

UC Santa Barbara

UC Santa Barbara Electronic Theses and Dissertations

Title

Towards Trustworthy Machine Learning: Fairness Under Dynamics and Counterfactuals

Permalink

<https://escholarship.org/uc/item/0r1882jr>

ISBN

9798186386117

Author

Guldogan, Ozgur

Publication Date

2026-06-10

Peer reviewed|Thesis/dissertation

University of California
Santa Barbara

Towards Trustworthy Machine Learning: Fairness Under Dynamics and Counterfactuals

A dissertation submitted in partial satisfaction
of the requirements for the degree

Doctor of Philosophy
in
Electrical and Computer Engineering

by

Ozgur Guldogan

Committee in charge:

Professor Ramtin Pedarsani, Chair
Professor Haewon Jeong
Professor Mahnoosh Alizadeh
Professor Upamanyu Madhow

June 2026

The Dissertation of Ozgur Guldogan is approved.

Professor Haewon Jeong

Professor Mahnoosh Alizadeh

Professor Upamanyu Madhow

Professor Ramtin Pedarsani, Committee Chair

May 2026

Towards Trustworthy Machine Learning: Fairness Under Dynamics and Counterfactuals

Copyright © 2026

by

Ozgur Guldogan

To my family and friends, whose love, support, and patience
made this journey possible

Acknowledgements

I would first like to thank my advisor, Ramtin Pedarsani. I could not have asked for a better advisor. Ramtin gave me the freedom to explore different research directions, while helping me find clarity when ideas were still uncertain. He always encouraged me to think deeply, communicate clearly, and trust the research process. I am deeply grateful for his guidance, patience, support, and trust. His mentorship has shaped the way I think about research, and the lessons I learned from him will stay with me throughout my career.

I would also like to thank my dissertation committee members, Upamanyu Madhow, Mahnoosh Alizadeh, and Haewon Jeong, for their time, feedback, and support. I am especially grateful to Upamanyu for serving on my committee, for our collaboration on long-term fairness, and for the sharp research questions that challenged me to think more deeply and clearly. I also thank Mahnoosh and Haewon for the valuable courses I took with them, which shaped my graduate education, and for their thoughtful feedback, encouragement, and perspective throughout this process.

I have been fortunate to collaborate with many wonderful researchers during my Ph.D. I would like to thank my coauthors and collaborators: Yuchen Zeng, Bhagyashree Puranik, Jy-yong Sohn, Neeraj Sarna, Yuanyuan Li, Michael Berger, Jackson Kunde, and Kangwook Lee. Each collaboration taught me something different, and I am grateful for the ideas, discussions, and shared effort that made this work possible. Yuchen, thank you for being a wonderful collaborator and for the many thoughtful discussions that helped shape my early research. Bhagya, thank you for being such a generous collaborator and friend. Working with you made the research process much more rewarding, and I am grateful for the many conversations that helped me think through my research and the next steps in my career. Kangwook, thank you for your mentorship, insight, and support

during all of our collaborations. I learned a great deal from your clarity, research taste, and ability to see the broader picture behind technical problems.

I am also grateful to the faculty, staff, and students at UCSB and Boğaziçi University who shaped my education and supported me along the way. I thank all of the faculty for the courses and discussions that influenced my graduate studies, and the students I had the opportunity to teach and work with as a teaching assistant. Teaching was one of the most rewarding parts of my Ph.D., and I am especially grateful to Jason Marden for making my experience as a teaching assistant so enjoyable and meaningful. I would especially like to thank Erika; this degree would have been inordinately more difficult and stressful without all of her help and support.

I am thankful to my office mates, Mark, Arghavan, Nanfei, Sajjad, Ziyue, Irmak, and Berkay, for their friendship, support, and many helpful conversations. The office discussions, meals, coffee breaks, and pickleball games made graduate school in Santa Barbara much more enjoyable.

I want to thank my friends in Santa Barbara for making this place feel like a home away from home. The dinners, conversations, hikes, trips, and all of the random small moments we shared made this journey much happier and much less lonely. Thank you to Spencer, Brandon, Nibodh, Nolan, Ursula, Kaylaa, Nathalie, Sasha, Charlie, Ben, Mathias, PK, Marianna, Kemal, Canan, Zeki, Katie, Hanna, Tyler, Olivia, Ash, and many others.

I also want to thank my friends from undergrad. Many of you have been with me for a very long time, even from far away, and I am grateful for the memories, support, and friendship we have shared over the years. Thank you to Gökberk, Burak, Berk, Alperen, Hamit, Yiğit, Ege, İrem, İbrahim, Gökhan, and many others.

A special thanks goes to Alex, whose love, patience, and support made this journey unforgettable. You were there through the stressful deadlines, the uncertain moments,

and the happiest parts of these years. The games, walks, and conversations we've shared made this chapter of my life warmer and more joyful.

Finally, I would like to thank my family. I am especially grateful to my brother, Ufuk, my dad, Hasan, and my mom, Ayşe, my first teacher in life. Their love, sacrifices, and belief in me gave me the foundation and motivation to be here today. Even from far away, their support was always with me, especially during the most difficult moments of this journey. They taught me the value of education, hard work, and kindness, and this accomplishment would not have been possible without them.

Curriculum Vitæ

Ozgur Guldogan

Education

2026	Ph.D. in Electrical and Computer Engineering (Expected) University of California, Santa Barbara.
2024	M.S. in Electrical and Computer Engineering University of California, Santa Barbara.
2021	B.S. in Electrical and Electronics Engineering Boğaziçi University.
2021	B.S. in Mathematics Boğaziçi University.

Publications

1. **O. Guldogan**, N. Sarna, Y. Li, and M. Berger, “Counterfactually Fair Conformal Prediction,” International Conference on Artificial Intelligence and Statistics (AISTATS), 2026.
2. **O. Guldogan**, J. Kunde, K. Lee, and R. Pedarsani, “Multi-Bin Batching for Increasing LLM Inference Throughput,” under review.
3. B. Puranik*, **O. Guldogan***, U. Madhow, and R. Pedarsani, “Long-Term Fairness in Sequential Multi-Agent Selection with Positive Reinforcement,” IEEE Journal on Selected Areas in Information Theory, 2024.
4. **O. Guldogan***, Y. Zeng*, J. Sohn, R. Pedarsani, and K. Lee, “Equal Improvability: A New Fairness Notion Considering the Long-term Impact,” International Conference on Learning Representations (ICLR), 2023.

* Equal contribution.

Awards and Service

2026	Electrical and Computer Engineering Department Dissertation Fellowship, University of California, Santa Barbara.
2023	Outstanding ECE Teaching Assistant Award, University of California, Santa Barbara.
2021	UC Regents Fellowship in Electrical and Computer Engineering.
2022–2026	Reviewer for FAccT, ICLR, NeurIPS, and IEEE Transactions on Networking.

Abstract

Towards Trustworthy Machine Learning: Fairness Under Dynamics and Counterfactuals

by

Ozgur Guldogan

Machine learning systems increasingly make decisions with long-term consequences for individuals and society. Designing trustworthy machine learning systems therefore requires going beyond static accuracy and one-shot fairness criteria. Decisions may alter future populations through feedback, individuals may improve and reapply, uncertainty may call for set-valued predictions rather than point estimates, and large-scale deployment may be constrained by the efficiency of modern generative models. This thesis explores these challenges through contributions spanning fairness under dynamics, counterfactual fairness under predictive uncertainty, and efficient inference serving for large language models.

We first study fairness in dynamic decision-making settings such as lending, hiring, and admissions, where rejected individuals may invest effort to improve and reapply. We introduce Equal Improvability, a fairness notion requiring that rejected individuals from different groups have comparable opportunities to become accepted after bounded effort. We develop optimization-based methods for learning classifiers that satisfy this criterion and show that they can promote more equitable long-term population dynamics. Next, we consider sequential selection by multiple decision-makers from a shared applicant pool. We propose fair-greedy policies that balance utility with long-term representation targets, characterize convergence under positive reinforcement dynamics, and show that policy design must account for more complex feedback mechanisms to avoid unintended negative reinforcement.

We then address fairness under counterfactual reasoning and predictive uncertainty. Counterfactually Fair Conformal Prediction constructs prediction sets that are both statistically valid and counterfactually fair. By aggregating conformity scores across counterfactual interventions on protected attributes, the method produces prediction sets that remain invariant under counterfactual changes to the protected attribute while preserving finite-sample marginal coverage guarantees.

Finally, we study computational efficiency in generative machine learning systems. Multi-Bin Batching is an inference-serving policy for large language models that groups requests by predicted generation lengths to reduce inefficiency from heterogeneous runtimes. Queueing-theoretic analysis characterizes the throughput benefits of length-aware batching and shows that the effective service rate approaches the idealized static-batching capacity as the number of bins increases. Experiments on simulated and real inference workloads demonstrate substantial throughput improvements over static batching.

Together, these contributions provide fairness notions and algorithms that remain meaningful under feedback, uncertainty, and counterfactual interventions, while also developing principled serving mechanisms that support scalable deployment of generative models.

The material in this dissertation is based in part on the following publications:

- O. Guldogan*, Y. Zeng*, J. Sohn, R. Pedarsani, and K. Lee, “Equal Improvability: A New Fairness Notion Considering the Long-term Impact,” *International Conference on Learning Representations (ICLR)*, 2023.
- © 2024 IEEE. Reprinted, with permission from B. Puranik*, O. Guldogan*, U. Madhow, and R. Pedarsani, “Long-Term Fairness in Sequential Multi-Agent Selection with Positive Reinforcement,” special issue on Information-Theoretic Methods for Trustworthy and Reliable ML, *IEEE Journal on Selected Areas in Information Theory*, 2024.
- O. Guldogan, N. Sarna, Y. Li, and M. Berger, “Counterfactually Fair Conformal Prediction,” *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2026.

Contents

Curriculum Vitae	viii
Abstract	ix
1 Introduction	1
1.1 Overview of Contributions	3
2 Equal Improvability for Long-Term Fairness	8
2.1 Introduction	8
2.2 Equal Improvability	11
2.3 Achieving Equal Improvability	17
2.4 Experiments	20
2.5 Related Work	28
2.6 Conclusion	29
3 Long-Term Fairness in Sequential Multi-Agent Selection	30
3.1 Introduction	30
3.2 Related Work	33
3.3 Modeling Multi-Agent Decision-Making	37
3.4 Variations on the Pool Evolution Model	46
3.5 Experimental Evaluation	52
3.6 Conclusion	59
4 Counterfactually Fair Conformal Prediction	61
4.1 Introduction	61
4.2 Background: Conformal Prediction	65
4.3 Counterfactual Fairness	67
4.4 Counterfactually Fair Prediction Sets	68
4.5 Experiments	75
4.6 Conclusion	81

5	Multi-Bin Batching for LLM Inference	83
5.1	Introduction	83
5.2	Related Work	86
5.3	Problem Setup and the Multi-Bin Batching Algorithm	89
5.4	Throughput Analysis	92
5.5	Latency Analysis	99
5.6	Analysis under Exponential Distribution	102
5.7	Experiments	105
5.8	Conclusion	112
A	Supplementary Materials for Chapter 2	113
A.1	Theoretical Results	113
A.2	Supplementary Materials on the Fairness Notions	119
A.3	Supplementary Materials on Experiments	130
A.4	Examples: Investigating the One-Step Impact of Fairness Notions	143
B	Supplementary Materials for Chapter 3	153
B.1	Score-Based Reward Optimization	153
B.2	Convergence Under MFG Policy	158
B.3	Role-Model Negative Feedback	164
B.4	Additional Experiments	167
C	Supplementary Materials for Chapter 4	170
C.1	Proofs	170
C.2	Experimental Details	172
C.3	Additional Results	176
D	Supplementary Materials for Chapter 5	180
D.1	Proofs	180
	Bibliography	187

Chapter 1

Introduction

Machine learning (ML) has rapidly moved from a collection of predictive tools to a central component of modern decision-making and computational infrastructure. ML systems now support decisions in domains such as hiring, lending, healthcare, recommendation, and scientific discovery, while deep learning has enabled substantial progress across perception and prediction tasks [1]. More recently, transformer architectures [2] and large language models [3] have expanded the role of ML systems to text generation, code generation, reasoning, and multimodal interaction. These developments are part of a broader shift toward foundation models, whose capabilities and risks have motivated renewed attention to responsible and trustworthy ML systems [4].

