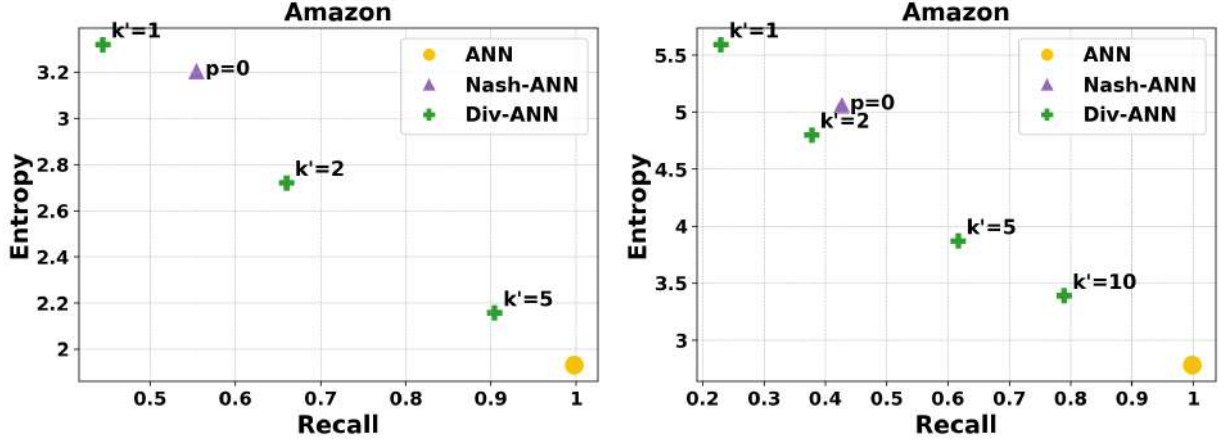


Figure 2.21: The plots show recall versus entropy trade-offs for $k = 10$ (Left) and $k = 50$ (Right) in the single-attribute setting on Amazon dataset.

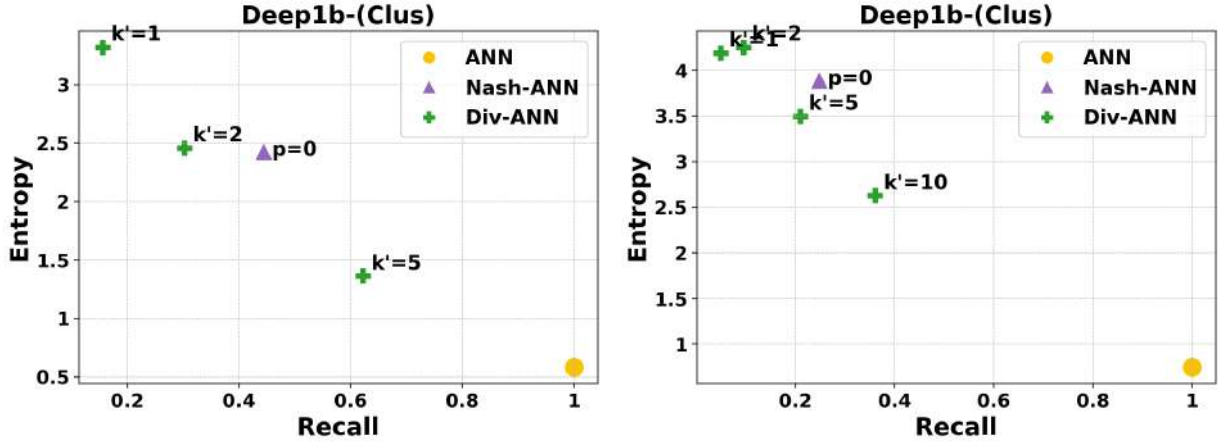


Figure 2.22: The plots show recall versus entropy trade-offs for $k = 10$ (Left) and $k = 50$ (Right) in the single-attribute setting on Deep1b-(Clus) dataset.

results for performance of Multi p -mean-ANN (an analogue of Multi Nash-ANN, detailed in Section 2.4.3) for $p \in \{-10, -1, -0.5, 0, 0.5, 1\}$ are shown in Figures 2.28 and 2.29. Interestingly, we observe that with decreasing p , the entropy (across C_1 or C_2) increases but the approximation ratio remains nearly the same and very close to 1. On the other hand, Multi Div-ANN with $k' = 1$ has very low approximation ratio. In fact, Multi p -mean-ANN with $p = -1$ and -10 Pareto dominates Multi Div-ANN with $k' = 1$.

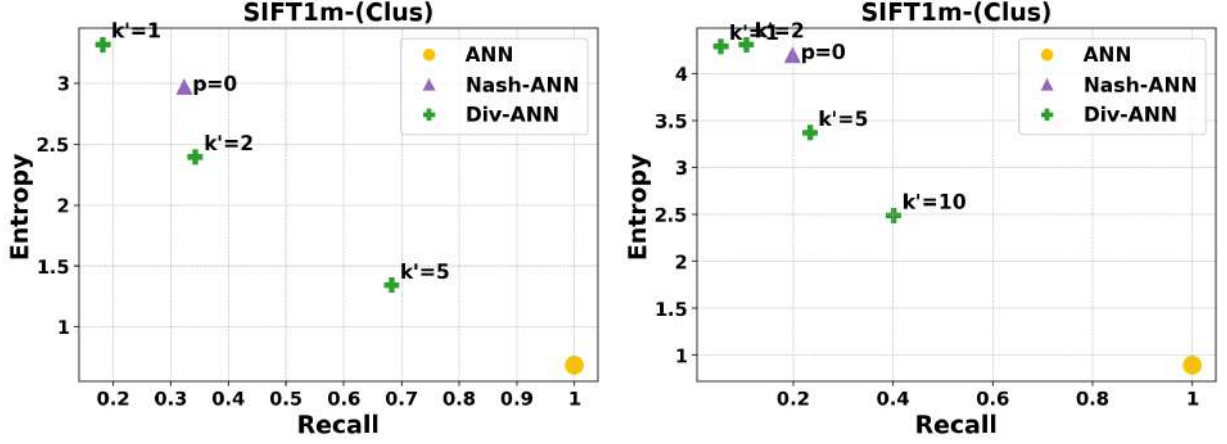


Figure 2.23: The plots show recall versus entropy trade-offs for $k = 10$ (Left) and $k = 50$ (Right) in the single-attribute setting on Sift1m-(Clus) dataset.

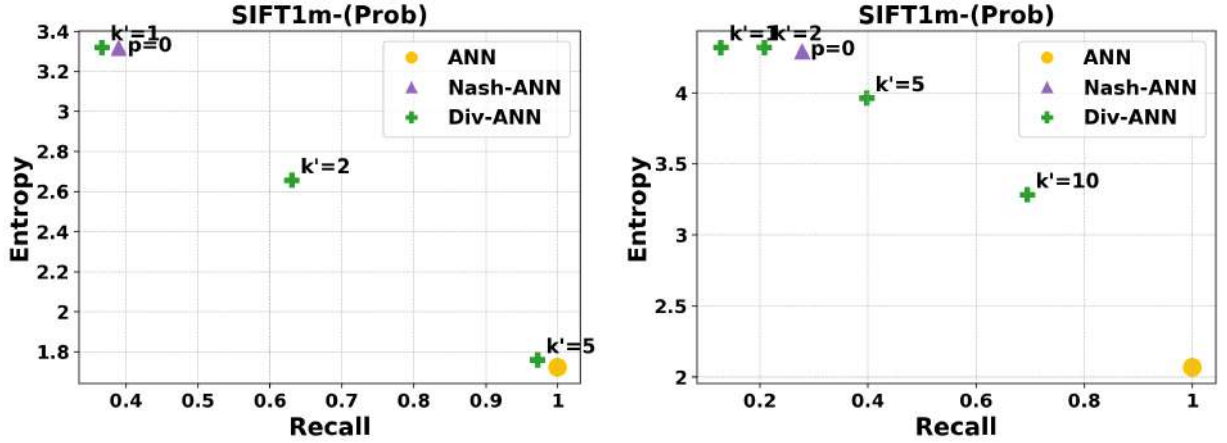


Figure 2.24: The plots show recall versus entropy trade-offs for $k = 10$ (Left) and $k = 50$ (Right) in the single-attribute setting on Sift1m-(Prob) dataset.

2.7.3 More Experiments for p -FetchUnion-ANN

In this section, we empirically study a faster heuristic algorithm for NSW and p -mean welfare formulations. Specifically, the heuristic—called p -FetchUnion-ANN—first fetches a sufficiently large candidate set of vectors (irrespective of their attributes) using the ANN algorithm. Then, it applies the Nash (or p -mean) selection (similar to Line 5 in Algorithm 1 or Lines 6–8 in Algorithm 3) within this set. That is, instead of starting out with k neighbors for each $\ell \in [c]$ (as in Line 1 of Algorithm 1), the alternative here is to work with sufficiently many neighbors from the set $\cup_{\ell=1}^c D_\ell$.

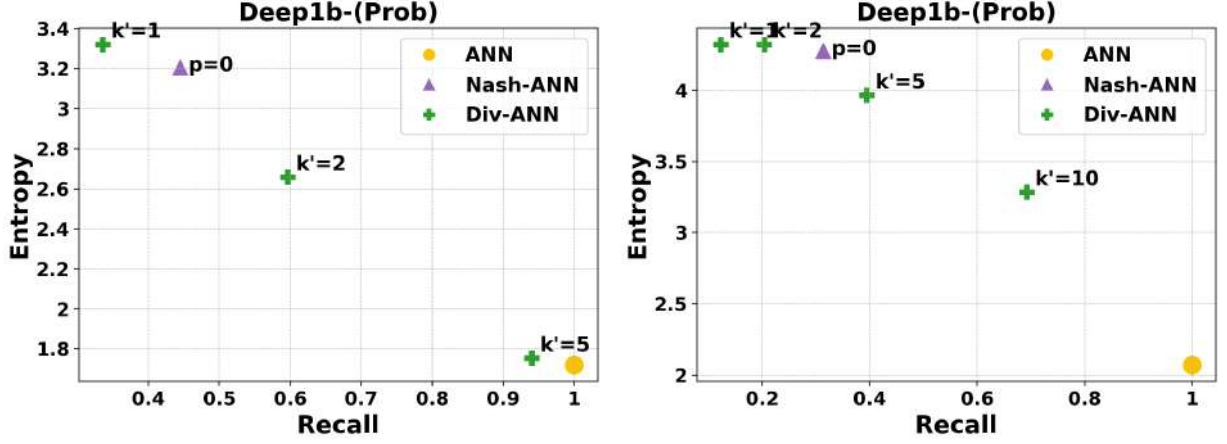


Figure 2.25: The plots show recall versus entropy trade-offs for $k = 10$ (Left) and $k = 50$ (Right) in the single-attribute setting on Deep1b-(Prob) dataset.

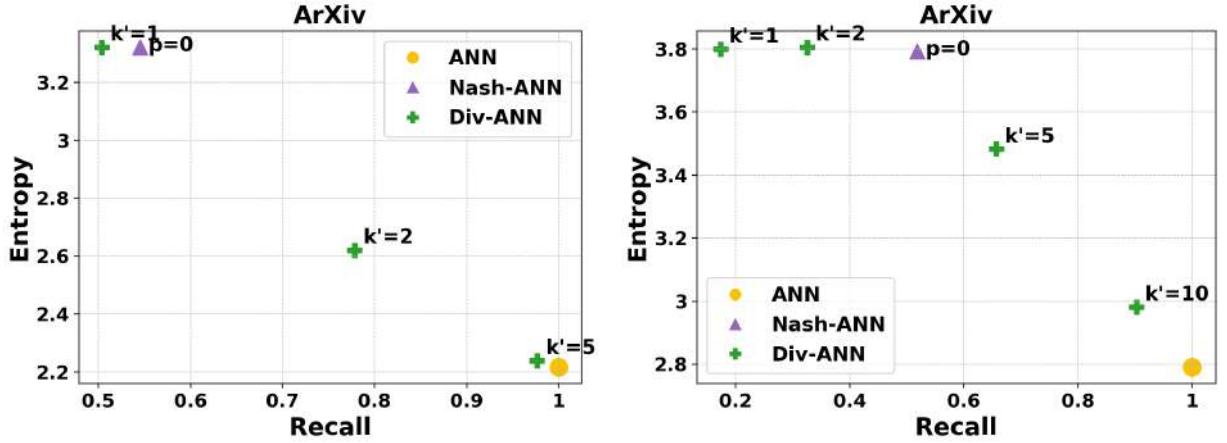


Figure 2.26: The plots show recall versus entropy trade-offs for $k = 10$ (Left) and $k = 50$ (Right) in the single-attribute setting on ArXiv dataset.

We empirically show (in Tables 2.4 to 2.8) that this heuristic consistently achieves performance comparable to p -Mean-ANN across nearly all datasets and evaluation metrics. Since p -FetchUnion-ANN starts with a large pool of vectors (with high similarity to the query) retrieved by the ANN algorithm without diversity considerations, it achieves improved approximation ratio over p -Mean-ANN. This trend is clearly evident in two datasets, namely Deep1b-(Clus) and Sift1m-(Clus). However, the improvement in approximation ratio comes at the cost of reduced entropy, which can be explained by the fact that in restricting its search to an initially fetched large pool of vectors, p -FetchUnion-ANN may miss out on a more diverse solution that exists over the entire dataset. Another important aspect of p -FetchUnion-ANN

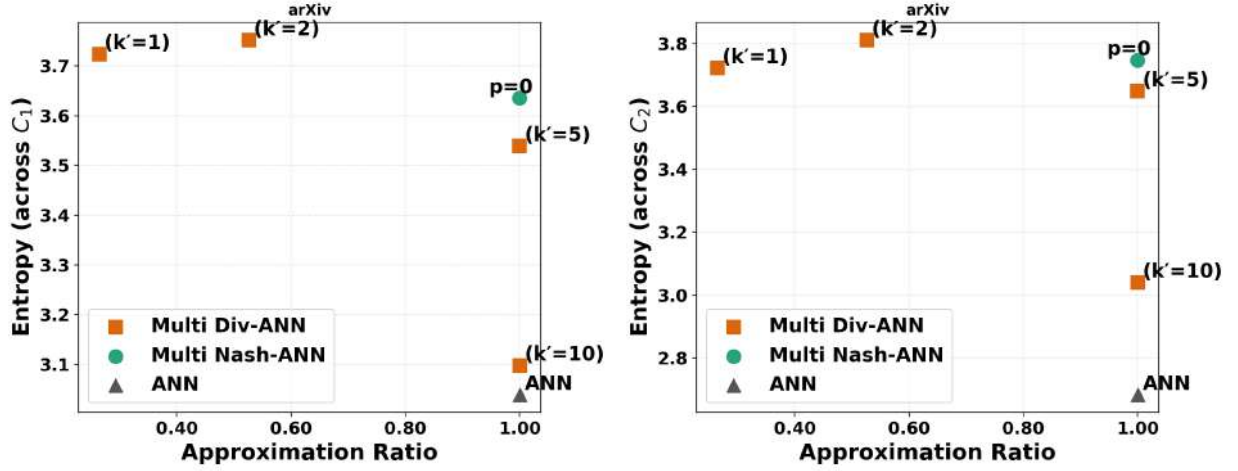


Figure 2.27: The plots show approximation ratio versus entropy trade-offs for various algorithms on ArXiv dataset across attribute classes C_1 (Left) and C_2 (Right) in the multi-attribute setting for $k = 50$.

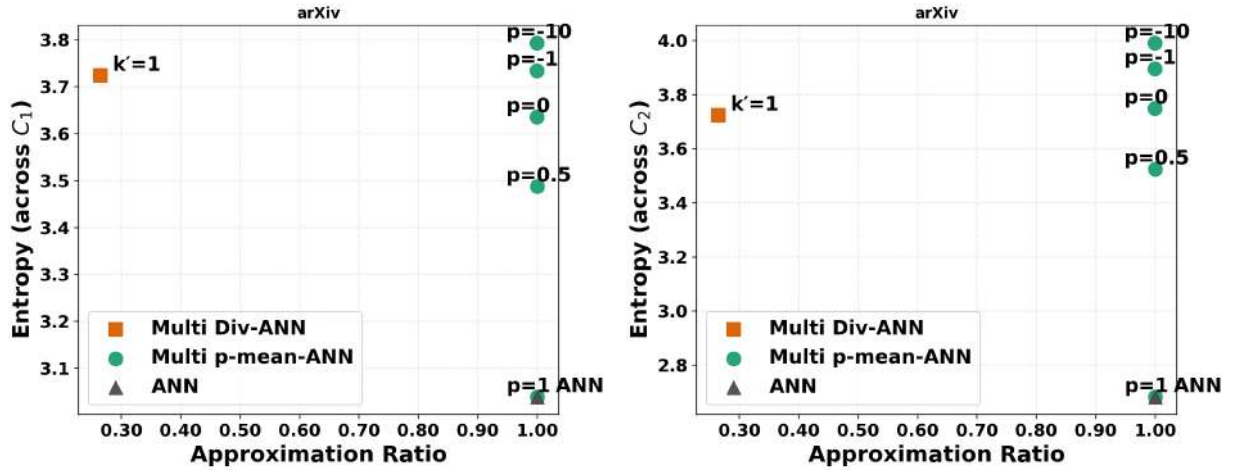


Figure 2.28: The plots show approximation ratio versus entropy trade-offs for p -mean-ANN, as p varies, for $k = 50$ across attribute classes C_1 (Left) and C_2 (Right) in the multi-attribute setting on ArXiv dataset.

is that, because it retrieves all neighbors from the union at once, the heuristic delivers substantially higher throughput (measured as queries answered per second, QPS) and therefore lower latency. The results validating these findings are reported in Tables 2.3 and 2.9 for the Sift1m-(Clus) and Amazon datasets, respectively. In particular, it serves almost $10\times$ more queries on Sift1m-(Clus) and $3\times$ more queries on Amazon dataset. The latency values exhibit a similar trend with reductions of similar magnitude. In summary, these observations position

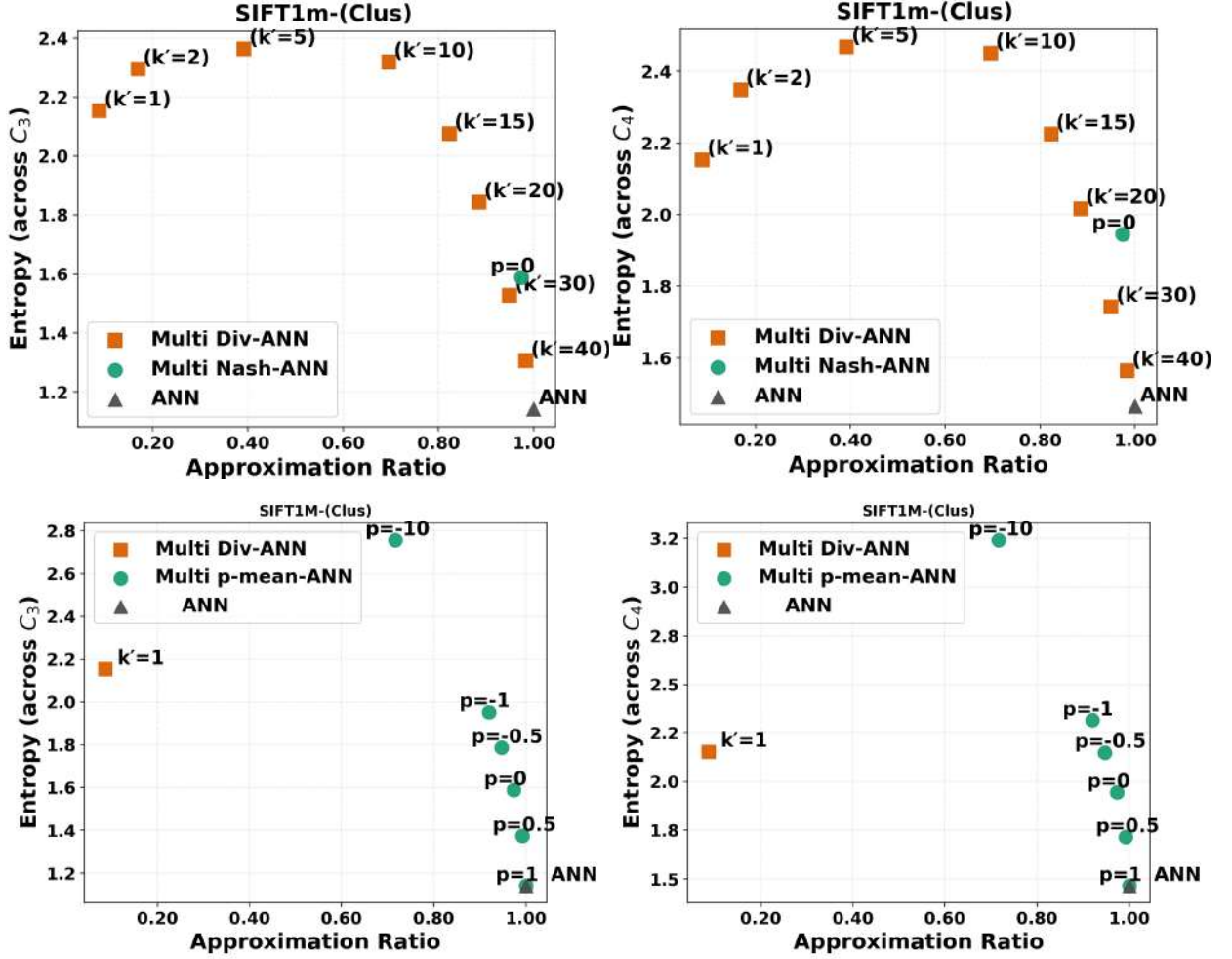


Figure 2.29: **Top Row:** The plots show approximation ratio versus entropy trade-offs for Nash-ANN against Div-ANN with varying values of k' for attribute class C_3 (left) and C_4 (right). **Bottom Row:** The plots show approximation ratio versus entropy trade-offs for p -mean-ANN, as p varies, across attribute classes C_3 (left) and C_4 (right). Both the rows correspond to $k = 50$ on Sift1m-(Clus) dataset in the multi-attribute setting.

the heuristic as a notably fast method for NaNNS and p -NNS, particularly when c is large.

2.8 Conclusion

In this chapter, we formulated diversity in neighbor search with a welfarist perspective, using Nash social welfare (NSW) and p -mean welfare as the underlying objectives. Our NSW formulation balances diversity and relevance in a query-dependent manner, satisfies several desirable axiomatic properties, and is naturally applicable in both single-attribute and multi-attribute settings. With these properties, our formulation overcomes key limitations of the prior hard-

Table 2.4: Comparison of relevance and diversity of p -mean-ANN, p -FetchUnion-ANN across different values of p against ANN and Div-ANN ($k' = 1$) for Amazon at $k = 50$.

Metric	Algorithm	$p = -10$	$p = -1$	$p = -0.5$	$p = 0$	$p = 0.5$	$p = 1$
Approx. Ratio	p -Mean-ANN	0.865 ± 0.045	0.909 ± 0.029	0.922 ± 0.027	0.938 ± 0.023	0.961 ± 0.018	1.000 ± 0.000
	p -FetchUnion-ANN	0.907 ± 0.033	0.912 ± 0.030	0.921 ± 0.027	0.935 ± 0.024	0.958 ± 0.019	1.000 ± 0.000
	ANN			1.000 ± 0.000			
	Div-ANN ($k' = 1$)			0.813 ± 0.053			
Entropy	p -Mean-ANN	5.644 ± 0.000	5.382 ± 0.135	5.252 ± 0.153	5.058 ± 0.178	4.687 ± 0.227	2.782 ± 0.684
	p -FetchUnion-ANN	5.364 ± 0.156	5.333 ± 0.149	5.261 ± 0.150	5.099 ± 0.171	4.736 ± 0.221	2.782 ± 0.684
	ANN			2.782 ± 0.684			
	Div-ANN ($k' = 1$)			5.594 ± 0.049			

Metric	Algorithm	$p = -10$	$p = -1$	$p = -0.5$	$p = 0$	$p = 0.5$	$p = 1$
Approx. Ratio	p -Mean-ANN	0.985 ± 0.010	0.985 ± 0.010	0.985 ± 0.010	0.986 ± 0.009	0.989 ± 0.008	1.000 ± 0.001
	p -FetchUnion-ANN	0.989 ± 0.007	0.989 ± 0.007	0.989 ± 0.007	0.990 ± 0.006	0.991 ± 0.006	1.000 ± 0.001
	ANN			1.000 ± 0.001			
	Div-ANN ($k' = 1$)			0.293 ± 0.007			
Entropy	p -Mean-ANN	3.793 ± 0.002	3.793 ± 0.002	3.793 ± 0.002	3.793 ± 0.002	3.793 ± 0.002	2.790 ± 0.510
	p -FetchUnion-ANN	3.704 ± 0.167	3.704 ± 0.166	3.704 ± 0.166	3.704 ± 0.166	3.704 ± 0.166	2.790 ± 0.510
	ANN			2.790 ± 0.510			
	Div-ANN ($k' = 1$)			3.799 ± 0.029			

Table 2.5: Comparison of relevance and diversity of p -mean-ANN, p -FetchUnion-ANN across different values of p against ANN and Div-ANN ($k' = 1$) for ArXiv at $k = 50$.

constrained approach [3]. Furthermore, the more general p -mean welfare interpolates between complete relevance ($p = 1$) and complete diversity ($p = -\infty$), offering practitioners a tunable parameter for real-world needs. Our formulations also admit provable and practical algorithms suited for low-latency scenarios. Experiments on real-world and semi-synthetic datasets validate their effectiveness in balancing diversity and relevance against existing baselines.

An important direction for future work is the design of sublinear-time approximation algorithms, in both single- and multi-attribute settings, that directly optimize our welfare objectives as part of ANN algorithms, thereby further improving efficiency. Another promising avenue is to extend welfare-based diversity objectives to settings without explicit attributes.

Metric	Algorithm	$p = -10$	$p = -1$	$p = -0.5$	$p = 0$	$p = 0.5$	$p = 1$
Approx. Ratio	p -Mean-ANN	0.784±0.071	0.815±0.065	0.831±0.063	0.858±0.060	0.904±0.049	1.000±0.000
	p -FetchUnion-ANN	0.958±0.033	0.961±0.030	0.962±0.029	0.963±0.028	0.968±0.024	1.000±0.000
	ANN			1.000±0.000			
	Div-ANN ($k'=1$)			0.286±0.041			
Entropy	p -Mean-ANN	4.293±0.000	4.200±0.052	4.105±0.091	3.887±0.155	3.349±0.267	0.746±0.717
	p -FetchUnion-ANN	2.101±1.214	2.101±1.214	2.099±1.212	2.095±1.207	2.068±1.179	0.746±0.717
	ANN			0.746±0.717			
	Div-ANN ($k'=1$)			4.191±0.234			

Table 2.6: Comparison of relevance and diversity of p -mean-ANN, p -FetchUnion-ANN across different values of p against ANN and Div-ANN ($k' = 1$) for Deep1b-(Clus) at $k = 50$.

Metric	Algorithm	$p = -10$	$p = -1$	$p = -0.5$	$p = 0$	$p = 0.5$	$p = 1$
Approx. Ratio	p -Mean-ANN	0.958±0.019	0.960±0.017	0.961±0.016	0.963±0.014	0.969±0.010	1.000±0.000
	p -FetchUnion-ANN	0.958±0.019	0.960±0.017	0.961±0.016	0.963±0.014	0.969±0.010	1.000±0.000
	ANN			1.000±0.000			
	Div-ANN ($k'=1$)			0.395±0.010			
Entropy	p -Mean-ANN	4.293±0.000	4.292±0.005	4.288±0.010	4.275±0.020	4.217±0.068	2.070±0.208
	p -FetchUnion-ANN	4.293±0.001	4.292±0.005	4.288±0.010	4.275±0.020	4.217±0.068	2.070±0.207
	ANN			2.070±0.207			
	Div-ANN ($k'=1$)			4.322±0.002			

Table 2.7: Comparison of relevance and diversity of p -mean-ANN, p -FetchUnion-ANN across different values of p against ANN and Div-ANN ($k' = 1$) for Deep1b-(Prob) at $k = 50$.

Metric	Algorithm	$p = -10$	$p = -1$	$p = -0.5$	$p = 0$	$p = 0.5$	$p = 1$
Approx. Ratio	p -Mean-ANN	0.975±0.010	0.977±0.008	0.979±0.008	0.980±0.008	0.982±0.006	1.000±0.000
	p -FetchUnion-ANN	0.975±0.010	0.977±0.008	0.979±0.008	0.980±0.008	0.982±0.006	1.000±0.000
	ANN			1.000±0.000			
	Div-ANN ($k'=1$)			0.404±0.004			
Entropy	p -Mean-ANN	4.292±0.006	4.292±0.003	4.293±0.002	4.293±0.002	4.269±0.020	2.068±0.205
	p -FetchUnion-ANN	4.292±0.006	4.292±0.003	4.293±0.002	4.293±0.003	4.269±0.020	2.068±0.205
	ANN			2.068±0.205			
	Div-ANN ($k'=1$)			4.322±0.005			

Table 2.8: Comparison of relevance and diversity of p -mean-ANN, p -FetchUnion-ANN across different values of p against ANN and Div-ANN ($k' = 1$) for Sift1m-(Prob) at $k = 50$.

Metric	Algorithm	$p = -10$	$p = -1$	$p = -0.5$	$p = 0$	$p = 0.5$	$p = 1$
Query per Second	p -Mean-ANN	198.08	195.97	199.08	179.03	171.22	189.31
	p -FetchUnion-ANN	620.27	610.62	551.02	608.76	572.57	591.76
Latency (μs)	p -Mean-ANN	161385.00	163121.00	160503.00	178555.00	186780.00	168856.00
	p -FetchUnion-ANN	51539.90	52362.30	58028.60	52521.60	55843.70	54030.80
99.9th percentile of Latency	p -Mean-ANN	433434.00	407151.00	418147.00	421725.00	475474.00	404477.00
	p -FetchUnion-ANN	146632.00	144989.00	145620.00	145657.00	143627.00	146464.00

Table 2.9: Comparison of Queries per second and Latency of p -mean-ANN, p -FetchUnion-ANN across different values of p against ANN and Div-ANN ($k' = 1$) for Amazon dataset for $k = 50$.

Chapter 3

Active Learning for Sample Efficient RLHF

3.1 Problem Setup

We have a set of contexts $\mathcal{X}$ and a set of possible actions per context $\mathcal{A}$. To learn using preference feedback, the agent selects $x \in \mathcal{X}$ and $a, a' \in \mathcal{A}$ to present to a human labeller, who then reveals a binary preference y that takes value 1 if a wins over a' and 0 otherwise. We assume that given (x, a, a') , y is sampled from the Bradley-Terry-Luce (BTL) preference model [17, 58] with r^* as the latent (unknown) reward function, i.e.,

$$\mathbb{P}[y=1|x, a, a'; r^*] = \frac{\exp(r^*(x, a))}{\exp(r^*(x, a)) + \exp(r^*(x, a'))} ,$$

The goal of the agent is to first learn r^* over T rounds of sequential interaction with the labeller, collecting dataset $\mathcal{D} = (x_s, a_s, a'_s, y_s)_{s=1}^T$, and then employ the learned reward to train a policy $\pi : \mathcal{X} \rightarrow \mathcal{A}$, which will eventually fetch high latent rewards $r^*(x, \pi(x))$.

In this work, we consider linear latent rewards $r^*(x, a) = \phi(x, a)^\top \theta^*$, where $\theta^* \in \mathbb{R}^d$ is the unknown reward parameter, and $\phi : \mathcal{X} \times \mathcal{A} \rightarrow \mathbb{R}^d$ is some known and fixed feature map. For instance, such a ϕ can be constructed by removing the last layer of a pre-trained language model, and in that case, θ^* corresponds to the weights of the last layer. With this model, for any $\theta \in \mathbb{R}^d$, one can equivalently write the probability of sampling $y_s = 1$ given (x_s, a_s, a'_s) as

$$\mathbb{P}[y_s=1|x_s, a_s, a'_s; \theta] = \sigma((\phi(x_s, a_s) - \phi(x_s, a'_s))^\top \theta) = \sigma(z_s^\top \theta) ,$$

where $\sigma(w) = \frac{1}{1+e^{-w}}$ is the sigmoid function and $z_s = \phi(x_s, a_s) - \phi(x_s, a'_s)$ is the feature difference

of actions a_s and a'_s for context x_s . With this, the latent reward parameter θ^* is typically estimated by minimizing the binary cross entropy loss (log-loss) [72], which is equivalent to *maximum likelihood estimation* (MLE). Specifically, At round t , the MLE of θ^* is computed as $\hat{\theta}_t = \operatorname{argmin}_{\theta \in \Theta} \mathcal{L}_t(\theta)$ using the preference dataset $\{(x_s, a_s, a'_s, y_s)\}_{s=1}^{t-1}$, where log-loss $\mathcal{L}_t(\theta)$ is given by

$$\mathcal{L}_t(\theta) = - \sum_{s=1}^{t-1} y_s \log(\sigma(z_s^\top \theta)) + (1 - y_s) \log(1 - \sigma(z_s^\top \theta)). \quad (3.1)$$

The above optimization problem is convex if we let the constraint set $\Theta \subset \mathbb{R}^d$ to be convex, and hence can be solved using standard algorithms [37].

Performance Measure. Our goal is to learn a policy over the collected data $\mathcal{D}$, which has high rewards or, equivalently, low sub-optimality. Formally, the sub-optimality gap of a policy π_T trained on the dataset $\mathcal{D}$ is defined as

$$R(T) = \max_{x \in \mathcal{X}} \max_{a \in \mathcal{A}} \{r^*(x, a) - r^*(x, \pi_T(x))\}. \quad (3.2)$$

Here, our policy π_T competes with the *Condorcet* winner for a given context – an action that fetches a higher reward than all other actions. The sub-optimality gap is the worst possible difference in rewards over the set of contexts. Prior work [67] competes against the *Borda* winner – an action that fetches a higher reward on average than another randomly chosen action – a weaker competitor (the *Condorcet* winner is also the *Borda* winner but not the other way around).

Remark 4 *To the best of our knowledge, common practical implementations of the RLHF pipeline use the following method: (i) remove the top layer of the LLM and convert it into an encoder, (ii) append a new linear layer on top of it, and (iii) output the logit score. Hence, the linear reward assumption is not restrictive in the sense that we only train the linear layer, keeping the encoder fixed. In practice, however, one needs to train the encoder, and hence, a more general function class needs to be considered. To this end, we generalize this setup to preference models with bounded class complexities, removing the need for explicit linear reward models (see Section 3.3 and Appendix 3.7 for details).*

3.2 Lower Bounds

We first illustrate the pitfall of a learner who samples contexts uniformly. We characterize such a learner in preference-based learning/RLHF and then show that such a learner can suffer a constant sub-optimality gap under budget constraints.

Definition 3.1 (Uniform Learner) Say an algorithm $\mathbf{Alg}$ samples T contexts uniformly at random from a set $\mathcal{X}$ and for each context x_t , picks two actions a_t, a'_t from a set $\mathcal{A}$. For each chosen triplet (x_t, a_t, a'_t) , $\mathbf{Alg}$ queries the preference model parameterized by θ^* and observes a stochastic preference $y_t \in \{0, 1\}$ between the actions. $\mathbf{Alg}$ then solves an MLE on this data to obtain $\hat{\theta}$, and learns a greedy policy with respect to $\hat{\theta}$. We call such an algorithm $\mathbf{Alg}$ a Uniform Learner.

Theorem 3.1 (Lower bound for uniform context sampling) There exists a problem instance $(\mathcal{X}, \mathcal{A}, \theta^*)$ for which the policy learnt by a Uniform Learner $\mathbf{Alg}$ under the budget $T \ll |\mathcal{X}|$ suffers $\Omega(1)$ sub-optimality gap with high probability.

Proof: We show the result for $d = 2$ for simplicity. The main idea is to divide the set of contexts into two groups—*good* and *bad*. Further, every context has only two actions, a and a' . The *good* group has a large number of contexts, so the uniform learner mostly samples contexts from this group. For all context x in the *good* group, the feature difference $\phi(x, a) - \phi(x, a') = z_g$ is the same. For *bad* contexts, this feature difference is z_b , such that $\langle z_b, z_g \rangle < 0$. Finally, θ^* is taken as the angle-bisector of z_g and z_b . From this construction (see Fig. 3.1), we have that a gets a higher reward for every context. Then, we show that the uniform learner only samples context-actions corresponding to z_g when the sample budget T is much smaller than the number of contexts. Under this scenario, the MLE estimate of the uniform learner correctly classifies the reward of a to be higher than that of a' for all the *good* contexts, but it wrongly classifies the rewards for the *bad* contexts. Thus, for *bad* contexts, the uniform learner suffers a constant suboptimality gap. We now present the details below.

Let the number of contexts be $|\mathcal{X}| = N$. Assume $T \ll N$. Otherwise, if the sample budget $T > N$, then one can just collect data for every context, and the setting becomes trivial. We divide $\mathcal{X}$ into two disjoint subsets: a *good* set $\mathcal{X}_g$ and a *bad* set $\mathcal{X}_b$. We assume w.l.o.g. that $|\mathcal{X}_b| = 1$, and we denote the *bad* context by b . Let the action set be $\mathcal{A} = \{a, a'\}$ for all contexts, and let a have a higher reward than a' . Let $\phi : \mathcal{X} \times \mathcal{A} \rightarrow \mathbb{R}^2$ be a feature map and $z_x = \phi(x, a) - \phi(x, a')$ be the feature difference vector at context x . Fix an $\alpha > 0$ and consider the problem instance:

$$\theta^* = \alpha \begin{bmatrix} \frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix}^\top, \quad z_b = \begin{bmatrix} -\frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix}^\top, \quad z_x = \begin{bmatrix} 1 & 0 \end{bmatrix}^\top, \quad \forall x \in \mathcal{X}_g.$$

Note that $\|\theta^*\|_2 = \alpha$. From this construction (see Fig. 3.1), it is clear that both for *good* and *bad* contexts, $z_x^\top \theta^* = \alpha/2 > 0$, which implies that indeed action a has higher reward than a' .