However, strong empirical performance alone is not sufficient for many real-world deployments. In high-stakes settings, ML systems interact with individuals, institutions, and populations over time. A lending or admissions decision can influence whether an individual is able or motivated to improve and reapply, while a hiring or admissions policy can change the composition of future applicant pools. A prediction about an individual may also be judged not only by whether it is accurate, but by whether it would remain appropriate under a counterfactual change to a protected attribute. At

the same time, modern generative models introduce another deployment challenge: even useful and accurate models can be difficult to serve efficiently at scale because inference is costly and request lengths are highly heterogeneous.

Classical group fairness notions have played an important role in formalizing disparities across sensitive groups. Demographic parity and related parity-based notions aim to control differences in selection rates across groups [5, 6], while equal opportunity and equalized odds focus on disparities in error rates conditioned on the true label [7]. These notions are useful, but often static. They evaluate a classifier at a single point in time, without modeling how individuals respond to decisions or how decisions affect future populations. A classifier may satisfy a parity constraint while still placing rejected individuals from one group much farther from acceptance than rejected individuals from another group. Similarly, a policy that appears fair in one round of selection may produce delayed effects that either improve or worsen outcomes for a group [8]. Such feedback effects can also change future data distributions and representation over time [9, 10], motivating fairness notions and algorithms that account for long-term consequences.

Another challenge is to reason about fairness when predictions are uncertain and protected attributes have causal effects. Counterfactual fairness asks whether a prediction would remain unchanged had an individual’s protected attribute been different in a causal model [11]. This individual-level viewpoint is especially relevant when protected attributes may influence predictions through complex causal pathways. However, many prediction problems require more than a single point estimate because uncertainty may be intrinsic, data may be limited, or overconfident predictions may be undesirable. Conformal prediction addresses uncertainty by producing prediction sets with finite-sample coverage guarantees under exchangeability [12], and has become a practical framework for uncertainty quantification in modern ML systems [13]. Combining counterfactual fairness with conformal prediction therefore raises a natural question: can prediction

sets be constructed to be both statistically valid and invariant to counterfactual changes in protected attributes?

Trustworthy deployment also requires efficient and scalable inference. This is particularly important for LLMs, where autoregressive generation is computationally expensive and output lengths vary substantially across requests. Static batching can waste computation when short requests are processed together with long ones, while serving systems such as Orca use iteration-level scheduling to improve utilization [14]. Memory management and batching mechanisms such as PagedAttention further illustrate the importance of system-level design for scalable LLM serving [15]. In this sense, efficient inference complements the fairness and uncertainty questions above: all three concern how ML systems behave under the constraints of real deployment.

This dissertation studies trustworthy machine learning through a deployment-aware lens, focusing on three directions: fairness under dynamics, counterfactual fairness under predictive uncertainty, and efficient inference serving for large language models.

1.1 Overview of Contributions

This section summarizes the four main contributions of the dissertation in the order in which they appear.

1.1.1 Equal Improvability for Long-Term Fairness

We begin with dynamic decision-making settings such as lending, hiring, and admissions, where rejected individuals may improve their qualifications and reapply in the future. In Chapter 2, we introduce Equal Improvability (EI), a group fairness notion designed to capture the future opportunity available to rejected individuals [16]. Standard fairness notions may equalize immediate outcomes, but they do not necessarily ensure

that rejected individuals from different groups are similarly positioned to become accepted after making reasonable effort. EI addresses this gap by requiring that, among rejected individuals, the probability of becoming accepted after bounded effort is comparable across groups.

The motivation behind EI is that rejection is not always the end of a decision-making process. A rejected individual who is close to the decision boundary may have a realistic path to acceptance, while one who is far from the boundary may face discouraging or impractical effort requirements. If these differences are systematically group-dependent, then a classifier can create unequal future opportunities even when it appears fair according to a static criterion. We develop methods for learning EI-regularized classifiers by incorporating differentiable penalties based on the improbability of rejected samples into empirical risk minimization. We also analyze the relationship between EI and existing fairness notions, and show through experiments on synthetic and real datasets that EI can reduce improbability disparities while maintaining competitive predictive performance. We further study a dynamic model in which individuals improve over multiple rounds, demonstrating that EI can reduce long-term unfairness in evolving feature distributions.

This chapter shows that fairness should consider not only who is accepted now, but also whether rejected individuals have a fair chance to be accepted in the future.

1.1.2 Long-Term Fairness in Sequential Multi-Agent Selection

We next consider sequential selection systems with multiple decision-makers competing for a shared applicant pool, such as universities selecting students or companies hiring from the same labor market. In Chapter 3, we study a model in which multiple agents select applicants over repeated rounds, and the composition of the applicant pool evolves through positive reinforcement [17]. The motivating hypothesis is that selecting a higher

proportion of applicants from an underrepresented group can increase the future participation of that group through mechanisms such as visibility, role models, and improved expectations.

We propose the Multi-agent Fair-Greedy (MFG) policy, under which agents balance score maximization with a long-term representation target. Under a pure positive reinforcement model, we characterize the optimal actions of agents and prove convergence of the applicant pool and admission proportions to the desired fairness target, showing that decentralized agents can collectively promote long-term representation goals under suitable dynamics. The chapter also shows that positive feedback is not guaranteed: under a role-model reinforcement model, uncoordinated decentralized decisions can produce negative feedback and reduce the representation of the underrepresented group over time. We show that a centralized version of the MFG policy can restore positive reinforcement, and experiments on synthetic and semi-synthetic datasets illustrate how long-term outcomes depend on both population dynamics and coordination among decision-makers.

These results show that positive reinforcement can promote long-term fairness, but only when the feedback model and coordination structure are carefully accounted for.

1.1.3 Counterfactually Fair Conformal Prediction

We then move from dynamic decision-making to prediction under uncertainty, where a single predicted label or value may not be sufficient. In Chapter 4, we introduce Counterfactually Fair Conformal Prediction (CF-CP), a framework for constructing prediction sets that are both statistically valid and counterfactually fair [18]. Conformal prediction provides a distribution-free framework for constructing prediction sets with finite-sample marginal coverage guarantees, but standard conformal prediction does not ensure that the resulting sets are counterfactually fair with respect to protected attributes.

CF-CP is motivated by the principle that a prediction set for an individual should not change under a counterfactual intervention on a protected attribute, when the individual’s latent background variables are held fixed. This extends counterfactual fairness from point predictions to set-valued predictions, where fairness should concern not only the model’s most likely prediction, but also the full set of plausible outcomes presented by the system. The proposed approach constructs conformity scores by aggregating model behavior across counterfactual versions of each calibration point. Calibrating with these aggregated scores produces prediction sets that are invariant across counterfactual interventions on the protected attribute while preserving the finite-sample marginal coverage guarantee of split conformal prediction. The framework is training-free in the sense that it can be applied on top of a fixed predictive model, together with a structural causal model used to generate counterfactuals.

This chapter shows that uncertainty quantification and counterfactual fairness can be integrated directly at the level of prediction sets.

1.1.4 Multi-Bin Batching for LLM Inference

The final contribution addresses the efficiency of generative ML systems, where inference can be expensive and request lengths can vary substantially. In Chapter 5, we propose Multi-Bin Batching, a length-aware batching policy that groups requests according to predicted generation lengths [19]. LLM inference is challenging because generation proceeds autoregressively: each output token depends on the previously generated tokens. As a result, serving many requests efficiently requires careful batching and scheduling.

In static batching, requests in the same batch are typically processed together until the longest generation finishes. This can lead to substantial inefficiency when short and long requests are mixed in the same batch, since shorter requests may waste computation

while waiting for longer ones. Multi-Bin Batching reduces this inefficiency by partitioning requests into bins based on estimated output length and forming batches within each bin. We analyze this policy using a queueing-theoretic framework that isolates the impact of output-length heterogeneity, showing that length-aware binning improves the effective batched service rate and approaches an idealized static-batching capacity as the number of bins increases. Experiments on simulated and real inference workloads demonstrate substantial throughput improvements over standard static batching and clarify its relationship to continuous batching baselines.

This chapter provides a practical mechanism for improving the scalability of LLM inference through simple length-aware scheduling.

Chapter 2

Equal Improvability for Long-Term Fairness

2.1 Introduction

Over the past decade, machine learning has been used in a wide variety of applications. However, these machine learning approaches are observed to be unfair to individuals across different ethnicity, race, and gender. As the implicit bias in artificial intelligence tools raised concerns over potential discrimination and equity issues, various researchers suggested defining fairness notions and developing classifiers that achieve fairness. One popular fairness notion is *demographic parity* (DP), which requires the decision-making system to provide output such that the groups are equally likely to be assigned to the desired prediction classes, *e.g.*, acceptance in the admission procedure. DP and related fairness notions are widely used to mitigate the bias in many realistic problems such as recruitment, credit lending, and university admissions [20, 7, 5, 21].

However, most of the existing fairness notions only focus on immediate fairness, without taking potential follow-up inequity risk into consideration. In Figure 2.1, we provide

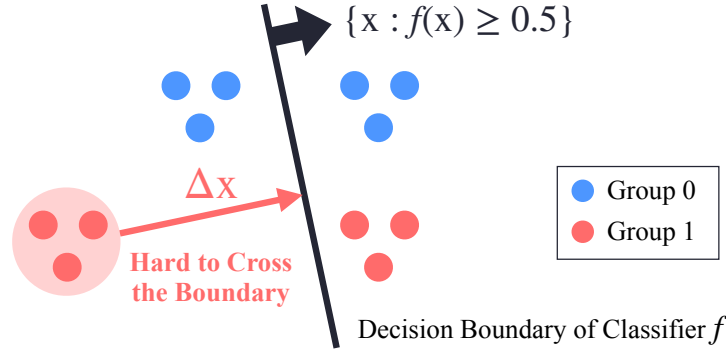


Figure 2.1: Toy example showing the insufficiency of fairness notion that does not consider improvability. We consider the binary classification (accept/reject) on 12 samples (dots), where $\mathbf{x}$ is the feature of the sample and the color of the dot represents the group. The given classifier f is fair in terms of a popular notion called demographic parity (DP), but does not satisfy equal improvability of rejected samples ($f(\mathbf{x}) < 0.5$) in two groups; the rejected samples in group 1 need more effort $\Delta \mathbf{x}$ to be accepted, *i.e.*, $f(\mathbf{x} + \Delta \mathbf{x}) \geq 0.5$, compared with the rejected samples in group 0.

an example scenario when using DP fairness has a long-term fairness issue, in a simple loan approval problem setting. Consider two groups (group 0 and group 1) with different distributions, where each individual has one label (approve the loan or not) and two features (credit score, income) that can be improved over time. Suppose each group consists of two clusters (with three samples each), and the distance between the clusters is different for two groups. Figure 2.1 visualizes the distributions of two groups and the decision boundary of a classifier f which achieves DP among the groups. We observe that the rejected samples (left-hand-side of the decision boundary) in group 1 are located further away from the decision boundary than the rejected samples in group 0. As a result, the rejected applicants in group 1 need to make more effort to cross the decision boundary and get approval. This improvability gap between the two groups can make the rejected applicants in group 1 less motivated to improve their features, which may increase the gap between different groups in the future.