Let $\mathcal{E}_1$ be the event that all the T sampled contexts are *good* (i.e., from $\mathcal{X}_g$). Since, under

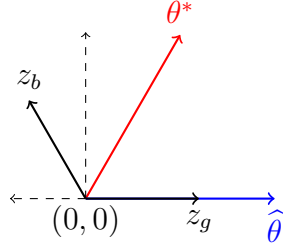


Figure 3.1: Visualization of the instance for Theorem 3.1. Here, z_g represents feature difference vectors for *good* contexts, and z_b represents feature difference for the *bad* context. θ^* and $\hat{\theta}$ are the true and learnt parameters respectively.

uniform sampling, for a random context X , $\mathbb{P}[X \in \mathcal{X}_g] = 1 - 1/N$, we have $\mathbb{P}[\mathcal{E}_1] = (1 - 1/N)^T$. Let $\mathcal{E}_2$ be the event that all observed preferences $y_1, \dots, y_T$ are equal to 1. Since, for a random preference Y given a context x , $\mathbb{P}[Y = 1|x] = \sigma(\alpha/2)$, we have $\mathbb{P}[\mathcal{E}_2|\mathcal{E}_1] = \sigma(\alpha/2)^T$.

Now, under the event $\mathcal{E}_1 \cap \mathcal{E}_2$, the MLE $\hat{\theta}$ constrained to the same norm as θ^* is given by

$$\hat{\theta} = \underset{\theta \in \mathbb{R}^2: \|\theta\|_2 \leq \alpha}{\operatorname{argmin}} \sum_{t=1}^T \log(1 + e^{-\alpha \theta_1}) = \underset{\theta \in \mathbb{R}^2: \|\theta\|_2 \leq \alpha}{\operatorname{argmin}} \log(1 + e^{-\alpha \theta_1}) .$$

It is easy to see that $\hat{\theta} = \begin{bmatrix} \alpha & 0 \end{bmatrix}^\top$. For any $x \in \mathcal{X}_g$, the predicted reward difference between actions a and a' is $z_x^\top \hat{\theta} = \alpha > 0$. Thus, $\hat{\theta}$ predicts the better action a correctly for all *good* contexts $\mathcal{X}_g$. However, for context b , the reward difference is $z_b^\top \hat{\theta} = -\frac{\alpha}{2} < 0$. Thus, $\hat{\theta}$ wrongly predicts a' as the better action for the *bad* context b . This yields a constant sub-optimality gap

$$R(T, b) = \phi(b, a)^\top \theta^* - \phi(b, a')^\top \theta^* = \alpha/2 = \Omega(1).$$

Finally, it remains to show that the event $\mathcal{E}_1 \cap \mathcal{E}_2$ happens with high probability. To this end, we choose $\alpha = 2 \log(N - 1)$ which yields $\sigma(\alpha/2) = 1 - 1/N$. This yields

$$\mathbb{P}[\mathcal{E}_1 \cap \mathcal{E}_2] = \mathbb{P}[\mathcal{E}_2|\mathcal{E}_1]\mathbb{P}[\mathcal{E}_1] = \left(1 - \frac{1}{N}\right)^{2T} \geq 1 - \frac{2T}{N} .$$

The last step uses $T \ll N$, which completes the proof. $\square$

Theorem 3.1 effectively shows that the uniform learner cannot make efficient use of the sampling budget because its performance may not increase with an increasing budget. Now, it is essential to characterize the limits of learning in this setting. To this end, we prove a lower bound on the sub-optimality gap of any algorithm with no restriction on how contexts and action pairs can be sampled. Note that standard regret lower bounds for dueling [78] and

logistic bandits [1] are not applicable here because we bound the sub-optimality gap and not regret.

Theorem 3.2 (Lower Bound for any sampling strategy) *Let $\mathcal{X}$ be a finite set of contexts, $\Theta = \{-\frac{1}{\sqrt{T}}, \frac{1}{\sqrt{T}}\}^d$, $\mathcal{A} = \{-\frac{1}{2}, \frac{1}{2}\}^d$. Then, for any algorithm, there exists a parameter $\theta^* \in \Theta$ such that sub-optimality gap of a policy learnt by the algorithm after collecting T samples satisfies¹*

$$\mathbb{E}_{\theta^*} [R(T)] \geq \Omega\left(d/\sqrt{T}\right) .$$

Proof: Let π be the policy learnt by any given sampling strategy. Without loss of generality, fix a context $x \in \mathcal{X}$ and denote by $\pi_T(x) \in \mathcal{A}$ the action predicted by the learned policy. For a preference parameter θ , denote by $\mathbb{P}_\theta$ the distribution over $(x_1, a_1, a'_1, y_1, x_2, \dots, x_T, a_T, a'_T, y_T)$. We also denote by $D_{KL}(P, Q)$ the KL-divergence between two probability distributions P and Q . By Divergence Decomposition Lemma (Lemma 3.1), we have, for $\theta, \theta' \in \Theta$,

$$D_{KL}(\mathbb{P}_\theta, \mathbb{P}_{\theta'}) \leq \frac{1}{8} \mathbb{E}_\theta \left[\sum_{t=1}^T \langle a_t - a'_t, \theta - \theta' \rangle^2 \right]$$

Note that for any $x \in \mathcal{X}$ and any two actions $a, a' \in \mathcal{A}$, $(a - a') \in [-1, 1]^d$. Now, we define the event

$$\mathcal{E}_{\theta,i} = \{\text{sign}(\pi_{T,i}(x)) \neq \text{sign}(\theta_i)\} ,$$

where $\pi_{T,i}(x)$ and θ_i are the i -th coordinates of $\pi_T(x)$ and θ respectively. Further, let $\theta' \in \Theta$ be such that $\theta'_i = -\theta_i$ and for $j \neq i$, $\theta'_j = \theta_j$. Note that $\mathcal{E}_{\theta',i} = \mathcal{E}_{\theta,i}^c$. Thus, by Bretagnolle-Huber Inequality (Lemma 3.2),

$$\begin{aligned} \mathbb{P}_\theta[\mathcal{E}_{\theta,i}] + \mathbb{P}_{\theta'}[\mathcal{E}_{\theta',i}] &\geq \frac{1}{2} \exp(-D_{KL}(\mathbb{P}_\theta, \mathbb{P}_{\theta'})) \\ &\geq \frac{1}{2} \exp\left(-\frac{1}{8} \mathbb{E}_\theta \left[\sum_{t=1}^T \langle a_t - a'_t, \theta - \theta' \rangle^2 \right]\right) \\ &\geq \frac{1}{2} \exp\left(-\frac{1}{8} \sum_{t=1}^T 4\theta_i^2\right) = \frac{1}{2} \exp(-\frac{1}{2}) . \end{aligned}$$

¹Expectation is over the randomness of $(x_1, a_1, a'_1, y_1, \dots, y_T)$ under hypothesis θ^* .

Now, since we have $|\Theta| = 2^d$,

$$\sum_{\theta \in \Theta} \frac{1}{|\Theta|} \sum_{i=1}^d \mathbb{P}_{\theta}[\mathcal{E}_{\theta,i}] = \frac{1}{|\Theta|} \sum_{i=1}^d \sum_{\theta \in \Theta} p_{\theta,i} \geq \frac{2^{d-1}}{2^d} \cdot d \cdot \frac{1}{2} \exp(-\frac{1}{2}) = \frac{d}{4} \exp(-\frac{1}{2}).$$

Hence, there exists a $\theta^* \in \Theta$ such that $\sum_{i=1}^d \mathbb{P}_{\theta^*}[\mathcal{E}_{\theta^*,i}] \geq \frac{d}{4} \exp(-\frac{1}{2})$. Thus, the expected suboptimality gap is lower bounded as

$$\begin{aligned} \mathbb{E}_{\theta^*}[R(T)] &\geq \mathbb{E}_{\theta^*} \left[\max_{a \in \mathcal{A}} \langle a - \pi_T(x), \theta^* \rangle \right] \\ &= \mathbb{E}_{\theta^*} \left[\sum_{i=1}^d |\theta_i^*| \cdot \mathbb{1}\{\text{sign}(\pi_{T,i}(x)) \neq \text{sign}(\theta_i^*)\} \right] \\ &\quad \text{(Best action has coordinate-wise same sign as } \theta^*) \\ &= \frac{1}{\sqrt{T}} \sum_{i=1}^d \mathbb{E}_{\theta^*} [\mathbb{1}\{\text{sign}(\pi_{T,i}(x)) \neq \text{sign}(\theta_i^*)\}] \\ &= \frac{1}{\sqrt{T}} \sum_{i=1}^d \mathbb{P}_{\theta^*}[\mathcal{E}_{\theta^*,i}] \geq \frac{1}{\sqrt{T}} \cdot \frac{d \exp(-1/2)}{4}, \end{aligned}$$

which completes the proof. $\square$

To the best of our knowledge, Theorem 3.2 gives the first lower bound for general active-learning algorithms, which was missing in prior work [67, 70]. Now, the immediate question is whether one can design an algorithm that learns a policy whose sub-optimality gap matches this lower bound. In the next section, we present an algorithm *Active Preference Optimization* (APO) that achieves this up to log factors and an instance-dependent non-linear factor.

3.2.1 Auxiliary Lemmas for Proof of Theorem 3.2

Lemma 3.1 (Divergence Decomposition) *Consider a logistic bandit instance with action space $\mathcal{A} \subset \mathbb{R}^d$ and parameter $\theta \in \mathbb{R}^d$. A policy π , over T rounds, takes actions $a_t \in \mathcal{A}$ and observes reward $r_t \sim \text{Ber}(\sigma(\langle a_t, \theta \rangle))$, for all $t \in [T]$. Let the sequence of random actions and observations be denoted by $\mathcal{O} = (a_1, r_1, a_2, \dots, a_T, r_T)$. Further, let $\mathbb{P}_{\theta}$ and $\mathbb{P}_{\theta'}$ be the two distributions of $\mathcal{O}$ under parameters θ and θ' . Then,*

$$D_{KL}(\mathbb{P}_{\theta}, \mathbb{P}_{\theta'}) \leq \frac{1}{8} \mathbb{E}_{\theta} \left[\sum_{t=1}^T \langle a_t, \theta - \theta' \rangle^2 \right].$$

Proof: This lemma is a direct adaptation of [52, Lemma 15.1, Exercise 15.8(b)] to the logistic

bandit setting. From these results, we directly have

$$D_{KL}(\mathbb{P}_\theta, \mathbb{P}_{\theta'}) = \mathbb{E}_\theta \left[\sum_{t=1}^T D_{KL}(\text{Ber}(\sigma(\langle a_t, \theta \rangle)), \text{Ber}(\sigma(\langle a_t, \theta' \rangle))) \right]. \quad (3.3)$$

Now, upon simplification of the KL-divergence between two logistic distributions, we have,

$$\begin{aligned} D_{KL}(\text{Ber}(\sigma(\langle a_t, \theta \rangle)), \text{Ber}(\sigma(\langle a_t, \theta' \rangle))) \\ = (1 - \sigma(\langle a_t, \theta \rangle))\langle a_t, \theta' - \theta \rangle + \log(\sigma(\langle a_t, \theta \rangle)) - \log(\sigma(\langle a_t, \theta' \rangle)). \end{aligned} \quad (3.4)$$

Let $p = \langle a_t, \theta \rangle$ and $q = \langle a_t, \theta' \rangle$. Now, via a Taylor expansion, we have,

$$\begin{aligned} \log(\sigma(q)) &= \log(\sigma(p)) + \frac{\dot{\sigma}(p)}{\sigma(p)}(q - p) + \frac{1}{2} \frac{d}{dp} \left(\frac{\dot{\sigma}(p)}{\sigma(p)} \right) \bigg|_{u \in (p, q)} (q - p)^2 \\ &= \log(\sigma(p)) + (1 - \sigma(p))(q - p) - \frac{1}{2} \dot{\sigma}(u)(q - p)^2. \quad (\dot{\sigma}(p) = \sigma(p)(1 - \sigma(p))) \end{aligned}$$

Therefore, substituting the Taylor expansion in 3.4, we get,

$$D_{KL}(\text{Ber}(\sigma(\langle a_t, \theta \rangle)), \text{Ber}(\sigma(\langle a_t, \theta' \rangle))) = \frac{1}{2} \dot{\sigma}(u) \langle a_t, \theta' - \theta \rangle^2 \leq \frac{1}{8} \langle a_t, \theta' - \theta \rangle^2. \quad (\dot{\sigma}(u) \leq \frac{1}{4})$$

Substituting this in 3.3 gives the desired result. $\square$

Lemma 3.2 (Bretagnolle-Huber Inequality [52, Theorem 14.2]) *Let $(\Omega, \mathcal{F}, P)$ and $(\Omega, \mathcal{F}, Q)$ be two probability spaces. Let $A \in \mathcal{F}$ and A^c be its complement event. Then,*

$$\mathbb{P}_P[A] + \mathbb{P}_Q[A^c] \geq \frac{1}{2} \exp(-D_{KL}(P, Q))$$

3.3 Our Approach: Active Preference Optimization

At each round t , APO (Algorithm 4) proceeds by computing the MLE estimate $\hat{\theta}_t$ based on the data obtained in the past $t - 1$ steps (3.1). Based on $\hat{\theta}_t$, our goal is to maximize exploration. To do this, for a context $x \in \mathcal{X}$, we compute the uncertainty $b_t(x, a, a')$ for each action (a, a') available for that context and choose the one which maximizes this, i.e., we choose the pair

$$(a_t(x), a'_t(x)) = \operatorname{argmax}_{(a, a') \in \mathcal{A} \times \mathcal{A}} b_t(x, a, a'), \quad (3.5)$$

Algorithm 4: AP0: Active Preference Optimization

Input: Context set $\mathcal{X}$, action set $\mathcal{A}$, feature map $\phi : \mathcal{X} \times \mathcal{A} \rightarrow \mathbb{R}^d$, regularization $\lambda > 0$, and failure probability $\delta \in (0, 1]$.

- 1 Initialize $\hat{\theta}_1 = 0$.
 - 2 **for** $t = 1, \dots, T$ **do**
 - 3 Choose the triplet (x_t, a_t, a'_t) using (3.5) and (3.7).
 - 4 Observe preference feedback $y_t \sim \text{Ber}(\sigma(z_t^\top \theta^*))$, where $z_t = \phi(x_t, a_t) - \phi(x_t, a'_t)$.
 - 5 Compute reward estimate $\hat{\theta}_{t+1}$ that minimizes the constrained log-loss (3.1).
 - 6 Compute (scaled) design matrix $H_{t+1}(\hat{\theta}_{t+1})$ via (3.6).
 - 7 Compute final policy $\pi_T(x)$ using (3.8).
-

where $b_t(x, a, a') = \|\phi(x, a) - \phi(x, a')\|_{H_t^{-1}(\hat{\theta}_t)}$ and $H_t(\hat{\theta}_t)$ is a matrix that describes a confidence ellipsoid around the unknown reward parameter θ^* after $t - 1$ steps of data collection. For any $\theta \in \Theta$, this is defined as

$$H_t(\theta) = \nabla^2 \mathcal{L}_t(\theta) + \lambda \mathbf{I}_d = \sum_{s=1}^{t-1} \dot{\sigma}(z_s^\top \theta) z_s z_s^\top + \lambda \mathbf{I}_d, \quad (3.6)$$

where $z_s = \phi(x_s, a_s) - \phi(x_s, a'_s)$ is the feature difference for the triplet (x_s, a_s, a'_s) . Intuitively, the confidence ellipsoid keeps shrinking along whichever direction (in $\mathbb{R}^d$) we decide to explore. Thus, for a given context x , choosing the pair $(a_t(x), a'_t(x))$ maximally reduces the uncertainty among all other possible action duels. However, our algorithm picks not only the action pair that maximizes uncertainty but also the context that increases it the most, i.e.,

$$x_t = \operatorname{argmax}_{x \in \mathcal{X}} b_t(x, a_t(x), a'_t(x)). \quad (3.7)$$

This is a crucial step in our approach that ensures that the uncertainty of the reward function over all contexts decreases at a fast rate, which in turn ensures a low sub-optimality gap. After T rounds, we define $\theta_T = \frac{1}{T} \sum_{s=1}^T \hat{\theta}_s$ as the average of all the past parameter estimates. Our final policy π_T for any context $x \in \mathcal{X}$ is to play the action that maximizes the reward parameterized by θ_T , i.e.,

$$\pi_T(x) = \operatorname{argmax}_{a \in \mathcal{A}(x)} \hat{r}_T(x, a) = \operatorname{argmax}_{a \in \mathcal{A}(x)} \phi(x, a)^\top \theta_T. \quad (3.8)$$

3.3.1 Suboptimality Gap of AP0

We make the following assumption, which is standard in the literature [101, 76].

Assumption 1 (Boundedness) (a) θ^* lies in the set $\Theta = \{\theta \in \mathbb{R}^d \mid \langle \mathbf{1}, \theta \rangle = 0, \|\theta\| \leq S\}$. (b) Features are bounded, i.e., $\|\phi(x, a)\| \leq 1, \forall (x, a) \in \mathcal{X} \times \mathcal{A}$.

The condition $\langle \mathbf{1}, \theta \rangle = 0$ ensures identifiability of θ^* . Now, we define a key quantity that captures learning complexity under the BTL preference model:

$$\kappa = \max_{x \in \mathcal{X}} \max_{a, a' \in \mathcal{A}} \max_{\theta \in \Theta} \frac{1}{\dot{\sigma}(\phi(x, a)^\top \theta - \phi(x, a')^\top \theta)} . \quad (3.9)$$

This parameter κ specifies difficulty in learning via the worst-case non-linearity in preference feedback. Note that we don't need the knowledge of κ beforehand. Next, we present the guarantee that our algorithm enjoys.

Theorem 3.3 (Sub-optimality gap of AP0) Let $\delta \in (0, 1]$. Under Assumption 1, setting $\lambda = \frac{1}{4S^2(2+2S)^2}$ and $\gamma = O\left(S\sqrt{d \log(ST/d) + \log(T/\delta)}\right)$, the policy π_T returned by AP0, with probability at least $1 - \delta$, has the suboptimality gap upper bounded as

$$R(T) \leq O\left(\gamma \sqrt{S \log\left(1 + (T/\lambda \kappa d)\right) \kappa d/T}\right) .$$

Comparison with lower bound and dependence on κ . Theorem 3.3 implies an $\tilde{O}(d\sqrt{\kappa/T})$ upper bound on the sub-optimality gap of AP0 policy. This matches the lower bound of Theorem 3.2 in parameter dimension d and number of samples T , implying optimal scaling w.r.t. these two terms (up to a log factor). There remains a gap in characterizing the optimal dependence on the non-linearity parameter κ , which, in the worst-case, can be exponential in the parameter norm S . In the logistic bandit literature, the state-of-the-art regret guarantee is (almost) κ -independent - the dependence is only in terms independent of T [81, 55]. We believe the $\sqrt{\kappa}$ dependence is unavoidable in the RLHF setting as the sub-optimality gap is w.r.t. real-valued rewards $r^*(x, a) = \phi(x, a)^\top \theta^*$ instead of the sigmoid rewards $\sigma(\phi(x, a)^\top \theta^*)$ in logistic bandits. Given this, we conjecture that it could be possible to improve the lower bound of Theorem 3.2 to $\Omega(d\sqrt{\kappa/T})$. We keep this as an interesting future direction.

Proof: (*Sketch*) First, we quantify the error in estimating θ^* in the following lemma. This is obtained by using a novel inequality derived from the self-concordance property of the sigmoid function (i.e., $|\ddot{\sigma}| \leq \dot{\sigma}$) and adapting the arguments from [55]. Proof of this lemma is deferred to Section 3.6.

Lemma 3.3 (Estimation error at round t) Let $\delta \in (0, 1]$. Under the hypothesis of Theo-

rem 3.3, with probability $\geq 1 - \delta$, for some universal constant $C > 0$,

$$\|\theta^* - \hat{\theta}_t\|_{H_t(\hat{\theta}_t)} \leq CS^{3/2} \sqrt{d \log(st/d) + \log(t/\delta)},$$

where $\hat{\theta}_t$ is the estimated reward parameter that minimizes the constrained log-loss (3.1) and $H_t(\hat{\theta}_t)$ is the (scaled) design matrix (3.6) at round t .

The proof of Theorem 3.3 proceeds by upper bounding the sub-optimality gap for every context with the error in parameter estimation times an arm-dependent quantity. Specifically, for context x , let $z_T(x) = \phi(x, a^*(x)) - \phi(x, \pi_T(x))$ denotes the feature difference for the triplet $(x, a^*(x), \pi_T(x))$, where $a^*(x)$ is the optimal action at context x . Then, from (3.2), the sub-optimality gap for context x can be bounded as $z_T(x)^\top \theta^* \leq z_T(x)^\top \theta^* - z_T(x)^\top \theta_T = \frac{1}{T} \sum_{t=1}^T z_T(x)^\top (\theta^* - \hat{\theta}_t)$. Here, the first inequality is because $\phi(x, \pi_T(x))^\top \theta_T \geq \phi(x, a^*(x))^\top \theta_T$, which follows from definition of π_T , and so $z_T(x)^\top \theta_T \leq 0$. Then, by Cauchy-Schwarz inequality, we get $z_T(x)^\top \theta^* \leq \frac{1}{T} \sum_{t=1}^T \|z_T(x)\|_{H_t(\hat{\theta}_t)^{-1}} \|\theta^* - \hat{\theta}_t\|_{H_t(\hat{\theta}_t)}$.

Now, we apply Lemma 3.3 to upper bound $\|\theta^* - \hat{\theta}_t\|_{H_t(\hat{\theta}_t)}$. Next, we note that $\|z_T(x)\|_{H_t(\hat{\theta}_t)^{-1}} \leq \|z_t\|_{H_t(\hat{\theta}_t)^{-1}}$ by the design of our algorithm. To bound this, consider the regularized sample covariance matrix of feature differences, defined as $V_t = \sum_{s=1}^{t-1} z_s z_s^\top + \kappa \lambda \mathbf{I}_d$. Compare this with $H_t(\theta)$, which scales each rank-one component inside the sum by its variance given that the parameter is θ (see Eq. 3.6). A key relation between these two matrices is that $H_t(\theta) \succcurlyeq V_t / \kappa$. Using this, we upper bound $\|z_t\|_{H_t(\hat{\theta}_t)^{-1}}$ by $\sqrt{\kappa} \|z_t\|_{V_t^{-1}}$. Finally, applying Elliptic Potential Lemma (Lemma 3.10) finishes the proof. We defer the details till Section 3.6. $\square$

Remark 5 To the best of our knowledge, [67] is the only work similar to ours with theoretical guarantees. Therefore, we highlight in detail the major differences. First, the algorithm design is entirely different as we choose both the actions for any context by maximizing the exploration bonus $b_t(x, a, a')$, while [67] chooses one action uniformly at random. This can be wasteful in practice as the choice of the second action is equally crucial in preference-based learning. Further, the context selection rule is entirely different. While we pick the context with the highest exploration bonus (Eq. 3.7), [67] uses an uncertainty estimate calculated via upper and lower confidence bounds of the rewards. Moreover, their suboptimality gap scales linearly with κ , while our guarantee only scales as $\sqrt{\kappa}$.

Next, [67] competes against the Borda winner, while we do so against the (stronger) Condorcet winner. Moreover, [67] assumes that the Borda function $g^*(x, a) = \mathbb{E}_{a' \sim \text{Unif}(A)} [\sigma(r^*(x, a) - r^*(x, a'))]$ lie in the same function space as the reward function $r^*(x, a) = \langle \theta^*, \phi(x, a) \rangle$. This assumption doesn't hold in general due to the non-linearity in σ and hence restricts the preference probabilities. The assumption holds trivially if each $\phi(x, a)$ is a one-hot vector $\mathbf{e}_{x,a}$, but it

pushes θ^* to $|\mathcal{X}| \cdot |\mathcal{A}|$ dimensions and blows up the suboptimality gap significantly. We remove this restriction and provide the guarantee in dimension $d \ll |\mathcal{X}| \cdot |\mathcal{A}|$, by crucially exploiting properties of the sigmoid function.

Remark 6 (Extension to Direct Preference Optimization (DPO)) Another popular alignment algorithm DPO [75] does not train a reward model separately, rather it uses the log-probability $r_\theta(x, a) = \log \pi_\theta(a|x) - \log \pi_{\text{ref}}(a|x)$ as the reward, where $\theta \in \mathbb{R}^d$ parameterizes the policy to be learnt. For example, Softmax policies take the form $\pi_\theta(a|x) \propto \exp(f_\theta(x, a))$, where f_θ is a differentiable function. Now if we assume f_θ to be linear, then π_θ becomes a log-linear policy, i.e., $\log \pi_\theta(a|x) \propto \langle \theta, \phi(x, a) \rangle$, which eventually makes r_θ a (shifted) linear function. Hence, one would be able to apply our proposed approach to learn the policy parameter θ directly from the preference dataset $\mathcal{D}$.

3.4 Generalization Beyond BTL Model

In this section, we remove the assumption of the BTL preference model and assume access to a non-parametric function class

$$\mathcal{F} = \{f : \mathcal{X} \times \mathcal{A} \times \mathcal{A} \rightarrow [0, 1] : f(x, a, a') + f(x, a', a) = 1\} ,$$

where $f(x, a, a')$ denotes the probability that action a wins over action a' (denoted by $a \succ a'$) given context x when the preference function is f , i.e., $f(x, a, a') = \mathbb{P}[a \succ a' | x, f]$. Note that in this case, there is no latent reward model, and hence, this is a strict generalization of the BTL model. Now, we assume that there is a realizable $f^* \in \mathcal{F}$ from which preferences are observed at each round t , i.e. $y_t \sim \text{Ber}(f^*(x_t, a_t, a'_t))$. Further, we assume a *Condorcet* winner at each context $x \in \mathcal{X}$ w.r.t. f^* :

Assumption 2 For all context $x \in \mathcal{X}$, there is an action $a^*(x) \in \mathcal{A}$ such that $f^*(x, a^*(x), a) \geq 1/2 \forall a \in \mathcal{A}$.

Accordingly, the sub-optimality gap of policy π_T is defined as

$$R(T) := \max_{x \in \mathcal{X}} f^*(x, a^*(x), \pi_T(x)) - 1/2 .$$

In this setting, we propose a generalization of APO, namely, APO-Gen (Algorithm 5) described next.

3.4.1 Algorithm

Our algorithm takes a function class $\mathcal{F}$ and a confidence level $\delta \in (0, 1]$ as its inputs. First, a regularized least square estimate of f^* is computed by minimizing the cumulative squared prediction error:

$$\hat{f}_t \in \operatorname{argmin}_{f \in \mathcal{F}} \sum_{s=1}^{t-1} (y_s - f(x_s, a_s, a'_s))^2. \quad (3.10)$$

The confidence set $\mathcal{C}_t(\mathcal{F}, \delta)$ is then defined as

$$\mathcal{C}_t(\mathcal{F}, \delta) := \left\{ f \in \mathcal{F} : \sum_{s=1}^{t-1} (f(x_s, a_s, a'_s) - \hat{f}_t(x_s, a_s, a'_s))^2 \leq \beta_t(\mathcal{F}, \delta) \right\}, \quad (3.11)$$

where $\beta_t(\mathcal{F}, \delta)$ is an appropriately chosen confidence parameter. Since $y_t \sim \text{Ber}(f^*(x_t, a_t, a'_t))$ given (x_t, a_t, a'_t) , We have $\text{Var}[y_t] \leq 1/4$. Thus, following [8], we set the confidence parameter

$$\beta_t(\mathcal{F}, \delta) = 2 \log \frac{2\mathcal{N}(\mathcal{F})}{\delta} + 2 \sqrt{\log \frac{4t(t+1)}{\delta}} + 4,$$

where $\mathcal{N}(\mathcal{F})$ denotes the $(1/T, \|\cdot\|_\infty)$ -covering number¹ of $\mathcal{F}$. This choice of confidence width ensures that f^* lies in the confidence set $\mathcal{C}_t(\mathcal{F}, \delta)$ at all time instant $t \geq 1$ with probability at least $1 - \delta$ (Lemma 3.7).

Next, for each triplet (x, a, a') , we define the exploration bonus $b_t(x, a, a')$ at round t as

$$b_t(x, a, a') = \max_{f_1, f_2 \in \mathcal{C}_t(\mathcal{F}, \delta)} |f_1(x, a, a') - f_2(x, a, a')|, \quad (3.12)$$

which measures the uncertainty of a pair of actions a, a' given a context x with respect to the confidence set $\mathcal{C}_t(\mathcal{F}, \delta)$. The near-optimal action set $\mathcal{A}_t(x)$ at round t is defined as the set of all actions in the previous set $\mathcal{A}_{t-1}(x)$ satisfying

$$\hat{f}_t(x, a, a_0) + b_t(x, a, a_0) \geq 1/2 \forall a_0 \in \mathcal{A}_{t-1}(x). \quad (3.13)$$

Intuitively speaking, we retain only those actions from the previous near-optimal set that are not significantly outperformed by other actions according to the estimates of the current round. Since $f^* \in \mathcal{C}_t(\mathcal{F}, \delta)$, the optimal action $a^*(x)$ lies in $\mathcal{A}_t(x)$ for each context x for all t with high probability (Lemma 3.8). By pruning out suboptimal actions every round, we make better use of samples. When the set $\mathcal{A}_t(x)$ becomes a singleton (i.e., $a^*(x)$ has been identified w.h.p), we

¹For any $\alpha > 0$, we call $\mathcal{F}^\alpha$ an $(\alpha, \|\cdot\|_\infty)$ cover of the function class $\mathcal{F}$ if for any $f \in \mathcal{F}$ there exists an f' in $\mathcal{F}^\alpha$ such that $\|f' - f\|_\infty := \sup_{x \in \mathcal{X}} |f'(x) - f(x)| \leq \alpha$.

Algorithm 5: AP0-Gen: Active Preference Optimization with General Function Approximation

Input: Context set $\mathcal{X}$, action set $\mathcal{A} = [K]$, function class $\mathcal{F}$, failure level $\delta \in (0, 1)$.

- 1 Set $\mathcal{X}_0 = \mathcal{X}$ and $\mathcal{A}_0(x) = \mathcal{A} \ \forall x \in \mathcal{X}$.
 - 2 **for** $t = 1, 2, \dots, T$ **do**
 - 3 Compute function estimate $\hat{f}_t$ using (3.10).
 - 4 Construct confidence set $\mathcal{C}_t(\mathcal{F}, \delta)$ using (3.11).
 - 5 Initialize the $\mathcal{X}_t = \mathcal{X}_{t-1}$.
 - 6 **for each context** $x \in \mathcal{X}_{t-1}$ **do**
 - 7 For each pair of actions $a, a' \in \mathcal{A}_{t-1}(x)$, compute the bonus $b_t(x, a, a')$ using (3.12).
 - 8 Find the near-optimal action set $\mathcal{A}_t(x)$ using (3.13).
 - 9 **if** $|\mathcal{A}_t(x)| = 1$ **then**
 - 10 Set $\mathcal{A}_T(x) = \mathcal{A}_t(x)$ and $\mathcal{X}_t \leftarrow \mathcal{X}_t \setminus \{x\}$.
 - 11 Choose context and pair of actions $(x_t, a_t, a'_t) = \operatorname{argmax}_{x \in \mathcal{X}_t, a, a' \in \mathcal{A}_t(x)} b_t(x, a, a')$.
 - 12 Observe preference $y_t \sim \operatorname{Ber}(f^*(x_t, a_t, a'_t))$
 - 13 Output final policy $\pi_T(x) = a$ for some arbitrary $a \in \mathcal{A}_T(x)$.
-

remove this context from the pool of contexts considered in future rounds.

To encourage exploration, we choose actions $(a_t(x), a'_t(x))$ which has the highest uncertainty in $\mathcal{A}_t(x)$, i.e., we choose

$$(a_t(x), a'_t(x)) = \operatorname{argmax}_{a, a' \in \mathcal{A}_t(x)} b_t(x, a, a') . \quad (3.14)$$

Next, we choose the context x_t that provides the maximum information about the unknown preference function f^* , i.e.,

$$x_t \in \operatorname{argmax}_{x \in \mathcal{X}} b_t(x, a_t(x), a'_t(x)) . \quad (3.15)$$

We play the actions $a_t = a_t(x_t)$ and $a'_t = a'_t(x_t)$ in round t and observe the preference feedback y_t . We repeat this until we have exhausted the budget T . Our final policy π_T samples an action uniformly at random from the set $\mathcal{A}_T(x)$ for every context $x \in \mathcal{X}$. Pseudocode is given in Algorithm 5.

3.4.2 Suboptimality Gap Guarantee of AP0-Gen

Now we state the suboptimality gap upper bound for Algorithm 5. We characterize the complexity of function class $\mathcal{F}$ by its *eluder dimension* [77] and the *covering number*.

Definition 3.2 (Eluder dimension) *The ε -eluder dimension $\dim_\varepsilon(\mathcal{F}, \varepsilon)$ of a function class $\mathcal{F}$ defined on a domain $\mathcal{X}$ is the length of the longest sequence $\{x_i\}_{i=1}^n \subseteq \mathcal{X}$ of input points such*

that for some $\varepsilon' \geq \varepsilon$ and for each $i \in \{2, \dots, n\}$,

$$\sup_{f_1, f_2 \in \mathcal{F}} \left\{ (f_1 - f_2)(x_i) \mid \sqrt{\sum_{j=1}^{i-1} (f_1 - f_2)^2(x_j)} \leq \varepsilon' \right\} > \varepsilon' .$$

We denote by $d_\varepsilon(\mathcal{F}) = \dim_\varepsilon(\mathcal{F}, 1/T)$, the $(1/T)$ -Eluder dimension of the function class $\mathcal{F}$.

Definition 3.3 (Covering number) *The α -covering number of a function class $\mathcal{F}$ is defined as the size of the smallest set $\mathcal{G} \subset \mathcal{F}$ such that for any $f \in \mathcal{F}$, $\min_{g \in \mathcal{G}} \|f - g\|_\infty \leq \alpha$.*

We denote by $\mathcal{N}(\mathcal{F})$ the $\frac{1}{T}$ -covering number of $\mathcal{F}$.

Now, we state the sub-optimality guarantee of the final policy using the eluder dimension and metric entropy of the function class $\mathcal{F}$. The proof is presented in Section 3.7.

Theorem 3.4 (Suboptimality Gap of APO-Gen) *Let $\delta \in (0, 1)$. Then, with probability at least $1 - \delta$, the policy returned by **APO-Gen** (Algorithm 5) has the suboptimality gap upper bounded as*

$$R(T) \leq \tilde{O} \left(\sqrt{\log(\mathcal{N}(\mathcal{F})T/\delta) d_\varepsilon(\mathcal{F})/T} \right) .$$

For the BTL preference model $f(x, a, a') = \mu(\phi(x, a)^\top \theta - \phi(x, a')^\top \theta)$. Define $r = \bar{h}/\underline{h}$, where

$$\begin{aligned} \bar{h} &= \sup_{x, a, a', \theta} \dot{\mu}(\phi(x, a)^\top \theta - \phi(x, a')^\top \theta) , \\ \underline{h} &= \inf_{x, a, a', \theta} \dot{\mu}(\phi(x, a)^\top \theta - \phi(x, a')^\top \theta) . \end{aligned}$$

Then the $\log \mathcal{N}(\mathcal{F})$ and Eluder dimension of $\mathcal{F}$ are at most $O(d \log(\bar{h}T))$ and $O(dr^2 \bar{h} \log(rS \bar{h}T))$, respectively. Note that $\underline{h} = 1/\kappa$ and $\bar{h} \leq 1/4$. This yields $\log \mathcal{N}(\mathcal{F}) = O(d \log T)$ and $d_\varepsilon(\mathcal{F}) = O(\kappa^2 d \log T)$. Hence, we get an $\tilde{O}(\kappa d / \sqrt{T})$ sub-optimality gap for **APO-Gen**, which is $\sqrt{\kappa}$ factor loose than Theorem 3.3. This is because we crucially use self-concordance of the sigmoid function in Theorem 3.3 to shave this extra $\sqrt{\kappa}$ factor. Nevertheless, this result is general enough to subsume other preference models (e.g., Thurstone) beyond the BTL model.

3.5 Experiments

We first present a practical version of **APO**, which largely follows the former with minor changes adapted for the computationally efficient implementation required in large-scale experiments. Next, we present experimental results that demonstrate the efficacy of **APO** over random sampling (hereafter denoted by **Random**) and baselines [67, 70]. Hyperparameter details are given in Appendix 3.5.4. The experiment code can be found [here](#).

Algorithm 6: APO (Practical version)

Input: Context-generation pairs $\mathcal{M} = \{(x, a, a')\}$, sample budget T , encoder ϕ , SFT policy π_{SFT} , log-loss $\mathcal{L}$, batch size B , uncertainty regularizer $\lambda > 0$, KL regularizer $\beta > 0$, learning rate $\eta > 0$.

- 1 Initialize $V_1 = \lambda I, \hat{\theta}_1 = 0, \mathcal{D} = \emptyset$. **for** batch $t = 1, \dots, \lfloor T/B \rfloor$ **do**
- 2 Set $b_t(x, a, a') = \|\phi(x, a) - \phi(x, a')\|_{V_t^{-1}}$ for each $(x, a, a') \in \mathcal{M}$. Set $\mathcal{M}_t = \emptyset$. **for**
 $j = 1, \dots, B$ **do**
- 3 Pick $(x_{t,j}, a_{t,j}, a'_{t,j}) = \underset{(x,a,a') \in \mathcal{M} \setminus \mathcal{M}_t}{\operatorname{argmax}} b_t(x, a, a')$; Observe preference $y_{t,j}$.
- 4 Update $\mathcal{M}_t \leftarrow \mathcal{M}_t \cup \{(x_{t,j}, a_{t,j}, a'_{t,j})\}$ and $\mathcal{D} \leftarrow \mathcal{D} \cup \{(x_{t,j}, a_{t,j}, a'_{t,j}, y_{t,j})\}$.
- 5 Update $\hat{\theta}_{t+1} \leftarrow \text{Gradient-step}(\mathcal{L}, \hat{\theta}_t, \mathcal{D}, \eta)$.
- 6 Update $V_{t+1} \leftarrow V_t + \sum_{j=1}^B z_{t,j} z_{t,j}^\top$, where $z_{t,j} = \phi(x_{t,j}, a_{t,j}) - \phi(x_{t,j}, a'_{t,j})$.
- 7 Set reward $\hat{r}_T(x, a) = \phi(x, a)^\top \hat{\theta}_{\lfloor T/B \rfloor + 1} \forall (x, a)$ and policy $\pi_T \leftarrow \text{PPO}(\pi_{\text{SFT}}, \hat{r}_T, \beta)$.

3.5.1 Practical Version of APO for RLHF

In this practical version of APO (Algorithm 6), we access preference data in batches instead of being fully online. At the start of each batch t , we first compute the uncertainty $b_t(x, a, a')$ of each triplet (x, a, a') (Step 2). This is similar to Algorithm 4 except that here we compute the norm w.r.t. the inverted sample covariance matrix of feature differences V_t^{-1} instead of $H_t(\hat{\theta}_t)^{-1}$ (Step 6). We do so since it is both compute and memory efficient for large scale experiments. To maximize exploration, only those B triplets (x, a, a') are sent for labeling in a batch that have the highest uncertainty $b_t(x, a, a')$, and those are stored in a buffer $\mathcal{D}$. At the end of each batch t , we update the parameter estimate $\hat{\theta}_t$ via a black-box gradient-descend-based algorithm (e.g. Adam [47]) on the log-loss (3.1) over the dataset $\mathcal{D}$. Finally, after the budget T is exhausted, we first learn an estimate $\hat{r}_T$ of the latent reward model r^* , and then align the policy via proximal policy optimization (PPO) [82], which takes as input the SFT policy π_{SFT} , the learnt reward model $\hat{r}_T$ and a KL-regularizer β , and returns π_T .

3.5.2 Results on Controlled Sentiment Generation Task

In this experiment, we consider a user group that prefers positive sentiment completions for movie reviews in the IMDB dataset [60]. The goal is to output generations a that exhibit positive sentiment, catering to the user group’s preferences for a given context x . For controlled evaluation, we generate preference pairs (a, a') utilizing a pre-trained sentiment classifier where $\mathbb{P}(\text{positive-sentiment} \mid x, a) > \mathbb{P}(\text{positive-sentiment} \mid x, a')$. We generate a total of 10000 preference samples (x, a, a') and use 4:1 train-test split. For the SFT policy, we fine-tune GPT-

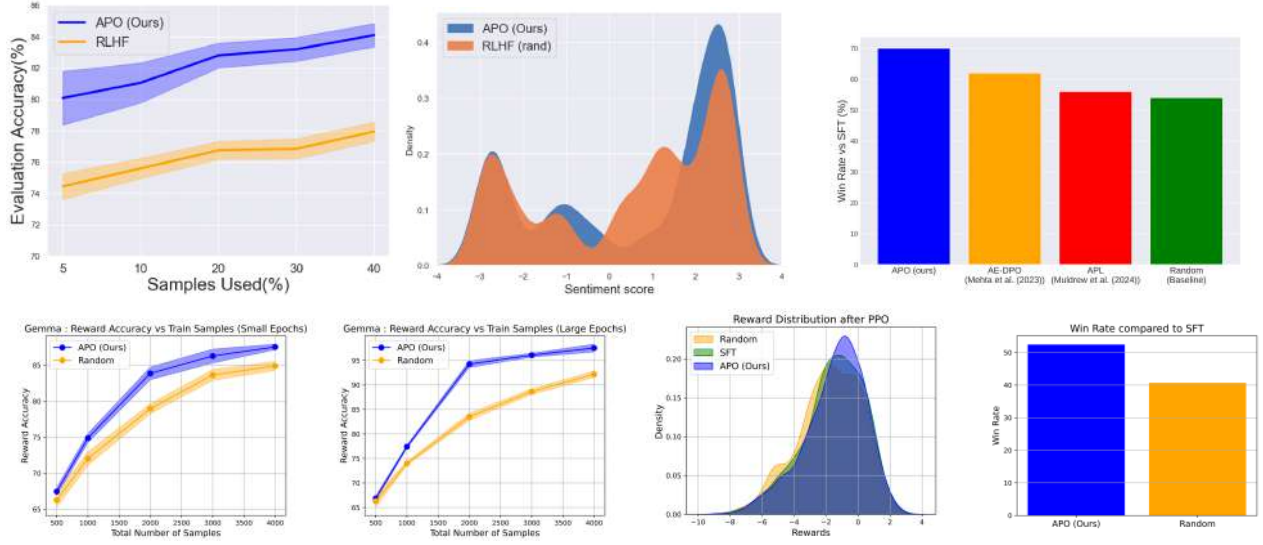


Figure 3.2: **Top Row: Controlled Sentiment Generation Task:** **Left:** Evaluation accuracy of trained reward model vs. no. of samples (in %) comparing APO with Random. **Middle:** Sentiment score distribution of aligned policies trained on reward model learned with APO and on Random’s highest accuracy reward model. Generations by APO-trained reward is more shifted towards positive, showing better alignment than Random. **Right:** Win rates of APO, AE-DPO [67] and APL [70] and Random against SFT policy. APO outperforms AE-DPO, APL and Random by 72 : 62 : 56 : 54 win rate. **Bottom Row: Single-turn Dialogue Task:** **Left and 2nd Left:** Evaluation accuracy of trained reward model vs. no. of samples comparing APO with Random, when the number of epochs is 5 (**Left**) and 20 (**2nd Left**). Evaluation accuracy of APO is higher than the Random in both cases. **2nd Right:** Reward distribution of APO-aligned, SFT and Random-aligned policies for generations on prompts in the test dataset. Clearly, APO’s alignment is better than Random. **Right:** Win rates of APO and Random aligned policies against SFT policy. APO outperforms Random by 55 : 40 win rate.

2 [73] on preferred reviews from the train set (8000 samples) and use this GPT-2 backbone for both reward learning and policy alignment.

For reward training, we adaptively select context and generation pairs from the train set. We use the feature representation $\phi(x, a)$, and estimate the uncertainty $b_t(x, a, a')$ for each (x, a, a') in the train set and select top- B samples to update the reward model. We repeat this process K times and return the final trained reward model. We evaluate the performance of the trained reward model against Random (where we select B samples randomly at every batch) on the test set of 2000 samples. Figure 3.2 (**Top Left**) shows the result: evaluation accuracy of the reward model learned by APO is much higher than the one learned via Random even when APO’s sample budget is only 5% of the data and Random’s is 40%.

For policy alignment, we align the SFT policy with respective trained reward models (via APO and Random) using PPO (step 7). To demonstrate the effectiveness of adaptive sampling, we use the reward model trained on a sample budget of only 10% for APO, while we use the

highest accuracy reward model (corresponding to 40% samples) for **Random**. Similar to [75], the generations of aligned policies are evaluated against the ground truth reward r^* for positive sentiment, which is provided by the pre-trained sentiment classifier. From Figure 3.2 (**Top Middle**), it can be seen that the reward distribution of the generations of **AP0**-aligned policy achieves a higher density of positive sentiment compared to that aligned by **Random**. Moreover, from Figure 3.2 (**Top Right**) it is evident that **AP0** outperforms **APL** [70], **AE-DPO** [67] and **Random** in terms of win-rate against the SFT policy demonstrating the efficiency of the proposed method.

3.5.3 Results on Single-turn Dialogue task

In this experiment, we use Anthropic-HH [9] preference dataset and instruction-tuned Gemma-2b [91] language model. We collect all the contexts with single-turn dialogues and split these into two sets in a 4:1 ratio. We put samples from the larger collection into three buckets based on reward difference between *chosen* and *rejected* responses using **Mistral-7b** reward model as latent reward r^* . These buckets contain data points that are progressively easier to classify: (B1) reward difference between -1 to 1 , (B2) reward difference between 1 to 3 and (B3) reward difference of more than 3 . Out of these three buckets, we take 4500 samples from (B1), 2500 from (B2), and 1000 from (B3) to carefully curate a collection of 8000 training samples. Such a collection (more samples taken from the buckets with a smaller reward difference and fewer samples from the bucket with a larger reward difference) highlights the importance of selecting prompts carefully to obtain useful information during reward training – randomly sampling contexts to collect feedback is more likely to hurt the performance in such a setting. For the test set, we sample 2000 data points from the smaller collection set aside.

Reward Evaluation. We compare the reward models learnt by **AP0** and **Random** by computing the % of samples in the test set for which the models assign higher reward to the *chosen* responses than to the *rejected* responses. We study how this accuracy changes with the number of batches or epochs, keeping the sample budget the same (Fig. 3.2 (**Bottom Left**) for 5 epochs and Fig. 3.2 (**Bottom 2nd Left**) for 20 epochs). We observe that **AP0** always outperforms **Random**. We also see that the reward accuracy increases with an increasing number of epochs for a given sample budget. We show results by varying training samples till 4000 as we want to demonstrate the effectiveness of **AP0** under budget constraint.

Win Rate. Based on reward models learnt by **AP0** and **Random**, we fine tune the SFT policy with **PPO** to obtain **AP0-policy** and **Random-policy** respectively. Then we generate responses for contexts in the test set using **AP0**, **Random**, and **SFT** policies, and get them evaluated by the **Mistral-7b** reward model. The reward distributions of these three policies are shown in Fig. 3.2

(**Bottom 2nd right**). It can be seen that APO has a higher density of positive rewards. Win rate of APO-policy and Random-policy against SFT policy is shown in Fig. 3.2 (**Bottom Right**), which shows that APO outperforms Random by a 55 : 40 win rate.

3.5.4 Hyperparameter Details

Any hyperparameters not mentioned use the default values in the TRL library.¹

Table 3.1: Hyperparameters used in the experiment

Parameter	Value
regularizer in APO	1e-5
beta	0.1
learning rate	1.41e-5
batch size	16
max length	512
max prompt length	128

Table 3.2: Hyperparameters used in PPO Training

Parameter	Value	Parameter	Value
learning rate	1e-4	max new tokens	256
lora-rank	8	top_k	80
lora_alpha	32	top_p	1
lora_dropout	0.05	temperature	1.1
batch size	16	do_sample	True
mini batch size	8		

Experiments on Controlled Sentiment Generation. All experiments here were run on a Tesla T4 (16GB) GPU. The hyperparameters for the experiments are outlined in Table 3.1.

Experiment with Anthropic Dataset. All experiments here were run on an A100 (80 GB) GPU. For reward learning, we use regularizer $\lambda = 1 \times 10^{-5}$. The learning rate for both APO and random is set to 1×10^{-2} with weight decay of 1×10^{-5} . After every epoch of data collection, the training step on the logistic loss is run for 10 epochs. The details for the PPO configuration are the same for both APO and random. The details are given in Table 3.2.

3.6 Proof of Theorem 3.3

We restate Theorem 3.3 and give its proof.

¹huggingface.co/docs/trl/index

Theorem 3.3 (Sub-optimality gap of APO) Let $\delta \in (0, 1]$. Under Assumption 1, setting $\lambda = \frac{1}{4S^2(2+2S)^2}$ and $\gamma = O\left(S\sqrt{d \log(ST/d) + \log(T/\delta)}\right)$, the policy π_T returned by APO, with probability at least $1 - \delta$, has the suboptimality gap upper bounded as

$$R(T) \leq O\left(\gamma \sqrt{S \log\left(1 + (T/\lambda \kappa d)\right) \kappa d/T}\right).$$

Proof: Let the suboptimality gap for a context $x \in \mathcal{X}$ be denoted as $R(T, x)$. Thus,

$$\begin{aligned} R(T, x) &= (\phi(x, a^*(x)) - \phi(x, \pi_T(x)))^\top \theta^* \\ &\leq (\phi(x, a^*(x)) - \phi(x, \pi_T(x)))^\top \theta^* + (\phi(x, \pi_T(x)) - \phi(x, a^*(x)))^\top \left(\frac{1}{T} \sum_{t=1}^T \hat{\theta}_t\right) \\ &\quad (\because \pi_T(x) = \operatorname{argmax}_{a \in \mathcal{A}} \phi(x, a)^\top \left(\frac{1}{T} \sum_{i=1}^T \hat{\theta}_i\right)) \end{aligned}$$

Therefore, we have,

$$\begin{aligned} R(T) &\leq (\phi(x, a^*(x)) - \phi(x, \pi_T(x)))^\top (\theta^* - \frac{1}{T} \sum_{t=1}^T \hat{\theta}_t) \\ &= \frac{1}{T} \sum_{t=1}^T (\phi(x, a^*(x)) - \phi(x, \pi_T(x)))^\top (\theta^* - \hat{\theta}_t) \\ &\leq \frac{1}{T} \sum_{t=1}^T \|\phi(x, a^*(x)) - \phi(x, \pi_T(x))\|_{H_t^{-1}(\hat{\theta}_t)} \|\theta^* - \hat{\theta}_t\|_{H_t(\hat{\theta}_t)}. \quad (\text{Holder's Inequality}) \end{aligned}$$

Now we use lemma 3.6 to upper bound $\|\theta^* - \hat{\theta}_t\|_{H_t(\hat{\theta}_t)}$ with $CS^{1/2}\gamma_t(\delta)$ which we further upper bound by $CS^{1/2}\gamma_T(\delta)$ after noting that $\gamma_t(\delta) \leq \gamma_{t+1}(\delta)$ for all $t \in [T]$. Thus, we now have,

$$\begin{aligned} R(T, x) &\leq \frac{CS^{1/2}\gamma_T(\delta)}{T} \sum_{t=1}^T \|\phi(x, a^*(x)) - \phi(x, \pi_T(x))\|_{H_t^{-1}(\hat{\theta}_t)} \\ &\leq \frac{CS^{1/2}\gamma_T(\delta)}{T} \sum_{t=1}^T \|\phi(x_t, a_t) - \phi(x_t, a'_t)\|_{H_t^{-1}(\hat{\theta}_t)} \\ &\quad ((x_t, a_t, a'_t) = \operatorname{argmax}_{x \in \mathcal{X}, a, a' \in \mathcal{A}} \|\phi(x, a) - \phi(x, a')\|_{H_t^{-1}(\hat{\theta}_t)}) \\ &\leq \frac{C\sqrt{\kappa S}\gamma_T(\delta)}{T} \sum_{t=1}^T \|\phi(x_t, a_t) - \phi(x_t, a'_t)\|_{V_t^{-1}} \quad (V_t \preceq \kappa H_t(\hat{\theta}_t)) \end{aligned}$$

$$\begin{aligned}
&\leq \frac{C\sqrt{\kappa S}\gamma_T(\delta)}{T} \sqrt{T \sum_{t=1}^T \|\phi(x_t, a_t) - \phi(x_t, a'_t)\|_{V_t^{-1}}^2} && \text{(Cauchy-Schwarz)} \\
&\leq \frac{C\sqrt{\kappa S}\gamma_T(\delta)}{T} \sqrt{2dT \log\left(1 + \frac{T}{\lambda\kappa d}\right)} && \text{(Lemma 3.10)} \\
&= CS^{3/2} \sqrt{\left(d \log\left(\frac{ST}{d}\right) + \log\left(\frac{T}{\delta}\right)\right) \log\left(1 + \frac{T}{\lambda\kappa d}\right) \frac{\kappa d}{T}} && \text{(Def. of } \gamma_T(\delta)\text{)}
\end{aligned}$$

Thus, $R(T) = \max_{x \in \mathcal{X}} R(T, x) \leq CS^{3/2} \sqrt{\left(d \log\left(\frac{ST}{d}\right) + \log\left(\frac{T}{\delta}\right)\right) \log\left(1 + \frac{T}{\lambda\kappa d}\right) \frac{\kappa d}{T}}$. $\square$

3.6.1 Auxiliary Lemmas for Proof of Theorem 3.3

Here we first state the result from the logistic bandit literature that characterizes the confidence set for the constrained maximum likelihood estimator. Here, we give one version of the confidence set from [55], but note that similar guarantees are also derived in [1].

Lemma 3.4 (Confidence Set for MLE [55, Theorem 1]) *Let $\hat{\theta}_t$ be the constrained maximum likelihood estimator after $t - 1$ time steps defined as follows:*

$$\hat{\theta}_t = \underset{\theta \in \Theta}{\operatorname{argmin}} \left\{ - \sum_{s=1}^{t-1} y_s \log(\sigma(z_s^\top \theta)) + (1 - y_s) \log(1 - \sigma(z_s^\top \theta)) \right\}.$$

Now define the set

$$\mathcal{C}_t(\delta) = \{\theta \in \Theta : \mathcal{L}_t(\theta) - \mathcal{L}_t(\hat{\theta}) \leq \beta_t(\delta)^2\}$$

where $\beta_t(\delta) = \sqrt{10d \log\left(\frac{St}{4d} + e\right) + 2(e - 2 + S) \log\left(\frac{1}{\delta}\right)}$. Then we have $P(\forall t \geq 1, \theta^* \in \mathcal{C}_t(\delta)) \geq 1 - \delta$.

The details of the proof can be found in [55, section 3.1]. Next, we present another lemma that quantifies the parameter estimation error. Using this lemma and a novel self-concordance property, we will prove Lemma 3.6.

Lemma 3.5 (Lemma 6 of [55]) *Let $\hat{\theta}_t$ be defined above. Further, let $\theta^* \in \mathcal{C}_t(\delta)$. Then,*

$$\|\hat{\theta}_t - \theta^*\|_{H_t(\theta^*)}^2 \leq \gamma_t(\delta)^2 := 2(2 + 2S)f(d, S, t, \delta)$$

where

$$f(d, S, t, \delta) := 2(e - 2)(2 + 2S)d \log\left(\frac{5St}{d}\right) + 2(e - 2)(2 + 2S) \log\left(\frac{t}{\delta}\right) + \frac{5d}{4} + \frac{d^2}{16St}$$

Simplifying the RHS above, we have, $\gamma_t(\delta)^2 = CS^2 \left(d \log \frac{St}{d} + \log \frac{t}{\delta} \right)$ for some $C > 0$.

The proof of the lemma can be found in [55, appendix C.4.4]. Now we present the proof of lemma 3.6.

Lemma 3.6 Suppose $\theta^* \in \mathcal{C}_t(\delta)$. Then, $\|\theta^* - \hat{\theta}_t\|_{H_t(\hat{\theta}_t)} \leq CS^{1/2}\gamma_t(\delta)$.

Proof: By Taylor's theorem, we have,

$$\begin{aligned} & \mathcal{L}_t(\hat{\theta}_t) - \mathcal{L}_t(\theta^*) \\ &= \nabla \mathcal{L}_t(\theta^*)^\top (\hat{\theta}_t - \theta^*) + \int_{v=0}^1 (1-v) (\hat{\theta}_t - \theta^*)^\top \nabla^2 \mathcal{L}_t(\theta^*) (\hat{\theta}_t - \theta^*) dv \\ &= \nabla \mathcal{L}_t(\theta^*)^\top (\hat{\theta}_t - \theta^*) + \sum_{s=1}^{t-1} \left[\int_{v=0}^1 (1-v) \dot{\sigma}(z_s^\top \theta^* + v(z_s^\top \hat{\theta}_t - z_s^\top \theta^*)) dv \right] (z_s^\top (\hat{\theta}_t - \theta^*))^2 \\ &= \nabla \mathcal{L}_t(\theta^*)^\top (\hat{\theta}_t - \theta^*) + \|\hat{\theta}_t - \theta^*\|_{\tilde{G}_t(\theta^*, \hat{\theta}_t)}^2 - \lambda \|\hat{\theta}_t - \theta^*\|^2 \end{aligned}$$

where we define $\tilde{G}_t(\theta^*, \hat{\theta}_t) = \lambda \mathbf{I}_d + \sum_{s=1}^{t-1} \left[\int_{v=0}^1 (1-v) \dot{\sigma}(z_s^\top \theta^* + v(z_s^\top \hat{\theta}_t - z_s^\top \theta^*)) dv \right] z_s z_s^\top$. Thus, we obtain,

$$\|\hat{\theta}_t - \theta^*\|_{\tilde{G}_t(\theta^*, \hat{\theta}_t)}^2 = \mathcal{L}_t(\theta^*) - \mathcal{L}_t(\hat{\theta}_t) + \nabla \mathcal{L}_t(\theta^*)^\top (\hat{\theta}_t - \theta^*) + \lambda \|\hat{\theta}_t - \theta^*\|^2$$

Now, from a novel self-concordant analysis (see lemma 3.9), $H_t(\hat{\theta}_t) \preceq C(2 + 2S)^2 \tilde{G}_t(\theta^*, \hat{\theta}_t)$ for some $C > 1.01$. Thus,

$$\begin{aligned} \|\hat{\theta}_t - \theta^*\|_{H_t(\hat{\theta}_t)}^2 &\leq C(2 + 2S)^2 \|\hat{\theta}_t - \theta^*\|_{\tilde{G}_t(\theta^*, \hat{\theta}_t)}^2 \\ &= C(2 + 2S)^2 \left[\mathcal{L}_t(\theta^*) - \mathcal{L}_t(\hat{\theta}_t) + \nabla \mathcal{L}_t(\theta^*)^\top (\hat{\theta}_t - \theta^*) + \lambda \|\hat{\theta}_t - \theta^*\|^2 \right] \\ &\leq C(2 + 2S)^2 \left[4\lambda S^2 + \beta_t(\delta)^2 + \nabla \mathcal{L}_t(\theta^*)^\top (\hat{\theta}_t - \theta^*) \right] \end{aligned} \tag{3.16}$$

where the last inequality is because (a) $\hat{\theta}_t, \theta^* \in \Theta$ which implies that $\|\theta^* - \hat{\theta}_t\| \leq \text{diam}(\Theta) = 2S$ and (b) by lemma 3.4, $\mathcal{L}_t(\theta^*) - \mathcal{L}_t(\hat{\theta}_t) \leq \beta_t(\delta)^2$ since $\theta^* \in \mathcal{C}_t(\delta)$ by assumption.

Thereafter, from the proof of Lemma 6 of [55] it can be extracted that $|\nabla \mathcal{L}_t(\theta^*)^\top (\hat{\theta}_t - \theta^*)| \leq \frac{\|\hat{\theta}_t - \theta^*\|_{H_t(\theta^*)}^2}{2(2+2S)} + f(d, S, t, \delta)$. Then using lemma 3.5, the R.H.S of 3.16 can be bounded by

$2f(d, S, t, \delta)$. Thus, we now obtain,

$$\begin{aligned}
\|\hat{\theta}_t - \theta^*\|_{H_t(\hat{\theta}_t)}^2 &\leq C(2 + 2S)^2 [4\lambda S^2 + \beta_t(\delta)^2 + 2f(d, S, t, \delta)] \\
&\leq C(2 + 2S)^2 \left[\frac{1}{(2 + 2S)^2} + \beta_t(\delta)^2 + \frac{\gamma_t(\delta)^2}{2(2 + 2S)} \right] \quad (\lambda = \frac{1}{4S^2(2+2S)^2}) \\
&\leq C(2 + 2S)^2 \left[\frac{1}{(2 + 2S)} + \beta_t(\delta) + \frac{\gamma_t(\delta)}{\sqrt{2(2 + 2S)}} \right]^2
\end{aligned}$$

Taking square-root on both sides,

$$\begin{aligned}
\|\hat{\theta}_t - \theta^*\|_{H_t(\hat{\theta}_t)} &\leq C(2 + 2S) \left[\frac{1}{(2 + 2S)} + \beta_t(\delta) + \frac{\gamma_t(\delta)}{\sqrt{2(2 + 2S)}} \right] \\
&= C(1 + (2 + 2S)\beta_t(\delta) + \sqrt{2 + 2S}\gamma_t(\delta)) \\
&= CS^{3/2} \sqrt{\left(d \log\left(\frac{St}{d}\right) + \log\left(\frac{t}{\delta}\right) \right)},
\end{aligned}$$

which proves the lemma. $\square$

3.7 Proof of Theorem 3.4

We restate Theorem 3.4 and present its proof.

Theorem 3.4 (Suboptimality Gap of AP0-Gen) *Let $\delta \in (0, 1)$. Then, with probability at least $1 - \delta$, the policy returned by AP0-Gen (Algorithm 5) has the suboptimality gap upper bounded as*

$$R(T) \leq \tilde{O}\left(\sqrt{\log(\mathcal{N}(\mathcal{F})T/\delta)d_{\varepsilon}(\mathcal{F})/T}\right).$$

Proof: The idea is to show that our arm elimination technique throws away arms with large suboptimality gaps in every round for every context. Thus, the set $\mathcal{A}_t(x)$ maintains a candidate set of good arms at every time instant. In the end, playing any action from $\mathcal{A}_T(x)$ ensures that we only play actions from a set of actions that are only $1/\sqrt{T}$ suboptimal. Formally, for any context $x \in \mathcal{X}$, the suboptimality $R(T, x)$ is upper bounded as follows:

$$R(T, x) = f^*(x, a^*(x), \pi_T(x)) - \frac{1}{2}$$

$$\begin{aligned}
&\leq \frac{1}{T} \sum_{t=1}^T \left[\widehat{f}_t(x, a^*(x), \pi_T(x)) + b_t(x, a^*(x), \pi_T(x)) - \frac{1}{2} \right] \\
&\hspace{25em} (a^*(x), \pi_T(x) \in \mathcal{A}_t(x) \ \forall \ t \in [T]) \\
&= \frac{1}{T} \sum_{t=1}^T \left[\frac{1}{2} - \widehat{f}_t(x, \pi_T(x), a^*(x)) + b_t(x, a^*(x), \pi_T(x)) \right] \\
&\hspace{25em} (f(x, a, a') + f(x, a', a) = 1 \ \forall \ f \in \mathcal{F}) \\
&= \frac{1}{T} \sum_{t=1}^T \left[\frac{1}{2} - \widehat{f}_t(x, \pi_T(x), a^*(x)) + b_t(x, \pi_T(x), a^*(x)) \right] \quad (b_t(x, a, a') = b_t(x, a', a)) \\
&\leq \frac{1}{T} \sum_{t=1}^T [b_t(x, \pi_T(x), a^*(x)) + b_t(x, \pi_T(x), a^*(x))] \\
&\hspace{25em} (\text{Since } \pi_T(x), a^*(x) \in \mathcal{A}_t(x), \text{ line 7 Algorithm 5}) \\
&= \frac{2}{T} \sum_{t=1}^T b_t(x, \pi_T(x), a^*(x)) \\
&\leq \frac{2}{T} \sum_{t=1}^T b_t(x_t, a_t, a'_t) . \hspace{10em} (\text{Line 9 of Algorithm 5})
\end{aligned}$$

Now we invoke lemma 3.12 to bound the RHS.

$$R(T, x) \leq \frac{2}{T} \sum_{t=1}^T b_t(x_t, a_t, a'_t) \leq \frac{2}{T} \left[\frac{1}{T} + \min\{d_\varepsilon(\mathcal{F}), T\} + 2\beta_T(\mathcal{F}, \delta/T) \sqrt{d_\varepsilon(\mathcal{F})T} \right]$$

Simplifying constants and using the fact that $\min\{a, b\} \leq \sqrt{ab}$ for $a, b > 0$, we get $R(T, x) \leq C\beta_T(\mathcal{F}, \delta/T) \sqrt{\frac{d_\varepsilon(\mathcal{F})}{T}}$. Now, using order notation, we have for all $x \in \mathcal{X}$ with probability at least $1 - \delta$,

$$R(T, x) \leq \tilde{O} \left(\sqrt{\frac{\log(\mathcal{N}(\mathcal{F})T/\delta) d_\varepsilon(\mathcal{F})}{T}} \right) .$$

Hence, $R(T) = \max_{x \in \mathcal{X}} R(T, x) \leq \tilde{O} \left(\sqrt{\frac{\log(\mathcal{N}(\mathcal{F})T/\delta) d_\varepsilon(\mathcal{F})}{T}} \right)$. □

3.7.1 Auxiliary Lemmas for Proof of Theorem 3.4

First we present a result that characterizes the confidence set around $\widehat{f}_t$.

Lemma 3.7 (Confidence Set for Function Approximation (Lemma A.1 of [21])) *Let $\delta \in$*

$(0, 1)$. Define the confidence set

$$\mathcal{C}_t(\mathcal{F}, \delta) = \{f \in \mathcal{F} \mid \sum_{s=1}^{t-1} (f(x_s, a_s, a'_s) - \hat{f}_t(x_s, a_s, a'_s))^2 \leq \beta_t(\mathcal{F}, \delta)\}$$

Let $\mathcal{E}_t(\delta)$ be the event that $f^* \in \mathcal{C}_t(\mathcal{F}, \delta)$. Then, $\mathbb{P}[\mathcal{E}_t(\delta)] \geq 1 - \delta$. Further, $\mathbb{P}[\cap_{t=1}^T \mathcal{E}_t(\delta/T)] \geq 1 - \delta$.

Proof: The proof is a direct extension of lemma 3.11 by observing that in our case, the subgaussianity parameter $\sigma = 1/4$ since our rewards are Bernoulli and by setting $\alpha = 1/t$. Moreover, $C = 1$ in our case. Finally, since $\mathcal{E}_t(\delta/t)$ holds with probability at least $1 - \delta/t$, by union bound we can show that $\mathbb{P}[\cap_{t=1}^T \mathcal{E}_t(\delta/T)] = 1 - \mathbb{P}[\cup_{t=1}^T \bar{\mathcal{E}}_t(\delta/T)] \geq 1 - \sum_{t=1}^T \mathbb{P}[\bar{\mathcal{E}}_t(\delta/T)] \geq 1 - \delta$. $\square$

Hereon, we will assume that $\mathcal{E}_t(\delta/T)$ holds for all $t \in [T]$. All subsequent guarantees will be proved under this event. The next result shows that for each context x , the optimal action $a^*(x)$ lies in $\mathcal{A}_t(x)$ for all t .

Lemma 3.8 For a given context $x \in \mathcal{X}$, let $\{\mathcal{A}_s(x)\}_{s=0}^t$ be defined as follows: (a) $\mathcal{A}_0(x) = \mathcal{A}$ and (b) $\mathcal{A}_s(x) = \{a \in \mathcal{A}_{s-1}(x) \mid \hat{f}_s(x, a, a') + b_s(x, a, a') \geq \frac{1}{2} \forall a' \in \mathcal{A}_{s-1}(x)\}$. Then, we have, $a^*(x) \in \mathcal{A}_s(x)$ for all $s \in [t]$.

Proof: The proof is by induction. First note that by definition of $a^*(x)$, $f^*(x, a^*(x), a') \geq 1/2$ for every $a' \in \mathcal{A}$, and $a^*(x) \in \mathcal{A}_0(x) = \mathcal{A}$. Suppose, for some $s > 0$, $a^*(x) \in \mathcal{A}_{s-1}(x)$. Now, we know that under event $\mathcal{E}_s(\delta/T)$, $f^* \in \mathcal{C}_s(\mathcal{F}, \delta/T)$ and thus from definition of $b_s(x, a, a')$, $f^*(x, a, a') - \hat{f}_s(x, a, a') \leq b_s(x, a, a')$. Thus, for any $a' \in \mathcal{A}_{s-1}(x)$,

$$\frac{1}{2} \leq f^*(x, a^*(x), a') \leq \hat{f}_s(x, a^*(x), a') + b_s(x, a^*(x), a')$$

Hence $a^*(x) \in \mathcal{A}_s(x)$. Thus by induction, $a^*(x) \in \mathcal{A}_s(x)$ for all $s \in [t]$. $\square$

3.8 Some Useful Results

Lemma 3.9 Let $z, z' \in \mathbb{R}$ and $\tilde{\alpha}(z, z') := \int_0^1 (1-v) \dot{\sigma}(z + v(z' - z)) dv$. Then, for some $C > 1$ (1.01 suffices),

$$\tilde{\alpha}(z, z') \geq \frac{\dot{\sigma}(z')}{C(2 + |z - z'|)^2}$$

Proof: Firstly, note that by property of definite integrals $\int_a^b f(x) dx = \int_a^b f(a + b - x) dx$, we

have

$$\int_0^1 (1-v)\dot{\sigma}(z+v(z'-z))dv = \int_0^1 v\dot{\sigma}(z'+v(z-z'))dv$$

Now, we use the fact that $\dot{\sigma}(x) \geq \dot{\sigma}(y) \exp(-|x-y|)$ (see appendix A of [29]). Let $a = |z-z'|$. Thus,

$$\begin{aligned} \int_0^1 v\dot{\sigma}(z'+v(z-z'))dv &\geq \int_0^1 v\dot{\sigma}(z')\exp(-va)dv \\ &= \dot{\sigma}(z') \int_0^1 v\exp(-va)dv \\ &= \dot{\sigma}(z') \left(\frac{1-(1+a)e^{-a}}{a^2} \right) \\ &\geq \dot{\sigma}(z') \left(\frac{1-1/a}{a^2} \right) \quad ((1+a)e^{-a} < \frac{1}{a} \forall a > 0) \\ &= \dot{\sigma}(z') \left(\frac{a-1}{a^3} \right) \end{aligned}$$

Again, from appendix A of [29], we have that $\tilde{\alpha}(z, z') \geq \dot{\sigma}(z)/(2+a)$ which can again be lower bounded with $\dot{\sigma}(z')e^{-a}/(2+a)$. Combining this lower bound with the above, we get,

$$\tilde{\alpha}(z, z') \geq \max \left\{ \frac{e^{-a}}{2+a}, \frac{a-1}{a^3} \right\} \dot{\sigma}(z')$$

Finally, we can lower bound the RHS with $\frac{\dot{\sigma}(z')}{C(2+a)^2}$ for some $C > 1.01$. To do this, let $f(x) = (2+x)e^{-x}$. Thus, $f'(x) = -(1+x)e^{-x}$ which implies that $f(x)$ is decreasing for $x > 0$. Thus, $f(x) = \frac{1}{C}$ is satisfied for only one value of x since $f(0) = 2 > 1/C$. For $C = 1.01$, this value is $x_0 = 1.1608$. Then, for $0 \leq x \leq x_0$, $e^{-x}/(2+x) \geq 1/C(2+x)^2$. Again, let $g(x) = (x-1)(x+2)^2/x^3$. Simplifying, we have, $g(x) = 1 + \frac{3}{x} - \frac{4}{x^3}$. It is easy to see that for $x \geq 2/\sqrt{3}$, $\frac{3}{x} \geq \frac{4}{x^3}$ which implies that $g(x) \geq 1$ for all $x \geq x_1 = 2/\sqrt{3} = 1.1547$. So, for $x \geq 1.1547$, $g(x) \geq 1/C$ (since $C > 1$) which is equivalent to $\frac{(x-1)}{x^3} \geq \frac{1}{C(x+2)^2}$. The numeric solution to $g(x) = 1/C$ for $C = 1.01$ is $x_2 = 1.1525$. It can be checked via the first derivative test that $g(x)$ is increasing in $x_2 \leq x \leq x_1$. Thus, indeed, $g(x) \geq 1/C$ for all $x \geq x_2$. Hence, we have established so far that for $C = 1.01$,

$$\begin{aligned} \frac{x-1}{x^3} &\geq \frac{1}{C(x+2)^2} \quad \forall x \geq x_2 = 1.1525 \\ \frac{e^{-x}}{2+x} &\geq \frac{1}{C(x+2)^2} \quad \forall x \leq x_0 = 1.1608 \end{aligned}$$

Since, $x_2 \leq x_0$, we have the required result that $\max \left\{ \frac{e^{-a}}{2+a}, \frac{a-1}{a^3} \right\} \dot{\sigma}(z') \geq \frac{\dot{\sigma}(z')}{C(2+a)^2}$ for all $a \geq 0$ which completes the proof. $\square$

Lemma 3.10 (Elliptic Potential Lemma) *Let $\{z_s\}_{s=1}^t$ be a sequence of vectors in $\mathbb{R}^d$ such that $\|z_s\| \leq L$ for any $s \in [t]$. Let $V_t = \sum_{s=1}^{t-1} z_s z_s^\top + \lambda I$. Then,*

$$\sum_{s=1}^t \|z_s\|_{V_s^{-1}}^2 \leq 2d \log \left(1 + \frac{tL^2}{\lambda d} \right).$$

Now, we present the confidence set properties of function approximation. We use the same notations as [8].

Let $(X_p, Y_p)_{p \geq 1}$ be a sequence of random elements, $X_p \in \mathcal{X}$ for some measurable set $\mathcal{X}$ and $Y_p \in \mathbb{R}$. Let $\mathcal{F}$ be a subset of the set of real-valued measurable functions with domain $\mathcal{X}$. Let $\mathbb{F} = (\mathbb{F}_p)_{p \geq 0}$ be a filtration such that for all $p \geq 1$, $(X_1, Y_1, \dots, X_{p-1}, Y_{p-1}, X_p)$ is $\mathbb{F}_{p-1}$ measurable and such that there exists some function $f^* \in \mathcal{F}$ such that $\mathbb{E}[Y_p \mid \mathbb{F}_{p-1}] = f^*(X_p)$ holds for all $p \geq 1$. The (nonlinear) least-squares predictor given $(X_1, Y_1, \dots, X_t, Y_t)$ is $\hat{f}_t = \operatorname{argmin}_{f \in \mathcal{F}} \sum_{p=1}^t (f(X_p) - Y_p)^2$. We say that Z is conditionally ρ -subgaussian given the σ -algebra $\mathbb{F}$ if for all $\lambda \in \mathbb{R}$, $\log \mathbb{E}[\exp(\lambda Z) \mid \mathbb{F}] \geq \frac{1}{2} \lambda^2 \rho^2$. For $\alpha > 0$, let N_α be the $\|\cdot\|_\infty$ -covering number of $\mathcal{F}$ at scale α . That is, N_α is the smallest integer for which there exist $\mathcal{G} \subset \mathcal{F}$ with N_α elements such that for any $f \in \mathcal{F}$, $\min_{g \in \mathcal{G}} \|f - g\|_\infty \leq \alpha$. For $\beta > 0$, define $\mathcal{F}_t(\beta) = \{f \in \mathcal{F} : \sum_{s=1}^t (f(X_s) - \hat{f}_t(X_s))^2 \leq \beta\}$.