This motivated the advent of fairness notions that consider dynamic scenarios when each rejected sample makes effort to improve its feature, and measure the group fairness

after such effort is made [22, 23, 24]. However, as shown in Table 2.1, they have various limitations *e.g.*, vulnerable to imbalanced group negative rates or outliers.

In this chapter, we introduce another fairness notion designed for dynamic scenarios, dubbed as *Equal Improvability* (EI), which does not suffer from these limitations. Let $\mathbf{x}$ be the feature of a sample and f be a score-based classifier, *e.g.*, predicting a sample as accepted if $f(\mathbf{x}) \geq 0.5$ holds and as rejected otherwise. We assume each rejected individual wants to get accepted in the future, thus improving its feature within a certain effort budget towards the direction that maximizes its score $f(\mathbf{x})$. Under this setting, we define EI fairness as the equity of the potential acceptance rate of the different rejected groups, once each individual makes the best effort within the predefined budget. This prevents the risk of exacerbating the gap between different groups in the long run.

The main contributions are summarized below. This chapter is based in part on our published work [16].

- We propose a new group fairness notion called *Equal Improvability* (EI), which aims to equalize the probability of rejected samples being qualified after a certain amount of feature improvement, for different groups. EI encourages rejected individuals in different groups to have an equal amount of motivation to improve their feature to get accepted in the future. We analyze the properties of EI and the connections of EI with other existing fairness notions.
- We provide three methods to find a classifier that is fair in terms of EI, each of which uses a unique way of measuring the inequity in the improvability. Each method is solving a min-max problem where the inner maximization problem is finding the best effort to measure the EI unfairness, and the outer minimization problem is finding the classifier that has the smallest fairness-regularized loss. Experiments on synthetic/real datasets demonstrate that our algorithms find classifiers having

low EI unfairness.

- We run experiments on dynamic scenarios where the data and the classifier evolve over multiple rounds, and show that training a classifier with EI constraints is beneficial for making the feature distributions of different groups identical in the long run.

2.2 Equal Improvability

Before defining our new fairness notion called *Equal Improvability* (EI), we first introduce necessary notations. For an integer n , let $[n] = \{0, \dots, n-1\}$. We consider a binary classification setting where each data sample has an input feature vector $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^d$ and a label $y \in \mathcal{Y} = \{0, 1\}$. In particular, we have a sensitive attribute $z \in \mathcal{Z} = [Z]$, where Z is the number of sensitive groups. As suggested in [25], we sort d features $\mathbf{x} \in \mathbb{R}^d$ into three categories: improvable features $\mathbf{x}_I \in \mathbb{R}^{d_I}$, manipulable features $\mathbf{x}_M \in \mathbb{R}^{d_M}$, and immutable features $\mathbf{x}_{IM} \in \mathbb{R}^{d_{IM}}$, where $d_I + d_M + d_{IM} = d$ holds. Here, improvable features $\mathbf{x}_I$ refer to the features that can be improved and can directly affect the outcome, *e.g.*, salary in the credit lending problem, and GPA in the school's admission problem. In contrast, manipulable features $\mathbf{x}_M$ can be altered, but are not directly related to the outcome, *e.g.*, marital status in the admission problem, and communication type in the credit lending problem. Although individuals may manipulate these manipulable features to get the desired outcome, we do not consider it as a way to make efforts as it does not affect the individual's true qualification status. Immutable features $\mathbf{x}_{IM}$ are features that cannot be altered, such as race, age, or date of birth. Note that if sensitive attribute z is included in the feature vector, then it belongs to immutable features. For ease of notation, we write $\mathbf{x} = (\mathbf{x}_I, \mathbf{x}_M, \mathbf{x}_{IM})$. Let $\mathcal{F} = \{f : \mathcal{X} \rightarrow [0, 1]\}$ be the set of classifiers, where each classifier is parameterized by $\mathbf{w}$, *i.e.*, $f = f_{\mathbf{w}}$. Given $f \in \mathcal{F}$, we consider the

following deterministic prediction: $\hat{y} \mid \mathbf{x} = \mathbf{1}\{f(\mathbf{x}) \geq 0.5\}$ where $\mathbf{1}\{A\} = 1$ if condition A holds, and $\mathbf{1}\{A\} = 0$ otherwise. We now introduce our new fairness notion.

Definition 2.1 (Equal Improvability) *Define a norm $\mu : \mathbb{R}^{d_I} \rightarrow [0, \infty)$. For a given constant $\delta > 0$, a classifier f is said to achieve equal improvability with δ -effort if*

$$\mathbb{P} \left(\max_{\mu(\Delta \mathbf{x}_I) \leq \delta} f(\mathbf{x} + \Delta \mathbf{x}) \geq 0.5 \mid f(\mathbf{x}) < 0.5, z = z \right) = \mathbb{P} \left(\max_{\mu(\Delta \mathbf{x}_I) \leq \delta} f(\mathbf{x} + \Delta \mathbf{x}) \geq 0.5 \mid f(\mathbf{x}) < 0.5 \right) \quad (2.1)$$

holds for all $z \in \mathcal{Z}$, where $\Delta \mathbf{x}_I$ is the effort for improvable features and $\Delta \mathbf{x} = (\Delta \mathbf{x}_I, \mathbf{0}, \mathbf{0})$.

Note that the condition $f(\mathbf{x}) < 0.5$ represents that an individual is unqualified, and $f(\mathbf{x} + \Delta \mathbf{x}) \geq 0.5$ implies that the effort $\Delta \mathbf{x}$ allows the individual to become qualified. The above definition of fairness in *equal improvability* requires that unqualified individuals from different groups $z \in [Z]$ are equally likely to become qualified if appropriate effort is made. Note that μ can be defined on a case-by-case basis. For example, we can use $\|\mathbf{x}_I\| = \sqrt{\mathbf{x}_I^\top \mathbf{C} \mathbf{x}_I}$, where $\mathbf{C} \in \mathbb{R}^{d_I \times d_I}$ is a cost matrix that is diagonal and positive definite. Here, the diagonal terms of $\mathbf{C}$ characterize how difficult to improve each feature. For instance, consider the graduate school admission problem where $\mathbf{x}_I$ contains features “number of publications” and “GPA”. Since publishing more papers is harder than raising the GPA, the corresponding diagonal term for the number of publications feature in $\mathbf{C}$ should be greater than that for the GPA feature. The constant δ in Definition 2.1 can be selected depending on the classification task and the features. Appendix A.2.1 contains the interpretation of each term in Definition 2.1. We introduce an equivalent definition of EI fairness below.

Proposition 2.2 *The EI fairness notion defined in Definition 2.1 has an equivalent*

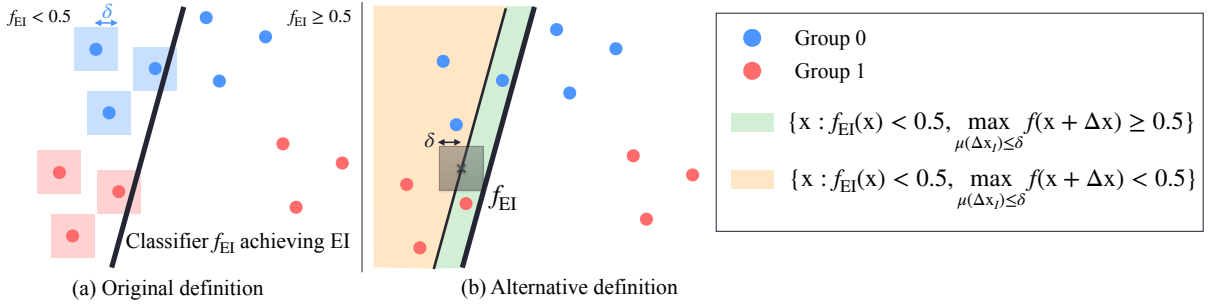


Figure 2.2: Visualization of EI fairness. For binary classification on 12 samples (dots) in two groups (red/blue), we visualize the fairness notion defined in this chapter: (a) shows the original definition in Definition 2.1, and (b) shows an equivalent definition in Prop. 2.2. Here we assume two-dimensional features (both are improvable) and L_∞ norm for $\mu(\mathbf{x}) = \|\mathbf{x}\|_\infty$. The classifier f_{EI} achieves equal improvability (EI) since the same portion (1 out of 3) of unqualified samples in each group can be improved to qualified samples.

format: a classifier f achieves equal improvability with δ -effort if and only if

$$\mathbb{P}(\mathbf{x} \in \mathcal{X}_-^{\text{imp}} \mid \mathbf{x} \in \mathcal{X}_-, z = z) = \mathbb{P}(\mathbf{x} \in \mathcal{X}_-^{\text{imp}} \mid \mathbf{x} \in \mathcal{X}_-) \quad (2.2)$$

holds for all $z \in \mathcal{Z}$ where $\mathcal{X}_- = \{\mathbf{x} : f(\mathbf{x}) < 0.5\}$ is the set of features $\mathbf{x}$ for unqualified samples, and $\mathcal{X}_-^{\text{imp}} = \{\mathbf{x} : f(\mathbf{x}) < 0.5, \max_{\mu(\Delta \mathbf{x}_I) \leq \delta} f(\mathbf{x} + \Delta \mathbf{x}) \geq 0.5\}$ is the set of features $\mathbf{x}$ for unqualified samples that can be improved to qualified samples by adding $\Delta \mathbf{x}$ satisfying $\mu(\Delta \mathbf{x}_I) \leq \delta$.

The proof of this proposition is trivial from the definition of $\mathcal{X}_-$ and $\mathcal{X}_-^{\text{imp}}$. Note that $\mathbb{P}(\mathbf{x} \in \mathcal{X}_-^{\text{imp}} \mid \mathbf{x} \in \mathcal{X}_-)$ in the above equation indicates the probability that unqualified samples can be improved to qualified samples by changing the features within budget δ . This is how we define the “improvability” of unqualified samples, and the EI fairness notion is equalizing this improvability for all groups.

Visualization of EI. Figure 2.2 shows the geometric interpretation of EI fairness notion in Definition 2.1 and Prop. 2.2, for a simple two-dimensional dataset having 12

samples in two groups $z \in \{\text{red}, \text{blue}\}$. Consider a linear classifier f_{EI} shown in the figure, where the samples at the right-hand-side of the decision boundary is classified as qualified samples ($f_{\text{EI}}(\mathbf{x}) \geq 0.5$). In Figure 2.2a, we have L_∞ ball at each unqualified sample, representing that these samples have a chance to improve their feature in a way that the improved feature $\mathbf{x} + \Delta\mathbf{x}$ allows the sample to be classified as qualified, *i.e.*, $f_{\text{EI}}(\mathbf{x} + \Delta\mathbf{x}) \geq 0.5$. One can confirm that $\mathbb{P}(\max_{\mu(\Delta\mathbf{x}_1) \leq \delta} f(\mathbf{x} + \Delta\mathbf{x}) \geq 0.5 \mid f(\mathbf{x}) < 0.5, z = z) = \frac{1}{3}$ holds for each group $z \in \{\text{red}, \text{blue}\}$, thus satisfying equal improvability according to Definition 2.1. In Figure 2.2b, we check this in an alternative way by using the EI fairness definition in Prop. 2.2. Here, instead of making a set of improved features at each sample, we partition the feature domain $\mathcal{X}$ into three parts: (i) the features for qualified samples $\mathcal{X}_+ = \{\mathbf{x} : f_{\text{EI}}(\mathbf{x}) \geq 0.5\}$, (ii) the features for unqualified samples that can be improved $\mathcal{X}_-^{\text{imp}} = \{\mathbf{x} : f_{\text{EI}}(\mathbf{x}) < 0.5, \max_{\mu(\Delta\mathbf{x}_1) \leq \delta} f(\mathbf{x} + \Delta\mathbf{x}) \geq 0.5\}$, and (iii) the features for unqualified samples that cannot be improved $\{\mathbf{x} : f_{\text{EI}}(\mathbf{x}) < 0.5, \max_{\mu(\Delta\mathbf{x}_1) \leq \delta} f(\mathbf{x} + \Delta\mathbf{x}) < 0.5\}$. In the figure, (ii) is represented as the green region and (iii) is shown as the yellow region. From Prop. 2.2, EI fairness means that $\frac{\# \text{ samples in (ii)}}{\# \text{ samples in (ii)} + \# \text{ samples in (iii)}}$ is identical at each group $z \in \{\text{red}, \text{blue}\}$, which is true for the example in Figure 2.2b.