Lemma 3.11 (Theorem 5 of [8]) *Let $\mathbb{F}$ be the filtration defined above and assume that the functions in $\mathcal{F}$ are bounded by the positive constant $C > 0$. Assume that for each $s \geq 1$, $(Y_p - f^*(X_p))_p$ is conditionally σ -subgaussian given $\mathbb{F}_{p-1}$. Then, for any $\alpha > 0$, with probability $1 - \delta$, for all $t \geq 1$, $f^* \in \mathcal{F}_t(\beta_t(\delta, \alpha))$, where*

$$\beta_t(\delta, \alpha) = 8\sigma^2 \log(2N_\alpha/\delta) + 4t\alpha \left(C + \sqrt{\sigma^2 \log(4t(t+1)/\delta)} \right).$$

Lemma 3.12 (Lemma 2 of [77]) *Let $\mathcal{F} \in B_\infty(\mathcal{X}, C)$ be a set of functions bounded by $C > 0$, $(\mathcal{F}_t)_{t \geq 1}$ and $(x_t)_{t \geq 1}$ be sequences such that $\mathcal{F}_t \subset \mathcal{F}$ and $x_t \in \mathcal{X}$ hold for $t \geq 1$. Let $\mathcal{F} \mid_{x_{1:t}} = \{(f(x_1), \dots, f(x_t)) : f \in \mathcal{F}\} \subset \mathbb{R}^t$ and for $S \subset \mathbb{R}^t$, let $\operatorname{diam}(S) = \sup_{u, v \in S} \|u - v\|_2$ be the diameter of S . Then, for any $T \geq 1$ and $\alpha > 0$ it holds that*

$$\sum_{t=1}^T \operatorname{diam}(\mathcal{F}_t \mid_{x_t}) \leq \alpha + C(d \wedge T) + 2\delta T \sqrt{dT}$$

where $\delta_T = \max_{1 \leq t \leq T} \operatorname{diam}(\mathcal{F}_t \mid_{x_t})$ and $d = \dim_{\mathcal{E}}(\mathcal{F}, \alpha)$ is the Eluder Dimension of $\mathcal{F}$.

3.9 Conclusion

In this work, we studied whether sampling prompts uniformly at random from a dataset to solicit feedback is sample efficient. We first showed that this method can suffer a constant suboptimality gap when aligning a language model policy with human preferences. Next, we characterized the sub-optimality gap lower bound for any active-learning algorithm with sample budget T and problem dimension d , showing it to be $\Omega(d/\sqrt{T})$. Then, we proposed an algorithm **AP0**, which actively samples prompts to achieve an $\tilde{O}(d/\sqrt{T})$ sub-optimality gap. We also extended the results of **AP0** to general function approximation to better capture modern-day RLHF training. Finally, we showed its efficacy over the random-sampling baseline on practical datasets. Although **AP0**'s sub-optimality gap is minimax optimal in d and T , the optimal dependence on κ is unknown and seems to be an interesting future work.

Chapter 4

Auditing Fairness under Partial Feedback

4.1 Problem Formulation

In the fairness audit problem, we are interested in testing the fairness of a given *classifier* $f : \mathcal{X} \times \mathcal{A} \rightarrow \{0, 1\}$. Here $\mathcal{X}$ denotes the set of *features* and $\mathcal{A}$ denotes the group labels of individuals. Every individual is characterized by a random tuple (X, A, Y) , $X \in \mathcal{X}$, $A \in \mathcal{A}$ and $Y \in \{0, 1\}$, where Y is the *true label* of the individual. If $f(X, A) = 1$ for an individual, the individual is classified as positive; otherwise, they are classified as negative. The distribution $\mathcal{D}$ of (X, A, Y) is unknown to the auditor. We adopt *equalized odds* as our primary fairness notion for the remainder of the thesis, although our techniques can also be applied to other fairness metrics such as demographic parity [28] and equal opportunity [36].

Throughout this chapter, we use the shorthand notation $f := f(X, A)$ whenever there is no ambiguity regarding X and A . We define $\mathbb{P}[f \mid Y, A] := \mathbb{P}_X[f(X, A) \mid Y, A]$. The *equalized odds difference* is defined as [36]:

$$\Delta := \max_{y \in \{0, 1\}} \max_{a, a' \in \mathcal{A}} |\mathbb{P}[f = 1 \mid Y = y, A = a] - \mathbb{P}[f = 1 \mid Y = y, A = a']|$$

Under this notion, a classifier f is said to be fair if $\Delta \leq \gamma$ and unfair if $\Delta > \gamma + \varepsilon$ for some $\gamma \geq 0, \varepsilon > 0$. We pose the fairness audit of f as a hypothesis testing problem:

Definition 4.1 (($(\gamma, \varepsilon, \delta)$ -Fairness Audit) Given $\gamma \geq 0, \varepsilon, \delta \in (0, 1)$, an algorithm/auditor is said to succeed in the (ε, δ) -fairness audit if with probability at least $1 - \delta$ it can identify correctly which of the following hypotheses is true: (i) FAIR : $\Delta \leq \gamma$, OR (ii) UNFAIR : $\Delta > \gamma + \varepsilon$.

The parameter γ can be seen as a fairness threshold which the classifier must comply with and whose value can be chosen according to the requirement of the problem.

Partial Feedback: We now describe the $(\gamma, \varepsilon, \delta)$ -fairness audit under the partial feedback (PF) model. In the vanilla PF model, the true label is revealed for positively classified individuals only, e.g., the loan default ($Y \in \{0, 1\}$) can only be observed after the loan is given ($f = 1$) by the bank. However, we allow the auditor to request true labels by overriding the negative classification made by the decision-maker. Hence the auditor can explore additional data, which is necessary for the following reason. In our model, an auditor can access (X, A, Y) of individuals who were classified positively in the past by f , but only (X, A) of individuals who were classified negatively. From this database, the probability $\mathbb{P}[f = 1, Y = y, A = a]$, can be estimated, but not $\mathbb{P}[f = 0, Y = y, A = a]$, since Y is unknown when $f = 0$. Hence, we let the auditor access online samples from $\mathcal{D}$ and request X and Y of negatively classified individuals. With this blend of past and online data, the auditor must succeed in $(\gamma, \varepsilon, \delta)$ -fairness audit.

Cost model: We now introduce a novel but natural cost model for exploring online data motivated by practical examples of the partial feedback setting like bank loan applications [49], criminal recidivism [5], and college admissions [24]. The auditor can observe the sensitive attribute of the individual but is charged a fixed cost $c_{feat} \geq 0$ every time they request only the feature of an individual for whom $f = 0$. If the auditor requests only the true label or both the feature and the true label of such an individual, they are charged $c_{feat} + c_{lab} \cdot \mathbb{1}\{Y = 0\}$, $c_{lab} \geq 0$. We illustrate the motivation of this cost definition through the example of bank loan application. Firstly, there may be a cost involved in collecting the personal details of an individual, which is captured by c_{feat} and incurred to the auditor when the bank does not approve the loan. Secondly, to obtain the true label, the auditor has to force the bank (since $f = 0$) to give loan to the individual. If the individual fails to return the loan ($Y = 0$), only then the bank suffers a monetary loss. To capture this asymmetry, we define the cost formally below.

Definition 4.2 (Audit Cost) *Let the auditor conclude the $(\gamma, \varepsilon, \delta)$ -fairness audit after running their algorithm on N iid individuals sampled from $\mathcal{D}$ and let g_i and h_i respectively denote whether the auditor observes the feature only, and the true label of the i -th individual. Then, the audit cost is:*

$$cost := \sum_{i=1}^N \mathbb{E} [c_{feat} \cdot \mathbb{1}\{g_i = 1 \text{ or } h_i = 1, f = 0\} + c_{lab} \cdot \mathbb{1}\{h_i = 1, Y = 0, f = 0\}] .$$

Typically, $c_{feat} \ll c_{lab}$, e.g. in the bank loan application, the cost of loan default is much

higher than the loan processing fee, credit risk assessment etc. For the partial feedback setting, the audit cost is a more effective performance measure than the usual sample complexity in capturing the asymmetry in the cost of true label realization. Further, since the auditor might request features and labels separately, it is not clear what should count towards sample complexity. To this end, the cost definition provides a unified measure of performance of fairness audit algorithms in the partial feedback setting. Hence, with this novel definition of audit cost, we now state the main question studied in this work:

Can we design an algorithm that succeeds in the $(\gamma, \varepsilon, \delta)$ -fairness audit with as small audit cost as possible?

In this work, we study this question under two models of data distribution—*blackbox* and *mixture*. In the blackbox model, no assumption about the data distribution is made. In the mixture model, the distribution of (X, Y) conditioned on A , is assumed to come from a mixture of exponential family distributions. Recall that an exponential family distribution over covariate Z (domain $\mathcal{Z}$) and with parameter $\theta \in \mathbb{R}^d$ is given by the probability density function $\mathcal{E}_\theta(Z) := h(Z) \exp(\theta^\top T(Z) - W(\theta))$, where $h : \mathcal{Z} \rightarrow \mathbb{R}$ is the base measure, $T : \mathcal{Z} \rightarrow \mathbb{R}^d$ is the sufficient statistics and $W : \mathbb{R}^d \rightarrow \mathbb{R}$ is the log-partition function. With different choices of h, T and W , one can capture many different distributions in this class, notably, multivariate Gaussian, Poisson, Exponential and Binomial.

In our mixture model, the class of distribution from the exponential family is known (*i.e.*, h, T and W are known). The data (X, A, Y) is generated is as follows: first a group label A is sampled, then true label Y is sampled as $Y \sim \text{Ber}(q_{1|A})$ and finally, feature X is sampled as $\mathbb{P}[X \mid Y, A] = \mathcal{E}_{\theta_{Y,A}^*}(X) = h(X) \exp(\theta_{Y,A}^{*\top} T(X) - W(\theta_{Y,A}^*))$, where $\theta_{y,a}^* \in \Theta \subset \mathbb{R}^d$, $y \in \{0, 1\}$, $a \in \mathcal{A}$, are unknown. Here Θ is a known convex set.

In the following section, we highlight our main results which includes the audit cost of a natural baseline algorithm, followed by an improved cost algorithm for the blackbox model and a matching lower bound. Next, for the mixture model, we design another algorithm that further improves the cost.

4.2 Our Results

We denote by $q_{y,a} := \mathbb{P}[Y = y, A = a]$, $q_{y|a} := \mathbb{P}[Y = y \mid A = a]$ for all $y \in \{0, 1\}$ and $a \in \mathcal{A}$. Note that $q_{y,a} = q_{y|a} \cdot \mathbb{P}[A = a]$, therefore $q_{y,a} \leq q_{y|a}$. Next, we state the key results of this work. All our results hold for arbitrary classifier f , and our techniques extend to other fairness metrics like demographic parity and equal opportunity.

Recall that due to partial feedback, true labels for negatively classified individuals are not

observed. Hence, a natural baseline audit policy is to request true label for every individual with $f = 0$. We have the following characterization of the audit cost of this policy (proof in Section 4.6.1).

Theorem 4.1 *Given $\gamma \geq 0$, $\varepsilon, \delta \in (0, 1)$, under the blackbox model, the audit policy that requests true labels of every individual with $f = 0$ (Algorithm 7) succeeds in the $(\gamma, \varepsilon, \delta)$ -fairness audit with $\text{cost} \leq \tilde{O}\left(\sum_{a \in \mathcal{A}} \sum_{y \in \{0,1\}} \frac{c_{\text{feat}}\mathbb{P}[f=0] + c_{\text{lab}}\mathbb{P}[f=0, Y=0]}{q_{y,a}\varepsilon^2}\right)$. Moreover, the number of true labels requested by this policy is at most $\tilde{O}(\sum_{a \in \mathcal{A}} \sum_{y \in \{0,1\}} \frac{\mathbb{P}[f=0]}{q_{y,a}\varepsilon^2})$.*

Next, we give an algorithm **RS-Audit** (Algorithm 8) that has improved audit cost using rejection sampling. This algorithm requests an optimal number of true labels, indicating that characterization of the blackbox case is essentially complete. In the design of **RS-Audit** (and the baseline), we use a threshold-based stopping time construction to decide when to stop requesting true labels. The analysis requires concentration properties of the Negative Binomial distribution, instead of the usual Chernoff bounds (Lemma 4.15). Below we formally state the guarantees (proof in Section 4.3.2).

Theorem 4.2 *For $\gamma \geq 0$, $\varepsilon, \delta \in (0, 1)$, in the blackbox model, Algorithm 8 (**RS-Audit**) succeeds in $(\gamma, \varepsilon, \delta)$ -fairness audit with $\text{cost} \leq \tilde{O}\left(\sum_{a \in \mathcal{A}} \sum_{y \in \{0,1\}} \frac{c_{\text{feat}}\mathbb{P}[f=0|A=a] + c_{\text{lab}}\mathbb{P}[f=0, Y=0|A=a]}{q_{y|a}\varepsilon^2}\right)$. Moreover, the number of true labels requested by this policy is at most $\tilde{O}(\sum_{a \in \mathcal{A}} \sum_{y \in \{0,1\}} \frac{\mathbb{P}[f=0|A=a]}{q_{y|a}\varepsilon^2})$.*

RS-Audit runs in polynomial time. The audit cost and the number of true labels requested by it is lower than the baseline since $\frac{\mathbb{P}[f=0, Y=0, A=a]}{q_{y,a}} \leq \frac{\mathbb{P}[f=0, Y=0]}{q_{y,a}}$ and $\frac{\mathbb{P}[f=0, A=a]}{q_{y,a}} \leq \frac{\mathbb{P}[f=0]}{q_{y,a}}$. Hence, the number of true labels requested by the baseline is more than **RS-Audit** by $\tilde{\Omega}(\sum_{y \in \{0,1\}, a \in \mathcal{A}} \frac{\mathbb{P}[f=0, A \neq a]}{q_{y,a}\varepsilon^2})$. and the audit cost by $\tilde{\Omega}(\sum_{y \in \{0,1\}, a \in \mathcal{A}} \frac{c_{\text{feat}}\mathbb{P}[f=0, A \neq a] + c_{\text{lab}}\mathbb{P}[f=0, Y=0, A \neq a]}{q_{y,a}\varepsilon^2})$. Next, we show a lower bound on the cost incurred by any algorithm for auditing (proof in Section 4.3.3).

Theorem 4.3 (Lower Bound) *Under the blackbox setting, for any $\varepsilon \in (0, 1/8)$ and $\delta \in (0, 1)$, there exists a family of instances characterized by $\gamma = 0$, $q_{y|a}$, and $\mathbb{P}[f = 0 | A = a]$ for $y \in \{0, 1\}$ and $a \in \mathcal{A}$, even with $|\mathcal{A}| = 2$, such that any algorithm with finite expected running time that succeeds in the (ε, δ) -fairness test must suffer at least $\tilde{\Omega}\left(\sum_{a \in \mathcal{A}} \sum_{y \in \{0,1\}} \frac{c_{\text{feat}}\mathbb{P}[f=0|A=a] + c_{\text{lab}}\mathbb{P}[f=0, Y=0|A=a]}{q_{y|a}\varepsilon^2}\right)$ cost in expectation.*

Note that the audit cost incurred by **RS-Audit** (Algorithm 8) is within log factors of the lower bound. This in turn indicates that the characterization of the blackbox model is complete. To enrich our understanding of the fairness audit problem under partial feedback, we consider

mixture models under some mild structural assumptions. In the mixture model setting, we make the following assumptions:

Assumption 3 (Structural) (i) $\mathbb{P}[f = 1 \mid Y = y, A = a] \geq \alpha > 0$ for some known constant α , (ii) $\mathcal{E}_\theta(X)$ is log-concave in X , $\kappa \mathbf{I} \preceq \nabla^2 W(\theta) \preceq \lambda \mathbf{I}$ with $0 < \kappa < \lambda$ and W is L -Lipschitz for all $\theta \in \Theta$, $T(X)$ has components polynomial in X , $\mathcal{B}(\theta_{y,a}^*, \frac{1}{\beta}) \subset \Theta$ for some $\beta > 0$ for all $y \in \{0, 1\}$ and $a \in \mathcal{A}$.

Without loss of generality, let $\kappa \leq 1$ and $L, \beta \geq 1$. All these assumptions except Lipschitzness are from [54] which are used to recover model parameters from truncated data. The positivity assumption (i) also appears in other partial feedback works like [45] and ensures that enough probability mass lies on the positive classification, allowing the distribution recovery. The assumptions in (ii) ensure computational efficiency and concentration bounds. These assumptions along with Lipschitzness are mild and are satisfied by several popular distributions like Gaussian, Exponential and continuous Bernoulli for bounded parameter sets (see Appendix 4.8).

Assumption 4 (Separation) With $q_{M,a} := q_{1|a} \vee q_{0|a}$ and $q_{m,a} := q_{1|a} \wedge q_{0|a}$, let for all $a \in \mathcal{A}$:

$$\|\theta_{1,a}^* - \theta_{0,a}^*\| \geq \sqrt{\frac{48(L \vee \beta)\beta + 3\lambda}{8\kappa\beta^2} \log\left(\frac{10q_{M,a}}{q_{m,a}\varepsilon}\right)} \quad (4.1)$$

This separation assumption between parameters enables us to construct an approximate oracle to learn the mixture weights without requesting labels. Such an assumption has been considered in the context of mixture of spherical Gaussians in [61] and has been shown to be mild (and tight) condition for low Bayes error rate (see their Table 1 & Remark 8). We provide further discussion in Section 4.4 Remark 11.

Given these aforementioned assumptions, we state our next result which provides an auditing algorithm and its cost bound for the structured setting of the mixture model (proof in Section 4.7).

Theorem 4.4 Given $\gamma \geq 0$, $\varepsilon, \delta \in (0, 1)$, under the mixture model and assumptions 3 and 4, **Exp-Audit** (Algorithm 9) succeeds in $(\gamma, \varepsilon, \delta)$ -fairness audit with **cost** $\leq \tilde{O}\left(\sum_{a \in \mathcal{A}} \left(\frac{c_{feat} \mathbb{P}[f=0|A=a]}{q_{m,a}\varepsilon^2} + \sum_{y \in \{0,1\}} \frac{c_{lab} \mathbb{P}[f=0, Y=0|A=a]}{q_{y|a}}\right)\right)$.

Moreover, **Exp-Audit** runs in polynomial time. The above theorem shows that under structural and separation assumptions on the data distribution, algorithm with better audit cost

exists. Notably, the cost term concerning c_{lab} is a problem-dependent constant and independent of ε . Our algorithm consists of two key components. First is to view the historical partial feedback data as a collection of truncated samples, thereby applying algorithms for learning distribution parameters from truncated samples [54]. Second is learning mixture weights with approximate MAP oracle for mixture of exponential family. We believe the first component to be a conceptual contribution since it motivates studying partial feedback, an important setting in fairness in decision-making, through the lens of truncated samples. For the second component, we generalize [61] from spherical Gaussians to exponential family, which we believe could be of independent interest.

4.3 Fairness Audit under Blackbox Model

In this section we describe the algorithms for fairness audit under partial feedback for the blackbox model. We first describe the working of the natural baseline policy (Algorithm 7) and then another algorithm **RS-Audit** (Algorithm 8) which improves upon the audit cost and sample complexity of the baseline policy. Finally, we close the section with the construction and the proof of the lower bound (Theorem 4.3).

Recall that in the audit model under partial feedback, the auditor has to request the true label Y of a negatively classified individual by paying a cost. For the auditor to succeed in fairness audit, it suffices to estimate $\mathbb{P}[f = 1 \mid Y = y, A = a]$ for all $y \in \{0, 1\}$ and $a \in \mathcal{A}$. For a given $a \in \mathcal{A}$ and $y \in \{0, 1\}$, $\mathbb{P}[f = 1 \mid Y = y, A = a] = \frac{\mathbb{P}[f=1, Y=y, A=a]}{\mathbb{P}[Y=y, A=a]}$. First, note that $p_{y,a} := \mathbb{P}[f = 1, Y = y, A = a]$ can be estimated from past database using a simple empirical estimate $\frac{1}{n} \sum_{i=1}^n \mathbb{1}\{f_i = 1, Y_i = y, A_i = a\}$ without incurring any cost. As the size of the database n grows, this estimate becomes more accurate. Henceforth, we will focus on estimation of $q_{y,a}$ which requires online data, although we will specify how large n is sufficient for the fairness audit.

Baseline (Algorithm 7). To estimate $q_{y,a}$, a natural policy is to observe the true labels of all negatively classified individuals. To correctly identify the true hypothesis, it is sufficient to estimate $p_{y,a}$ and $q_{y,a}$ within a multiplicative factor of $1 \pm \frac{\varepsilon}{12}$ for all $y \in \{0, 1\}$ and $a \in \mathcal{A}$ (see proof of Theorem 4.1, Eq. (4.6) in Section 4.6.1). For a fixed $y \in \{0, 1\}$ and $a \in \mathcal{A}$, $\tilde{O}(\frac{1}{p_{y,a}\varepsilon^2})$ and $\tilde{O}(\frac{1}{q_{y,a}\varepsilon^2})$ samples are sufficient, respectively, via Chernoff bound (Lemma 4.15). Therefore, the auditor can stop after observing $\tilde{O}(\frac{1}{q_{y,a}\varepsilon^2})$ true labels from online data (including $f = 1$ individuals, and those the auditor requests). However, this insight does not translate into an algorithm since $p_{y,a}$ and $q_{y,a}$ are unknown.

Hence, we base our algorithm on a random stopping time that depends on an appropriately chosen threshold τ which is a function of ε , δ and $|\mathcal{A}|$ only. The stopping time for $q_{y,a}$ estimation

is defined as $N_{y,a} := \min\{t \geq 1 : \sum_{i=1}^t \mathbb{1}\{Y_i = y, A_i = a\} \geq \tau\}$ and likewise define $N'_{y,a}$ for $p_{y,a}$ but with $\mathbb{1}\{f_i = 1, Y_i = y, A_i = a\}$. In other words, for every $y \in \{0, 1\}$ and $a \in \mathcal{A}$, the auditor keeps requesting true labels of individuals till a total (including $f = 1$ individuals) of τ individuals with $Y = y$ and $A = a$ are observed. Thereafter, $\hat{q}_{y,a} = \frac{\tau}{N_{y,a}}$ and $\hat{p}_{y,a} = \frac{\tau}{N'_{y,a}}$ act as an estimate of $q_{y,a}$ and $p_{y,a}$ respectively. Using concentration of Negative Binomials (Lemma 4.13), we show that these estimates are within $1 \pm \frac{\varepsilon}{12}$ factor of $q_{y,a}$ and $p_{y,a}$. The auditor then computes $\hat{\Delta} = \max_{y \in \{0,1\}, a, a' \in \mathcal{A}} |\frac{\hat{p}_{y,a}}{\hat{q}_{y,a}} - \frac{\hat{p}_{y,a'}}{\hat{q}_{y,a'}}|$ and concludes **UNFAIR** if $\hat{\Delta} > \gamma + \frac{\varepsilon}{2}$ and **FAIR** otherwise. The details of the algorithm are presented in Algorithm 7.

Algorithm 7: Baseline Audit Algorithm

Input: $\varepsilon, \delta \in (0, 1)$, past database D .

```

1 Set  $\tau = 576 \log(8|\mathcal{A}|/\delta)/\varepsilon^2$  for  $y \in \{0, 1\}$  do
2   for  $a \in \mathcal{A}$  do
3     Observe the true labels  $Y$  of individuals till  $\tau$  individuals with  $Y = y$  and  $A = a$ 
       have been observed (including individuals with  $f = 1$ ). Let the number of
       individuals whose true labels are observed be  $N$ .
4     Iterate over the past database  $D$  of the decision-maker till  $\tau$  individuals with
        $f = 1, Y = y$  and  $A = a$  are observed. Let the number of individuals iterated
       over be  $N'$ .
5     Set  $\hat{q}_{y,a} = \frac{\tau}{N}$  and  $\hat{p}_{y,a} = \frac{\tau}{N'}$ .
6 Set  $\hat{\Delta} = \max_{y \in \{0,1\}} \max_{a, a' \in \mathcal{A}} \left| \frac{\hat{p}_{y,a}}{\hat{q}_{y,a}} - \frac{\hat{p}_{y,a'}}{\hat{q}_{y,a'}} \right|$ .
7 if  $\hat{\Delta} > \frac{\varepsilon}{2}$  then
8   return UNFAIR.
9 else
10  return FAIR.

```

4.3.1 RS-Audit: Rejection Sampling based Audit

We start with the observation that $\mathbb{P}[f = 1 \mid Y = y, A = a]$ can also be written as:

$$\mathbb{P}[f = 1 \mid Y = y, A = a] = \frac{\mathbb{P}[f = 1, Y = y \mid A = a]}{\mathbb{P}[Y = y \mid A = a]}.$$

The key insight from this expression is that it suffices to estimate the conditional quantities $p_{y|a} := \mathbb{P}[f = 1, Y = y \mid A = a]$ and $q_{y|a}$ instead of the joint probabilities $p_{y,a}$ and $q_{y,a}$. With this, we describe **RS-Audit** (Algorithm 8). For a fixed $a \in \mathcal{A}$, **RS-Audit** overlooks all samples with $A \neq a$, hence simulating samples from distribution conditioned on $A = a$ via rejection sampling. In this distribution conditioned on $A = a$, **RS-Audit** estimates $q_{y|a}$ via a similar stopping time

Algorithm 8: RS-Audit: Fairnes Audit Algorithm for Blackbox Model

Input: $\varepsilon, \delta \in (0, 1)$ and past database D , then set $\tau = 576 \log(8|\mathcal{A}|/\delta)/\varepsilon^2$.

```

1 for  $y \in \{0, 1\}$  do
2   for  $a \in \mathcal{A}$  do
3      $N \leftarrow \text{Online}(\tau, y, a)$ ,  $N' \leftarrow \text{Past}(D, \tau, y, a)$ .
4     Set  $\hat{q}_{y|a} = \frac{\tau}{N}$  and  $\hat{p}_{y|a} = \frac{\tau}{N'}$ .
5    $\hat{\Delta} = \max_{y \in \{0, 1\}, a, a' \in \mathcal{A}} \left| \frac{\hat{p}_{y|a}}{\hat{q}_{y|a}} - \frac{\hat{p}_{y|a'}}{\hat{q}_{y|a'}} \right|$ .
6   if  $\hat{\Delta} > \gamma + \frac{\varepsilon}{2}$  then
7     return UNFAIR.
8 else
9   return FAIR.

```

construction: $N_{y|a} = \min\{t_a \geq 1 : \sum_{i=1}^{t_a} \mathbb{1}\{Y_i = y\} \geq \tau\}$, where t_a denotes the number of individuals with $A = a$. This is done via the subroutine $\text{Online}(\tau, y, a)$ that returns the number of true labels $N_{y|a}$ observed (including both $f = 1$ individuals, and those the auditor requests) till τ individuals with label $Y = y$ are observed. Similarly, $p_{y|a}$ is estimated via subroutine $\text{Past}(D, \tau, y, a)$ from past database D after removing all individuals with $A \neq a$ to obtain D_a and then using similar stopping time $N'_{y|a} = \min\{t'_a \geq 1 : \sum_{i=1}^{t'_a} \mathbb{1}\{f_i = 1, Y_i = y\} \geq \tau\}$, *i.e.*, counting the data points in D_a sequentially till τ individuals with $\{f = 1, Y = y\}$ are observed. The estimates $\hat{q}_{y|a} = \frac{\tau}{N_{y|a}}$ and $\hat{p}_{y|a} = \frac{\tau}{N'_{y|a}}$ are within a factor of $(1 \pm \frac{\varepsilon}{12})$ of their true values (proved using concentration of Negative Binomials (Lemma 4.13)). Finally, the algorithm computes $\hat{\Delta} = \max_{y \in \{0, 1\}, a, a' \in \mathcal{A}} \left| \frac{\hat{p}_{y|a}}{\hat{q}_{y|a}} - \frac{\hat{p}_{y|a'}}{\hat{q}_{y|a'}} \right|$ and concludes UNFAIR if $\hat{\Delta} > \gamma + \frac{\varepsilon}{2}$ and FAIR otherwise.

Remark 7 Although *RS-Audit* and baseline (Algorithm 7) are superficially similar, these are two conceptually different algorithms. The novel insight of not observing the true labels for individuals with $A \neq a$ when trying to estimate $q_{y|a}$ exploits the structure of the audit cost under partial feedback where A is observable without cost, which accrues only when the auditor requests the true labels. Moreover, we crucially identify the right amount of information needed to estimate $\mathbb{P}[f = 1 \mid Y = y, A = a]$, namely, samples of true labels from the distribution $\mathbb{P}[Y \mid A = a]$. Combining these two observations leads to *RS-Audit*, which is nearly optimal in cost complexity (see Theorem 4.3). Moreover, due to these observations, the gap in sample complexity of baseline over *RS-Audit* is roughly $\tilde{\Omega} \left(\sum_{y \in \{0, 1\}, a \in \mathcal{A}} \frac{\mathbb{P}[f=0, A \neq a]}{q_{y,a} \varepsilon^2} \right)$ and the difference in audit cost is roughly $\tilde{\Omega} \left(\sum_{y \in \{0, 1\}, a \in \mathcal{A}} \frac{c_{feat} \mathbb{P}[f=0, A \neq a] + c_{lab} \mathbb{P}[Y=0, f=0, A \neq a]}{q_{y,a} \varepsilon^2} \right)$.

4.3.2 Proof of Theorem 4.2

In this section we present the analysis of **RS-Audit** (Algorithm 8) under the blackbox model. Recall that the quantities of interest are $p_{y,a} := \mathbb{P}[f = 1, Y = y, A = a]$, $q_{y,a} := \mathbb{P}[Y = y, A = a]$, $p_{y|a} := \mathbb{P}[f = 1, Y = y \mid A = a]$, and $q_{y|a} := \mathbb{P}[Y = y \mid A = a]$ for all $y \in \{0, 1\}$ and $a \in \mathcal{A}$. Note that $p_{y,a} = p_{y|a} \cdot \mathbb{P}[A = a]$, therefore $p_{y,a} \leq p_{y|a}$, and similarly, $q_{y,a} \leq q_{y|a}$. We now state the proof of Theorem 4.2 restated below.

Theorem 4.2 *For $\gamma \geq 0$, $\varepsilon, \delta \in (0, 1)$, in the blackbox model, Algorithm 8 (**RS-Audit**) succeeds in $(\gamma, \varepsilon, \delta)$ -fairness audit with $\text{cost} \leq \tilde{O}\left(\sum_{a \in \mathcal{A}} \sum_{y \in \{0,1\}} \frac{c_{feat}\mathbb{P}[f=0|A=a] + c_{lab}\mathbb{P}[f=0,Y=0|A=a]}{q_{y|a}\varepsilon^2}\right)$. Moreover, the number of true labels requested by this policy is at most $\tilde{O}(\sum_{a \in \mathcal{A}} \sum_{y \in \{0,1\}} \frac{\mathbb{P}[f=0|A=a]}{q_{y|a}\varepsilon^2})$.*

Proof: The proof essentially follows similar arguments as in the proof of Theorem 4.1 with the crucial difference that we now consider the conditional distributions $\mathbb{P}[Y \mid A]$. This is because Algorithm 8 iterates over all $y \in \{0, 1\}$ and $a \in \mathcal{A}$ with the key difference that in a particular iteration, *i.e.*, for a fixed $y \in \{0, 1\}$ and $a \in \mathcal{A}$, the algorithm only requests the true labels of individuals with $A = a$. This is akin to collecting samples from the conditional distribution obtained by conditioning on $A = a$. Therefore, the arguments have to be adapted to the conditional distribution.

In particular, let us fix $y \in \{0, 1\}$ and $a \in \mathcal{A}$. Let $N_{y|a}$ be the number of samples collected by the algorithm in the corresponding iteration (Step 4 of the algorithm). Consider the event $\mathcal{E}_{y,a} := \{N_{y|a} \in (1 \pm \frac{\varepsilon}{12}) \frac{576 \log(8|\mathcal{A}|/\delta)}{\varepsilon^2 q_{y|a}}\}$ (recall that $q_{y|a} = \mathbb{P}[Y = y \mid A = a]$). From Lemma 4.13, with $\tau = \frac{576 \log(8|\mathcal{A}|/\delta)}{\varepsilon^2}$, we have that $\mathbb{P}[\mathcal{E}_{y,a}^c] \leq \frac{\delta}{4|\mathcal{A}|}$. Therefore, the event $\mathcal{E} := \cap_{(y,a) \in \{0,1\} \times \mathcal{A}} \mathcal{E}_{y,a}$ happens with probability at least $1 - \frac{\delta}{2}$ by Union Bound. We also define $\mathcal{F} := \{N'_{y|a} \in (1 \pm \frac{\varepsilon}{12}) \frac{576 \log(8|\mathcal{A}|/\delta)}{\varepsilon^2 p_{y|a}}\}$, where $N'_{y|a}$ is the number of samples collected in Step 5 of the algorithm. Once again, by application of Lemma 4.13 and Union Bound, the event $\mathcal{F} := \cap_{(y,a) \in \{0,1\} \times \mathcal{A}} \mathcal{F}_{y,a}$ happens with probability at least $1 - \frac{\delta}{2}$. Therefore, $\mathbb{P}[\mathcal{E} \cap \mathcal{F}] \geq 1 - \delta$.

Under $\mathcal{E}$, the total number of online samples collected by Algorithm 8 is

$$\begin{aligned} N &:= \sum_{(y,a) \in \{0,1\} \times \mathcal{A}} N_{y|a} \leq \sum_{(y,a) \in \{0,1\} \times \mathcal{A}} \left(1 + \frac{\varepsilon}{12}\right) \frac{576 \log(8|\mathcal{A}|/\delta)}{\varepsilon^2 q_{y|a}} \\ &\leq \sum_{(y,a) \in \{0,1\} \times \mathcal{A}} \frac{13}{12} \cdot \frac{576 \log(8|\mathcal{A}|/\delta)}{\varepsilon^2 q_{y|a}} \quad (\varepsilon < 1) \\ &= \sum_{a \in \mathcal{A}} \sum_{y \in \{0,1\}} \frac{624 \log(8|\mathcal{A}|/\delta)}{\varepsilon^2 q_{y|a}}. \end{aligned}$$

Note that $N_{y|a}$ samples are collected by the algorithm under the conditional distribution $\mathbb{P}[Y \mid A = a]$. Therefore, the expected number of samples actually requested by the algorithm is $N_{y|a} \cdot \mathbb{P}[f = 0 \mid A = a]$. Similarly, the audit cost of $N_{y|a}$ samples is $N_{y|a}(c_{feat}\mathbb{P}[f = 0 \mid A = a] + c_{lab}\mathbb{P}[f = 0, Y = 0 \mid A = a])$. Therefore, the expected number of samples requested by the algorithm is at most

$$\sum_{y \in \{0,1\}} \sum_{a \in \mathcal{A}} N_{y|a} \cdot \mathbb{P}[f = 0 \mid A = a] \leq \sum_{a \in \mathcal{A}} \sum_{y \in \{0,1\}} \frac{624 \cdot \mathbb{P}[f = 0 \mid A = a] \log(8|\mathcal{A}|/\delta)}{\varepsilon^2 q_{y|a}}.$$

Similarly, the audit cost of Algorithm 8 can be bounded as

$$\begin{aligned} \text{cost} &= \sum_{y \in \{0,1\}} \sum_{a \in \mathcal{A}} N_{y|a} (c_{feat}\mathbb{P}[f = 0 \mid A = a] + c_{lab}\mathbb{P}[f = 0, Y = 0 \mid A = a]) \\ &\leq \sum_{a \in \mathcal{A}} \sum_{y \in \{0,1\}} \frac{624(c_{feat}\mathbb{P}[f = 0 \mid A = a] + c_{lab}\mathbb{P}[f = 0, Y = 0 \mid A = a]) \log(8|\mathcal{A}|/\delta)}{\varepsilon^2 q_{y|a}}. \end{aligned}$$

The correctness of Algorithm 8 can be shown in the same way as in the proof of Theorem 4.1. This is because under event $\mathcal{E} \cap \mathcal{F}$, we have $\widehat{q}_{y|a} = \frac{\tau}{N_{y|a}} = \frac{576 \log(8|\mathcal{A}|/\delta)}{\varepsilon^2 N_{y|a}}$. Hence, $\frac{q_{y|a}}{1 + \frac{\varepsilon}{12}} \leq \widehat{q}_{y|a} \leq \frac{q_{y|a}}{1 - \frac{\varepsilon}{12}}$. Similarly, $\frac{p_{y|a}}{1 + \frac{\varepsilon}{12}} \leq \widehat{p}_{y|a} \leq \frac{p_{y|a}}{1 - \frac{\varepsilon}{12}}$. Hence,

$$\frac{(1 - \frac{\varepsilon}{12})p_{y|a}}{(1 + \frac{\varepsilon}{12})q_{y|a}} \leq \frac{\widehat{p}_{y|a}}{\widehat{q}_{y|a}} \leq \frac{(1 + \frac{\varepsilon}{12})p_{y|a}}{(1 - \frac{\varepsilon}{12})q_{y|a}}. \quad (4.2)$$

Using Lemma 4.3 and the fact that $0 \leq \frac{p_{y|a}}{q_{y|a}} \leq 1$, we finally have,

$$\frac{p_{y|a}}{q_{y|a}} - \frac{\varepsilon}{6} \leq \frac{\widehat{p}_{y|a}}{\widehat{q}_{y|a}} \leq \frac{p_{y|a}}{q_{y|a}} + \frac{\varepsilon}{3}.$$

Since the above holds for all $(y, a) \in \{0, 1\} \times \mathcal{A}$ under $\mathcal{E} \cap \mathcal{F}$, we therefore have, for a particular $y \in \{0, 1\}$ and $a, a' \in \mathcal{A}$,

$$\frac{p_{y|a}}{q_{y|a}} - \frac{p_{y|a'}}{q_{y|a'}} - \frac{\varepsilon}{2} \leq \frac{\widehat{p}_{y|a}}{\widehat{q}_{y|a}} - \frac{\widehat{p}_{y|a'}}{\widehat{q}_{y|a'}} \leq \frac{p_{y|a}}{q_{y|a}} - \frac{p_{y|a'}}{q_{y|a'}} + \frac{\varepsilon}{2}.$$

From this, we can conclude that

$$\left| \frac{\widehat{p}_{y|a}}{\widehat{q}_{y|a}} - \frac{\widehat{p}_{y|a'}}{\widehat{q}_{y|a'}} \right| - \left| \frac{p_{y|a}}{q_{y|a}} - \frac{p_{y|a'}}{q_{y|a'}} \right| \leq \frac{\varepsilon}{2}. \quad (4.3)$$

Suppose the true hypothesis is **UNFAIR**. Let $y^* \in \{0, 1\}$ and $a_1, a_2 \in \mathcal{A}, a_1 \neq a_2$ be such that $|\mathbb{P}[f = 1 \mid Y = y^*, A = a_1] - \mathbb{P}[f = 1 \mid Y = y^*, A = a_2]| = \left| \frac{p_{y^*|a_1}}{q_{y^*|a_1}} - \frac{p_{y^*|a_2}}{q_{y^*|a_2}} \right| > \gamma + \varepsilon$. Then, we have,

$$\begin{aligned}
\widehat{\Delta} &\geq \left| \frac{\widehat{p}_{y^*|a_1}}{\widehat{q}_{y^*|a_1}} - \frac{\widehat{p}_{y^*|a_2}}{\widehat{q}_{y^*|a_2}} \right| && \text{(Definition of } \widehat{\Delta} \text{ (Step 9 of Algorithm 8))} \\
&\geq \left| \frac{p_{y^*|a_1}}{q_{y^*,a_1}} - \frac{p_{y^*|a_1}}{q_{y^*|a_2}} \right| - \frac{\varepsilon}{2} && \text{(Equation (4.7))} \\
&> \gamma + \varepsilon - \frac{\varepsilon}{2} && \left(\left| \frac{p_{y^*|a_1}}{q_{y^*|a_1}} - \frac{p_{y^*|a_2}}{q_{y^*|a_2}} \right| > \gamma + \varepsilon \right) \\
&= \gamma + \frac{\varepsilon}{2}.
\end{aligned}$$

The algorithm therefore outputs **UNFAIR** (Step 10). Hence, under the hypothesis **UNFAIR**, the algorithm's output is correct.

Next, suppose the true hypothesis is **FAIR**. Then, we have for all $y \in \{0, 1\}$ and $a, a' \in \mathcal{A}$,

$$|\mathbb{P}[f = 1 \mid Y = y, A = a] - \mathbb{P}[f = 1 \mid Y = y, A = a']| = \left| \frac{p_{y|a}}{q_{y|a}} - \frac{p_{y|a'}}{q_{y|a'}} \right| \leq \gamma.$$

Hence, by Equation (4.3), we can conclude that for all $y \in \{0, 1\}$ and $a, a' \in \mathcal{A}$,

$$\begin{aligned}
\left| \frac{\widehat{p}_{y|a}}{\widehat{q}_{y|a}} - \frac{\widehat{p}_{y|a'}}{\widehat{q}_{y|a'}} \right| &\leq \left| \frac{p_{y|a}}{q_{y|a}} - \frac{p_{y|a'}}{q_{y|a'}} \right| + \frac{\varepsilon}{2} \\
&\leq \gamma + \frac{\varepsilon}{2}. && \left(\left| \frac{p_{y|a}}{q_{y|a}} - \frac{p_{y|a'}}{q_{y|a'}} \right| \leq \gamma \right)
\end{aligned}$$

In particular, we have $\widehat{\Delta} = \max_{y \in \{0,1\}} \max_{a,a' \in \mathcal{A}} \left| \frac{\widehat{p}_{y|a}}{\widehat{q}_{y|a}} - \frac{\widehat{p}_{y|a'}}{\widehat{q}_{y|a'}} \right| \leq \frac{\varepsilon}{2}$. Hence, the algorithm outputs **FAIR** implying that the algorithm's output is correct.

Since the output of the algorithm is correct under both the hypotheses, one can conclude the output of the algorithm is always correct. Note that we assumed event $\mathcal{E} \cap \mathcal{F}$ to hold, under which the correctness of our algorithm was derived. Therefore, our algorithm is correct with probability at least $1 - \delta$. The theorem stands proved. $\square$

4.3.3 Lower Bound for the Blackbox Model: Proof of Theorem 4.3

In this section, we describe the construction of the lower bound instance and prove Theorem 4.3. In order to do so, we first formalize the sequential interaction between the auditor and the data.

Canonical Partial Feedback Model. Let us consider the sequential interaction of an algorithm π with a partial feedback system given $A = a$, that is, the algorithm only observes

samples from the conditional distribution $A = a$. At round t , the environment first generates an iid sample (Y_t, f_t) , where Y_t is the true label and f_t is the classifiers decision. First, the f_t is visible to the algorithm. If $f_t = 1$, the algorithm takes no action (denote by $x_t = \odot$) on the environment (*i.e.*, only internal computations). Then $Z_t = Y_t$ is revealed to the algorithm. The round ends and the algorithm can update its policy from π_t to π_{t+1} . On the other hand, if $f_t = 0$, the algorithm must decide whether to observe Y_t (action $x_t = 1$) or not (action $x_t = 0$). If $x_t = 1$, the label Y_t is revealed to the algorithm, *i.e.*, $Z_t = Y_t$, else $Z_t = \star$. After this, the algorithm updates its policy to π_{t+1} .

This gives rise to a natural filtration $\mathcal{F}_t = \sigma(f_1, x_1, Z_1, f_2, x_2, Z_2, \dots, f_t, x_t, Z_t)$. Let us consider the probability of this random vector $\{f_1, \dots, Z_t\}$ under some hypothesis M generating the data.

$$\begin{aligned} & \mathbb{P}_{M\pi}(f_1, x_1, Z_1, \dots, f_t, x_t, Z_t) \\ &= \prod_{s=1}^t \mathbb{P}_{M\pi}[Z_s \mid f_1, \dots, x_s] \cdot \mathbb{P}_{M\pi}[x_s \mid f_1, x_1, Z_1, \dots, f_s] \cdot \mathbb{P}_{M\pi}[f_s \mid f_1, \dots, Z_{s-1}] \\ &= \prod_{s=1}^t \mathbb{P}_{M\pi}[Z_s \mid f_s, x_s] \cdot \pi_s(x_s \mid f_1, x_1, Z_1, \dots, f_s) \cdot \mathbb{P}_M[f_s] \end{aligned}$$

Here, we have used the fact that the samples f_t 's are iid and Z_t is conditionally independent of the past given f_t and x_t , so that $\mathbb{P}_{M\pi}[Z_s \mid f_1, \dots, x_s] = \mathbb{P}_{M\pi}[Z_s \mid f_s, x_s]$ and $\mathbb{P}_{M\pi}[f_s \mid f_1, \dots, Z_{s-1}] = \mathbb{P}_M[f_s]$.

Now suppose there is an alternate hypothesis that also generates the data such that $\mathbb{P}_N[f_t] = \mathbb{P}_M[f_t]$. Then, we have

$$\frac{\mathbb{P}_M(f_1, x_1, Z_1, \dots, f_t, x_t, Z_t)}{\mathbb{P}_N(f_1, x_1, Z_1, \dots, f_t, x_t, Z_t)} = \prod_{s=1}^t \frac{\mathbb{P}_M[Z_s \mid f_s, x_s] \cdot \pi_s(x_s \mid f_1, \dots, f_s) \cdot \mathbb{P}_M[f_s]}{\mathbb{P}_N[Z_s \mid f_s, x_s] \cdot \pi_s(x_s \mid f_1, \dots, f_s) \cdot \mathbb{P}_N[f_s]} = \prod_{s=1}^t \frac{\mathbb{P}_M[Z_s \mid f_s, x_s]}{\mathbb{P}_N[Z_s \mid f_s, x_s]} \quad (4.4)$$

Lemma 4.1 (Divergence Decomposition for Canonical Partial Feedback Model) *Suppose M and N are two hypotheses for the canonical partial feedback model defined above and let π be an algorithm interacting with this feedback model till round $t \in \mathbb{N}$. Let $M\pi$ and $N\pi$ denote the probability measures under M and N respectively for the random vector $(f_1, x_1, Z_1, \dots, f_t, x_t, Z_t)$. If it holds that $\mathbb{P}_N[f_t] = \mathbb{P}_M[f_t]$, then, with $V_t = (f_1, x_1, Z_t, \dots, Z_t)$*

$$D_{KL}(\mathbb{P}_{M\pi}[V_t] \parallel \mathbb{P}_{N\pi}[V_t]) = t \cdot \mathbb{P}_M[f = 1] \cdot D_{KL}(\mathbb{P}_M[Y \mid f = 1] \parallel \mathbb{P}_N[Y \mid f = 1])$$

$$+ \mathbb{E}_{M\pi}[L_t] \cdot D_{KL}(\mathbb{P}_{M\pi}[Y \mid f = 0] \parallel \mathbb{P}_{N\pi}[Y \mid f = 0]),$$

where $L_t := \sum_{s=1}^t \mathbb{1}\{f_s = 0, x_s = 1\}$.

Proof: We have that

$$\begin{aligned}
& D_{KL}(\mathbb{P}_{M\pi}[V_t] \parallel \mathbb{P}_{N\pi}[V_t]) \\
&= \mathbb{E}_{M\pi} \left[\log \left(\frac{\mathbb{P}_M(f_1, x_1, Z_1, \dots, f_t, x_t, Z_t)}{\mathbb{P}_N(f_1, x_1, Z_1, \dots, f_t, x_t, Z_t)} \right) \right] \\
&= \mathbb{E}_{M\pi} \left[\sum_{s=1}^t \log \left(\frac{\mathbb{P}_M[Z_s \mid f_s, x_s]}{\mathbb{P}_N[Z_s \mid f_s, x_s]} \right) \right] \quad (\text{via Equation (4.4) since } \mathbb{P}_M[f] = \mathbb{P}_M[f]) \\
&= \sum_{s=1}^t \mathbb{E}_{M\pi} \left[\mathbb{E}_{M\pi} \left[\log \left(\frac{\mathbb{P}_M[Z_s \mid f_s, x_s]}{\mathbb{P}_N[Z_s \mid f_s, x_s]} \right) \middle| (f_s, x_s) \right] \right] \\
&= \sum_{s=1}^t \mathbb{E}_{M\pi} [D_{KL}(\mathbb{P}_{M\pi}[Z_s \mid f_s, x_s] \parallel \mathbb{P}_{N\pi}[Z_s \mid f_s, x_s])] \\
&= \sum_{s=1}^t \mathbb{E}_{M\pi} [\mathbb{1}\{f_s = 1, x_s = \odot\} \cdot D_{KL}(\mathbb{P}_{M\pi}[Z_s \mid f_s = 1, x_s = \odot] \parallel \mathbb{P}_{N\pi}[Z_s \mid f_s = 1, x_s = \odot]) \\
&\quad + \mathbb{1}\{f_s = 0, x_s = 0\} \cdot D_{KL}(\mathbb{P}_{M\pi}[Z_s \mid f_s = 0, x_s = 0] \parallel \mathbb{P}_{N\pi}[Z_s \mid f_s = 0, x_s = 0]) \\
&\quad + \mathbb{1}\{f_s = 0, x_s = 1\} \cdot D_{KL}(\mathbb{P}_{M\pi}[Z_s \mid f_s = 0, x_s = 1] \parallel \mathbb{P}_{N\pi}[Z_s \mid f_s = 0, x_s = 1])] \\
&= \sum_{s=1}^t \mathbb{E}_{M\pi} [\mathbb{1}\{f_s = 1\} \cdot D_{KL}(\mathbb{P}_{M\pi}[Y_s \mid f_s = 1] \parallel \mathbb{P}_{N\pi}[Y_s \mid f_s = 1]) \\
&\quad + \mathbb{1}\{f_s = 0, x_s = 0\} \cdot D_{KL}(\mathbb{P}_{M\pi}[\star \mid f_s = 0, x_s = 0] \parallel \mathbb{P}_{N\pi}[\star \mid f_s = 0, x_s = 0]) \\
&\quad + \mathbb{1}\{f_s = 0, x_s = 1\} \cdot D_{KL}(\mathbb{P}_{M\pi}[Y_s \mid f_s = 0, x_s = 1] \parallel \mathbb{P}_{N\pi}[Y_s \mid f_s = 0, x_s = 1])] \\
&= \sum_{s=1}^t \mathbb{E}_M [\mathbb{1}\{f = 1\}] \cdot D_{KL}(\mathbb{P}_M[Y \mid f = 1] \parallel \mathbb{P}_N[Y \mid f = 1]) \\
&\quad + 0 + \mathbb{E}_{M\pi} \left[\sum_{s=1}^t \mathbb{1}\{f_s = 0, x_s = 1\} \right] \cdot D_{KL}(\mathbb{P}_M[Y \mid f = 0] \parallel \mathbb{P}_N[Y \mid f = 0]) \\
&\hspace{20em} ((f_s, Y_s) \text{ are iid samples}) \\
&= t \cdot \mathbb{P}_M[f = 1] \cdot D_{KL}(\mathbb{P}_M[Y \mid f = 1] \parallel \mathbb{P}_N[Y \mid f = 1]) \\
&\quad + \mathbb{E}_{M\pi}[L_t] \cdot D_{KL}(\mathbb{P}_M[Y \mid f = 0] \parallel \mathbb{P}_N[Y \mid f = 0])
\end{aligned}$$

□

Lemma 4.2 (Divergence Decomposition with Stopping Time) *With notations same as in Lemma 4.1, let π be an algorithm that interacting with the model till a $(\mathcal{F}_t)_{t=1}^\infty$ -measurable stopping time τ . Further, let $\mathbb{E}_{M\pi}[\tau] < \infty$. Let $\mathcal{O}$ be $\mathcal{F}_\tau$ -measurable random variable. Then,*

$$D_{KL}(\mathbb{P}_{M\pi}[\mathcal{O}] \parallel \mathbb{P}_{N\pi}[\mathcal{O}]) \leq \mathbb{E}_{M\pi}[\tau] \cdot \mathbb{P}_M[f = 1] \cdot D_{KL}(\mathbb{P}_M[Y \mid f = 1] \parallel \mathbb{P}_N[Y \mid f = 1]) \\ + \mathbb{E}_{M\pi}[L_\tau] \cdot D_{KL}(\mathbb{P}_{M\pi}[Y \mid f = 0] \parallel \mathbb{P}_{N\pi}[Y \mid f = 0])$$

where L_τ is the stopped version of L_t .

Proof: The proof can be obtained by combining Lemma 4.1 with data processing inequality and equation relating KL-divergences between push-forward measures. For reference see Proof of Lemma 2.6 in [85]. $\square$

Now we restate Theorem 4.3 for the sample complexity lower bound in blackbox model and present its proof.

Theorem 4.3 (Lower Bound) *Under the blackbox setting, for any $\varepsilon \in (0, 1/8)$ and $\delta \in (0, 1)$, there exists a family of instances characterized by $\gamma = 0$, $q_{y|a}$, and $\mathbb{P}[f = 0 \mid A = a]$ for $y \in \{0, 1\}$ and $a \in \mathcal{A}$, even with $|\mathcal{A}| = 2$, such that any algorithm with finite expected running time that succeeds in the (ε, δ) -fairness test must suffer at least $\tilde{\Omega}\left(\sum_{a \in \mathcal{A}} \sum_{y \in \{0,1\}} \frac{c_{feat}\mathbb{P}[f=0|A=a] + c_{lab}\mathbb{P}[f=0, Y=0|A=a]}{q_{y|a}\varepsilon^2}\right)$ cost in expectation.*

Proof: We will construct two instances with $\mathcal{A} = \{0, 1\}$ such that they are close, yet in one case the classifier is fair and in the other, it is unfair. However since the instances are close, any algorithm using less than a certain number of samples will not be able to distinguish between the two.

Let $p, q \in (0, 1)$ be constants to be decided later. We let $\mathbb{P}[Y = 1 \mid A = a] = q$. The classifier f is given as follows. Under the FAIR hypothesis, given $Y = 1$ and $A = a$, $f \sim \text{Ber}(\frac{1}{2})$ and given $Y = 0$ and $A = a$, $f \sim \text{Ber}(\frac{p}{2(1-q)})$ for both $a \in \{0, 1\}$. On the other hand, in the UNFAIR hypothesis, the distribution of f is as specified in Table 4.1.

Using the description of f above, we can generate Table 4.2 that shows the values of various quantities of interest.

From Table 4.2, it can be seen that given $A = a$, $\mathbb{P}_{\text{FAIR}}[f = 1 \mid A = a] = \mathbb{P}_{\text{UNFAIR}}[f = 1 \mid A = a]$. Therefore, $\mathbb{P}_{\text{FAIR}}[f = 0 \mid A = a] = \mathbb{P}_{\text{UNFAIR}}[f = 0 \mid A = a] = 1 - \frac{p+q}{2}$ for all $a \in \mathcal{A}$. Moreover, the $\mathbb{P}_{\text{FAIR}}[Y = 1 \mid f = 1, A = a] = \mathbb{P}_{\text{UNFAIR}}[Y = 1 \mid f = 1, A = a]$. Hence, $D_{KL}(\mathbb{P}_{\text{FAIR}}[Y \mid f = 1, A = a] \parallel \mathbb{P}_{\text{UNFAIR}}[Y \mid f = 1, A = a]) = 0$. Finally, $\mathbb{P}_{\text{FAIR}}[Y = 1 \mid f = 0, A = 1] = \frac{q}{2(1-\frac{p+q}{2})}$ and $\mathbb{P}_{\text{UNFAIR}}[Y = 1 \mid f = 0, A = 1] = \frac{q(1+8\varepsilon)}{2(1-\frac{p+q}{2})}$.

Given	Distribution of f
$Y = 1, A = 0$	$f \sim \text{Ber}\left(\frac{1}{2}\right)$
$Y = 1, A = 1$	$f \sim \text{Ber}\left(\frac{1}{2(1+4\varepsilon)}\right)$
$Y = 0, A = 0$	$f \sim \text{Ber}\left(\frac{p}{2(1-q)}\right)$
$Y = 0, A = 1$	$f \sim \text{Ber}\left(\frac{p}{2(1-q-4\varepsilon q)}\right)$

Table 4.1: Distribution of f under **UNFAIR** hypothesis.

Probability	FAIR (same for both a)	UNFAIR	
		$a = 0$	$a = 1$
$\mathbb{P}[f = 1, Y = 1 \mid A = a]$	$\frac{q}{2}$	$\frac{q}{2}$	$\frac{q}{2}$
$\mathbb{P}[f = 1, Y = 0 \mid A = a]$	$\frac{p}{2}$	$\frac{p}{2}$	$\frac{p}{2}$
$\mathbb{P}[f = 0, Y = 1 \mid A = a]$	$\frac{q}{2}$	$\frac{q}{2}$	$\frac{q}{2}(1 + 8\varepsilon)$
$\mathbb{P}[f = 0, Y = 0 \mid A = a]$	$1 - q - \frac{p}{2}$	$1 - q - \frac{p}{2}$	$1 - q - \frac{p}{2} - 4\varepsilon q$

Table 4.2: Probability values of interest under the constructed f .

We set $\varepsilon \in (0, \frac{1}{8})$, $p \in (0, \frac{1}{3})$ and $q \in (0, \frac{1}{3})$ so that $q(1 + 4\varepsilon) \leq 1$. It is easy to verify that the equalized odds difference under the **FAIR** hypothesis is 0 (intuitively, the probabilities of the classifier is the same for both the groups $a \in \{0, 1\}$). The equalized odds disparity gap of the **UNFAIR** hypothesis is more than ε as can be seen below.

$$\begin{aligned}
& \mathbb{P}[f = 1 \mid Y = 1, A = 0] - \mathbb{P}[f = 1 \mid Y = 1, A = 1] \\
&= \frac{\mathbb{P}[f = 1, Y = 1 \mid A = 0]}{\mathbb{P}[Y = 1 \mid A = 0]} - \frac{\mathbb{P}[f = 1, Y = 1 \mid A = 1]}{\mathbb{P}[Y = 1 \mid A = 1]} \\
&= \frac{q/2}{q} - \frac{q/2}{q(1 + 4\varepsilon)} \\
&= \frac{1}{2} \left(1 - \frac{1}{1 + 4\varepsilon} \right) = \frac{2\varepsilon}{1 + 4\varepsilon} > \varepsilon \quad (\varepsilon < \frac{1}{4})
\end{aligned}$$

Next we lower bound the number of samples required by any algorithm to be able to conclude which of the hypothesis is true. Suppose the auditing algorithm π is informed that the classifier is biased only for $a = 1$ group. Then, it makes sense for the algorithm to sample only from the distribution conditioned on $A = 1$.

Consider the canonical partial feedback model with hypotheses $M = \text{FAIR}$ and $N = \text{UNFAIR}$ when the distribution is conditioned on $A = 1$.

Let π interact with the model till some stopping time τ such that $\mathbb{E}_{\text{FAIR},\pi}[\tau] < \infty$ and $\mathbb{E}_{\text{UNFAIR},\pi}[\tau] < \infty$. Here, $\mathbb{P}_{\text{FAIR},\pi}$ and $\mathbb{P}_{\text{UNFAIR},\pi}$ denote distribution of outcome generated by interaction of π with the distributions $\mathbb{P}_{\text{FAIR}}$ and $\mathbb{P}_{\text{UNFAIR}}$ respectively. Let $\text{ALG} \in \{\text{FAIR}, \text{UNFAIR}\}$ be the output of the algorithm. Moreover, let $\mathbb{P}_{\text{FAIR},\pi}[\text{ALG} = \text{UNFAIR}] < \delta$ and $\mathbb{P}_{\text{UNFAIR},\pi}[\text{ALG} = \text{FAIR}] < \delta$, *i.e.*, the algorithm π succeeds in the (ε, δ) -fairness audit. Then, we have:

$$\begin{aligned}
2\delta &> \mathbb{P}_{\text{FAIR},\pi}[\text{ALG} = \text{UNFAIR}] + \mathbb{P}_{\text{UNFAIR},\pi}[\text{ALG} = \text{FAIR}] \\
&\geq \frac{1}{2} \exp(-D_{KL}(\mathbb{P}_{\text{FAIR},\pi}[\text{ALG}] \parallel \mathbb{P}_{\text{UNFAIR},\pi}[\text{ALG}])) \quad (\text{Lemma 4.4}) \\
&\geq \frac{1}{2} \exp(-\mathbb{E}_{\text{FAIR}}[L_\tau] \cdot D_{KL}(\mathbb{P}_{\text{FAIR}}[Y \mid f = 0, A = 1] \parallel \mathbb{P}_{\text{UNFAIR}}[Y \mid f = 0, A = 1])) \\
&\quad (\text{Lemma 4.2, } D_{KL}(\mathbb{P}_{\text{FAIR}}[Y \mid f = 1, A = a] \parallel \mathbb{P}_{\text{UNFAIR}}[Y \mid f = 1, A = a]) = 0) \\
&= \frac{1}{2} \exp\left(-\mathbb{E}_{\text{FAIR}}[L_\tau] \cdot D_{KL}\left(\text{Ber}\left(\frac{q}{2(1-\frac{p+q}{2})}\right) \parallel \text{Ber}\left(\frac{q(1+8\varepsilon)}{2(1-\frac{p+q}{2})}\right)\right)\right) \\
&\geq \frac{1}{2} \exp\left(-64 \cdot \mathbb{E}_{\text{FAIR}}[L_\tau] \cdot \varepsilon^2 \cdot \frac{q}{2(1-\frac{p+q}{2})}\right) \\
&\quad (\text{Lemma 4.5, } \frac{q}{2-p-q} \leq \frac{1}{3} \text{ and } \frac{8q\varepsilon}{2-p-q} \leq \frac{1}{2} \text{ for } p, q \in (0, \frac{1}{3}) \text{ and } \varepsilon \in (0, \frac{1}{4}))
\end{aligned}$$

Therefore, $\mathbb{E}_{\text{FAIR}}[L_\tau] > \frac{(1-\frac{p+q}{2})\log(1/4\delta)}{32q\varepsilon^2}$. Moreover, since Lemma 4.4 is symmetric, we also have $\mathbb{E}_{\text{UNFAIR}}[L_\tau] > \frac{(1-\frac{p+q}{2})\log(1/4\delta)}{64q\varepsilon^2}$ using Lemma 4.5. Hence, $\min\{\mathbb{E}_{\text{FAIR}}[L_\tau], \mathbb{E}_{\text{UNFAIR}}[L_\tau]\} > \frac{(1-\frac{p+q}{2})\log(1/4\delta)}{64q\varepsilon^2}$. Therefore, under both the hypothesis, the algorithm must request $\Omega\left(\frac{\mathbb{P}[f=0|A=1]\log(1/\delta)}{\mathbb{P}[Y=1|A=1]\varepsilon^2}\right)$ true labels in expectation to be able to output the correct hypothesis. Further, each requested true label is associated with a cost of $c_{\text{feat}} + c_{\text{lab}}\mathbb{P}[Y = 0 \mid f = 0]$. Thus, the total cost is at least $\Omega\left(\frac{(c_{\text{feat}} + c_{\text{lab}}\mathbb{P}[Y=0|f=0]) \cdot \mathbb{P}[f=0|A=1]\log(1/\delta)}{\mathbb{P}[Y=1|A=1]\varepsilon^2}\right) = \Omega\left(\frac{(c_{\text{feat}}\mathbb{P}[f=0|A=1] + c_{\text{lab}}\mathbb{P}[f=0, Y=0|A=a])\log(1/\delta)}{\varepsilon^2}\right)$.

Repeating this argument over all combinations of $y \in \{0, 1\}$ and $a \in \mathcal{A}$ concludes the proof of the theorem. $\square$

4.4 Fairness Audit under Mixture Model

Recall that in the mixture model, given $A = a$, the true label is sampled according to $Y \sim \text{Ber}(q_{1|a})$, and given both $A = a$ and $Y = y$, the feature of the individual is sampled according to $X \sim \mathcal{E}_{\theta_{y,a}^*}$ (see Section 4.1 & 4.2). Also, we denote $\mathbf{c} := (L, \beta, \kappa, \alpha)$, which are some known constants of the problem. Now we describe **Exp-Audit** (Algorithm 9) solving fairness audit under partial feedback in this model with a significantly lower audit cost than the blackbox

Algorithm 9: Exp-Audit: Audit for Mixture Model

Input: $\varepsilon, \delta \in (0, 1)$, past database D , classifier f , $\mathbf{c} = (L, \beta, \kappa, \alpha)$, parameter set Θ .

- 1 Set $\varepsilon' = \frac{1}{10}$, $\tau' = \frac{4 \log(14|A|/\delta)}{\varepsilon'^2}$, $\tau = \frac{576 \log(14|A|/\delta)}{\varepsilon^2}$.
- 2 **for** $a \in \mathcal{A}$ **do**
- 3 **for** $y \in \{0, 1\}$ **do**
- 4 Set $\hat{\theta}_{y,a} = \text{TruncEst}(D_{y,a}, f, \frac{\varepsilon}{2\beta}, \frac{\delta}{14|A|}, \mathbf{c}, \Theta)$.
- 5 $N' \leftarrow \text{Online}(\tau', y, a)$, $N \leftarrow \text{Past}(D, \tau, y, a)$.
- 6 Set $\tilde{q}_{y|a} = \frac{\tau'}{N'}$ and $\hat{p}_{y|a} = \frac{\tau}{N}$.
- 7 Create MAP oracle: $M_a(x) = \underset{y \in \{0,1\}}{\text{argmax}} \tilde{q}_{y|a} \mathcal{E}_{\hat{\theta}_{y,a}}(x)$.
- 8 Let $\tilde{q}_{m,a} = \tilde{q}_{1|a} \wedge \tilde{q}_{0|a}$, $R_0 = R_1 = 0$.
- 9 Observe $R = \frac{3430 \log(12|A|/\delta)}{\tilde{q}_{m,a} \varepsilon^2}$ features X with $A=a$ and set $R_{M_a(X)} \leftarrow R_{M_a(X)} + 1$ for each X . Set $\hat{q}_{y|a} = \frac{R_y}{R}$.
- 10 $\hat{\Delta} = \max_{y \in \{0,1\}, a, a' \in \mathcal{A}} \left| \frac{\hat{p}_{y|a}}{\hat{q}_{y|a}} - \frac{\hat{p}_{y|a'}}{\hat{q}_{y|a'}} \right|$.
- 11 **if** $\hat{\Delta} > \gamma + \frac{\varepsilon}{2}$ **then**
- 12 return UNFAIR.
- 13 **else**
- 14 return FAIR.

case.

Exp-Audit first splits the past database of positively classified individuals into subsets $D_{y,a}$, $a \in \mathcal{A}$, $y \in \{0, 1\}$, so that $D_{y,a}$ only contains individuals with $A = a$ and $Y = y$. Note that samples in $D_{y,a}$ are iid samples of $X \sim \mathcal{E}_{\theta_{y,a}^*}$ truncated by f , *i.e.*, $D_{y,a}$ contains only those samples from $\mathcal{E}_{\theta_{y,a}^*}$ for whom $f = 1$. On every subset $D_{y,a}$, **Exp-Audit** applies the subroutine $\text{TruncEst}(D_{y,a}, f, \frac{\varepsilon}{2\beta}, \frac{\delta}{14|A|}, \mathbf{c}, \Theta)$ (adaptation of [54] and polynomial time, see Appendix 4.7.2). From $D_{y,a}$ this subroutine learns an estimate $\hat{\theta}_{y,a}$ of $\theta_{y,a}^*$ such that $\|\hat{\theta}_{y,a} - \theta_{y,a}^*\| \leq \frac{\varepsilon}{2\beta}$ with probability at least $1 - \frac{\delta}{14|A|}$ if $|D_{y,a}| \geq \tilde{\Omega}(\frac{d}{\varepsilon^2})$. Note that the condition $|D_{y,a}| \geq \tilde{\Omega}(\frac{d}{\varepsilon^2})$ can be assumed true without loss of generality because of Assumption 3(i) which states $\mathbb{P}[f = 1 \mid Y = y, A = a] > 0$, ensuring that the condition is satisfied in finite time. Crucially, there is no cost for waiting for samples which being iid are unaffected by delay.

After obtaining $\hat{\theta}_{y,a}$, the algorithm uses the rejection-sampling based approach (line 5) similar to **RS-Audit** to obtain $\hat{p}_{y|a} \in (1 \pm \frac{\varepsilon}{12})p_{y|a}$ and $\tilde{q}_{y|a} \in (1 \pm \varepsilon')q_{y|a}$, where $\varepsilon' = \frac{1}{10}$ is a constant. Note that the cost incurred in estimating $\tilde{q}_{y,a}$ is therefore $\tilde{O}(1)$. This is a crucial difference of **Exp-Audit** with **RS-Audit** which estimates $q_{y|a}$ upto a multiplicative $1 \pm \varepsilon$ error incurring $\tilde{O}(\frac{1}{\varepsilon^2})$ cost.

Then, for every $a \in \mathcal{A}$, **Exp-Audit** constructs a maximum *a posteriori* (MAP) oracle for the mixture model $M_a : \mathcal{X} \rightarrow \{0, 1\}$, $M_a(X) = \operatorname{argmax}_{y \in \{0, 1\}} \tilde{q}_{y|a} \mathcal{E}_{\hat{\theta}_{y,a}}(X)$. In other words, for $A = a$ and feature X , the MAP oracle returns which is the most likely true label Y . Next, for every $a \in \mathcal{A}$, with $\tilde{q}_{m,a} = \tilde{q}_{0|a} \wedge \tilde{q}_{1|a}$, the algorithm observes only the features X for $R = \tilde{O}(\frac{1}{\tilde{q}_{m,a}\varepsilon^2})$ individuals from group $A = a$ and uses the MAP oracle M_a to assign a proxy label to each X . With R_y as the number of individuals with proxy label y , **Exp-Audit** computes $\hat{q}_{y|a} = \frac{R_y}{R}$. Under the separation condition (4.1), we have $\hat{q}_{y|a} \in (1 \pm \frac{\varepsilon}{12})q_{y|a}$ (see Section 4.7.1). Using $\hat{p}_{y|a}$ and $\hat{q}_{y|a}$, **Exp-Audit** sets $\hat{\Delta} = \max_{y \in \{0, 1\}, a, a' \in \mathcal{A}} \left| \frac{\hat{p}_{y|a}}{\hat{q}_{y|a}} - \frac{\hat{p}_{y|a'}}{\hat{q}_{y|a'}} \right|$ and returns **UNFAIR** if $\hat{\Delta} > \gamma + \frac{\varepsilon}{2}$, else **FAIR**.

Remark 8 *The cost incurred by **Exp-Audit** is significantly lower than that of **RS-Audit** (see Theorems 4.2 and 4.4) because the term involved with c_{lab} , the cost of requesting a negative label ($Y = 0$), is a problem-dependent constant and independent of ε for **Exp-Audit**. Note that in most applications, $c_{\text{lab}} \gg c_{\text{feat}}$. This motivates the study of scenarios where the audit cost is lower than the black-box case, as an interesting future direction.*

Remark 9 *The design of **Exp-Audit** has two novel aspects. The first is to view the past database under partial feedback as a collection of truncated samples which allows us to learn the distribution parameters using techniques for learning from truncated samples [54, 26] using past data. The second is to construct approximate MAP oracles as a proxy for true labels under a suitable separation condition. For this, we generalize the result of [61] from spherical Gaussians to general exponential family (may be of independent interest), leading to improvement in the audit cost.*

Remark 10 *In **Exp-Audit**, besides solving the audit problem, we also recover the parameters $q_{y|a}$ and $\theta_{y,a}^*$ within ε error which can be used to obtain a classifier with error rate at most $O(\varepsilon)$ more than that of the Bayes optimal classifier. Moreover, fairness-accuracy tradeoff can be studied by expressing the Bayes error and unfairness metrics using learnt distribution parameters with high accuracy [23, 27, 100].*

Remark 11 (Separation Condition) *The separation condition (4.1) assumed to obtain Theorem 4.4 is significantly milder than what is required to obtain good clustering of a mixture of exponential family. We require that the separation is $\Omega(\sqrt{\log(1/\varepsilon)})$ whereas that required for good clustering is $\Omega(\frac{1}{\varepsilon})$ (see [61] Table 1). Moreover, this separation is the minimum required for the Bayes optimal classifier to have low ($O(\varepsilon)$) error rate (see [61], Remark 8). We also highlight that the condition (4.1) can be tested after the parameters $\hat{\theta}_{y,a}$ and $\tilde{q}_{y|a}$ are learnt (lines*

2-7 in Algorithm 9), and if the condition is violated, the auditor can switch to Algorithm 8. In our experiments on the Adult Income dataset, we do not enforce this condition but we observe that Algorithm 9 has significantly lower cost while being able to perform the fairness audit successfully.

4.4.1 Proof of Theorem 4.4

In this section, we provide the proof of Theorem 4.4 which provides the bound on the audit cost of **Exp-Audit** (Algorithm 9). We first restate the theorem and then present its proof.

Theorem 4.4 *Given $\gamma \geq 0$, $\varepsilon, \delta \in (0, 1)$, under the mixture model and assumptions 3 and 4, **Exp-Audit** (Algorithm 9) succeeds in $(\gamma, \varepsilon, \delta)$ -fairness audit with $\text{cost} \leq \tilde{O}\left(\sum_{a \in \mathcal{A}} \left(\frac{c_{\text{feat}} \mathbb{P}[f=0|A=a]}{q_{m,a} \varepsilon^2} + \sum_{y \in \{0,1\}} \frac{c_{lab} \mathbb{P}[f=0, Y=0|A=a]}{q_{y|a}}\right)\right)$.*

Proof: Firstly, for $\hat{\theta}_{y,a} = \text{TruncEst}(D_{y,a}, f, \frac{\varepsilon}{2\beta}, \frac{\delta}{14|\mathcal{A}|})$, we have $\|\hat{\theta}_{y,a} - \theta_{y,a}^*\| \leq \frac{\varepsilon}{2\beta}$ with probability at least $1 - \frac{\delta}{14|\mathcal{A}|}$ due to the guarantees of the **TruncEst** subroutine as stated in Theorem 4.5 and Remark 13. Therefore, the event $\xi_1 = \{\forall y \in \{0, 1\}, a \in \mathcal{A}, \|\hat{\theta}_{y,a} - \theta_{y,a}^*\| \leq \frac{\varepsilon}{2\beta}\}$ happens with probability at least $1 - \frac{2\delta}{14}$ via Union Bound.

Next, we have by arguments essentially similar to proof of Theorem 4.2, the following event ξ_2 holds with probability at least $1 - \frac{8\delta}{14}$: for all $y \in \{0, 1\}$ and $a \in \mathcal{A}$, $(1 - \frac{\varepsilon}{12})p_{y|a} \leq \hat{p}_{y|a} \leq (1 + \frac{\varepsilon}{12})p_{y|a}$ and $(1 - \varepsilon')q_{y|a} \leq \tilde{q}_{y|a} \leq (1 + \varepsilon')q_{y|a}$.

Therefore, $\xi_1 \cap \xi_2$ happens with probability at least $1 - \frac{10\delta}{14}$. Hereon, we will assume that $\xi_1 \cap \xi_2$ holds.

Let us now analyze the MAP oracle for a fixed group label $A = a$. Under event $\xi_1 \cap \xi_2$, we have $\|\hat{\theta}_{1,a} - \hat{\theta}_{0,a}\| \geq \|\theta_{1,a}^* - \theta_{0,a}^*\| - 2 \cdot \frac{\varepsilon}{2\beta}$ by triangle inequality. Therefore, by separation condition (4.1), we have,

$$\|\hat{\theta}_{1,a} - \hat{\theta}_{0,a}\| \geq \sqrt{\frac{3(16(L \vee \beta)\beta + \lambda)}{4\kappa\beta^2} \log\left(\frac{10q_{M,a}}{q_{m,a}\varepsilon}\right)} - \frac{\varepsilon}{\beta} \geq \sqrt{\frac{3(16(L \vee \beta)\beta + \lambda)}{8\kappa\beta^2} \log\left(\frac{10q_{M,a}}{q_{m,a}\varepsilon}\right)},$$

where we have used the fact that $L, \beta \geq 1$ and $\kappa \leq 1$ and then applied Lemma 4.7 (1). Hence under $\xi_1 \cap \xi_2$, we have the following for all $a \in \mathcal{A}$:

1. $\|\hat{\theta}_{y,a} - \theta_{y,a}^*\| \leq \frac{\varepsilon}{2\beta}$ for $y \in \{0, 1\}$,
2. $\|\hat{\theta}_{1,a} - \hat{\theta}_{0,a}\| \geq \sqrt{\frac{3(16(L \vee \beta)\beta + \lambda)}{16\kappa\beta^2} \log\left(\frac{10q_{M,a}}{q_{m,a}\varepsilon}\right)}$, and

3. $\mathcal{B}\left(\frac{\hat{\theta}_{0,a} + \hat{\theta}_{1,a}}{2}, \frac{1}{2\beta}\right) \subset \Theta$ by Lemma 4.10.