Comparison of EI with other fairness notions. The suggested fairness notion *equal improvability* (EI) is in stark difference with existing popular fairness notions that do not consider dynamics, *i.e.*, demographic parity (DP), equal opportunity (EO) [7], and equalized odds (EOD) [7], which can be “myopic” and focuses only on achieving classification fairness in the current status. Our notion instead, aims at achieving classification fairness in the long run when each sample improves its feature over time.

On the other hand, EI also has differences with existing fairness notions that capture the dynamics of samples [23, 26, 22, 24]. Table 2.1 compares our fairness notion with the related existing notions. In particular, Bounded Effort (BE) fairness proposed by

[23] equalizes “the available reward after each individual making a bounded effort” for different groups, which is very similar to EI when we set a proper reward function (see Appendix A.2.2). To be more specific, the BE fairness can be represented as in Table 2.1. Comparing this BE expression with EI in Definition 2.1, one can confirm the difference: the inequality $f(\mathbf{x}) < 0.5$ is located at the conditional part for EI, which is not true for BE. EI and BE are identical if the negative prediction rates are equal across the groups, but in general, they are different. The condition $f(\mathbf{x}) < 0.5$ here is very important since only looking into the unqualified members makes more sense when we consider *improvability*. More importantly, the BE definition is based on reward functions and we are presenting BE in a form that is closest to our EI fairness expression. Besides, Equal Recourse (ER) fairness proposed by [22] suggests equalizing the average effort of different groups without limiting the amount of effort that each sample can make. Note that ER is vulnerable to outliers. For example, when we have an unqualified outlier sample that is located far away from the decision boundary, ER disparity will be dominated by this outlier and fail to reflect the overall unfairness. Prior work [24] suggested Individual-Level Fair Causal Recourse (ILFCR), a fairness notion that considers a more general setting that allows causal influence between the features. This notion aims to equalize the cost of recourse required for a rejected individual to obtain an improved outcome if the individual is from a different group. Individual-level equal recourse shares a similar spirit with EI since both aim to equalize the potential to improve the decision outcome for the rejected samples. However, introducing individual-level fairness with respect to different groups inherently requires counterfactual fairness, which has its own limitation, as described in [27], and it is also vulnerable to outliers. Another causal-based fairness notion was proposed in [26] to equalize the minimum level of effort such that the expected prediction score of the groups is equal to each other. Note that, their definition is specific to causal settings and it considers the whole sensitive groups instead of the rejected samples of the

Name of fairness	Definition	Consider efforts?	Limitations
Equal Improvability	$\mathbb{P}\left(\max_{\mu(\Delta\mathbf{x}_1) \leq \delta} f(\mathbf{x} + \Delta\mathbf{x}) \geq 0.5 \mid f(\mathbf{x}) < 0.5, z = z\right) = \mathbb{P}\left(\max_{\mu(\Delta\mathbf{x}_1) \leq \delta} f(\mathbf{x} + \Delta\mathbf{x}) \geq 0.5 \mid f(\mathbf{x}) < 0.5\right)$	Yes	-
Demographic Parity	$\mathbb{P}(f(\mathbf{x}) \geq 0.5 \mid z = z) = \mathbb{P}(f(\mathbf{x}) \geq 0.5)$	No	-
Equal Opportunity [7]	$\mathbb{P}(f(\mathbf{x}) \geq 0.5 \mid y = 1, z = z) = \mathbb{P}(f(\mathbf{x}) \geq 0.5 \mid y = 1)$	No	-
Equalized Odds [7]	$\mathbb{P}(f(\mathbf{x}) \geq 0.5 \mid y = y, z = z) = \mathbb{P}(f(\mathbf{x}) \geq 0.5 \mid y = y)$	No	-
Bounded Effort[23]	$\mathbb{P}\left(\max_{\mu(\Delta\mathbf{x}_1) \leq \delta} f(\mathbf{x} + \Delta\mathbf{x}) \geq 0.5, f(\mathbf{x}) < 0.5 \mid z = z\right) = \mathbb{P}\left(\max_{\mu(\Delta\mathbf{x}_1) \leq \delta} f(\mathbf{x} + \Delta\mathbf{x}) \geq 0.5, f(\mathbf{x}) < 0.5\right)$	Yes	Cannot handle imbalanced group negative rates (Appendix A.3.4)
Equal Recourse[22]	$\mathbb{E}\left[\min_{f(\mathbf{x} + \Delta\mathbf{x}) \geq 0.5} \mu(\Delta\mathbf{x}) \mid f(\mathbf{x}) < 0.5, z = z\right] = \mathbb{E}\left[\min_{f(\mathbf{x} + \Delta\mathbf{x}) \geq 0.5} \mu(\Delta\mathbf{x}) \mid f(\mathbf{x}) < 0.5\right]$	Yes	Vulnerable to outliers (Appendix A.2.3 and A.3.4)
Individual-Level Fair Causal Recourse [24]	$\min_{f(\mathbf{x}') \geq 0.5} \mu(\mathbf{x}' - \mathbf{x}(z, \mathbf{u})) = \min_{f(\mathbf{x}') \geq 0.5} \mu(\mathbf{x}' - \mathbf{x}(z', \mathbf{u}))$ for all latent variable $\mathbf{u}$, all groups z, z'	Yes	Limitations of counterfactual fairness

Table 2.1: Comparison of our EI fairness with existing fairness notions.

sensitive groups. In addition to fairness notions, we also discuss other related works such as fairness-aware algorithms in Section 2.5.

Compatibility of EI with other fairness notions. Here we prove the compatibility of three fairness notions (EI, DP, and BE), under two mild assumptions. Assumption 2.3 ensures that EI is well-defined, while Assumption 2.4 implies that the norm μ and the effort budget δ are chosen such that we have nonzero probability that unqualified individuals can become qualified after making efforts.

Assumption 2.3 *For any classifier f , the probability of unqualified samples for each demographic group is not equal to 0, i.e., $\mathbb{P}(f(\mathbf{x}) < 0.5, z = z) \neq 0$ for all $z \in \mathcal{Z}$.*

Assumption 2.4 *For any classifier f , the probability of being qualified after the effort for unqualified samples is not equal to 0, i.e., $\mathbb{P}\left(\max_{\mu(\Delta\mathbf{x}_1) \leq \delta} f(\mathbf{x} + \Delta\mathbf{x}) \geq 0.5, f(\mathbf{x}) < 0.5\right) \neq 0$.*

Under these assumptions, the following theorem reveals the relationship between DP, EI and BE.

Theorem 2.5 *If a classifier f achieves two of the following three fairness notions, DP, EI, and BE; then it has to achieve the remaining fairness notion as well.*

The proof of the Theorem 2.5 is provided in Appendix A.1.1. This theorem immediately implies the following corollary, which provides a condition such that EI and BE conflict with each other.

Corollary 2.6 *The above theorem says that if a classifier f achieves EI and BE, it has to achieve DP. Thus, by contraposition, if f does not achieve DP, then it cannot achieve EI and BE simultaneously.*

Besides, we also investigate the connections between EI and ER in Appendix A.1.2.

2.3 Achieving Equal Improvability

In this section, we discuss methods for finding a classifier that achieves EI fairness. Following existing in-processing techniques [6, 28, 21, 29], we focus on finding a fair classifier by solving a fairness-regularized optimization problem. To be specific, we first derive a differentiable penalty term U_δ that approximates the unfairness with respect to EI, and then solve a regularized empirical minimization problem having the unfairness as the regularization term. This optimization problem can be represented as

$$\min_{f \in \mathcal{F}} \left\{ \frac{(1 - \lambda)}{N} \sum_{i=1}^N \ell(y_i, f(\mathbf{x}_i)) + \lambda U_\delta \right\}, \quad (2.3)$$

where $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ is the given dataset, $\ell : \{0, 1\} \times [0, 1] \rightarrow \mathbb{R}$ is the loss function, $\mathcal{F}$ is the set of classifiers we are searching over, and $\lambda \in [0, 1)$ is a hyperparameter that balances fairness and prediction loss. Here we consider three different ways of defining the penalty term U_δ , which are (a) covariance-based, (b) kernel density estimator (KDE)-

based, and (c) loss-based methods. We first introduce how we define U_δ in each method, and then discuss how we solve (2.3).

Covariance-based EI penalty. Our first method is inspired by [6], which measures the unfairness of a score-based classifier f by the covariance of the sensitive attribute z and the score $f(\mathbf{x})$, when the demographic parity (DP) fairness condition ($\mathbb{P}(f(\mathbf{x}) > 0.5 | z = z) = \mathbb{P}(f(\mathbf{x}) > 0.5)$ holds for all z) is considered. The intuition behind this idea of measuring the covariance is that a perfect fair DP classifier should have zero correlation between z and $f(\mathbf{x})$. By applying similar approach to our fairness notion in Definition 2.1, the EI unfairness is measured by the covariance between the sensitive attribute z and the maximally improved score of rejected samples within the effort budget. In other words, $(\text{Cov}(z, \max_{\|\Delta \mathbf{x}_I\| \leq \delta} f(\mathbf{x} + \Delta \mathbf{x}) \mid f(\mathbf{x}) < 0.5))^2$ represents the EI unfairness of a classifier f where we took the square to penalize negative correlation case as well. Let $I_- = \{i : f(\mathbf{x}_i) < 0.5\}$ be the set of indices of unqualified samples, and $\bar{z} = \sum_{i \in I_-} z_i / |I_-|$. Then, EI unfairness can be approximated by the square of the empirical covariance, *i.e.*,

$$U_\delta \triangleq \left(\frac{1}{|I_-|} \sum_{i \in I_-} (z_i - \bar{z}) \left(\max_{\|\Delta \mathbf{x}_{I_i}\| \leq \delta} f(\mathbf{x}_i + \Delta \mathbf{x}_i) - \sum_{j \in I_-} \max_{\|\Delta \mathbf{x}_{I_j}\| \leq \delta} f(\mathbf{x}_j + \Delta \mathbf{x}_j) / |I_-| \right) \right)^2. \quad (2.4)$$

Since $\sum_{i \in I_-} (z_i - \bar{z}) \left(\sum_{j \in I_-} \max_{\|\Delta \mathbf{x}_{I_j}\| \leq \delta} f(\mathbf{x}_j + \Delta \mathbf{x}_j) / |I_-| \right) = 0$ from $\sum_{i \in I_-} (z_i - \bar{z}) = 0$, we have

$$U_\delta = \left(\frac{1}{|I_-|} \sum_{i \in I_-} (z_i - \bar{z}) \max_{\|\Delta \mathbf{x}_{I_i}\| \leq \delta} f(\mathbf{x}_i + \Delta \mathbf{x}_i) \right)^2.$$

KDE-based EI penalty. The second method is inspired by [29], which suggests to first approximate the probability density function of the score $f(\mathbf{x})$ via kernel density estimator (KDE) and then put the estimated density formula into the probability term

in the unfairness penalty. Recall that given m samples $y_1, \dots, y_m$, the true density g_y on y is estimated by KDE as $\hat{g}_y(\hat{y}) \triangleq \frac{1}{mh} \sum_{i=1}^m g_k\left(\frac{\hat{y}-y_i}{h}\right)$, where g_k is a kernel function and h is a smoothing parameter.