Hence, we can now apply Lemma 4.6 with $\theta^* = \theta_{y,a}^*$, $\theta_1 = \hat{\theta}_{y,a}$, $\theta_2 = \hat{\theta}_{1-y,a}$, $\gamma = \frac{10q_{M,a}}{q_{m,a}} \geq e^2$ and $t = \frac{\tilde{q}_{1-y|a}}{\tilde{q}_{y|a}}$. Note that all the conditions on W are satisfied by assumption. Suppose $X \sim \mathcal{E}_{\theta_{y,a}^*}$. Then, we have the following:

$$\begin{aligned}
\mathbb{P}_X \left[\tilde{q}_{y|a} \mathcal{E}_{\hat{\theta}_{y,a}}(X) < \tilde{q}_{1-y|a} \mathcal{E}_{\hat{\theta}_{1-y,a}}(X) \right] &= \mathbb{P}_X \left[\sqrt{\frac{\mathcal{E}_{\hat{\theta}_{1-y,a}}(X)}{\mathcal{E}_{\hat{\theta}_{y,a}}(X)}} > \sqrt{\frac{\tilde{q}_{y|a}}{\tilde{q}_{1-y|a}}} \right] \\
&\leq \mathbb{P}_X \left[\sqrt{\frac{\mathcal{E}_{\hat{\theta}_{1-y,a}}(X)}{\mathcal{E}_{\hat{\theta}_{y,a}}(X)}} > \sqrt{\frac{(1 - \varepsilon')q_{y|a}}{(1 + \varepsilon')q_{1-y|a}}} \right] \\
&\quad (\text{Event } \xi_1 \cap \xi_2 \text{ holds, } \tilde{q}_{y|a} \in (1 \pm \varepsilon')q_{y|a} \ \forall y \in \{0, 1\}, a \in \mathcal{A}) \\
&\leq \sqrt{\frac{(1 + \varepsilon')q_{1-y|a}}{(1 - \varepsilon')q_{y|a}}} \cdot \left(\frac{\varepsilon q_{m,a}}{10q_{M,a}} \right)^2 \quad (\text{by Lemma 4.6}) \\
&\leq \frac{\varepsilon q_{m,a}}{36} \quad (\text{Plugging } \varepsilon' = \frac{1}{10})
\end{aligned}$$

Therefore, the approximate MAP oracle $M_a(X) = \operatorname{argmax}_{y \in \{0,1\}} \tilde{q}_{y|a} \mathcal{E}_{\hat{\theta}_{y|a}}(X)$ correctly classifies at least $1 - \frac{\varepsilon q_{m,a}}{36}$ fraction of the samples generated from $\mathcal{E}_{\theta_{y|a}^*}$ for every $y \in \{0, 1\}$. For a given y , recall that R_y (line 10 of Algorithm 9) counts the number of samples assigned label y by the approximate MAP oracle. For $X_1, X_2, \dots, X_R$ drawn iid according to the mixture model $\mathcal{M}_a := \sum_{y \in \{0,1\}} q_{y|a} \mathcal{E}_{\theta_{y,a}^*}$, we have

$$\begin{aligned}
\mathbb{E}[R_y] &= \mathbb{E} \left[\sum_{i=1}^R \mathbb{1}\{M_a(X_i) = y\} \right] \\
&= \sum_{i=1}^R \mathbb{E}_{X_i \sim \mathcal{M}_a} [\mathbb{1}\{M_a(X_i) = y\}] \\
&= \sum_{i=1}^R \mathbb{P}_{X \sim \mathcal{M}_a} [M_a(X) = y] \quad (X_i \text{'s are iid}) \\
&= R \cdot \mathbb{P}_{X \sim \mathcal{M}_a} [M_a(X) = y] \\
&= R \cdot (\mathbb{P}_{X \sim \mathcal{M}_a} [M_a(X) = y \mid Y = y] \cdot \mathbb{P}[Y = y] + \mathbb{P}_{X \sim \mathcal{M}_a} [M_a(X) = y \mid Y = 1 - y] \cdot \mathbb{P}[Y = 1 - y]) \\
&= R \cdot \left(\mathbb{P}_{X \sim \mathcal{E}_{\theta_{y,a}^*}} [M_a(X) = y] \cdot \mathbb{P}[Y = y] + \mathbb{P}_{X \sim \mathcal{E}_{\theta_{1-y,a}^*}} [M_a(X) = y] \cdot \mathbb{P}[Y = 1 - y] \right) \\
&\geq R \cdot \mathbb{P}_{X \sim \mathcal{E}_{\theta_{y,a}^*}} [M_a(X) = y] \cdot \mathbb{P}[Y = y]
\end{aligned}$$

$$\geq R \cdot \left(1 - \frac{\varepsilon q_{m,a}}{36}\right) q_{y|a} \geq R \cdot \left(1 - \frac{\varepsilon}{36}\right) q_{y|a}$$

Similarly, we also have,

$$\begin{aligned} \mathbb{E}[R_y] &= R \cdot \left(\mathbb{P}_{X \sim \mathcal{E}_{\theta^*,a}}[M_a(X) = y] \cdot \mathbb{P}[Y = y] + \mathbb{P}_{X \sim \mathcal{E}_{\theta^*,1-y,a}}[M_a(X) = y] \cdot \mathbb{P}[Y = 1 - y] \right) \\ &\leq R \cdot \left(1 \cdot q_{y|a} + \frac{\varepsilon q_{m,a}}{36} \cdot q_{1-y|a} \right) \leq R \cdot \left(1 + \frac{\varepsilon}{36} \right) q_{y|a} \end{aligned}$$

Therefore, via an application of the Chernoff Bound (Lemma 4.15),

$$\begin{aligned} \mathbb{P} \left[\left(1 - \varepsilon_o\right) \left(1 - \frac{\varepsilon}{36}\right) q_{y|a} R \leq R_y \leq \left(1 + \varepsilon_o\right) \left(1 + \frac{\varepsilon}{36}\right) q_{y|a} R \right] &\geq 1 - 2 \exp \left(-\frac{\varepsilon_o^2 \cdot \mathbb{E}[R_y]}{3} \right) \\ &\geq 1 - 2 \exp \left(-\frac{\varepsilon_o^2 \left(1 - \frac{\varepsilon}{36}\right) R \cdot q_{y|a}}{3} \right) \end{aligned} \quad (4.5)$$

In **Exp-Audit** (Algorithm 9), we have set $R = \frac{3430 \log(14|\mathcal{A}|/\delta)}{\min\{\tilde{q}_{1|a}, \tilde{q}_{0|a}\} \varepsilon^2}$. Note that under event $\xi_1 \cap \xi_2$, $\tilde{q}_{y|a} \leq 1.1 q_{y|a}$. Therefore, $\min\{\tilde{q}_{1|a}, \tilde{q}_{0|a}\} \leq 1.1 \min\{q_{1|a}, q_{0|a}\} = 1.1 q_{m,a}$. Hence, $R \geq \frac{3 \log(14|\mathcal{A}|/\delta)}{q_{m,a} (1 - \frac{\varepsilon}{36}) (\frac{\varepsilon}{18.5})^2}$, since $\varepsilon \in (0, 1)$. This implies that in Eq. (4.5), if we substitute $\varepsilon_o = \frac{\varepsilon}{18.5}$, we have,

$$\mathbb{P} \left[\left(1 - \frac{\varepsilon}{18.5}\right) \left(1 - \frac{\varepsilon}{36}\right) q_{y|a} R \leq R_y \leq \left(1 + \frac{\varepsilon}{18.5}\right) \left(1 + \frac{\varepsilon}{36}\right) q_{y|a} R \right] \geq 1 - \frac{2\delta}{14|\mathcal{A}|}.$$

Hence, the estimate $\hat{q}_{y|a}$ obtained in Line 11 of **Exp-Audit** (Algorithm 9) for all $a \in \mathcal{A}$ satisfies with probability at least $1 - \frac{4\delta}{14}$ (Union Bound): $(1 - \frac{\varepsilon}{12}) q_{y|a} \leq \hat{q}_{y|a} \leq (1 + \frac{\varepsilon}{12}) q_{y|a}$ for all $y \in \{0, 1\}$, where we have used $1 - \frac{\varepsilon}{12} \leq (1 - \frac{\varepsilon}{36})(1 - \frac{\varepsilon}{18.5})$ and $1 + \frac{\varepsilon}{12} \geq (1 + \frac{\varepsilon}{36})(1 + \frac{\varepsilon}{18.5})$ for all $\varepsilon \in (0, 1)$.

Having found an estimate $\hat{q}_{y|a} \in (1 \pm \frac{\varepsilon}{12}) q_{y|a}$, the guarantee in terms of the correctness of the hypothesis test follows similarly as in the proof of Theorem 4.2. Hence, the algorithm is successful in the (ε, δ) -fairness audit is with probability at least $1 - \delta$.

The audit cost can be calculated by observing that the audit cost is incurred in two steps of the algorithm—first to obtain $\tilde{q}_{y|a}$ and then to observe R features. For estimating $\tilde{q}_{y|a}$, the auditor pays $\tilde{O} \left(\frac{c_{feat} \mathbb{P}[f=0|A=a] + c_{lab} \mathbb{P}[f=0, Y=0|A=a]}{q_{y|a} \varepsilon'^2} \right)$ (similar to proof of Theorem 4.2). Since $\varepsilon' = \frac{1}{10}$ is a constant, the cost of estimating $\tilde{q}_{y|a}$ is $\tilde{O} \left(\frac{c_{feat} \mathbb{P}[f=0|A=a] + c_{lab} \mathbb{P}[f=0, Y=0|A=a]}{q_{y|a}} \right)$. Adding this cost for both $y = 0$ and $y = 1$ gives that the audit cost of this stage of the algorithm for a particular group $A = a$ is $\tilde{O} \left(\sum_{y \in \{0,1\}} \frac{c_{feat} \mathbb{P}[f=0|A=a] + c_{lab} \mathbb{P}[f=0, Y=0|A=a]}{q_{y|a}} \right)$.

For observing R features, the auditor incurs an audit cost of $R \cdot c_{feat} \cdot \mathbb{P}[f = 0 | A = a]$.

Plugging $R = \tilde{O}\left(\frac{1}{q_{m,a}\varepsilon^2}\right)$, we have that the audit cost for this phase is $\tilde{O}\left(\frac{c_{feat}\mathbb{P}[f=0|A=a]}{q_{m,a}\varepsilon^2}\right)$.

Summing the two costs up for all group labels $a \in \mathcal{A}$ we have the desired result. $\square$

4.5 Experiments & Results

In the interest of space, all details and an exhaustive set of experiments are presented in Appendix 4.10, with some key results here. The code is provided in the supplementary.

Blackbox Model: We compare the Baseline (Algorithm 7) and **RS-Audit** on the Adult Income [13] and Law School Admission [96] datasets (details in Appendix 4.10). For a given dataset, we first train a classifier on a subset. Thereafter, we run our algorithms with this classifier. We simulate past database using a portion of the dataset that is classified positively by the trained classifier, and online samples by drawing random data points from the dataset. We repeat the experiments across 5 seeds and plot the average and standard deviation. The details of the experimental setup are presented in Appendix 4.10. In our experiments, we set $\gamma = 0$ and vary the stopping threshold τ over a range and study the behavior of the two algorithms. For every τ , we obtain the mean number of samples, the mean predicted equalized odds difference (EOD), and the mean cost. In Fig. 4.1, we show the results for a logistic regression classifier trained on all the features on the Adult Dataset. The first plot shows the predicted EOD vs the average number of samples collected by the two algorithms. We observe that **RS-Audit** (black line) is able to estimate the true EOD (horizontal solid line) with much fewer samples than Baseline (red line). The second plot shows that **RS-Audit**'s cost (black line) is significantly smaller than Baseline (red line) at various values of τ .

Mixture Model: In the mixture model, we perform experiments on synthetic and Adult Income datasets. For the synthetic experiments, we choose spherical Gaussians as our exponential family and ensure the separation condition (4.1) during construction. For the Adult Income dataset, we first train a neural network to predict the labels, and then use the embedding from the penultimate layer of the network. We do not enforce any structure or separation on the embeddings, instead we instantiate **Exp-Audit** with the exponential family being multivariate Gaussian. Note that the true data may not actually follow this distribution. Moreover, the past data and online samples are simulated similarly as in the experiments for the blackbox model. We experiment with various values of γ and ε (with $\delta = 0.01$) and repeat the experiments over 5 seeds. In Fig. 4.1, we show the comparison between **RS-Audit** and **Exp-Audit** (Algorithm 9). We observe that on both Adult dataset (second column) and the synthetic dataset (third column), while the estimated EOD of **RS-Audit** is slightly more closer to the true EOD than **Exp-Audit**, the cost of **Exp-Audit** is significantly smaller than that of **RS-Audit** (here we show for $c_{feat} = 0$ and $c_{lab} = 1$, while other values are shown in Appendix 4.10). Moreover, the trend

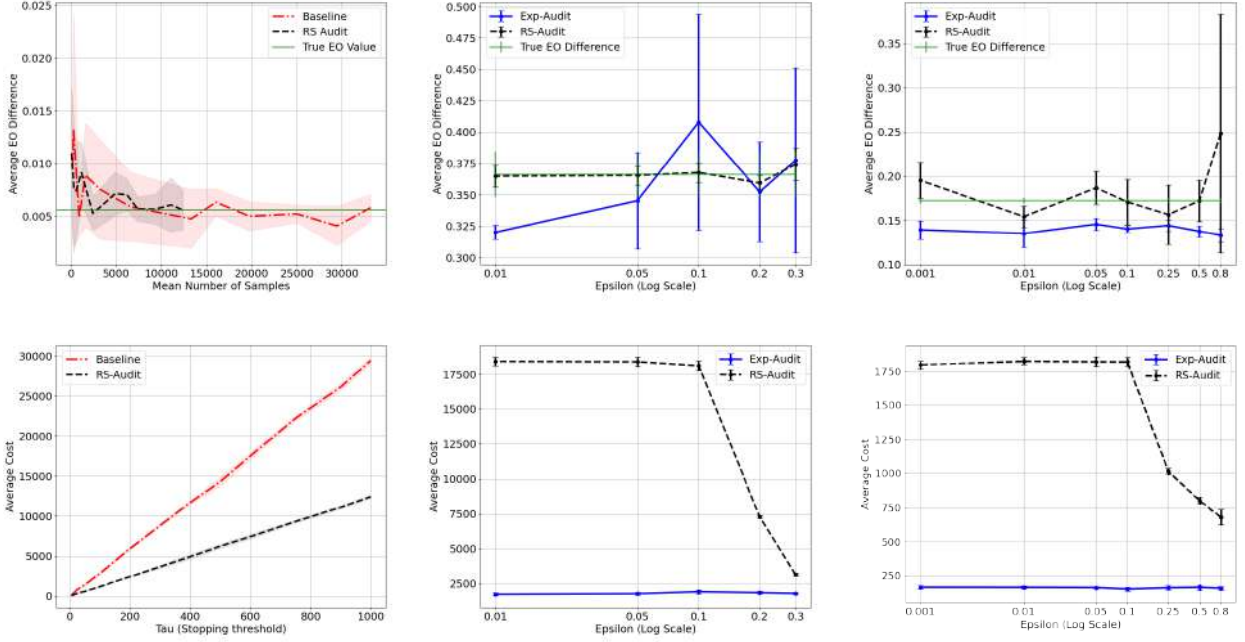


Figure 4.1: **Leftmost Column:** Results for the Baseline (Algorithm 7) and the RS-Audit (Algorithm 8) on the Adult Income dataset. **Middle and Right Columns:** Comparison of RS-Audit and Exp-Audit (Algorithm 9) on the Adult Income (middle column) and synthetic (right column) dataset.

emerges that the cost of Exp-Audit is independent of ϵ while the cost of RS-Audit increases with ϵ (note that we had to cap τ for RS-Audit to a maximum of 1000 for timely execution of the code, hence flattening of the curve).

Remark 12 (Empirical Choices) *Our algorithms and their guarantees are agnostic to the choice of classifier. Further, the blackbox setting is agnostic to the data distribution. Therefore, we choose logistic regression as our classifier which is also preferred in fairness applications due to its explainability over complicated models, and we experiment with popular fairness datasets to validate our theoretical results empirically. Note that in the mixture model experiment with Adult dataset, we do not enforce any of the structural or separation assumptions required to prove Theorem 4.4. Yet, we observe that Exp-Audit performs better than RS-Audit in terms of the audit cost while giving good EOD estimates, further indicating its robustness.*

4.6 Proof of Theorem 4.1 and Other Auxiliary Lemmas

Recall that the quantities of interest are $p_{y,a} := \mathbb{P}[f = 1, Y = y, A = a]$, $q_{y,a} := \mathbb{P}[Y = y, A = a]$, $p_{y|a} := \mathbb{P}[f = 1, Y = y \mid A = a]$, and $q_{y|a} := \mathbb{P}[Y = y \mid A = a]$ for all $y \in \{0, 1\}$ and $a \in \mathcal{A}$. Note that $p_{y,a} = p_{y|a} \cdot \mathbb{P}[A = a]$, therefore $p_{y,a} \leq p_{y|a}$, and similarly, $q_{y,a} \leq q_{y|a}$.

4.6.1 Proof of Theorem 4.1

First we restate Theorem 4.1 and then present its proof. Recall that $h : \mathcal{X} \times \mathcal{A} \times \{0, 1\} \rightarrow \{0, 1\}$ denotes the policy of the auditor whether to observe the true label of an individual and $g : \mathcal{A} \rightarrow \{0, 1\}$ denotes the policy whether to observe the feature of the individual. Note that setting $h = 1$ reveals the features of the individual anyway so when $h = 1$, one can set $g = 0$.

Theorem 4.1 *Given $\gamma \geq 0$, $\varepsilon, \delta \in (0, 1)$, under the blackbox model, the audit policy that requests true labels of every individual with $f = 0$ (Algorithm 7) succeeds in the $(\gamma, \varepsilon, \delta)$ -fairness audit with $\text{cost} \leq \tilde{O}\left(\sum_{a \in \mathcal{A}} \sum_{y \in \{0, 1\}} \frac{c_{feat}\mathbb{P}[f=0] + c_{lab}\mathbb{P}[f=0, Y=0]}{q_{y,a}\varepsilon^2}\right)$. Moreover, the number of true labels requested by this policy is at most $\tilde{O}(\sum_{a \in \mathcal{A}} \sum_{y \in \{0, 1\}} \frac{\mathbb{P}[f=0]}{q_{y,a}\varepsilon^2})$.*

Proof: Fix an $y \in \{0, 1\}$ and an $a \in \mathcal{A}$. The algorithm keeps observing (A, Y) till $\tau = \frac{576 \log(8|\mathcal{A}|/\delta)}{\varepsilon^2}$ samples of $Y = y$ and $A = a$ are observed. Let this number of samples collected be $N_{y,a}$. We define the event $\mathcal{E}_{y,a} := \{N_{y,a} \in (1 \pm \frac{\varepsilon}{12}) \frac{576 \log(8|\mathcal{A}|/\delta)}{\varepsilon^2 q_{y,a}}\}$. From Lemma 4.13, with $\tau = \frac{576 \log(8|\mathcal{A}|/\delta)}{\varepsilon^2}$, we have $\mathbb{P}[\mathcal{E}_{y,a}^c] \leq \frac{\delta}{4|\mathcal{A}|}$. Applying a Union Bound over all $y \in \{0, 1\}$ and $a \in \mathcal{A}$, we have for the event $\mathcal{E} := \cap_{(y,a) \in \{0,1\} \times \mathcal{A}} \mathcal{E}_{y,a}$,

$$\begin{aligned} \mathbb{P}[\mathcal{E}] &= \mathbb{P}[\cap_{(a,y) \in \mathcal{A} \times \{0,1\}} \mathcal{E}_{y,a}] = 1 - \mathbb{P}[\cup_{(a,y) \in \mathcal{A} \times \{0,1\}} \mathcal{E}_{y,a}^c] && \text{(De Morgan's law)} \\ &\geq 1 - \sum_{(a,y) \in \mathcal{A} \times \{0,1\}} \mathbb{P}[\mathcal{E}_{y,a}^c] && \text{(Union Bound)} \\ &\geq 1 - \sum_{(a,y) \in \mathcal{A} \times \{0,1\}} \frac{\delta}{4|\mathcal{A}|} = 1 - \frac{\delta}{2}. \end{aligned}$$

Similarly, we also define the event $\mathcal{F}_{y,a} := \{N'_{y,a} \in (1 \pm \frac{\varepsilon}{12}) \frac{576 \log(8|\mathcal{A}|/\delta)}{\varepsilon^2 p_{y,a}}\}$, where $N'_{y,a}$ is the number of individuals iterated over by the algorithm in the past database while trying to collect τ individuals with $f = 1, Y = y$ and $A = a$. By essentially similar arguments as above (application of Lemma 4.13), we can show that the event $\mathcal{F} := \cap_{(y,a) \in \{0,1\} \times \mathcal{A}} \mathcal{F}_{y,a}$ happens with probability at least $1 - \frac{\delta}{2}$. Therefore, via Union Bound, $\mathbb{P}[\mathcal{E} \cap \mathcal{F}] \geq 1 - \delta$.

Under event $\mathcal{E}$, we have that the total number of samples collected by the algorithm:

$$\begin{aligned} N &:= \sum_{(a,y) \in \mathcal{A} \times \{0,1\}} N_{y,a} \leq \sum_{(a,y) \in \mathcal{A} \times \{0,1\}} \left(1 + \frac{\varepsilon}{12}\right) \frac{576 \log(8|\mathcal{A}|/\delta)}{\varepsilon^2 q_{y,a}} \\ &\leq \sum_{(a,y) \in \mathcal{A} \times \{0,1\}} \frac{13}{12} \cdot \frac{576 \log(8|\mathcal{A}|/\delta)}{\varepsilon^2 q_{y,a}} && (\varepsilon < 1) \\ &= \sum_{a \in \mathcal{A}} \sum_{y \in \{0,1\}} \frac{624 \log(8|\mathcal{A}|/\delta)}{\varepsilon^2 q_{y,a}} \end{aligned}$$

Finally, the expected number of individuals for whom $h = 1$ is $N \cdot \mathbb{P}[h = 1] = N \cdot \mathbb{P}[f = 0]$. This is the actual number of individuals for whom the algorithm requests true labels. Hence, the number of true labels requested by this algorithm is at most

$$N \cdot \mathbb{P}[f = 0] \leq \sum_{a \in \mathcal{A}} \sum_{y \in \{0,1\}} \frac{624 \mathbb{P}[f = 0] \log(8|\mathcal{A}|/\delta)}{\varepsilon^2 q_{y,a}}.$$

Further, the audit cost of observing the true labels of N individuals is $N \cdot (c_{feat} \mathbb{P}[f = 0] + c_{lab} \mathbb{P}[f = 0, Y = 0])$. Hence, the audit cost of the algorithm is bounded by

$$\text{cost} \leq \sum_{a \in \mathcal{A}} \sum_{y \in \{0,1\}} \frac{624(c_{feat} \mathbb{P}[f = 0] + c_{lab} \mathbb{P}[f = 0, Y = 0]) \log(8|\mathcal{A}|/\delta)}{\varepsilon^2 q_{y,a}}.$$

Finally, it remains to show that the algorithm also successfully performs the hypothesis test. Under the event $\mathcal{E} \cap \mathcal{F}$, we have $\hat{q}_{y,a} = \frac{\tau}{N_{y,a}} = \frac{576 \log(8|\mathcal{A}|/\delta)}{\varepsilon^2 N_{y,a}}$. Hence, $\frac{q_{y,a}}{1+\frac{\varepsilon}{12}} \leq \hat{q}_{y,a} \leq \frac{q_{y,a}}{1-\frac{\varepsilon}{12}}$. Similarly, $\frac{p_{y,a}}{1+\frac{\varepsilon}{12}} \leq \hat{p}_{y,a} \leq \frac{p_{y,a}}{1-\frac{\varepsilon}{12}}$. Hence,

$$\frac{(1 - \frac{\varepsilon}{12})p_{y,a}}{(1 + \frac{\varepsilon}{12})q_{y,a}} \leq \frac{\hat{p}_{y,a}}{\hat{q}_{y,a}} \leq \frac{(1 + \frac{\varepsilon}{12})p_{y,a}}{(1 - \frac{\varepsilon}{12})q_{y,a}}. \quad (4.6)$$

Using Lemma 4.3 and the fact that $0 \leq \frac{p_{y,a}}{q_{y,a}} \leq 1$, we finally have,

$$\frac{p_{y,a}}{q_{y,a}} - \frac{\varepsilon}{6} \leq \frac{\hat{p}_{y,a}}{\hat{q}_{y,a}} \leq \frac{p_{y,a}}{q_{y,a}} + \frac{\varepsilon}{3}.$$

Since the above holds for all $(y, a) \in \{0, 1\} \times \mathcal{A}$ under $\mathcal{E} \cap \mathcal{F}$, we therefore have, for a particular $y \in \{0, 1\}$ and $a, a' \in \mathcal{A}$,

$$\frac{p_{y,a}}{q_{y,a}} - \frac{p_{y,a'}}{q_{y,a'}} - \frac{\varepsilon}{2} \leq \frac{\hat{p}_{y,a}}{\hat{q}_{y,a}} - \frac{\hat{p}_{y,a'}}{\hat{q}_{y,a'}} \leq \frac{p_{y,a}}{q_{y,a}} - \frac{p_{y,a'}}{q_{y,a'}} + \frac{\varepsilon}{2}.$$

From this, we can conclude that

$$\left| \frac{\hat{p}_{y,a}}{\hat{q}_{y,a}} - \frac{\hat{p}_{y,a'}}{\hat{q}_{y,a'}} \right| - \left| \frac{p_{y,a}}{q_{y,a}} - \frac{p_{y,a'}}{q_{y,a'}} \right| \leq \frac{\varepsilon}{2}. \quad (4.7)$$

Suppose the true hypothesis is **UNFAIR**. Let $y^* \in \{0, 1\}$ and $a_1, a_2 \in \mathcal{A}, a_1 \neq a_2$ be such that $|\mathbb{P}[f = 1 \mid Y = y^*, A = a_1] - \mathbb{P}[f = 1 \mid Y = y^*, A = a_2]| = \left| \frac{p_{y^*,a_1}}{q_{y^*,a_1}} - \frac{p_{y^*,a_2}}{q_{y^*,a_2}} \right| > \gamma + \varepsilon$. Then, we

have,

$$\begin{aligned}
\widehat{\Delta} &\geq \left| \frac{\widehat{p}_{y^*,a_1}}{\widehat{q}_{y^*,a_1}} - \frac{\widehat{p}_{y^*,a_2}}{\widehat{q}_{y^*,a_2}} \right| && \text{(Definition of } \widehat{\Delta} \text{ (Step 9 of Algorithm 7))} \\
&\geq \left| \frac{p_{y^*,a_1}}{q_{y^*,a_1}} - \frac{p_{y^*,a_2}}{q_{y^*,a_2}} \right| - \frac{\varepsilon}{2} && \text{(Equation (4.7))} \\
&> \gamma + \varepsilon - \frac{\varepsilon}{2} && \left(\left| \frac{p_{y^*,a_1}}{q_{y^*,a_1}} - \frac{p_{y^*,a_2}}{q_{y^*,a_2}} \right| > \gamma + \varepsilon \right) \\
&= \gamma + \frac{\varepsilon}{2} .
\end{aligned}$$

The algorithm therefore outputs **UNFAIR** (Step 10). Hence, under the hypothesis **UNFAIR**, the algorithm's output is correct.

Next, suppose the true hypothesis is **FAIR**. Then, we have for all $y \in \{0, 1\}$ and $a, a' \in \mathcal{A}$,

$$|\mathbb{P}[f = 1 \mid Y = y, A = a] - \mathbb{P}[f = 1 \mid Y = y, A = a']| = \left| \frac{p_{y,a}}{q_{y,a}} - \frac{p_{y,a'}}{q_{y,a'}} \right| \leq \gamma .$$

Hence, by Equation (4.7), we can conclude that for all $y \in \{0, 1\}$ and $a, a' \in \mathcal{A}$,

$$\begin{aligned}
\left| \frac{\widehat{p}_{y,a}}{\widehat{q}_{y,a}} - \frac{\widehat{p}_{y,a'}}{\widehat{q}_{y,a'}} \right| &\leq \left| \frac{p_{y,a}}{q_{y,a}} - \frac{p_{y,a'}}{q_{y,a'}} \right| + \frac{\varepsilon}{2} \\
&\leq \gamma + \frac{\varepsilon}{2} . && \left(\left| \frac{p_{y,a}}{q_{y,a}} - \frac{p_{y,a'}}{q_{y,a'}} \right| \leq \gamma \right)
\end{aligned}$$

In particular, we have $\widehat{\Delta} = \max_{y \in \{0,1\}} \max_{a,a' \in \mathcal{A}} \left| \frac{\widehat{p}_{y,a}}{\widehat{q}_{y,a}} - \frac{\widehat{p}_{y,a'}}{\widehat{q}_{y,a'}} \right| \leq \gamma + \frac{\varepsilon}{2}$. Hence, the algorithm outputs **FAIR** implying that the algorithm's output is correct.

Since the output of the algorithm is correct under both the hypotheses, one can conclude the output of the algorithm is always correct. Note that we assumed event $\mathcal{E} \cap \mathcal{F}$ to hold, under which the correctness of our algorithm was derived. Therefore, our algorithm is correct with probability at least $1 - \delta$. This finishes the proof. $\square$

4.6.2 Auxiliary Lemmas

Lemma 4.3 *Suppose $\varepsilon \in (0, \frac{1}{2})$. Then we have, (i) $\frac{1-\varepsilon}{1+\varepsilon} \geq 1 - 2\varepsilon$ and (ii) $\frac{1+\varepsilon}{1-\varepsilon} \leq 1 + 4\varepsilon$.*

Proof: For (i), we have

$$\begin{aligned}
\frac{1-\varepsilon}{1+\varepsilon} &\geq 1 - 2\varepsilon \\
\iff 1 - \varepsilon &\geq (1 - 2\varepsilon)(1 + \varepsilon)
\end{aligned}$$

$$\begin{aligned}
&\Longleftrightarrow (1 - 2\varepsilon)(1 + \varepsilon) = 1 + \varepsilon - 2\varepsilon - 2\varepsilon^2 \\
&\Longleftrightarrow 1 + \varepsilon - 2\varepsilon - 2\varepsilon^2 = 1 - \varepsilon - 2\varepsilon^2 \\
&\Longleftrightarrow 1 - \varepsilon \geq 1 - \varepsilon - 2\varepsilon^2 \\
&\Longleftrightarrow 0 \geq -2\varepsilon^2 .
\end{aligned}$$

This is always true for $\varepsilon \in (0, \frac{1}{2})$, so the inequality holds.

Similarly, for (ii) we have

$$\begin{aligned}
&\frac{1 + \varepsilon}{1 - \varepsilon} \leq 1 + 4\varepsilon \\
&\Longleftrightarrow 1 + \varepsilon \leq (1 + 4\varepsilon)(1 - \varepsilon) \\
&\Longleftrightarrow 1 + \varepsilon \leq 1 + 3\varepsilon - 4\varepsilon^2 \\
&\Longleftrightarrow 2\varepsilon(1 - 2\varepsilon) \geq 0 .
\end{aligned}$$

This is clearly true for $\varepsilon \in (0, \frac{1}{2})$, so the inequality holds. $\square$

Lemma 4.4 (Bretagnolle-Huber Inequality (Theorem 14.2 [53])) *Let P and Q be two probability measures on some probability space $(\Omega, \mathcal{F})$. Let A be a measurable event in this probability space. Then,*

$$P(A) + Q(A^c) \geq \frac{1}{2} \exp(-D_{KL}(P \parallel Q)) .$$

Lemma 4.5 *Let $q \in (0, \frac{1}{3})$ and $\varepsilon \in (0, \frac{1}{8})$. Then, we have*

$$D_{KL}(\text{Ber}(q) \parallel \text{Ber}(q(1 + 8\varepsilon))) \leq 64q\varepsilon^2 \quad \text{and} \quad D_{KL}(\text{Ber}(q(1 + 4\varepsilon)) \parallel \text{Ber}(q)) \leq 128q\varepsilon^2 .$$

Proof: First note that $\frac{8q\varepsilon}{1-q} \leq \frac{1}{2}$.

$$\begin{aligned}
D_{KL}(\text{Ber}(q) \parallel \text{Ber}(q(1 + 8\varepsilon))) &= q \log \left(\frac{q}{q(1 + 8\varepsilon)} \right) + (1 - q) \log \left(\frac{1 - q}{1 - q(1 + 8\varepsilon)} \right) \\
&= -q \log(1 + 8\varepsilon) - (1 - q) \log \left(1 - \frac{8q\varepsilon}{1 - q} \right) \\
&\leq -q \left(8\varepsilon - \frac{(8\varepsilon)^2}{2} \right) - (1 - q) \left(-\frac{8q\varepsilon}{1 - q} - \frac{(8q\varepsilon)^2}{(1 - q)^2} \right) \\
&\quad (\log(1 + x) \geq x - \frac{x^2}{2} \text{ and } \log(1 - x) \geq -x - x^2 \text{ for } x \in (0, \frac{1}{2})) \\
&= \frac{q(8\varepsilon)^2}{2} \left(\frac{2q}{1 - q} + 1 \right)
\end{aligned}$$

$$\begin{aligned}
&\leq \frac{64q\varepsilon^2}{2}(1+1) && (q \in (0, 1/3), \text{ hence } \frac{q}{1-q} \leq \frac{1}{2}) \\
&\leq 64q\varepsilon^2
\end{aligned}$$

For the other inequality, we have,

$$\begin{aligned}
D_{KL}(\text{Ber}(q(1+8\varepsilon)) \parallel \text{Ber}(q)) &= q(1+8\varepsilon) \log \left(\frac{q(1+8\varepsilon)}{q} \right) + (1-q(1+8\varepsilon)) \log \left(\frac{1-q(1+8\varepsilon)}{1-q} \right) \\
&= q(1+8\varepsilon) \log(1+8\varepsilon) + (1-q(1+8\varepsilon)) \log \left(1 - \frac{8q\varepsilon}{1-q} \right) \\
&\leq q(1+8\varepsilon)(8\varepsilon) + (1-q(1+8\varepsilon)) \left(-\frac{8q\varepsilon}{1-q} \right) \\
&\quad (\log(1+x) \leq x \text{ for } x \in (-1, 1)) \\
&= 8q\varepsilon + 64q\varepsilon^2 - 8q\varepsilon + \frac{64q^2\varepsilon^2}{1-q} \\
&= 64q\varepsilon^2 \left(1 + \frac{q}{1-q} \right) \\
&\leq 128q\varepsilon^2 && (q \in (0, \frac{1}{2}))
\end{aligned}$$

□

4.7 Missing Proofs from Section 4.4

Recall that in the mixture model, given $A = a$, the true label is sampled as $Y \sim \text{Ber}(q_{1|a})$ and then the feature X is sampled as $X \sim \mathcal{E}_{\theta_{y,a}^*}$, where $\mathcal{E}_{\theta_{y,a}^*}(X) = h(X) \exp(\theta_{y,a}^* T(X) - W(\theta))$. We denote by $\mathcal{B}(a, b)$ the d -dimensional ball of radius b around a point $a \in \mathbb{R}^d$. In the mixture model setting, we make the following assumptions: (i) $\mathbb{P}[f = 1 \mid Y = y, A = a] \geq \alpha > 0$ for some known constant α , (ii) $\mathcal{E}_\theta(X)$ is log-concave in X , $\kappa \mathbf{I} \preceq \nabla^2 W(\theta) \preceq \lambda \mathbf{I}$ with $0 < \kappa < \lambda$ and W is L -Lipschitz for all $\theta \in \Theta$, $T(X)$ has components polynomial in X , $\mathcal{B}(\theta_{y,a}^*, \frac{1}{\beta}) \subset \Theta$ for some $\beta > 0$ for all $y \in \{0, 1\}$ and $a \in \mathcal{A}$. Without loss of generality, let $\kappa \leq 1$ and $L, \beta \geq 1$.