Here we apply this KDE-based method for estimating the EI penalty term in Definition 2.1. Let $y_i^{\max} = \max_{\|\Delta \mathbf{x}_i\| \leq \delta} f(\mathbf{x}_i + \Delta \mathbf{x}_i)$ be the maximum score achievable by improving feature $\mathbf{x}_i$ within budget δ , and $I_{-,z} = \{i : f(\mathbf{x}_i) < 0.5, z_i = z\}$ be the set of indices of unqualified samples of group z . Then, the density of $y_i^{\max}$ for the unqualified samples in group z can be approximated as¹

$$\hat{g}_{y^{\max}|f(\mathbf{x})<0.5,z}(\hat{y}^{\max}) \triangleq \frac{1}{|I_{-,z}|h} \sum_{i \in I_{-,z}} g_k\left(\frac{\hat{y}^{\max} - y_i^{\max}}{h}\right). \quad (2.5)$$

Then, the estimate on the left-hand-side (LHS) probability term in Definition 2.1 is represented as

$$\begin{aligned} \hat{\mathbb{P}}\left(\max_{\mu(\Delta \mathbf{x}_1) \leq \delta} f(\mathbf{x} + \Delta \mathbf{x}) \geq 0.5 \mid f(\mathbf{x}) < 0.5, z = z\right) &= \int_{0.5}^{\infty} \hat{g}_{y^{\max}|f(\mathbf{x})<0.5,z}(\hat{y}^{\max}) d\hat{y}^{\max} \\ &= \frac{1}{|I_{-,z}|h} \sum_{i \in I_{-,z}} G_k\left(\frac{0.5 - y_i^{\max}}{h}\right) \end{aligned}$$

where $G_k(\tau) \triangleq \int_{\tau}^{\infty} g_k(y) dy$. Similarly, we can estimate the right-hand-side (RHS) probability term in Definition 2.1, and the EI-penalty U_{δ} is computed as the summation of the absolute difference of the two probability values (LHS and RHS) among all groups z .

Loss-based EI penalty. Another common way of approximating the fairness violation as a differentiable term is to compute the absolute difference of group-specific losses [31, 32]. Following the spirit of EI notion in Definition 2.1, we define EI loss of group z as $\tilde{L}_z \triangleq \frac{1}{|I_{-,z}|} \sum_{i \in I_{-,z}} \ell\left(1, \max_{\|\Delta \mathbf{x}_i\| \leq \delta} f(\mathbf{x}_i + \Delta \mathbf{x}_i)\right)$. Here, $\tilde{L}_z$ measures how far the rejected

¹This term is differentiable with respect to model parameters, since g_k is differentiable w.r.t $y_i^{\max}$, and $y_i^{\max} = \max_{\|\Delta \mathbf{x}_i\| \leq \delta} f(\mathbf{x}_i + \Delta \mathbf{x}_i)$ is differentiable w.r.t. model parameters from [30].

samples in group z are away from being accepted after the feature improvement within budget δ . Similarly, EI loss for all groups is written as $\tilde{L} \triangleq \sum_{z \in \mathcal{Z}} \frac{I_{-z}}{I_{-}} \tilde{L}_z$. Finally, the EI penalty term is defined as $U_\delta \triangleq \sum_{z \in \mathcal{Z}} \left| \tilde{L}_z - \tilde{L} \right|$.

Solving (2.3). For each approach defined above (covariance-based, KDE-based and loss-based), the penalty term U_δ is defined uniquely. Note that in all cases, we need to solve a maximization problem $\max_{\|\Delta_{\mathbf{x}_I}\| \leq \delta} f(\mathbf{x} + \Delta \mathbf{x})$ in order to get U_δ . Since (2.3) is a minimization problem containing U_δ in the cost function, it is essentially a minimax problem. We leverage adversarial training techniques to solve (2.3). The inner maximization problem is solved using one of two methods: (i) derive the closed-form solution for generalized linear models, (ii) use projected gradient descent for general settings. The details can be found in Appendix A.2.4.

2.4 Experiments

This section presents the empirical results of our EI fairness notion. To measure the fairness violation, we use EI disparity = $\max_{z \in [\mathcal{Z}]} |\mathbb{P}(\max_{\mu(\Delta_{\mathbf{x}_I)} < \delta} f(\mathbf{x} + \Delta \mathbf{x}) \geq 0.5 \mid f(\mathbf{x}) < 0.5, z = z) - \mathbb{P}(\max_{\mu(\Delta_{\mathbf{x}_I)} < \delta} f(\mathbf{x} + \Delta \mathbf{x}) \geq 0.5 \mid f(\mathbf{x}) < 0.5)|$. First, in Section 2.4.1, we show that our methods suggested in Section 2.3 achieve EI fairness in various real/synthetic datasets. Second, in Section 2.4.2, focusing on the dynamic scenario where each individual can make effort to improve its outcome, we demonstrate that training an EI classifier at each time step promotes achieving the long-term fairness, *i.e.*, the feature distribution of two groups become identical in the long run. In Appendix A.3.4, we include additional experimental results comparing the robustness of EI and related notions including ER and BE. Specifically, we compare (i) the outlier-robustness of EI and ER, and (ii) the robustness of EI and BE to imbalanced group negative rates.

2.4.1 Suggested methods achieve EI fairness

Recall that Section 2.3 provided three approaches for achieving EI fairness. Here we check whether such methods successfully find a classifier with a small EI disparity, compared with ERM which does not have fairness constraints. We train all algorithms using logistic regression (LR) in this experiment. Due to the page limit, we defer the presentation of results for (1) a two-layer ReLU neural network with four hidden neurons and (2) a five-layer ReLU network with 200 hidden neurons per layer to Appendix A.3.3, and provide more details of hyperparameter selection in Appendix A.3.2.

Experiment setting. For all experiments, we use the Adam optimizer and cross-entropy loss. We perform cross-validation on the training set to find the best hyperparameter. We provide statistics for five trials having different random seeds. For the KDE-based approach, we use the Gaussian kernel.

Datasets. We perform the experiments on one synthetic dataset, and two real datasets: German Statlog Credit [33], and ACSIncome-CA [34]. The synthetic dataset has two non-sensitive attributes $\mathbf{x} = (x_1, x_2)$, one binary sensitive attribute z , and a binary label y . Both features x_1 and x_2 are assumed to be improvable. We generate 20,000 samples where $(\mathbf{x}, y, z)$ pair of each sample is generated independently as below. We define z and $(y|z = z)$ as Bernoulli random variables for all $z \in \{0, 1\}$, and define $(\mathbf{x}|y = y, z = z)$ as multivariate Gaussian random variables for all $y, z \in \{0, 1\}$. The numerical details are in Appendix A.3.1 The maximum effort δ for this dataset is set to 0.5. The ratio of the training versus test data is 4:1.

German Statlog Credit dataset contains 1,000 samples and the ratio of the training versus test data is 4:1. The task is to predict the credit risk of an individual given its financial status. Following [35], we divide the samples into two groups using the age of thirty

Dataset	Metric	Methods			
		ERM	Covariance-based	KDE-based	Loss-based
Synthetic	Error Rate ($\downarrow$)	$.221 \pm .001$	$.253 \pm .003$	$.250 \pm .001$	$.246 \pm .001$
	EI Disp. ($\downarrow$)	$.117 \pm .007$	$.003 \pm .001$	$.005 \pm .003$	$.002 \pm .001$
German Stat.	Error Rate ($\downarrow$)	$.220 \pm .009$	$.262 \pm .009$	$.243 \pm .024$	$.237 \pm .008$
	EI Disp. ($\downarrow$)	$.041 \pm .008$	$.021 \pm .019$	$.035 \pm .026$	$.015 \pm .009$
ACSIIncome-CA	Error Rate ($\downarrow$)	$.184 \pm .000$	$.200 \pm .000$	$.196 \pm .000$	$.193 \pm .000$
	EI Disp. ($\downarrow$)	$.031 \pm .001$	$.008 \pm .001$	$.005 \pm .001$	$.006 \pm .001$

Table 2.2: Error rate and EI disparities of ERM and three proposed EI-regularized methods on logistic regression (LR). For each dataset, the lowest EI disparity (disp.) value is in boldface. Classifiers obtained by our three methods have much smaller EI disparity values than the ERM solution, without having much additional error.

as the boundary, *i.e.*, $z = 1$ for samples with age above thirty. Four features $\mathbf{x}$ are considered as improvable: `checking account`, `saving account`, `housing` and `occupation`, all of which are ordered categorical features. For example, the `occupation` feature has four levels: (1) unemployed, (2) unskilled, (3) skilled, and (4) highly qualified. We set the maximum effort $\delta = 1$, meaning that an unskilled man (with level 2) can become a skilled man, but cannot be a highly qualified man.

ACSIIncome-CA dataset consists of data for 195,665 people and is split into training/test sets in the ratio of 4:1. The task is predicting whether a person’s income would exceed 50K USD per year. We use sex as the sensitive attribute; we have two sensitive groups, male and female. We select education level (ordered categorical feature) as the improvable feature. We set the maximum effort $\delta = 3$.

Results. Table 2.2 shows the test error rate and test EI disparity (disp.) for ERM and our three EI-regularized methods (covariance-based, KDE-based, and loss-based) suggested in Section 2.3. For all three datasets, our EI-regularized methods successfully reduce the EI disparity without increasing the error rate too much, compared with ERM.

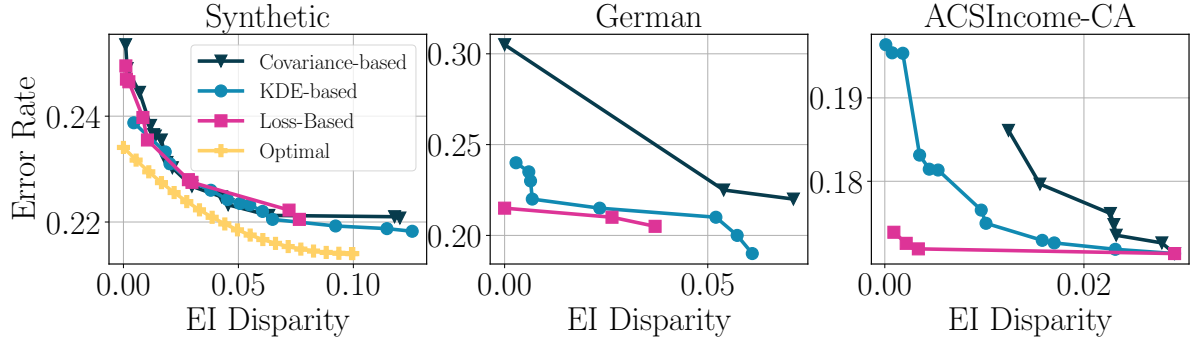


Figure 2.3: Tradeoff between EI disparity and error rate. We run three EI-regularized methods suggested in Section 2.3 for different regularizer coefficient λ and plot the frontier lines. For the synthetic dataset, the tradeoff curve for the ideal classifier is located at the bottom left corner, which is similar to the curves of proposed EI-regularized methods. This shows that our methods successfully find classifiers balancing EI disparity and error rate.

Figure 2.3 shows the tradeoff between the error rate and EI disparity of our EI-regularized methods. We marked the dots after running each method multiple times with different penalty coefficients λ , and plotted the frontier line. For the synthetic dataset with the Gaussian feature, we numerically obtained the performance of the optimal EI classifier, which is added in the yellow line at the bottom left corner of the first column plot. The details of finding the optimal EI classifier is in Appendix A.2.5. One can confirm that our EI-regularized methods have similar tradeoff curves for the synthetic dataset. Especially, for the synthetic dataset, the tradeoff curve of our methods nearly achieves that of the optimal EI classifier. For German and ACSIncome-CA datasets, the loss-based method is having a slightly better tradeoff curve than other methods.