4.7.1 Approximate Maximum-a-posteriori Oracle for Exponential Family

Lemma 4.6 *Let $\Theta \subset \mathbb{R}^d$ be a convex set containing parameters θ_1, θ_2 and θ^* . Let $\varepsilon \in (0, 1)$ and $L, \beta \geq 1$ and $\kappa \leq 1$. Further, assume that (i) $\kappa \mathbf{I} \preceq \nabla^2 W(\theta) \preceq \lambda \mathbf{I}$, (ii) $W(\theta)$ is L -Lipschitz for all $\theta \in \Theta$, (iii) $\|\theta_1 - \theta_2\| \geq \sqrt{\frac{3(16(L\sqrt{\beta})\beta + \lambda)}{16\kappa\beta^2}} \log\left(\frac{\gamma}{\varepsilon}\right)$ for some $\gamma \geq e^2$, (iv) $\|\theta_1 - \theta^*\| \leq \frac{\varepsilon}{2\beta}$, and (v) $\mathcal{B}\left(\frac{\theta_1 + \theta_2}{2}, \frac{1}{2\beta}\right) \subset \Theta$. Then we have $\mathbb{P}_{X \sim \mathcal{E}_{\theta^*}}[\mathcal{E}_{\theta_1}(X) > t \cdot \mathcal{E}_{\theta_2}(X)] \geq 1 - \sqrt{t} \left(\frac{\varepsilon}{\gamma}\right)^2$.*

Proof: We have the following:

$$\begin{aligned}
& \mathbb{P}_{X \sim \mathcal{E}_{\theta^*}} [\mathcal{E}_{\theta_1}(X) < t \cdot \mathcal{E}_{\theta_2}(X)] \\
&= \mathbb{P}_{X \sim \mathcal{E}_{\theta^*}} \left[\sqrt{\frac{\mathcal{E}_{\theta_2}(X)}{\mathcal{E}_{\theta_1}(X)}} > \sqrt{\frac{1}{t}} \right] \\
&\leq \sqrt{t} \cdot \mathbb{E}_{X \sim \mathcal{E}_{\theta^*}} \left[\sqrt{\frac{\mathcal{E}_{\theta_2}(X)}{\mathcal{E}_{\theta_1}(X)}} \right] \quad (\text{Markov's Inequality}) \\
&= \sqrt{t} \int_{x \in \mathcal{X}} h(x) \exp(\theta^{*\top} T(x) - W(\theta^*)) \cdot \sqrt{\frac{h(x) \exp(\theta_2^\top T(x) - W(\theta_2))}{h(x) \exp(\theta_1^\top T(x) - W(\theta_1))}} \cdot dx \\
&= \sqrt{t} \int_{x \in \mathcal{X}} h(x) \exp(\theta^{*\top} T(x) - W(\theta^*)) \cdot \exp \left(\frac{1}{2} ((\theta_2 - \theta_1)^\top T(x) - W(\theta_2) + W(\theta_1)) \right) \cdot dx \\
&= \sqrt{t} \int_{x \in \mathcal{X}} h(x) \exp \left(\left(\frac{\theta_1 + \theta_2}{2} \right)^\top T(x) + (\theta^* - \theta_1)^\top T(x) - \frac{W(\theta_1) + W(\theta_2)}{2} + W(\theta_1) - W(\theta^*) \right) \cdot dx
\end{aligned}$$

We now explain the strategy to bound the above quantity. Firstly, we will use the Lipschitzness of W to bound $W(\theta_1) - W(\theta^*)$ by leveraging the fact that θ_1 and θ^* are close by assumption of the lemma. Next, the goal is to bound the remaining expressions so that we obtain a valid probability density function from the exponential family, possibly with a different parameter than what we started with. At this point, one may be tempted to use the convexity of W to lower bound $\frac{W(\theta_1) + W(\theta_2)}{2} \geq W\left(\frac{\theta_1 + \theta_2}{2}\right)$. Although this give us the probability density whose parameter is $\frac{\theta_1 + \theta_2}{2}$, the bound is not tight and does not allow us to obtain meaningful results. Therefore, we resort to bounding $\frac{W(\theta_1) + W(\theta_2)}{2}$ via Taylor's theorem. The steps are illustrated below.

$$\begin{aligned}
& \mathbb{P}_X [\mathcal{E}_{\theta_1}(X) < t \cdot \mathcal{E}_{\theta_2}(X)] \\
&\leq \sqrt{t} e^{\frac{L\varepsilon}{2\beta}} \int_{x \in \mathcal{X}} h(x) \exp \left(\left(\frac{\theta_1 + \theta_2}{2} \right)^\top T(x) - \frac{W(\theta_1) + W(\theta_2)}{2} \right) \cdot \exp((\theta^* - \theta_1)^\top T(x)) \cdot dx \\
&\quad (\|\theta^* - \theta_1\| \leq \frac{\varepsilon}{2\beta}, W \text{ is } L\text{-Lipschitz}) \\
&= \sqrt{t} e^{\frac{L\varepsilon}{2\beta}} \int_{x \in \mathcal{X}} h(x) \exp \left(\left(\frac{\theta_1 + \theta_2}{2} \right)^\top T(x) - W \left(\frac{\theta_1 + \theta_2}{2} \right) - \frac{1}{8} \|\theta_1 - \theta_2\|_{(\nabla^2 W(\bar{\theta}) + \nabla^2 W(\tilde{\theta}))}^2 \right) \\
&\quad \cdot \exp((\theta^* - \theta_1)^\top T(x)) \cdot dx \\
&\quad (\text{Lemma 4.8, } \bar{\theta} \in [\theta_1, \frac{\theta_1 + \theta_2}{2}], \tilde{\theta} \in [\theta_2, \frac{\theta_1 + \theta_2}{2}]) \\
&= \sqrt{t} \exp \left(\frac{L\varepsilon}{2\beta} - \frac{1}{8} \|\theta_1 - \theta_2\|_{(\nabla^2 W(\bar{\theta}) + \nabla^2 W(\tilde{\theta}))}^2 \right)
\end{aligned}$$

$$\cdot \int_{x \in \mathcal{X}} h(x) \exp \left(\left(\frac{\theta_1 + \theta_2}{2} \right)^\top T(x) - W \left(\frac{\theta_1 + \theta_2}{2} \right) \right) \cdot \exp ((\theta^* - \theta_1)^\top T(x)) \cdot dx$$

After this, we will use $\nabla^2 W(\theta) \succeq \kappa \mathbf{I}$ for all $\theta \in \Theta$ to lower bound the quadratic term. On the other hand, the quantity in the integral is actually an expectation with respect to the distribution whose parameter is $\frac{\theta_1 + \theta_2}{2}$. To bound the $\exp ((\theta^* - \theta_1)^\top T(x))$ in expectation, we will use the sub-exponential property of the exponential family distribution combined with the fact that for the exponential family we have $\mathbb{E}_{X \sim \mathcal{E}_\theta}[T(X)] = \nabla W(\theta)$.

$$\begin{aligned} & \mathbb{P}_X[\mathcal{E}_{\theta_1}(X) < t \cdot \mathcal{E}_{\theta_2}(X)] \\ & \leq \sqrt{t} \exp \left(\frac{L\varepsilon}{2\beta} - \frac{1}{8} \|\theta_1 - \theta_2\|_{(\nabla^2 W(\bar{\theta}) + \nabla^2 W(\tilde{\theta}))}^2 + (\theta^* - \theta_1)^\top \mu \right) \cdot \mathbb{E}_{X \sim \mathcal{E}_{\frac{\theta_1 + \theta_2}{2}}} [\exp ((\theta^* - \theta_1)^\top (T(X) - \mu))] \\ & \quad (\mu := \mathbb{E}_{X \sim \mathcal{E}_{\frac{\theta_1 + \theta_2}{2}}}[T(X)]) \\ & \leq \sqrt{t} \cdot \exp \left(\frac{L\varepsilon}{2\beta} - \frac{1}{8} \|\theta_1 - \theta_2\|_{2\kappa \mathbf{I}}^2 + (\theta^* - \theta_1)^\top \mu \right) \cdot \mathbb{E}_{X \sim \mathcal{E}_{\frac{\theta_1 + \theta_2}{2}}} [\exp ((\theta^* - \theta_1)^\top (T(X) - \mu))] \\ & \quad (\nabla^2 W(\theta) \succeq \kappa \mathbf{I} \text{ for all } \theta \in \Theta) \\ & \leq \sqrt{t} \cdot \exp \left(\frac{L\varepsilon}{2\beta} - \frac{\kappa}{4} \|\theta_1 - \theta_2\|^2 + \|\theta^* - \theta_1\| \cdot \|\mu\| \right) \cdot \mathbb{E}_{X \sim \mathcal{E}_{\frac{\theta_1 + \theta_2}{2}}} [\exp ((\theta^* - \theta_1)^\top (T(X) - \mu))] \\ & \quad (\text{Cauchy-Schwarz Inequality}) \\ & \leq \sqrt{t} \cdot \exp \left(\frac{L\varepsilon}{\beta} - \frac{\kappa}{4} \|\theta_1 - \theta_2\|^2 \right) \cdot \mathbb{E}_{X \sim \mathcal{E}_{\frac{\theta_1 + \theta_2}{2}}} [\exp ((\theta^* - \theta_1)^\top (T(X) - \mu))] \\ & \quad (\|\theta^* - \theta_1\| \leq \frac{\varepsilon}{2\beta}; \|\mu\| = \|\nabla W(\frac{\theta_1 + \theta_2}{2})\| \leq L \text{ since } W \text{ is } L\text{-Lipschitz}) \\ & \leq \sqrt{t} \cdot \exp \left(\frac{L\varepsilon}{\beta} - \frac{\kappa}{4} \|\theta_1 - \theta_2\|^2 \right) \cdot \exp \left(\frac{\|\theta^* - \theta_1\|^2 \lambda}{2} \right) \\ & \quad (\text{Lemma 4.9 as by assumption } \mathcal{B} \left(\frac{\theta_1 + \theta_2}{2}, \frac{1}{2\beta} \right) \subset \Theta \text{ and } \|\theta^* - \theta_1\| < \frac{1}{2\beta}) \\ & \leq \sqrt{t} \cdot \exp \left(\frac{(L \vee \beta)\varepsilon}{\beta} + \frac{\lambda \varepsilon^2}{8\beta^2} - \frac{\kappa}{4} \|\theta_1 - \theta_2\|^2 \right) \quad (\|\theta^* - \theta_1\| \leq \frac{\varepsilon}{2\beta} \text{ and } L \leq (L \vee \beta)) \end{aligned}$$

Finally, we will use the assumption in the lemma statement that $\|\theta_1 - \theta_2\| \geq \sqrt{\frac{3(16(L \vee \beta)\beta + \lambda)}{8\kappa\beta^2} \log \left(\frac{\gamma}{\varepsilon} \right)}$. Hence, we obtain

$$\begin{aligned} & \mathbb{P}_X[\mathcal{E}_{\theta_1}(X) < t \cdot \mathcal{E}_{\theta_2}(X)] \\ & \leq \sqrt{t} \cdot \exp \left(\frac{(L \vee \beta)\varepsilon}{\beta} + \frac{\lambda \varepsilon^2}{8\beta^2} - \frac{\kappa}{4} \cdot \frac{6(16(L \vee \beta)\beta + \lambda)}{8\kappa\beta^2} \log \left(\frac{\gamma}{\varepsilon} \right) \right) \\ & \quad (\|\theta_1 - \theta_2\| \geq \sqrt{\frac{3(16(L \vee \beta)\beta + \lambda)}{8\kappa\beta^2} \log \left(\frac{\gamma}{\varepsilon} \right)}) \end{aligned}$$

$$\begin{aligned}
&\leq \sqrt{t} \cdot \exp \left(-\frac{16(L \vee \beta)\beta + \lambda}{8\beta^2} \left(\log \left(\frac{\gamma}{\varepsilon} \right) - \varepsilon \right) - \frac{16(L \vee \beta)\beta + \lambda}{16\beta^2} \log \left(\frac{\gamma}{\varepsilon} \right) \right) \\
&\hspace{25em} (\varepsilon^2 \leq \varepsilon \text{ for } \varepsilon \in (0, 1)) \\
&\leq \sqrt{t} \cdot \exp \left(-\frac{16(L \vee \beta)\beta + \lambda}{16\beta^2} \log \left(\frac{\gamma}{\varepsilon} \right) - \frac{16(L \vee \beta)\beta + \lambda}{16\beta^2} \log \left(\frac{\gamma}{\varepsilon} \right) \right) \quad (\text{Lemma 4.7 (2)}) \\
&= \sqrt{t} \cdot \left(\frac{\varepsilon}{\gamma} \right)^{\frac{16(L \vee \beta)\beta + \lambda}{8\beta^2}} \leq \sqrt{t} \cdot \left(\frac{\varepsilon}{\gamma} \right)^2
\end{aligned}$$

□

4.7.2 Parameter Estimation from Truncated Samples

In this section we state the **TruncEst** subroutine and its associated guarantees. The subroutine is same as Algorithm 1 of [54] and its guarantee is a restatement of Theorem 3.1 of [54].

Algorithm 10: **TruncEst**($D, f, \varepsilon, \delta, \mathbf{c}, \Theta$): Procedure to estimate parameters from truncated samples

- 1 Set $\varepsilon \leftarrow \varepsilon/\sqrt{3}$. Set n, m and η as defined in Theorem 4.5 (requires $\mathbf{c}$). Initialize $\hat{\Theta} = \emptyset$.
 - 2 Using n samples from D , find the vanilla maximum likelihood estimate of the parameter. Let this estimate be θ_0 , that is, $\mathbb{E}_{X \sim \mathcal{E}_{\theta_0}}[T(X)] = \frac{1}{n} \sum_{i=1}^n T(X_i)$ where $\{X_i\}_{i=1}^n$ are the n samples from D .
 - 3 **for** $j = 1, \dots, 10m \lceil \log(1/\delta) \rceil$ **do**
 - 4 $Z_j = (n + j)$ -th sample from D .
 - 5 $v_{j-1} = \text{SampleGradient}(Z_j, \theta_{j-1}, f)$.
 - 6 $\theta_j = \theta_{j-1} - \eta \cdot v_{j-1}$.
 - 7 Project θ_j to $\{\theta \in \mathbb{R}^d : \|\theta - \theta_0\| \leq \frac{\varepsilon^2 + 2\beta \log(1/\alpha)}{\kappa}\} \cap \Theta$.
 - 8 **if** $j \bmod m = 0$ **then**
 - 9 Set $\hat{\Theta} \leftarrow \hat{\Theta} \cup \{\theta_j\}$ and $\theta_j \leftarrow \theta_0$.
 - 10 Return $\hat{\theta} \in \hat{\Theta}$ such that $|\{\theta \in \hat{\Theta} : \|\hat{\theta} - \theta\| \leq 2\varepsilon\sqrt{3}\}| > \frac{1}{2}|\hat{\Theta}|$.
-

Algorithm 11: **SampleGradient**(Z, θ, f)

- 1 **while** *True* **do**
 - 2 Sample $Z' \sim \mathcal{E}_{\theta}$
 - 3 **if** $f(Z') = 1$ **then**
 - 4 Return $T(Z') - T(Z)$
-

Let there be a convex set of parameters Θ such that the data D is generated from an exponential family distribution $\mathcal{E}_{\theta^*}$, $\theta^* \in \Theta$. Further, suppose the following holds:

1. $\kappa \mathbf{I} \preceq \nabla^2 W(\theta) \preceq \lambda \mathbf{I}$ for all $\theta \in \Theta$.
2. There exists a finite β such that $\mathcal{B}(\theta^*, \frac{1}{\beta}) \in \Theta$.¹ Moreover, wlog assume $\beta \geq 1$.
3. $\mathbb{P}_{X \sim \mathcal{E}_{\theta^*}}[f(X) = 1] \geq \alpha$ for some constant $\alpha > 0$.
4. The probability density function $\mathcal{E}_{\theta^*}(X)$ is log-concave in X .
5. $T(X) \in \mathbb{R}^d$ has components which are polynomial in X . Further, the maximum degree of such polynomials is at most k .

Now we define the following quantities: $d(\alpha) := \varepsilon^2 + 2\beta \log(\frac{1}{\alpha})$, $\kappa_f := \frac{1}{2} \left(\frac{\alpha^2 \exp(-6 \frac{\lambda}{\kappa^2} d(\alpha)^2)}{4Ck} \right)^{2k}$.
 $\kappa, \lambda_f := \frac{\exp(6 \frac{\lambda}{\kappa^2} d(\alpha)^2)}{\alpha^2} \cdot \lambda$, $\rho^2 := d(\lambda_f + \lambda) + (1 + \frac{2\lambda}{\kappa})^2 \left(\frac{12\beta\lambda}{\kappa^2} \cdot d(\alpha)^2 - 4\beta \log(\alpha) + \varepsilon^2 \right)^2$, $G(d, \lambda, \kappa, \alpha, \beta) := \frac{\rho^2}{\kappa_f^2 \varepsilon^2}$. Here, C is the same constant as described in [54]. We denote by $\mathbf{c} := (L, \beta, \kappa, \alpha)$.

Theorem 4.5 ([54, Theorem 3.1]) *Under the above assumptions, when $\text{TruncEst}(D, f, \varepsilon, \delta)$ is run with the values of $n \geq \frac{2\beta \log(1/\delta)}{\varepsilon^2}$, $\eta = \frac{\kappa_f \varepsilon^2}{2\rho^2} \wedge \frac{1}{\kappa_f}$ and $m \geq (G(d, \lambda, \kappa, \alpha, \beta) \vee \frac{1}{2}) \log\left(\frac{d(\alpha)}{\kappa \varepsilon^2}\right)$, then we have $\mathbb{E}[\|\theta - \theta^*\|^2] \leq \varepsilon^2$ for all $\theta \in \hat{\Theta}$. Therefore, via Markov's inequality, we also have for any $\theta \in \hat{\Theta}$, $\mathbb{P}[\|\theta - \theta^*\|^2 \geq 3\varepsilon^2] \leq \frac{1}{3}$.*

Remark 13 *By repeating the projected gradient steps (lines 3-8) $O(\log(1/\delta))$ times with fresh data every time, the probability of success can be boosted to $1 - \delta$. Suppose $\hat{\Theta} = \{\theta_{m,1}, \dots, \theta_{m,v}\}$, where $v = O(\log(1/\delta))$, be set of candidate solutions. We choose $\hat{\theta} = \theta_{m,i}$ such that $\|\theta_{m,i} - \theta_{m,j}\| \leq 2\varepsilon\sqrt{3}$ for at least $v/2$ such j 's, $j \neq i$ (see Section 3.4.5 of [26]). This ensures that with probability at least $1 - \delta$, $\|\hat{\theta} - \theta^*\| \leq \varepsilon\sqrt{3}$.*

4.7.3 Some Technical Lemmas

Lemma 4.7 *Suppose $\varepsilon \in (0, 1]$. Then we have the following:*

1. $(\sqrt{12} - \sqrt{6}) \sqrt{a \log(\frac{10b}{\varepsilon})} \geq \varepsilon$ for all $a, b \geq 1$.
2. $\frac{1}{2} \log(\frac{\gamma}{\varepsilon}) \geq \varepsilon$ for all $\gamma \geq e^2$.

Proof: For (1), let $c := \sqrt{12} - \sqrt{6} \geq 1$. Define $f(\varepsilon) := c\sqrt{a \log(10b/\varepsilon)} - \varepsilon$. Then we have $f(1) = c\sqrt{a \log(10b)} - 1 \geq 0$ since $\log(10) \geq 2$ and $a, b, c \geq 1$. Moreover, we have $f'(\varepsilon) = -\frac{c\sqrt{a}}{2\varepsilon\sqrt{\log(10b/\varepsilon)}} - 1 < 0$. Therefore, $f(\varepsilon) \geq f(1) \geq 0$ for all $\varepsilon \in (0, 1]$.

For (2), we observe that for the LHS, $\frac{1}{2} \log(\gamma/\varepsilon) = \frac{1}{2} \log(\gamma) + \frac{1}{2} \log(1/\varepsilon) \geq 1 + \frac{1}{2} \log(1/\varepsilon)$ since $\gamma \geq e^2$. However, for the RHS, $\varepsilon \leq 1$. Hence, the inequality holds. $\square$

¹This assumption is implicit in [54]. The constant in the $O(\cdot)$ depends on β in their work.

Lemma 4.8 Let $W : \Theta \rightarrow \mathbb{R}$ be a convex function defined on a convex set Θ . For $\theta_1, \theta_2 \in \Theta$, we have

$$\frac{W(\theta_1) + W(\theta_2)}{2} = W\left(\frac{\theta_1 + \theta_2}{2}\right) + \frac{1}{4} \|\theta_1 - \theta_2\|_{\nabla^2 W(\bar{\theta}) + \nabla^2 W(\tilde{\theta})}^2 ,$$

where $\bar{\theta}$ lies on the line joining θ_1 and $\frac{\theta_1 + \theta_2}{2}$, and $\tilde{\theta}$ lies on the line joining θ_2 and $\frac{\theta_1 + \theta_2}{2}$.

Proof: We have the following due to Taylor's Theorem,

$$\begin{aligned} W(\theta_1) &= W\left(\frac{\theta_1 + \theta_2}{2}\right) + \nabla W\left(\frac{\theta_1 + \theta_2}{2}\right)^\top \left(\theta_1 - \frac{\theta_1 + \theta_2}{2}\right) + \frac{1}{2} \left\| \theta_1 - \frac{\theta_1 + \theta_2}{2} \right\|_{\nabla^2 W(\bar{\theta})}^2 \\ &= W\left(\frac{\theta_1 + \theta_2}{2}\right) + \nabla W\left(\frac{\theta_1 + \theta_2}{2}\right)^\top (\theta_1 - \theta_2) + \frac{1}{8} \|\theta_1 - \theta_2\|_{\nabla^2 W(\bar{\theta})}^2 \end{aligned}$$

where $\bar{\theta}$ lies on the line joining θ_1 and $\frac{\theta_1 + \theta_2}{2}$. Similarly, we have

$$W(\theta_2) = W\left(\frac{\theta_1 + \theta_2}{2}\right) + \nabla W\left(\frac{\theta_1 + \theta_2}{2}\right)^\top (\theta_2 - \theta_1) + \frac{1}{8} \|\theta_2 - \theta_1\|_{\nabla^2 W(\tilde{\theta})}^2$$

for some $\tilde{\theta}$ on the line joining θ_2 and $\frac{\theta_1 + \theta_2}{2}$. Adding the two equations and dividing by 2 gives the desired result. $\square$

Lemma 4.9 ([54, Claim 1]) Let θ^* lie in a convex set $\Theta \subset \mathbb{R}^d$ and suppose there exists (wlog) $\beta \geq 1$ such that $\mathcal{B}(\theta^*, \frac{1}{\beta}) \subset \Theta$. Let $\mu := \mathbb{E}_{X \sim \mathcal{E}_{\theta^*}}[T(X)]$ and assume that $\nabla^2 W(\theta) \preceq \lambda \mathbf{I}$ for all $\theta \in \Theta$. Then for any vector $\mathbf{u} \in \mathbb{R}^d$ with $\|\mathbf{u}\| < \frac{1}{\beta}$,

$$\mathbb{E}_{X \sim \mathcal{E}_{\theta^*}}[\exp(\mathbf{u}^\top (T(X) - \mu))] \leq \exp\left(\frac{\|\mathbf{u}\|^2 \lambda}{2}\right) .$$

Lemma 4.10 Suppose θ_1^* and θ_2^* lie in a convex set $\Theta \subset \mathbb{R}^d$ such that $\mathcal{B}(\theta_i^*, \frac{1}{\beta}) \subset \Theta$ for all $i \in \{1, 2\}$. Let $\theta_1, \theta_2 \in \Theta$ be such that $\|\theta_i^* - \theta_i\| \leq \frac{\varepsilon}{2\beta}$, $\varepsilon \in (0, 1)$, for $i \in \{1, 2\}$. Then, $\mathcal{B}\left(\frac{\theta_1 + \theta_2}{2}, \frac{1}{2\beta}\right) \subset \Theta$.

Proof: Firstly, observe that $\mathcal{B}\left(\frac{\theta_1^* + \theta_2^*}{2}, \frac{1}{\beta}\right) \subset \Theta$ because for any point $\frac{\theta_1^* + \theta_2^*}{2} + r\mathbf{u} \in \mathcal{B}\left(\frac{\theta_1^* + \theta_2^*}{2}, \frac{1}{\beta}\right)$, where $r \in [0, \frac{1}{\beta}]$ and $\mathbf{u}$ is a unit vector, we have,

$$\frac{\theta_1^* + \theta_2^*}{2} + r\mathbf{u} = \frac{1}{2}(\theta_1^* + r\mathbf{u}) + \frac{1}{2}(\theta_2^* + r\mathbf{u}) .$$

Note that $\theta_i^* + r\mathbf{u} \in \mathcal{B}(\theta_i^*, \frac{1}{\beta})$ for $i \in \{1, 2\}$. Hence we have $\theta_i^* + r\mathbf{u} \in \Theta$ for $i \in \{1, 2\}$. Finally by convexity of Θ , we have $\frac{1}{2}(\theta_1^* + r\mathbf{u}) + \frac{1}{2}(\theta_2^* + r\mathbf{u}) \in \Theta$. This implies that $\mathcal{B}\left(\frac{\theta_1^* + \theta_2^*}{2}, \frac{1}{\beta}\right) \subset \Theta$.

Now, observe that $\left\|\left(\frac{\theta_1^* + \theta_2^*}{2}\right) - \left(\frac{\theta_1 + \theta_2}{2}\right)\right\| \leq \frac{\varepsilon}{2\beta}$ by triangle inequality. Therefore, for any unit vector $\mathbf{u}$ and $r \in [0, \frac{1}{2\beta}]$, we have,

$$\left\|\left(\frac{\theta_1^* + \theta_2^*}{2}\right) - \left(\frac{\theta_1 + \theta_2}{2} + r\mathbf{u}\right)\right\| \leq \left\|\left(\frac{\theta_1^* + \theta_2^*}{2}\right) - \left(\frac{\theta_1 + \theta_2}{2}\right)\right\| + \|r\mathbf{u}\| \leq r + \frac{\varepsilon}{2\beta} \leq \frac{1 + \varepsilon}{2\beta} \leq \frac{1}{\beta}.$$

Hence, $\frac{\theta_1 + \theta_2}{2} + r\mathbf{u} \in \mathcal{B}\left(\frac{\theta_1^* + \theta_2^*}{2}, \frac{1}{\beta}\right) \forall$ unit vectors $\mathbf{u}$ and $r \in [0, \frac{1}{\beta}]$. In other words, $\mathcal{B}\left(\frac{\theta_1 + \theta_2}{2}, \frac{1}{2\beta}\right) \subset \mathcal{B}\left(\frac{\theta_1^* + \theta_2^*}{2}, \frac{1}{\beta}\right) \subset \Theta$.

□

4.8 Discussion on Exponential Family

Here we discuss some nice properties of the exponential family and also list in Table 4.3 some examples which satisfy the set of assumptions this work makes.

Proposition 4.1 *Let $\mathcal{E}_\theta(X) := h(X) \exp(\theta^\top T(X) - W(\theta))$ be an exponential family distribution. Then we have, (i) $\nabla W(\theta) = \mathbb{E}_{X \sim \mathcal{E}_\theta}[T(X)]$, and (ii) $\nabla^2 W(\theta) = \mathbf{Cov}_{X \sim \mathcal{E}_\theta}[T(X)]$, where $\mathbf{Cov}[x] = \mathbb{E}[(x - \mathbb{E}[x])(x - \mathbb{E}[x])^\top]$.*

Remark 14 (Lipschitzness of W) *From the above lemma, it is clear that Lipschitzness of $W(\theta)$ for $\theta \in \Theta$ is equivalent to assuming that $\|\mathbb{E}_{X \sim \mathcal{E}_\theta}[T(X)]\|$ is bounded for all $\theta \in \Theta$. This is a natural assumption that is satisfied for several distributions under mild assumption on Θ . Moreover, without Lipschitzness it may be difficult to show that if θ and θ' are close, then their distributions are close, a property that we expect to hold in a learning problem. Finally, Table 4.3 enlists some common exponential family distributions and shows what the several assumptions on the exponential family look like.*

4.9 Concentration Lemmas

4.9.1 Concentration of the Negative Binomial Distribution

Lemma 4.11 ([40, Theorem 2.1]) *Let $N = \sum_{i=1}^\tau N_i$, where $\tau \geq 1$, and N_i , $i \in [\tau]$ are independent and identically distributed Geometric random variables with parameter $p \in (0, 1)$, or in other words, $N_i \stackrel{iid}{\sim} \text{Geom}(p)$. Then, for any $\lambda \geq 1$,*

$$\mathbb{P}\left[N \geq \frac{\lambda\tau}{p}\right] \leq e^{-\tau(\lambda-1-\log \lambda)}$$

Distribution	$\mathcal{E}_\theta(X)$	$\nabla W(\theta)$	$\nabla^2 W(\theta)$	Θ
Exponential	$\exp(\theta X + \log(-\theta)),$ $\theta < 0$	$-\frac{1}{\theta}$	$\frac{1}{\theta^2}$	$\kappa \leq \frac{1}{\theta^2} \leq \lambda \Leftarrow \Theta = [\frac{-1}{\sqrt{\kappa}}, \frac{-1}{\sqrt{\lambda}}],$ $L = \sqrt{\lambda} \vee 1$
Weibull (known k)	$X^{k-1} \exp(\theta \cdot X^k +$ $\log(k) + \log(-\theta)),$ $\theta < 0$	$-\frac{1}{\theta}$	$\frac{1}{\theta^2}$	$\kappa \leq \frac{1}{\theta^2} \leq \lambda \Leftarrow \Theta = [\frac{-1}{\sqrt{\kappa}}, \frac{-1}{\sqrt{\lambda}}],$ $L = \sqrt{\lambda} \vee 1$
Continuous Bernoulli	$\exp\left(\theta X - \log\left(\frac{e^\theta - 1}{\theta}\right)\right)$	$\frac{e^\theta(\theta - 1) + 1}{\theta(e^\theta - 1)}$	$\frac{(e^\theta - 1)^2 - \theta^2 e^\theta}{\theta^2(e^\theta - 1)^2}$	$ W'(\theta) \leq 1,$ $ W''(\theta) \leq \frac{1}{12} \forall \theta \in \mathbb{R},$ $W''(\theta) \geq \kappa \Leftarrow L = 1,$ $\Theta = [-f(\kappa), f(\kappa)]$ for some f
Continuous Poisson	$\frac{\exp(\theta X - W(\theta))}{\Gamma(X+1)}$	e^θ	e^θ	$\kappa \leq e^\theta \leq \lambda \Leftarrow \Theta =$ $[\log(\kappa), \log(\lambda)], L = \lambda \vee 1$
Multivariate Gaussian	$\frac{\exp(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu))}{(\det(2\pi\Sigma))^{1/2}},$ $\theta = \begin{bmatrix} \Sigma^{-1}\mu \\ \Sigma^{-1} \end{bmatrix},$ $T(X) = \begin{bmatrix} X \\ -\frac{1}{2}X X^\top \end{bmatrix}$	$\begin{bmatrix} \mu \\ -\frac{1}{2}(\Sigma + \mu\mu^\top) \end{bmatrix}$	Complicated, see Lemma 3 in [26]	$\ [\mu, -\frac{1}{2}(\Sigma + \mu\mu^\top)]^\top\ \leq L \Leftarrow$ $\mu \in \mathcal{B}(0, r_1(L))$ and $\Sigma \in \mathcal{B}(0, r_2(L))$ for some $r(L) > 0$ depending on L

Table 4.3: Different Exponential Family Distributions that satisfy the assumption under corresponding Θ as defined.

Lemma 4.12 ([40, Theorem 3.1]) *Using notations from Lemma 4.11 and for any $\lambda \leq 1$,*

$$\mathbb{P}\left[N \leq \frac{\lambda\tau}{p}\right] \leq e^{-\tau(\lambda - 1 - \log \lambda)}$$

Lemma 4.13 *Using notations from Lemma 4.11, for any $\varepsilon \in [0, 1]$,*

$$\mathbb{P}\left[(1 - \varepsilon)\frac{\tau}{p} \leq N \leq (1 + \varepsilon)\frac{\tau}{p}\right] \geq 1 - 2e^{-\frac{\tau}{4}\varepsilon^2}$$

Proof:

$$\mathbb{P}\left[(1 - \varepsilon)\frac{\tau}{p} \leq N \leq (1 + \varepsilon)\frac{\tau}{p}\right] = 1 - \mathbb{P}\left[N \geq (1 + \varepsilon)\frac{\tau}{p}\right] - \mathbb{P}\left[N \leq (1 - \varepsilon)\frac{\tau}{p}\right]$$

$$\begin{aligned}
&\geq 1 - e^{\tau(\ln(1+\varepsilon)-\varepsilon)} - e^{\tau(\varepsilon+\ln(1-\varepsilon))} \\
&\quad (\text{setting } \lambda = 1 + \varepsilon \text{ in Lemma 4.11 and } \lambda = 1 - \varepsilon \text{ in Lemma 4.12}) \\
&\geq 1 - 2e^{-\frac{\tau}{4}\varepsilon^2} \quad (\text{via Lemma 4.17})
\end{aligned}$$

□

Lemma 4.14 *Given some $\varepsilon, \delta \in (0, 1)$, let Algorithm 12 be run on a $\text{Ber}(p)$ distribution, with $\tau = \frac{4\log(2/\delta)}{\varepsilon^2}$. Then the following holds with probability at least $1 - \delta$:*

$$(1 - \varepsilon)\frac{\tau}{p} \leq N \leq (1 + \varepsilon)\frac{\tau}{p}. \quad (4.8)$$

Proof: We start by noting that Algorithm 12 keeps sampling from $\text{Ber}(p)$ till τ 1's are observed. Therefore, the number of iterations needed by the algorithm is $N = \sum_{i=1}^{\tau} N_i$, where $N_i \stackrel{\text{iid}}{\sim} \text{Geom}(p)$, are i.i.d geometric random variables, with parameter p . Essentially, N_i is the number of samples needed between observing the $(i - 1)$ -th and i -th 1.

Applying Lemma 4.13 finishes the proof. □

Corollary 4.1 *Using notations of Lemma 4.14, the estimate $\hat{p} := \frac{\tau}{N}$ returned by Algorithm 12 satisfies $(1 - \varepsilon) \leq \hat{p} \leq (1 + \varepsilon)$ with probability at least $1 - \delta$.*

Proof: This follows from straightforward rearrangement of (4.8) and the fact that $\frac{1}{1-\varepsilon} \leq 1+\varepsilon$ and $\frac{1}{1+\varepsilon} \geq 1-\varepsilon$ for all $\varepsilon \in [0, 1]$. □

Lemma 4.15 (Chernoff Bound) *Let $X = \sum_{i=1}^n X_i$, where $X_i \sim \text{Ber}(p_i)$, and all X_i are independent. Let $\mu = \mathbb{E}[X] = \sum_{i=1}^n p_i$. Then, for all $0 < \delta < 1$,*

$$\mathbb{P}[|X - \mu| \geq \varepsilon\mu] \leq 2 \exp\left(-\frac{\mu\varepsilon^2}{3}\right).$$

4.9.2 Supporting Lemmas

Lemma 4.16 *For any $\varepsilon \in [0, 1]$, we have the following inequalities: $(1 + \varepsilon) \leq e^{\varepsilon - \frac{\varepsilon^2}{4}}$ and $(1 - \varepsilon) \leq e^{-\varepsilon - \frac{\varepsilon^2}{4}}$*

Algorithm 12: Sampling from Bernoulli distribution till τ successes

Input: Desired number of successes τ

```
1 Set  $m = \tau$ 
2 Initialize:  $N = 0$  while  $m > 0$  do
3    $Y \sim \text{Ber}(p)$ 
4    $N \leftarrow N + 1$  if  $Y = 1$  then
5      $m \leftarrow m - 1$ 
6 Return:  $\tau/N$ 
```

Proof: Let $f_1(\varepsilon) = \varepsilon - \frac{\varepsilon^2}{4} - \ln(1 + \varepsilon)$. Taking derivative of f_1 , we get

$$\begin{aligned} f_1'(\varepsilon) &= 1 - \left(\frac{\varepsilon}{2} + \frac{1}{1 + \varepsilon} \right) \\ &= 1 - \left(\frac{\varepsilon^2 + \varepsilon + 2}{2 + 2\varepsilon} \right) \\ &= \frac{\varepsilon - \varepsilon^2}{2 + 2\varepsilon} \geq 0 \end{aligned} \quad (0 \leq \varepsilon \leq 1)$$

Further, $f_1(0) = 0$. Therefore, $f_1(\varepsilon) \geq 0 \forall \varepsilon \in [0, 1]$. Similarly, one can argue that $f_2(\varepsilon) := -\varepsilon - \frac{\varepsilon^2}{4} - \ln(1 - \varepsilon) \geq 0 \forall \varepsilon \in [0, 1]$. $\square$

Lemma 4.17 For any non-negative integer τ and for an $\varepsilon \in [0, 1]$, $e^{\tau(\varepsilon + \ln(1 - \varepsilon))} + e^{\tau(\ln(1 + \varepsilon) - \varepsilon)} \leq 2e^{-\frac{\tau}{4}\varepsilon^2}$

Proof: Let $f(\varepsilon) := 2e^{-\frac{\tau}{4}\varepsilon^2} - e^{\tau(\varepsilon + \ln(1 - \varepsilon))} - e^{\tau(\ln(1 + \varepsilon) - \varepsilon)}$. Simplifying,

$$\begin{aligned} f(\varepsilon) &= 2e^{-\frac{\tau}{4}\varepsilon^2} - e^{\varepsilon\tau}(1 - \varepsilon)^\tau - e^{-\varepsilon\tau}(1 + \varepsilon)^\tau \\ &\geq 2e^{-\frac{\tau}{4}\varepsilon^2} - e^{\varepsilon\tau}e^{\tau(-\varepsilon - \frac{\varepsilon^2}{4})} - e^{-\varepsilon\tau}e^{\tau(\varepsilon - \frac{\varepsilon^2}{4})} \quad (\text{Using Lemma 4.16}) \\ &\geq 0 \end{aligned}$$

$\square$

4.10 Further Experiments

All codes for experiments can be found at https://github.com/nirjhar-das/fairness_audit_partial_feedback.

Feature Name	Feature Type	Feature Range	Description
age	Numerical	[17 - 90]	Age of an individual
workclass	Categorical	7	Employment status
fnlwgt	Numerical	[13,492 - 1,490,400]	Final weight
education	Categorical	16	Highest education level
educational-num	Numerical	1 - 16	Education level (numeric)
marital-status	Categorical	7	Marital status
occupation	Categorical	14	Occupation type
relationship	Categorical	6	Relationship to others
race	Categorical	5	Race
gender	Binary	{Male, Female}	Biological sex
capital-gain	Numerical	[0 - 99,999]	Capital gains
capital-loss	Numerical	[0 - 4,356]	Capital loss
hours-per-week	Numerical	[1 - 99]	Hours worked per week
native-country	Categorical	41	Country of origin
income	Binary	$\{\leq 50K, > 50K\}$	Income level

Table 4.4: Adult Income Dataset description.

4.10.1 Experiments for the Blackbox Model

Datasets

We use the Adult Income [13] and the Law School [96] datasets to evaluate and compare the performance of our proposed **RS-Audit** (Algorithm 8) with the Baseline (Algorithm 7). On these classification datasets, we use a classifier to act as a decision-maker. The datasets already contain the true labels.

Adult Income Dataset

- The Adult Income dataset is a popular classification benchmark dataset for evaluating fairness. The description of features of the dataset is provided in Table 4.4. The true label Y is a binary variable which determines whether the annual income of a person exceeds \$50000 based on various demographic characteristics. For our setting, we choose the ‘sex’ of the person as the sensitive attribute A , and the remaining covariates as features X .
- The dataset consists of 48842 instances, each described via 15 attributes, of which 6 are numerical, 7 are categorical and 2 are binary attributes.

Law School Dataset

Feature Name	Feature Type	Feature Range	Description
decile1b	Numerical	[1.0 - 10.0]	Student's decile in Year 1
decile3	Numerical	[1.0 - 10.0]	Student's decile in Year 3
lsat	Numerical	[11.0 - 48.0]	Student's LSAT score
ugpa	Numerical	[1.5 - 4.0]	Student's undergraduate GPA
zfygpa	Numerical	[-3.35 - 3.48]	First-year law school GPA
zgpa	Numerical	[-6.44 - 4.01]	Cumulative law school GPA
fulltime	Binary	{1, 2}	Full-time or part-time status
fam inc	Categorical	5	Family income bracket
male	Binary	{0, 1}	Male or female
tier	Categorical	6	Tier
racetxt	Categorical	6	Race
pass bar	Binary	{0, 1}	Passed bar exam on first try

Table 4.5: Law School Dataset description.

- The Law school dataset (feature description in Table 4.5) contains the law school admission records across 163 law schools in the United States in the year 1991. The true label Y is a binary variable which determines whether a candidate would pass the bar exam in the first attempt or not. We define the sensitive attribute A to be the sex of the student (male or female). The remaining features act as X .
- The dataset contains information of 20798 students characterized by 12 attributes (3 categorical, 3 binary and 6 numerical attributes).