2.4.2 EI promotes long-term fairness in dynamic scenarios

Here we provide simulation results on the dynamic setting, showing that EI classifier encourages the *long-term fairness*, *i.e.*, equalizes the feature distribution of different

groups in the long run ².

Dynamic system description

We consider a binary classification problem under the dynamic scenario with T rounds, where the improvable feature $\mathbf{x}_t \in \mathbb{R}$ and the label $\mathbf{y}_t \in \{0, 1\}$ of each sample as well as the classifier f_t evolve at each round $t \in \{0, \dots, T-1\}$. We denote the sensitive attribute as $z \in \{0, 1\}$, and the estimated label as $\hat{y}_t$. We assume $z \sim \text{Bern}(0.5)$ and $(\mathbf{x}_t \mid z = z) \sim \mathcal{P}_t^{(z)} = \mathcal{N}(\mu_t^{(z)}, \{\sigma_t^{(z)}\}^2)$. To mimic the admission problem, we only accept a fraction $\alpha \in (0, 1)$ of the population, *i.e.*, the true label is modeled as $y_t = \mathbf{1}\{\mathbf{x}_t \geq \chi_\alpha^{(t)}\}$, where $\chi_\alpha^{(t)}$ is the $(1 - \alpha)$ percentile of the feature distribution at round t . We consider z -aware linear classifier outputting $\hat{y}_t = \mathbf{1}\{\mathbf{x}_t \geq \tau_t^{(z)}\}$, which is parameterized by the thresholds $(\tau_t^{(0)}, \tau_t^{(1)})$ for two groups. Note that this classification rule is equivalent to defining score function $f_t(x, z) = 1/(\exp(\tau_t^{(z)} - x) + 1)$ and $\hat{y}_t = \mathbf{1}\{f(\mathbf{x}_t, z) \geq 0.5\}$.

Updating data parameters $(\mu_t^{(z)}, \sigma_t^{(z)})$. At each round t , we allow each sample to improve its feature from x to $x + \epsilon(x)$. Here we model $\epsilon(x) = \nu(x; z) = \frac{1}{(\tau_t^{(z)} - x + \beta)^2} \mathbf{1}\{x < \tau_t^{(z)}\}$ for a constant $\beta > 0$. In this model, the rejected samples with larger gap $(\tau_t^{(z)} - x)$ with the decision boundary are making less effort Δx , which is inspired by the intuition that a rejected sample is less motivated to improve its feature if it needs to take a large amount of effort to get accepted in a single step. More detailed justification of the $\epsilon(\cdot)$ selection is provided in Appendix A.3.5. After such effort is made, we compute the mean and standard deviation of each group: $\mu_{t+1}^{(z)} = \int_{-\infty}^{\infty} (x + \nu(x; z)) \phi(x; \mu_t^{(z)}, \sigma_t^{(z)}) dx$ and $\sigma_{t+1}^{(z)} = \sqrt{\int_{-\infty}^{\infty} (x + \nu(x; z) - \mu_{t+1}^{(z)})^2 \phi(x; \mu_t^{(z)}, \sigma_t^{(z)}) dx}$, where $\phi(\cdot; \mu, \sigma)$ is the pdf of

²Appendix A.4 provides analysis on the effect of fairness notions on the long-term fairness when a single step of feature improvement is applied, for toy examples. Our results show that EI better equalizes the feature distribution (compared with other fairness notions), which coincides with our empirical results in Section 2.4.2.

$\mathcal{N}(\mu, \sigma^2)$. We assume that the feature $\mathbf{x}_{t+1}$ in the next round follows a Gaussian distribution parameterized by $(\mu_{t+1}^{(z)}, \sigma_{t+1}^{(z)})$ for ease of simulation.

Updating classifier parameters $(\tau_t^{(0)}, \tau_t^{(1)})$. At each round t , we update the classifier depending on the current feature distribution $\mathbf{x}_t$. The EI classifier considered in this work updates $(\tau_t^{(0)}, \tau_t^{(1)})$ as below. Note that the maximized score $\max_{\|\Delta \mathbf{x}_I\| \leq \delta} f(\mathbf{x} + \Delta \mathbf{x})$ in Definition 2.1 can be written as $f_t(\mathbf{x}_t + \delta_t, z)$, and the equation $\max_{\|\Delta \mathbf{x}_I\| \leq \delta} f(\mathbf{x} + \Delta \mathbf{x}) \geq 0.5$ is equivalent to $\mathbf{x}_t + \delta_t > \tau_t^{(z)}$. Consequently, EI classifier obtains $(\tau_t^{(0)}, \tau_t^{(1)})$ by solving

$$\begin{aligned} \min_{\tau_t^{(0)}, \tau_t^{(1)}} \quad & \left| \mathbb{P}(\mathbf{x}_t + \delta_t > \tau_t^{(0)} \mid z = 0, \mathbf{x}_t < \tau_t^{(0)}) - \mathbb{P}(\mathbf{x}_t + \delta_t > \tau_t^{(1)} \mid z = 1, \mathbf{x}_t < \tau_t^{(1)}) \right| \\ \text{s.t.} \quad & \mathbb{P}(\hat{y}_t \neq y_t) \leq c. \end{aligned} \quad (2.6)$$

where $c \in [0, 1)$ is the maximum classification error rate we allow, and δ_t is the effort level at iteration t . In our experiments, δ_t is chosen as the mean efforts the population makes, *i.e.*, $\delta_t = 0.5 \sum_{z=0}^1 \int_{-\infty}^{\infty} \nu(x; z) \phi(x; \mu_t^{(z)}, \sigma_t^{(z)}) dx$. Meanwhile, we can similarly obtain the classifier for DP, BE, ER, and ILFCR constraints, details of which are in Appendix A.3.6. In the experiments, we numerically obtain the solution of this optimization problem.

Experiments on long-term fairness

We first initialize the feature distribution in a way that both sensitive groups have either different mean (*i.e.*, $\mu_0^{(0)} \neq \mu_0^{(1)}$) or different variance (*i.e.*, $\sigma_0^{(0)} \neq \sigma_0^{(1)}$). At each round $t \in \{1, \dots, T\}$, we update the data parameter $(\mu_t^{(z)}, \sigma_t^{(z)})$ for group $z \in \{0, 1\}$ and the classifier parameter $(\tau_t^{(0)}, \tau_t^{(1)})$, following the rule described in Section 2.4.2. At each round $t \in \{1, \dots, T\}$, we measure the *long-term unfairness* defined as the total variation

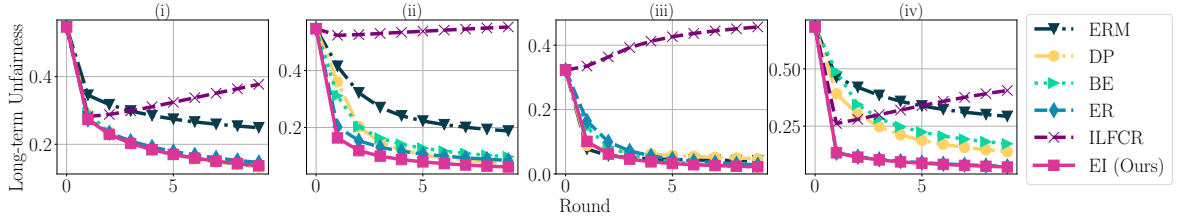


Figure 2.4: Long-term unfairness $d_{\text{TV}}(\mathcal{P}_t^{(0)}, \mathcal{P}_t^{(1)})$ at each round t for various algorithms. Consider the binary classification problem over two groups, under the dynamic scenario where the data distribution and the classifier for each group evolve over multiple rounds. We plot how the long-term unfairness (measured by the total variation distance between two groups) changes as round t increases. Here, each column shows the result for different initial feature distributions, details of which are given in Section 2.4.2. The long-term unfairness of EI classifier reduces faster than other existing fairness notions, showing that EI proposed in this chapter is helpful for achieving long-term fairness.

distance between the two group distributions:

$$d_{\text{TV}}(\mathcal{P}^{(0)}, \mathcal{P}^{(1)}) = \frac{1}{2} \int_{-\infty}^{\infty} \left| \phi(x; \mu_t^{(0)}, \sigma_t^{(0)}) - \phi(x; \mu_t^{(1)}, \sigma_t^{(1)}) \right| dx. \quad (2.7)$$

We run experiments on four different initial feature distributions:

- (i) $(\mu_0^{(0)}, \sigma_0^{(0)}, \mu_0^{(1)}, \sigma_0^{(1)}) = (0, 1, 1, 0.5)$, (ii) $(\mu_0^{(0)}, \sigma_0^{(0)}, \mu_0^{(1)}, \sigma_0^{(1)}) = (0, 0.5, 1, 1)$,
 (iii) $(\mu_0^{(0)}, \sigma_0^{(0)}, \mu_0^{(1)}, \sigma_0^{(1)}) = (0, 2, 0, 1)$, (iv) $(\mu_0^{(0)}, \sigma_0^{(0)}, \mu_0^{(1)}, \sigma_0^{(1)}) = (0, 0.5, 1, 0.5)$.

We set $\alpha = 0.2$, $c = 0.1$, and $\beta = 0.25$.

Baselines. We compare our EI classifier with multiple baselines, including the empirical risk minimization (ERM) and algorithms with fairness constraints: demographic parity (DP), bounded effort (BE) [23], equal recourse (ER) [22], and individual-level fair causal recourse (ILFCR) [24]. In particular, we assume a causal model for solving the ILFCR classifier for this specific data distribution, which is described in detail in Appendix A.3.6.

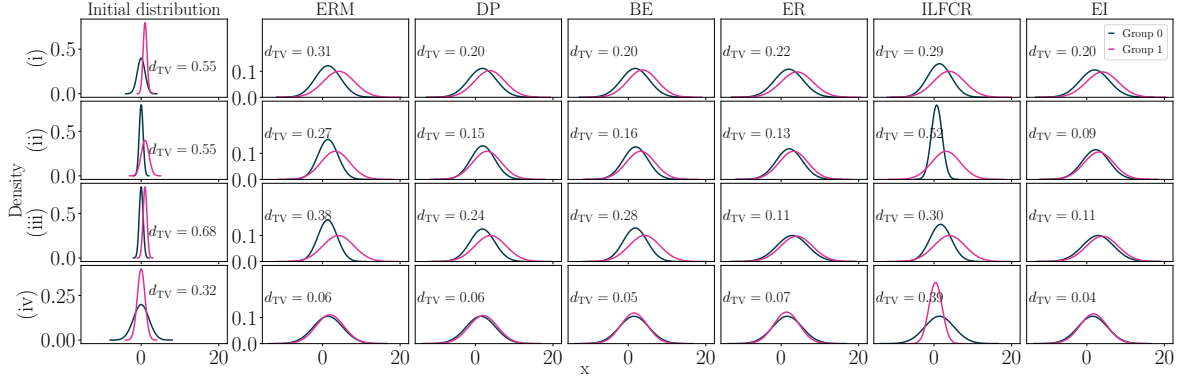


Figure 2.5: Evolution of the feature distribution, when we apply each algorithm for $t = 3$ rounds. At each row, the leftmost column shows the initial distribution and the rest of the columns show the evolved distribution for each algorithm, under the dynamic setting. Compared with existing fairness notions (DP, BE, ER, and ILFCR), EI achieves a smaller feature distribution gap between groups.