4.10.1.1 Experimental Setup

We first train a classifier f on a subsample of the dataset obtained by randomly selecting some samples from the dataset (100 for `Adult`, 5000 for `Law`). The classifier is trained to predict Y . We consider a wide range of classifiers, namely, logistic regression (LR) based classifiers trained on all the covariates including A (`all_LR`), all covariates except A (`wo_A_LR`), and a fair LR classifier trained via Exponentiated Gradient method [2] (`fair`). We also consider a random classifier that assigns outcome based on an independent draw from $\text{Ber}(1/2)$ (`random`) and a classifier that assigns the outcome equal to A (`sense_attr`). Once trained, these classifiers are frozen across various runs of the two algorithms of interest.

Next we take an inference of the classifier on the whole dataset and add the classification as a column to the dataset. To simulate historical data, we use the inference on the whole of the dataset (since past data is available for free). To simulate online samples, we randomly

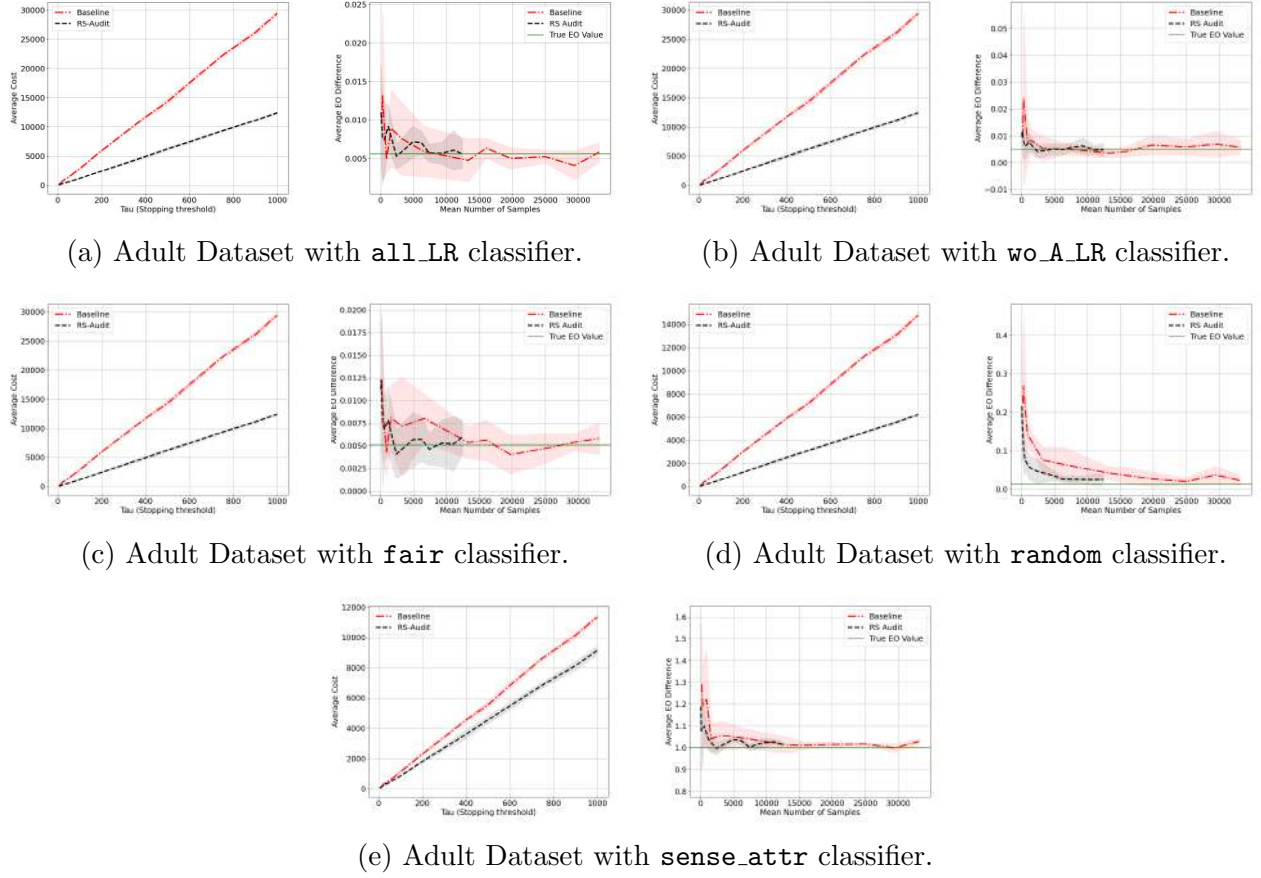
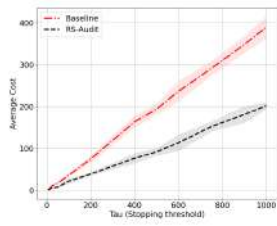


Figure 4.2: Results on the Adult Income dataset with different classifiers

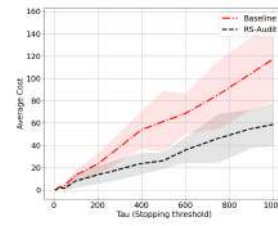
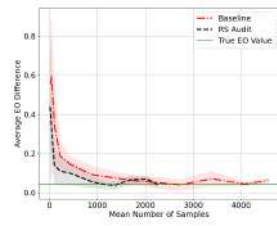
sample a row of the data and feed it to the algorithms. We vary the threshold τ over the set $\{5, 10, 50, 100, 200, 500, 1000\}$ and repeat the execution of the two algorithms over 5 seeds $\{1092, 42, 13, 729, 333\}$. At the end of execution of every algorithm, we collect the number of labels requested by the algorithm and the cost of the algorithm. We set $c_{lab} = 1$, hence cost corresponds to the number of $Y = 0$ labels requested by the algorithm. We also collect the predicted EOD ($\hat{\Delta}$) of the algorithm. The true EOD is calculated over the entire dataset.

4.10.1.2 Results

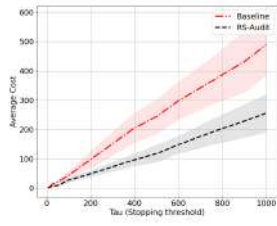
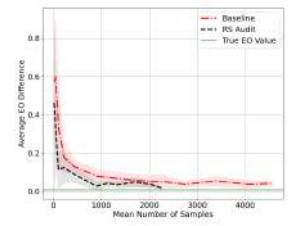
The results of the experiments are shown in Figures 4.2 and 4.3. We show two plots for each pair of classifier and dataset choice. On the left of each subplot, we show the average cost of the two algorithms `RS-Audit` and `Baseline` with respect to the stopping threshold τ . We also plot the standard deviation of the cost but it is too small to be significantly visible in the plots. On the right of each subplot, we show the average $\hat{\Delta}$ (predicted EOD) versus the average number of requested labels for every τ . The plot also shows the true EOD. From the plots, we observe that



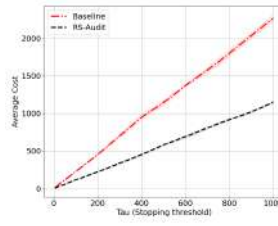
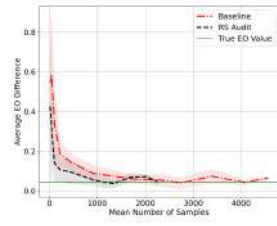
(a) Law Dataset with `all_LR` classifier.



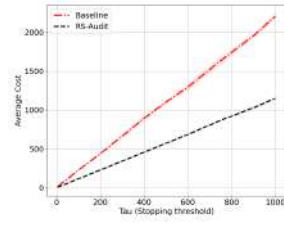
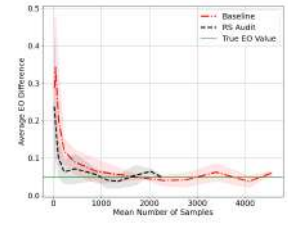
(b) Law Dataset with `wo_A_LR` classifier.



(c) Law Dataset with `fair` classifier.



(d) Law Dataset with `random` classifier.



(e) Law Dataset with `sense_attr` classifier.

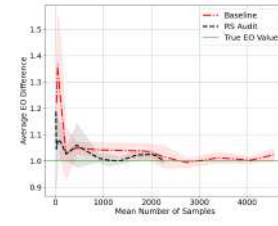


Figure 4.3: Results on the Law School dataset with different classifiers

the cost of **RS-Audit** is less than of Baseline by about 50% and that **RS-Audit** converges to the true EOD value with a much smaller number of samples, thus providing empirical validation to our theoretical results.

4.10.2 Experiments for the Mixture Model

4.10.2.1 Instance Construction

Synthetic Dataset. For the experiments under the mixture model, we construct a synthetic instance that follows the modeling assumptions (see Section 4.1 and 4.2). In our experiments, we fix $\mathcal{A} = \{0, 1\}$. We set $\mathbb{P}[A = 1] = 0.7$ and $\mathbb{P}[Y = 1 \mid A = 1] = 0.4$. We choose spherical Gaussian in $\mathbb{R}^d$, $d = 5$, as our exponential family, with covariance $\sigma^2 = 4$ across all subgroups $(y, a) \in \{0, 1\} \times \mathcal{A}$. The means are generated as follows. For a particular $a \in \mathcal{A}$, we first choose $\mu_{1,a} \in \mathbb{R}^d$ with each coordinate of $\mu_{1,a}$ from uniform distribution over $[-1, 1]$. Thereafter, we sample unit vector $\mathbf{u}$ uniformly randomly and set $\mu_{0,a} := \mu_{1,a} + r \cdot \mathbf{u}$, where $r = \Omega\left(\sqrt{\log\left(\frac{1}{\varepsilon}\right)}\right)$ as specified in the separation condition (4.1).

We set $\gamma = 0$ vary ε over the range of values $\{0.8, 0.5, 0.25, 0.1, 0.05, 0.01, 0.001\}$ and (c_{feat}, c_{lab}) over the set $\{(0.5, 0.25), (0.5, 0.5), (0.5, 1), (0.5, 3)\}$. For every combination of ε and (c_{feat}, c_{lab}) , we repeat the runs of the algorithm 5 times. We compare the algorithms **RS-Audit** (Algorithm 8) and **Exp-Audit** (Algorithm 9). To be able to complete the experiment withing reasonable time, we capped τ for **RS-Audit** at 1000 and R in **Exp-Audit** at 20000. To simulate past data, we first sample 200000 points as follows: first sample $A \sim \text{Ber}(0.7)$, then sample $Y \sim \text{Ber}(q_{1|A})$ and then $X \sim \mathcal{E}_{\theta_{Y,A}^*}$. Here, $q_{1|1} = 0.7$ and $q_{1|0} = 0.4$. Then, we take an inference of the classifier on these points. The positively classified points were sent to the **TruncEst** subroutine (Line 4, Algorithm 9). This past data generation process is also repeated with every repetition of the algorithm.

Adult Income Dataset. With the Adult Income dataset, we first use 70% of the dataset to train a 4-layer neural network with hidden layers having dimensions 64, 32 and 10 (64 is closer to input layer and 10 is closer to output layer) to predict the true labels. We train the neural net with Adam [48] optimizer for 1000 epochs. Thereafter, we obtain the embedding of the entire dataset by removing the last layer of the neural network and inferring on it. The resulting embedding is 10 dimensional. This way, we map the whole dataset to the embedding space. Hereon, by the term dataset, we will refer to this embedded dataset rather than the original one. We sample random rows from the dataset to simulate online data. We first use 1000 random samples to train a logistic regression classifier which serves as our classifier f to be audited. For simulating past database, we sample 20000 random rows from the dataset and

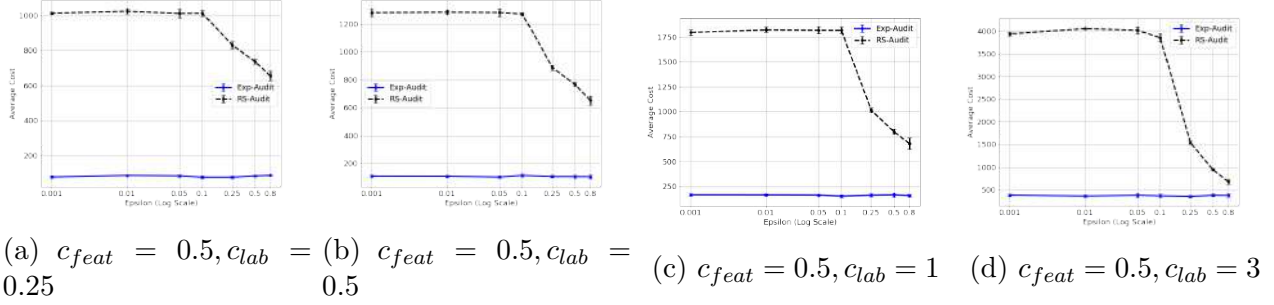


Figure 4.4: Results on the Mixture Model for different choices of c_{feat} and c_{lab} on synthetic dataset.

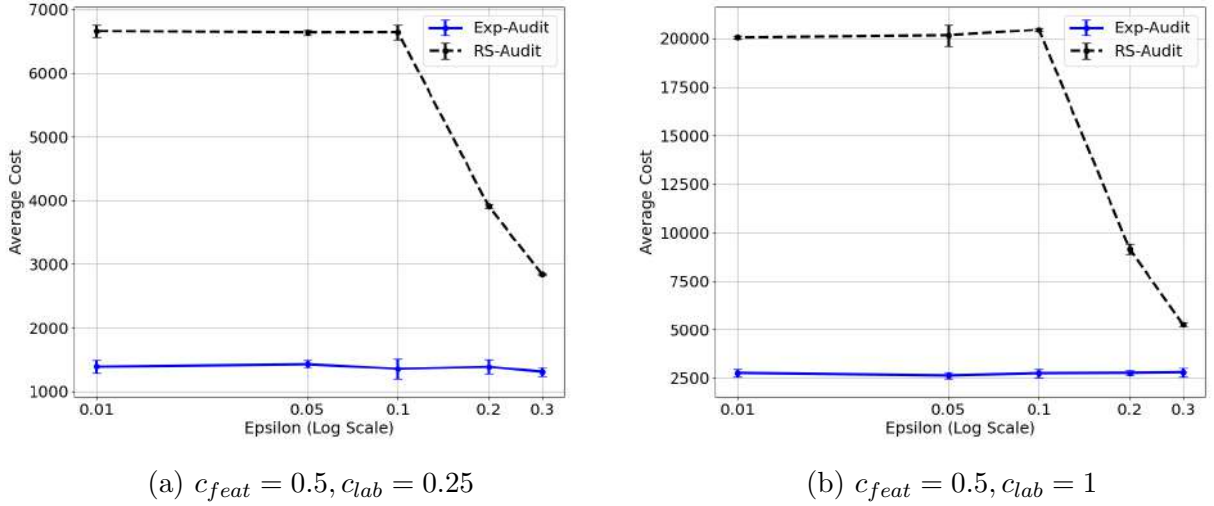


Figure 4.5: Results on the Mixture Model for different choices of c_{feat} and c_{lab} on Adult Income Dataset.

collect a subset of those rows that correspond to $f = 1$. Note that in this experiment, no structural or separation assumptions are enforced.

We set various choices of (γ, ε) , namely, $\{(0.5, 0.3), (0.4, 0.2), (0, 0.1), (0, 0.05), (0, 0.01)\}$. The values of (c_{feat}, c_{lab}) we experiment with are $\{(0.5, 0.25), (0, 1), (0.5, 1)\}$. We repeat every choice of $((\gamma, \varepsilon), (c_{feat}, c_{lab}))$ over 5 runs and report the average and standard deviations. We compare RS-Audit and Exp-Audit. To be able to complete the experiment withing reasonable time, we capped τ for RS-Audit at 1000 and R in Exp-Audit at 20000.

4.10.2.2 Results

Synthetic Dataset. In Fig. 4.4, we plot the cost of the two algorithms averaged over the different runs against ε . In every subplot, the plot shows the average cost of the two algorithms. We also show the standard deviation of each observation with error bars. We observe that the cost of **Exp-Audit** is nearly independent of ε whereas the cost of **RS-Audit** increases with decrease in ε . We also measure the fraction of the times the algorithms are able to predict the correct hypothesis in Table 4.6. From the table, we observe that **Exp-Audit** is correct at almost all times whereas **RS-Audit** has some error in the high $\varepsilon = 0.8$ case. Note that in the case that $\varepsilon = 0.25$, the assumption of the hypothesis test is not satisfied, *i.e.*, the true EOD is neither 0 nor $> \varepsilon$. Moreover, in other cases, where the algorithms are mostly correct, the true EOD is either $< \frac{\varepsilon}{2}$ or $> \varepsilon$. This is supported by our theory since we show that both the algorithms satisfy $|\hat{\Delta} - \Delta| \leq \frac{\varepsilon}{2}$ (see Eq. (4.3)).

Adult Dataset. In Fig. 4.5, we plot the the cost of the two algorithms averaged over the different runs against ε (note that cost is independent of γ as can be seen from our Theorem statements). We observe a trend similar to the synthetic dataset for the case of the Adult dataset. Moreover, the audit correctness of both **Exp-Audit** and **RS-Audit** is observed to be always 1, *i.e.*, both of these algorithms are able to correctly identify the hypothesis. This shows that our algorithm **Exp-Audit** is quite robust to the violation of structural and separation conditions.

Table 4.6: Audit correctness of **Exp-Audit** and **RS-Audit** across different values of ε and true hypotheses on Synthetic dataset. Note $\gamma = 0$.

Epsilon (ε)	True Hypothesis	Exp-Audit Correctness	RS-Audit Correctness
0.8	FAIR	1.0 ± 0.0	0.8 ± 0.4
0.5	FAIR	1.0 ± 0.0	1.0 ± 0.0
0.25	FAIR	0.0 ± 0.0	0.2 ± 0.4
0.1	UNFAIR	1.0 ± 0.0	1.0 ± 0.0
0.05	UNFAIR	1.0 ± 0.0	1.0 ± 0.0
0.01	UNFAIR	1.0 ± 0.0	1.0 ± 0.0
0.001	UNFAIR	1.0 ± 0.0	1.0 ± 0.0

4.11 Conclusion and Future Work

In this work we initiated the study of fairness audit under the partial feedback model and introduced a novel cost model to capture real-world concerns of exploring additional data. With

this cost model, we studied blackbox and mixture of exponential family models of data distributions. Our framework combines historical partial feedback data along with online samples. We provided a complete characterization of the blackbox setting with matching upper and lower bounds. In the mixture model setting, we leveraged techniques from learning with truncated samples and approximate MAP oracles to design an algorithm with significantly lower cost than the blackbox case. We complement our theoretical findings with empirical results that demonstrate the cost efficiency of our proposed algorithms. An interesting future work is to explore the audit cost complexity of unknown classifiers from a known hypothesis class, e.g., threshold classifiers. Moreover, one can also study non-uniform cost models.

Bibliography

- [1] Marc Abeille, Louis Faury, and Clément Calauzènes. Instance-wise minimax-optimal algorithms for logistic bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 3691–3699. PMLR, 2021. [69](#), [84](#)
- [2] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. A reductions approach to fair classification. In *International conference on machine learning*, pages 60–69. PMLR, 2018. [130](#)
- [3] Piyush Anand, Piotr Indyk, Ravishankar Krishnaswamy, Sepideh Mahabadi, Vikas C. Raykar, Kirankumar Shiragur, and Haike Xu. Graph-based algorithms for diverse similarity search. In *Forty-second International Conference on Machine Learning*, 2025. URL <https://openreview.net/forum?id=dmN2fQ3woH>. [3](#), [4](#), [6](#), [17](#), [27](#), [29](#), [32](#), [33](#), [34](#), [62](#)
- [4] Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Communications of the ACM*, 51(1):117–122, 2008. [3](#)
- [5] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias. In *Ethics of data and analytics*, pages 254–264. Auerbach Publications, 2022. [3](#), [93](#)
- [6] Sunil Arya, David M Mount, Nathan S Netanyahu, Ruth Silverman, and Angela Y Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM (JACM)*, 45(6):891–923, 1998. [3](#)
- [7] Giorgio Ausiello, Alessandro D. D’Atri, and Marco Protasi. Structure preserving reductions among convex optimization problems. *Journal of Computer and System Sciences*, 21(1):136–153, 1980. ISSN 0022-0000. doi: [https://doi.org/10.1016/0022-0000\(80\)90046-X](https://doi.org/10.1016/0022-0000(80)90046-X). URL <https://www.sciencedirect.com/science/article/pii/002200008090046X>. [23](#)

BIBLIOGRAPHY

- [8] Alex Ayoub, Zeyu Jia, Csaba Szepesvari, Mengdi Wang, and Lin Yang. Model-based reinforcement learning with value-targeted regression. In *International Conference on Machine Learning*, pages 463–474. PMLR, 2020. 76, 90
- [9] Yuntao Bai et al. Training a helpful and harmless assistant with reinforcement learning from human feedback, 2022. 9, 81
- [10] Dmitry Baranchuk, Artem Babenko, and Yury Malkov. Revisiting the inverted indices for billion-scale approximate nearest neighbors. In *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part XII*, page 209–224, Berlin, Heidelberg, 2018. Springer-Verlag. ISBN 978-3-030-01257-1. doi: 10.1007/978-3-030-01258-8_13. URL https://doi.org/10.1007/978-3-030-01258-8_13. 3
- [11] Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and machine learning: Limitations and opportunities*. MIT press, 2023. 9
- [12] Yahav Bechavod, Katrina Ligett, Aaron Roth, Bo Waggoner, and Steven Z Wu. Equal opportunity in online classification with partial feedback. *Advances in Neural Information Processing Systems*, 32, 2019. 10
- [13] Barry Becker and Ronny Kohavi. Adult. UCI Machine Learning Repository, 1996. DOI: <https://doi.org/10.24432/C5XW20>. 113, 129
- [14] Rachel KE Bellamy, Kuntal Dey, Michael Hind, Samuel C Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilović, et al. Ai fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias. *IBM Journal of Research and Development*, 63(4/5):4–1, 2019. 9
- [15] Alina Beygelzimer, Sham Kakade, and John Langford. Cover trees for nearest neighbor. In *Proceedings of the 23rd International Conference on Machine Learning, ICML '06*, page 97–104, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933832. doi: 10.1145/1143844.1143857. URL <https://doi.org/10.1145/1143844.1143857>. 3
- [16] Sarah Bird, Miro Dudík, Richard Edgar, Brandon Horn, Roman Lutz, Vanessa Milan, Mehrnoosh Sameki, Hanna Wallach, and Kathleen Walker. Fairlearn: A toolkit for assessing and improving fairness in ai. *Microsoft, Tech. Rep. MSR-TR-2020-32*, 2020. 9

BIBLIOGRAPHY

- [17] Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952. 8, 65
- [18] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pages 77–91. PMLR, 2018. 9
- [19] A. Camerra, E. Keogh, T. Palpanas, and J. Shieh. isax 2.0: Indexing and mining one billion time series. In *2013 IEEE 13th International Conference on Data Mining*, pages 58–67, Los Alamitos, CA, USA, dec 2010. IEEE Computer Society. doi: 10.1109/ICDM.2010.124. URL <https://doi.ieeecomputersociety.org/10.1109/ICDM.2010.124>. 2
- [20] Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336, 1998. 3
- [21] Xiaoyu Chen, Han Zhong, Zhuoran Yang, Zhaoran Wang, and Liwei Wang. Human-in-the-loop: Provably efficient preference-based reinforcement learning with general function approximation. In *International Conference on Machine Learning*, pages 3773–3793, 2022. 7, 87
- [22] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017. 7
- [23] Sam Corbett-Davies, Emma Pierson, Avi Feller, Sharad Goel, and Aziz Huq. Algorithmic decision making and the cost of fairness. In *Proceedings of the 23rd acm sigkdd international conference on knowledge discovery and data mining*, pages 797–806, 2017. 109
- [24] Sam Corbett-Davies, Johann D Gaebler, Hamed Nilforoshan, Ravi Shroff, and Sharad Goel. The measure and mismeasure of fairness. *Journal of Machine Learning Research*, 24(312):1–117, 2023. 93
- [25] Kunal Dahiya, Deepak Saini, Anshul Mittal, Ankush Shaw, Kushal Dave, Akshay Soni, Himanshu Jain, Sumeet Agarwal, and Manik Varma. Deepxml: A deep extreme multi-label learning framework applied to short text documents. In *Proceedings of the 14th*

BIBLIOGRAPHY

- International Conference on Web Search and Data Mining, WSDM '21*, New York, NY, USA, 2021. Association for Computing Machinery. 2
- [26] Constantinos Daskalakis, Themis Gouleakis, Chistos Tzamos, and Manolis Zampetakis. Efficient Statistics, in High Dimensions, from Truncated Samples . In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 639–649, Los Alamitos, CA, USA, October 2018. IEEE Computer Society. doi: 10.1109/FOCS.2018.00067. URL <https://doi.ieeecomputersociety.org/10.1109/FOCS.2018.00067>. 11, 109, 123, 126
- [27] Sanghamitra Dutta, Dennis Wei, Hazar Yueksel, Pin-Yu Chen, Sijia Liu, and Kush Varshney. Is there a trade-off between fairness and accuracy? a perspective using mismatched hypothesis testing. In *International conference on machine learning*, pages 2803–2813. PMLR, 2020. 10, 109
- [28] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226, 2012. 9, 92
- [29] Louis Faury, Marc Abeille, Kwang-Sung Jun, and Clément Calauzènes. Jointly efficient and optimal algorithms for logistic bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 546–580. PMLR, 2022. 89
- [30] Evelyn Fix and J. L. Hodges. Discriminatory analysis. nonparametric discrimination: Consistency properties. *International Statistical Review / Revue Internationale de Statistique*, 57(3):238–247, 1989. ISSN 03067734, 17515823. URL <http://www.jstor.org/stable/1403797>. 2
- [31] Cong Fu, Chao Xiang, Changxu Wang, and Deng Cai. Fast approximate nearest neighbor search with the navigating spreading-out graphs. *PVLDB*, 12(5):461 – 474, 2019. doi: 10.14778/3303753.3303754. URL <http://www.vldb.org/pvldb/vol12/p461-fu.pdf>. 3
- [32] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., USA, 1990. ISBN 0716710455. 23
- [33] Pratyush Garg, John Villasenor, and Virginia Foggo. Fairness metrics: A comparative analysis. In *2020 IEEE international conference on big data (Big Data)*, pages 3662–3666. IEEE, 2020. 9

BIBLIOGRAPHY

- [34] Marie Al Ghossein, Ching-Wei Chen, and Jason Tang. Shopping queries image dataset (sqid): An image-enriched esqi dataset for exploring multimodal learning in product search. *arXiv preprint arXiv:2405.15190*, 2024. [30](#)
- [35] Amelia Glaese et al. Improving alignment of dialogue agents via targeted human judgements. *arXiv preprint arXiv:2209.14375*, 2022. [7](#)
- [36] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29, 2016. [9](#), [92](#)
- [37] Elad Hazan et al. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016. [66](#)
- [38] Kenneth Holstein, Jennifer Wortman Vaughan, Hal Daumé III, Miro Dudik, and Hanna Wallach. Improving fairness in machine learning systems: What do industry practitioners need? In *Proceedings of the 2019 CHI conference on human factors in computing systems*, pages 1–16, 2019. [9](#)
- [39] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613, 1998. [3](#)
- [40] Svante Janson. Tail bounds for sums of geometric and exponential variables. *Statistics & Probability Letters*, 135:1–6, 2018. [125](#), [126](#)
- [41] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*, 2017. [3](#)
- [42] Nathan Kallus and Angela Zhou. Residual unfairness in fair machine learning from prejudiced data. In *International Conference on Machine Learning*, pages 2439–2448. PMLR, 2018. [10](#)
- [43] Michael Kearns and Aaron Roth. *The ethical algorithm: The science of socially aware algorithm design*. Oxford University Press, 2019. [3](#)
- [44] 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. [9](#), [10](#)

BIBLIOGRAPHY

- [45] Vijay Keswani, Anay Mehrotra, and L. Elisa Celis. Fair classification with partial feedback: an exploration-based data collection approach. In *Proceedings of the 41st International Conference on Machine Learning*, ICML’24. JMLR.org, 2024. 10, 96
- [46] Niki Kilbertus, Manuel Gomez Rodriguez, Bernhard Schölkopf, Krikamol Muandet, and Isabel Valera. Fair decisions despite imperfect predictions. In *International Conference on Artificial Intelligence and Statistics*, pages 277–287. PMLR, 2020. 10
- [47] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, 2015. 79
- [48] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015*, 2015. URL <http://arxiv.org/abs/1412.6980>. 133
- [49] I Elizabeth Kumar, Keegan E Hines, and John P Dickerson. Equalizing credit opportunity in algorithms: Aligning algorithmic fairness research with us fair lending regulation. In *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*, pages 357–368, 2022. 93
- [50] Himabindu Lakkaraju, Jon Kleinberg, Jure Leskovec, Jens Ludwig, and Sendhil Mullainathan. The selective labels problem: Evaluating algorithmic predictions in the presence of unobservables. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 275–284, 2017. 10
- [51] Richard N Landers and Tara S Behrend. Auditing the ai auditors: A framework for evaluating fairness and bias in high stakes ai predictive models. *American Psychologist*, 78(1):36, 2023. 9
- [52] Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020. 70, 71
- [53] Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020. 118
- [54] Jane Lee, Andre Wibisono, and Emmanouil Zampetakis. Learning exponential families from truncated samples. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 34826–34851. Curran Associates, 2023.

BIBLIOGRAPHY

- Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/6d5f304fb4ed0243851e41699dca4287-Paper-Conference.pdf. 11, 96, 97, 108, 109, 122, 123, 124
- [55] Junghyun Lee, Se-Young Yun, and Kwang-Sung Jun. Improved regret bounds of (multinomial) logistic bandits via regret-to-confidence-set conversion. In *International Conference on Artificial Intelligence and Statistics*, pages 4474–4482. PMLR, 2024. 73, 84, 85
- [56] Search Liaison. Google announces site diversity change to search results, 2019. URL <https://www.searchenginejournal.com/google-site-diversity-change/311557/>. 3
- [57] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982. 32
- [58] R Duncan Luce. *Individual choice behavior*. John Wiley, 1959. 8, 65
- [59] Xinsong Ma, Zekai Wang, and Weiwei Liu. On the tradeoff between robustness and fairness. *Advances in Neural Information Processing Systems*, 35:26230–26241, 2022. 10
- [60] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150. Association for Computational Linguistics, 2011. 9, 79
- [61] Satyaki Mahalanabis. Learning mixtures of gaussians with maximum-a-posteriori oracle. In Geoffrey Gordon, David Dunson, and Miroslav Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 489–497, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR. URL <https://proceedings.mlr.press/v15/mahalanabis11a.html>. 11, 96, 97, 109
- [62] Yury A. Malkov and D. A. Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *CoRR*, abs/1603.09320, 2016. URL <http://arxiv.org/abs/1603.09320>. 3
- [63] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, USA, 2008. ISBN 0521865719. 2

BIBLIOGRAPHY

- [64] Magdalen Dobson Manohar, Zheqi Shen, Guy Blelloch, Laxman Dhulipala, Yan Gu, Harsha Vardhan Simhadri, and Yihan Sun. Parlayann: Scalable and deterministic parallel graph-based approximate nearest neighbor search algorithms. In *Proceedings of the 29th ACM SIGPLAN Annual Symposium on Principles and Practice of Parallel Programming*, pages 270–285, 2024. 2
- [65] James B McQueen. Some methods of classification and analysis of multivariate observations. In *Proc. of 5th Berkeley Symposium on Math. Stat. and Prob.*, pages 281–297, 1967. 32
- [66] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM computing surveys (CSUR)*, 54(6):1–35, 2021. 9
- [67] Viraj Mehta, Vikramjeet Das, Ojash Neopane, Yijia Dai, Ilija Bogunovic, Jeff Schneider, and Willie Neiswanger. Sample efficient reinforcement learning from human feedback via active exploration. *arXiv preprint arXiv:2312.00267*, 2023. 9, 66, 70, 74, 78, 80, 81
- [68] Jakob Mökander, Jonas Schuett, Hannah Rose Kirk, and Luciano Floridi. Auditing large language models: a three-layered approach. *AI and Ethics*, 4(4):1085–1115, 2024. 9
- [69] Hervé Moulin. *Fair division and collective welfare*. MIT press, 2004. 3, 4
- [70] William Muldrew, Peter Hayes, Mingtian Zhang, and David Barber. Active preference learning for large language models. In *Proceedings of the 41st International Conference on Machine Learning*, pages 36577–36590. PMLR, 21–27 Jul 2024. 9, 70, 78, 80, 81
- [71] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions–i. *Math. Program.*, 14(1):265–294, December 1978. ISSN 0025-5610. doi: 10.1007/BF01588971. URL <https://doi.org/10.1007/BF01588971>. 27
- [72] Long Ouyang et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 2022. 7, 66
- [73] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019. 80
- [74] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger,

BIBLIOGRAPHY

- and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021. URL <https://proceedings.mlr.press/v139/radford21a.html>. 30
- [75] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 75, 81
- [76] Sayak Ray Chowdhury, Anush Kini, and Nagarajan Natarajan. Provably robust DPO: Aligning language models with noisy feedback. In *Proceedings of the 41st International Conference on Machine Learning*, pages 42258–42274. PMLR, 21–27 Jul 2024. 72
- [77] Daniel Russo and Benjamin Van Roy. Eluder dimension and the sample complexity of optimistic exploration. *Advances in Neural Information Processing Systems*, 26, 2013. 77, 90
- [78] Aadirupa Saha. Optimal algorithms for stochastic contextual preference bandits. *Advances in Neural Information Processing Systems*, 34:30050–30062, 2021. 68
- [79] Aadirupa Saha, Aldo Pacchiano, and Jonathan Lee. Dueling RL: Reinforcement learning with trajectory preferences. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, pages 6263–6289. PMLR, 25–27 Apr 2023. 7
- [80] Pedro Saleiro, Benedict Kuester, Loren Hinkson, Jesse London, Abby Stevens, Ari Anisfeld, Kit T Rodolfa, and Rayid Ghani. Aequitas: A bias and fairness audit toolkit. *arXiv preprint arXiv:1811.05577*, 2018. 9
- [81] Ayush Sawarni, Nirjhar Das, Siddharth Barman, and Gaurav Sinha. Generalized linear bandits with limited adaptivity. In *Advances in Neural Information Processing Systems*, volume 37, pages 8329–8369, 2024. 73
- [82] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 7, 79
- [83] Nian Si, Karthyek Murthy, Jose Blanchet, and Viet Anh Nguyen. Testing group fairness via optimal transport projections. In *International Conference on Machine Learning*, pages 9649–9659. PMLR, 2021. 9

BIBLIOGRAPHY

- [84] Harsha Vardhan Simhadri, Martin Aumüller, Amir Ingber, Matthijs Douze, George Williams, Magdalen Dobson Manohar, Dmitry Baranchuk, Edo Liberty, Frank Liu, Ben Landrum, et al. Results of the big ann: Neurips’23 competition. *arXiv preprint arXiv:2409.17424*, 2024. [3](#)
- [85] Connor J Stephens. Pure exploration in Multi-Armed bandits, 2023. [105](#)
- [86] Nisan Stiennon et al. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020. [7](#)
- [87] Suhas Jayaram Subramanya, Fnu Devvrit, Rohan Kadekodi, Ravishankar Krishnawamy, and Harsha Vardhan Simhadri. Diskann: Fast accurate billion-point nearest neighbor search on a single node. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d’Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 13748–13758, 2019. [3](#), [33](#)
- [88] Kohei Sugawara, Hayato Kobayashi, and Masajiro Iwasaki. On approximately searching for similar word embeddings. In *Annual Meeting of the Association for Computational Linguistics*, pages 2265–2275, 01 2016. doi: 10.18653/v1/P16-1214. [3](#)
- [89] Philip Sun, David Simcha, Dave Dopson, Ruiqi Guo, and Sanjiv Kumar. Soar: Improved indexing for approximate nearest neighbor search. In *Neural Information Processing Systems*, 2023. URL <https://arxiv.org/abs/2404.00774>. [3](#)
- [90] Bahar Taskesen, Jose Blanchet, Daniel Kuhn, and Viet Anh Nguyen. A statistical test for probabilistic fairness. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 648–665, 2021. [9](#)
- [91] Gemma Team et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024. [81](#)
- [92] TensorFlow. Tensorflow datasets. <https://www.tensorflow.org/datasets/catalog/>, 2025. Accessed: 2025-07-10. [32](#), [33](#)
- [93] J. Wang, J. Wang, G. Zeng, Z. Tu, R. Gan, and S. Li. Scalable k-nn graph construction for visual descriptors. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1106–1113, June 2012. doi: 10.1109/CVPR.2012.6247790. [2](#)

BIBLIOGRAPHY

- [94] Dennis Wei. Decision-making under selective labels: Optimal finite-domain policies and beyond. In *International conference on machine learning*, pages 11035–11046. PMLR, 2021. 10
- [95] August Wester. arxiv openai embeddings. <https://www.kaggle.com/datasets/awester/arxiv-embeddings>, 2022. Accessed: 2025-07-10. 30
- [96] L.F. Wightman, H. Ramsey, and Law School Admission Council. *LSAC National Longitudinal Bar Passage Study*. LSAC research report series. Law School Admission Council, 1998. URL <https://books.google.co.in/books?id=dtA7AQAAIAAJ>. 113, 129
- [97] Shangyu Wu, Ying Xiong, Yufei Cui, Haolun Wu, Can Chen, Ye Yuan, Lianming Huang, Xue Liu, Tei-Wei Kuo, Nan Guan, et al. Retrieval-augmented generation for natural language processing: A survey. *arXiv preprint arXiv:2407.13193*, 2024. 2
- [98] Songkai Xue, Mikhail Yurochkin, and Yuekai Sun. Auditing ml models for individual bias and unfairness. In *International Conference on Artificial Intelligence and Statistics*, pages 4552–4562. PMLR, 2020. 9, 10
- [99] Wenqian Ye, Guangtao Zheng, Xu Cao, Yunsheng Ma, and Aidong Zhang. Spurious correlations in machine learning: A survey. *arXiv preprint arXiv:2402.12715*, 2024. 9
- [100] Han Zhao and Geoffrey J Gordon. Inherent tradeoffs in learning fair representations. *Journal of Machine Learning Research*, 23(57):1–26, 2022. 109
- [101] Banghua Zhu, Michael Jordan, and Jiantao Jiao. Principled reinforcement learning with human feedback from pairwise or k-wise comparisons. In *Proceedings of the 40th International Conference on Machine Learning*, pages 43037–43067. PMLR, 2023. 7, 72