Results. Figure 2.4 shows how the long-term unfairness $d_{TV}(\mathcal{P}_t^{(0)}, \mathcal{P}_t^{(1)})$ changes as a function of round t , for cases (i) – (iv) having different initial feature distribution. Note that except ILFCR, other fairness notions (DP, BE, ER, and EI) yield lower long-term unfairness compared with ERM, for cases (i), (ii), and (iv). More importantly, EI accelerates the process of mitigating long-term unfairness, compared to other fairness notions. This observation highlights the benefit of EI in promoting true equity of groups in the long run. Figure 2.5 visualizes the initial distribution (at the leftmost column) and the evolved distribution at round $t = 3$ for multiple algorithms (at the rest of the columns). Each row represents different initial feature distribution, for cases (i) – (iv). One can confirm that EI brings the distribution of the two groups closer, compared with baselines. Moreover, in Appendix A.3.7, we explore the long-term impact of fairness notions on a different dynamic model, where most existing methods have an adverse effect on long-term fairness, while EI continues to enhance it.

2.5 Related Work

Fairness-aware algorithms. Most of the existing fair learning techniques fall into three categories: i) pre-processing approaches [36, 37, 38, 35], which primarily involve massaging the dataset to remove bias; ii) in-processing approaches [39, 40, 41, 6, 21, 42, 29, 43, 31, 32], which adjust model training for fairness; and iii) post-processing approaches [41, 44, 45, 7], which achieve fairness by modifying a given unfair classifier. Prior work [46] showed that the in-processing approach generally outperforms other approaches due to its flexibility. Hence, we focus on the in-processing approach and propose three methods to achieve EI. These methods achieve EI by solving fairness-regularized optimization problems. In particular, our proposed fairness regularization terms are inspired by prior work [6, 29, 31, 32].

Fairness notions related with EI. See Table 2.1 and the corresponding explanation given in Section 2.2.

Fairness dynamics. There are also a few attempts to study the long-term impact of different fairness policies [47, 23, 9]. In particular, prior work [9] studies how ERM amplifies the unfairness of a classifier in the long run. The key idea is that if the classifier of the previous iteration favors a certain candidate group, then the candidate groups will be more unbalanced since fewer individuals from the unfavored group will apply for this position. Thus, the negative feedback leads to a more unfair classifier. In contrast, other works [23, 47] focus more on long-term impact instead of classification fairness. To be specific, one line of work [23] studies how fairness intervention affects different groups in terms of evenness, centralization, and clustering by simulating the population’s response through effort. Another work [47] investigates how different fairness policies affect the gap between the qualification rates of different groups under a partially observed Markov

decision process. Besides, there are a few works which study how individuals may take strategic actions to improve their outcomes given a classifier [25]. However, this line of work [25] aims to address this problem by designing an optimization problem that is robust to strategic manipulation, which is orthogonal to our focus.

2.6 Conclusion

In this chapter, we introduce Equal Improvability (EI), a group fairness notion that equalizes the potential acceptance of rejected samples in different groups when appropriate effort is made by the rejected samples. We analyze the properties of EI and provide three approaches to finding a classifier that achieves EI. While our experiments demonstrate the effectiveness of the proposed approaches in reducing EI disparity, the theoretical analysis of the approximated EI penalties remains open. Moreover, we formulate a simplified dynamic model with one-dimensional features and a binary sensitive attribute to showcase the benefits of EI in promoting equity in feature distribution across different groups. We identify extending this model to settings with multiple sensitive attributes and high-dimensional features as an interesting future direction.

Chapter 3

Long-Term Fairness in Sequential Multi-Agent Selection

3.1 Introduction

With the increasing use of machine learning models for decision-making systems with significant societal impact, such as recruitment [48], criminal justice [49], and credit lending [50], there is also growing concern that such models may inherit existing bias in data, perpetuating and potentially exacerbating discrimination against certain groups in the population. This has prompted a growing body of research focused on developing fair and unbiased models, with much of the early literature focused on imposing notions of statistical fairness, such as equal selection rates or equal true positive rates, in static frameworks through pre-processing [51, 38, 36], in-processing [42, 21], or post-processing [7] mechanisms.

Going beyond concepts of one-shot fairness and unbiasedness, there is also increasing interest in the long-term impacts of automated decisions under various dynamical models. For example, the potentially adverse consequences of myopic fairness are pointed out

in [8]: unanticipated feedback dynamics due to the decisions made may change population statistics in an undesirable manner. For example, credit lending decisions which equalize true positive rates across groups to satisfy statistical fairness might lead to loans being offered to less creditworthy applicants from a disadvantaged group. Lower repayment rates from this group may then end up further decreasing its creditworthiness. On the other hand, feedback dynamics could also be used to shape population statistics in a desired direction. We take a step in this direction by hypothesizing that biasing slightly in favor of an underrepresented group in a selection problem (e.g., hiring or college admissions) provides positive reinforcement, increasing the proportion of applicants from that group in future selection rounds.

While most prior works on exploring such dynamics consider sequential decision-making by a single agent, we take a first step towards exploring the dynamics and long-term impact of multiple decision-making agents competing for a common pool of resources. We consider a selection problem, where we wish to use positive feedback to increase the proportion of underrepresented applicants in the presence of multiple agents (e.g, universities or companies) selecting from among a common pool of applicants. In order to focus attention on feedback dynamics, we consider a simplified model in which the agents are strictly rank ordered in terms of desirability from the applicants' point of view. For this model, we provide a mathematical framework for studying whether some level of cooperation between the agents helps promote long-term fairness, and whether the concept of positive reinforcement is robust to variations in the evolution model. We propose the evolution model of *positive reinforcement* where if a higher proportion of applicants from a specific group is selected, it will lead to an increase in the proportion of applicants from that group in subsequent rounds. We study variants of this basic model of reinforcement, a particularly interesting one being the notion of *role model reinforcement*. Here only a certain fraction of the admitted applicants (as opposed to all

admitted applicants) are in a position to influence applicant proportions in the future, by virtue of being role models. The idea that role models in society, to which a group can relate, could positively influence more aspirants to enter a field is supported by several works in social sciences and economics [52, 53].

Contributions The main contributions of this chapter are summarized below. This chapter is based in part on our published work [17].

- We propose the Multi-agent Fair-Greedy (MFG) policy, in which agents operate in a decentralized fashion, maximizing a greedy utility (based on the scores of selected applicants) while minimizing the disparity from a fixed long-term fairness target (based on the deviation of the selected proportion of minority applicants from a target proportion deemed to be socially fair). We characterize optimal actions under this policy and theoretically demonstrate the convergence of the overall applicant pool and the admission proportions to the desired long-term fairness target under the *pure positive reinforcement model* for the evolution of the composition of the applicant pool.
- Different population dynamics are studied empirically to evaluate the impact of variations in population behavior. We find that while the decentralized MFG policy attains long-term fairness under pure positive reinforcement, if the population dynamics follow a simple variant, termed the *role model reinforcement*, uncoordinated behavior by agents can result in an overall *negative feedback*, leading to a steady decrease in the number of underrepresented applicants in the pool. We propose a centralized version of the MFG policy which can restore positive feedback.
- We illustrate our mathematical framework through comprehensive experimental results based on synthetic and semi-synthetic datasets, highlighting variations in

system behavior under different evolution models, and under decentralized and centralized MFG policies. These results show that positive reinforcement indeed has potential for promoting long-term fairness even with multiple agents, but that policy design must be carefully considered in order to be robust to changes in the evolution model.

3.2 Related Work

There is a rich and rapidly growing literature on *fair* strategies that mitigate bias in one-shot algorithmic decision making, including pre-processing the labels or data and reweighting costs based on groups [36], reducing mutual information between sensitive attributes and predictions [54, 40], adversarial de-biasing [42], addition of constraints that satisfy fairness criteria [21], learning representations that obfuscate group information [51], and other information theoretic methods [55, 56, 57, 58, 59]. However, recent research has introduced an intriguing possibility: the design of sequential decision-making strategies that can have a positive long-term impact on social fairness and a study of their consequence on the population.

The effects of fairness-aware decisions on underlying population statistics were first studied in [8] for two-stage models: in the first stage, the algorithmic decisions are designed such that fairness constraints such as statistical parity or equal opportunity are satisfied, while the second stage examines the impact of these fairness interventions on the groups. Each sample is associated with a score corresponding to the probability of a positive outcome, sampled from group-specific distributions. The policy-maker or institution chooses selection policies to maximize utility subject to statistical fairness. The measure of interest in [8] is the expected change in the mean of the score distributions for the two groups as a result of one step of feedback. It was found that imposing

equal selection rates or true positives could lead to either improvement or cause harm, particularly to the minority group, depending on certain regimes. The results of [8] highlight the importance of going beyond static notions of fairness in algorithm/policy design.

Several works propose notions of statistical fairness focused on improving feature distributions among groups. For example, [23] presents a fairness notion that equalizes the maximum change in reward for groups with the same effort budget for improving their features. The authors examine how fairness interventions impact evenness, centralization, and clustering in the groups through their efforts, affecting score distributions. In another example, [16] proposes to equalize the proportion of unqualified candidates from different groups that can be qualified with a limited effort for improving their features. The authors investigate how statistical fairness notions change feature distributions among groups in the long term through modeling feature evolution.

Formal investigation of temporal effects of decision feedback and their equilibrium is typically performed in reinforcement learning [60, 61, 62, 63] or bandit settings [64, 65, 66, 67, 68]. The environment is described through a Markov Decision Process (MDP) framework where at each time, the decision-maker in a particular state takes an action and receives a reward. State transitions are governed by update models, and fairness constraints are included within reward definitions.

The long-term effects of fair decisions on the qualification rate of the group, which is defined as the probability of an individual from a particular group being qualified, is investigated in [47] under a partially observable MDP setting. The group proportions over time are fixed, but the decisions affect the feature distributions which in turn change their true qualification state, which is modeled as a hidden state. Decisions are performed on the features to maximize myopic instantaneous utility, subject to statistical fairness constraints. Threshold-based policies and their equilibrium are studied under two regimes:

“lack of motivation,” where the probability of remaining qualified on receiving a positive decision is less than that on being rejected, and “leg-up,” which is the opposite, where an accepted individual becomes inspired to become more qualified. Studies such as [69, 70] also examine the effects of fair policies on the distributions of the features. In particular, [69] studies how the qualification profiles of groups are influenced by a policy that imposes demographic parity (equal selection rates) across two groups of the population. They assume that social equity is achieved through equalized qualification profiles in equilibrium. The dynamic model in [70] is motivated by credit lending. The authors model the distributions of loan repayment likelihood (payback probabilities) by group, and examine the dynamics governed by the hypothesis that granting loans leads to upward mobility for the population if they are repaid. They study the impact of fair decisions on loan repayment likelihood and the negative effects of unequal misestimation of payback probabilities across groups, even if the decisions are fair. In contrast to studying the effects on group qualification, it is also imperative to understand how group representation could vary over time. It was first shown in [9] that empirical risk minimization can exacerbate the disparity in group representation. In the context of long-term fairness, [71] study how algorithmic decisions which are constrained by statistical fairness could degrade the representation of a minority group, and eventually cause the loss of minority representation in the system.

The work reported here builds upon preliminary results shown in [72], where the authors first introduce the problem of selecting applicants from a pool under the setting of a single institution. We introduce the positive reinforcement model for evolution of the applicant pool, which is similar in spirit to the “leg-up” model in [47], except that it applies at the level of a population rather than an individual: selection of a larger proportion of applicants from a specific group feeds back into society and leads to an increase in the proportion of applicants from that group in future rounds. The score of

an applicant, drawn from a group-specific distribution, is taken to represent the level of qualification of that individual. The Fair-Greedy policy, proposed in a single institution setting, greedily balances the sum of the scores of selected applicants against a fairness measure which increases with the deviation of the selected proportion of applicants from the under-represented group from a specified long-term target proportion. Theoretical results for identical score distributions across groups, and empirical results for other models, show convergence to a long-term fairness target set by the decision-maker. While the preliminary results in [72] show long-term fairness through positive feedback under a single institution, in this chapter, we extend the promise of positive reinforcement to multiple-agents, consider variations in the evolution model, and find that careful design of feedback mechanism is required.

All of the preceding works on long-term fairness focus on settings with a single decision-maker accessing a pool of samples from the population. However, in real-world selection processes such as college admissions or hiring, there are often multiple agents or institutions competing for a common pool of applicants. The composition of the applicant pool can therefore be affected by the collective decisions of these agents. To the best of our knowledge, ours is the first work to study the long-term evolution of fairness in such a multi-agent setting. While two-sided matching problems, where multiple agents and resources are to be matched to each other, are well studied in resource allocation problems and game theoretic settings [73, 74], the notion of long-term fairness in such settings is yet to be examined, and the definitions of fairness considered are quite distinct from those in fairness literature. Since our focus is on dynamical evolution, we sidestep the problem of matching between institutions (agents) and applicants (resources) by assuming that each applicant has the same fixed preference among the institutions.

3.3 Modeling Multi-Agent Decision-Making

Consider the problem of K agents, or institutions, selecting from a common pool of resources, or applicants. The applicant pool is composed of applicants from two groups $g = \{0, 1\}$, based on a sensitive attribute. Without loss of generality, we usually refer to $g = 0$ as the minority group and $g = 1$ as the majority group. We denote the number of applicants in round t , belonging to group g by N_t^g , and the total number of applicants as $N_t^0 + N_t^1 = N_t$. Every applicant is associated with a score which is sampled from a group-dependent distribution $\mathcal{P}^g$.

In many applications, the institutions have a growth in the program size, which is typically proportional to the population. For instance, the University of California system intends to serve top one-eighth of California's high school graduating class population [75]. Motivated by this line of thought, we consider that every institution fixes the total number of applicants it can admit, through the *capacity* c_k for institution k , set as a fraction of applicants it admits, out of the total number of applicants in the pool in that round. Thus we have $\sum_{k=1}^K c_k < 1$.

Each institution has to determine how to fill its admission slots, particularly by deciding how many minority and majority applicants are to be admitted, while satisfying two-fold objectives: (i) to accept applicants with the largest scores (ii) to achieve a long-term fairness target (denoted by $\alpha \in [0, 1]$) in the admission of applicants, which measures the proportion of admitted applicants belonging to the minority group, $g = 0$. The goal of the K institutions is to move towards the long-term fairness target, while admitting applicants with the highest scores. We denote the number of applicants admitted by institution k as $A_{k,t} = c_k N_t$, out of which $A_{k,t}^g$ belong to group g . Thus, we have $A_{k,t} = A_{k,t}^0 + A_{k,t}^1$, and the institutions hope to make decisions such that $A_{k,t}^0/A_{k,t}$ approaches α in equilibrium for all k .

We assume that the institutions are ranked, and that every applicant prefers institutions in the same order. Without loss of generality, we assume that the institutions are indexed by rank order, with $k = 1$ corresponding to the highest-ranked institution. Since the preference order is fixed, the matching of applicants to institutions simplifies to institutions picking from the application pool sequentially, in their ranked order. The decision variable at each institution in each round is the fraction of admitted applicants from each group (selecting from the highest scoring applicants in each group, chosen from among the pool that remains after higher-ranked institutions have made their selections). We, therefore, have an effectively sequential selection process in each round, where the selections are started by the highest-ranked institution and completed by the lowest-ranked institution.

The problem is formulated as a Markov Decision Process (MDP). The state $s_t \in [0, 1]$ is the fraction of applicants from group $g = 0$ in the applicant pool. Thus, at round t we have $s_t = N_t^0/N_t$. An institution k is associated with its respective action $a_t^k \in [0, 1]$, which is the fraction of applicants admitted from group $g = 0$ among the total number of applicants the institution k admits. Thus, $a_t^k = A_{k,t}^0/A_{k,t}$. The actions or the *decision variables* of each of the institutions are to be determined based on optimizing their respective *fairness-aware utilities*, defined below.

First, a set of N_t applicants in round t are associated with a set of scores $\{X_i^g\}$, where each X_i^g , $i \in [N_t^g]$, is sampled independently from $\mathcal{P}^g$. Additionally, $X_{(i)}^g$ denotes the i^{th} top score out of N_t^g applicants from group g . We define the score-based reward for an institution with rank k as the expectation of the sum of scores of the applicants admitted by the institution normalized by the number of total applicants it admits. Note that for an institution k , this depends upon the actions of the preceding $[1, k - 1]$ -ranked institutions, as each institution selects sequentially from the remnant pool of applicants. For $k \in \{2, 3, \dots, K\}$, we define $m_{k,t}^g$ as the number of applicants from group $g \in \{0, 1\}$

that have already been selected by higher ranked institutions $1, \dots, k-1$, with $m_{1,t}^g = 0$.

Thus, we have the score-based reward:

$$\begin{aligned} R_k(s_t, a_t^k) &= \frac{1}{A_{k,t}} \mathbb{E} \left[\sum_{i=m_{k,t}^0+1}^{m_{k,t}^0+A_{k,t}^0} X_{(i)}^0 + \sum_{i=m_{k,t}^1+1}^{m_{k,t}^1+A_{k,t}^1} X_{(i)}^1 \right] \\ &= \frac{1}{A_{k,t}} \mathbb{E} \left[\sum_{i=m_{k,t}^0+1}^{m_{k,t}^0+a_t^k A_{k,t}} X_{(i)}^0 + \sum_{i=m_{k,t}^1+1}^{m_{k,t}^1+(1-a_t^k)A_{k,t}} X_{(i)}^1 \right] \end{aligned} \quad (3.1)$$

with the group-specific offsets given by

$$m_{k,t}^0 = \sum_{j=1}^{k-1} c_j N_t a_t^j \text{ and } m_{k,t}^1 = \sum_{j=1}^{k-1} c_j N_t (1 - a_t^j). \quad (3.2)$$

Each institution is associated with its fairness loss, which is the squared difference between the proportion of group $g = 0$ out of the total admitted, and the long-term fairness target α , given by $L_k(a_t^k) = (a_t^k - \alpha)^2$. The considered fairness loss is a simple measure of disparity from the long-term fairness target, and penalizes the bias against both groups on the long run.

The “multi-agent fairness aware” utility of each institution is the sum of its *score-based reward* and *fairness loss*, expressed as

$$U_k(s_t, a_t^k) = R_k(s_t, a_t^k) - \lambda L_k(a_t^k). \quad (3.3)$$

where $\lambda \geq 0$ is a parameter that governs balancing fairness objective with score-based reward.

Thus each institution aims to maximize its score-based reward by admitting applicants with the highest scores, while minimizing its disparity from the long-term fairness target.

Multi-agent Fair-Greedy Policy We propose the *Multi-agent Fair-Greedy (MFG) policy*, which is a set of policies $\pi_{MFG}(s_t) = \{\pi_t^1, \pi_t^2, \dots, \pi_t^K\}$, where each institution optimizes its own multi-agent fairness aware utility as follows:

$$\pi_t^k = \arg \max_{a_t^k \in \mathcal{A}_t^k} U_k(s_t, a_t^k). \quad (3.4)$$

where $\mathcal{A}_t^k$ is the set of feasible actions for institution k in round t , which depends on the actions of higher ranked institutions and the state s_t . After higher ranked institutions have made their selections, the feasible action space for institution k is determined by the remaining capacity of the institution and the remaining applicants in the pool, and can be written as $\mathcal{A}_t^k = [\max(0, 1 - \frac{1-s_t - \sum_{j=1}^{k-1} (1-\pi_t^j)c_j}{c_k}), \min(1, \frac{s_t - \sum_{j=1}^{k-1} \pi_t^j c_j}{c_k})]$. For each institution, optimization of the instantaneous multi-agent fairness aware utility is equivalent to finding group-specific set of thresholds on the score distributions. In our rank-ordered model for institutions, each institution admits applicants whose scores exceed its threshold, choosing from the pool remaining after higher-ranked institutions have made their selections. However, we reiterate that the state s_t represents the original minority group proportion in the pool at round t .

Resource pool evolution A key ingredient of our formulation is modeling the manner in which the collective decisions of the institutions affect the pool of resources seen in the subsequent rounds of the problem. This is a complex relationship requiring data collected from carefully designed long-term experiments. While such data is not yet available, we can develop valuable insights by exploring different models for how the applicant pool might be shaped by institutional policies.

We assume that the state s_t is a random variable with mean θ_t and bounded variance. We model the number of applicants from group $g = 0$ and $g = 1$ as being sampled from

Poisson distributions, with $N_t^0 \sim \text{Poisson}(\theta_t N)$ and $N_t^1 \sim \text{Poisson}((1 - \theta_t)N)$, where N is the expected number of applicants in the pool. The total number of applicants in round t is given by $N_t = N_t^0 + N_t^1$.¹ The choice of the Poisson distribution is for the sake of simplicity, and the outcomes of our analysis remain independent of this particular choice of distribution. We consider different models for the evolution of θ_t in this chapter, focusing first on the model of *pure positive reinforcement*. In this model, a higher proportion of admission of a particular group in comparison to its application proportion, motivates more applicants from the group to participate in future selection rounds. Here the evolution follows:

$$\theta_{t+1} = [\theta_t + \eta_t(\pi_t^W - s_t)]_{\mathcal{C}} \quad (3.5)$$

where $[\cdot]_{\mathcal{C}}$ is the projection onto the set $\mathcal{C} = [0 + \epsilon, 1 - \epsilon]$ (or, simply clipping the mean parameter) where ϵ is a small positive number to avoid the mean parameter from reaching the boundary of the set and it implies that the any group cannot be completely eliminated from the pool. Here, η_t is a step size parameter and π_t^W represents the weighted actions of all the institutions.

In particular, under the MFG policy, π_t^W depends upon on the policies of all the institutions weighted by their capacities, which is also equivalent to the proportion of minority group among all the admitted applicants, defined as:

$$\pi_t^W = \frac{\sum_{k=1}^K c_k \pi_t^k}{\sum_{k=1}^K c_k}. \quad (3.6)$$

Thus, the collective decisions of all institutions, in particular their admission of applicants from a particular group, promote more such applicants to participate in the process, and

¹When both N_t^0 and N_t^1 are equal to zero, the round is completed without any admissions.

shape the evolution dynamics.

In the remainder of this section, we characterize the MFG policy in a decentralized setting, under the pure positive reinforcement model. We show that the optimal policy for each institution depends on actions that optimize the score-based reward alone, and the fairness loss alone. We therefore first derive the reward-optimal and fairness-optimal actions for each institution, and then characterize the MFG policy $\pi_t^k, k = 1, \dots, K$ in terms of these actions. We then show that, for pure positive reinforcement and identical score distributions $\mathcal{P}^g$, the applicant pool (state) and the admission proportions (actions) for all institutions converge to the long-term fairness target. We use the following assumptions in our convergence analysis:

Assumption 3.1 *The expected number of applicants in the pool N is large enough that the empirical distribution of scores can be replaced by statistical distributions using the law of large numbers.*

This assumption is required for the asymptotic analysis of the applicant pool and admission proportions. It is also used to derive the reward-optimal actions for the institutions.

Assumption 3.2 *The score distributions $\mathcal{P}^g$ are identical across groups.*

This assumption implies that the only difference in the groups is the proportion of applicants from each group in the pool at each round. We relax this assumption in Section 3.5.1 and provide empirical evidence for the existence of equilibria of the applicant pool proportion and the admission proportions.

Fairness-optimal action Each institution can minimize its fairness loss by setting its action to be equal to the long-term fairness target:

$$a_{F,t}^k = \arg \min_{a_t \in \mathcal{A}_t^k} L_k(a_t) = [\alpha]_{\mathcal{A}_t^k}, \quad \forall k \in \{1, 2, \dots, K\}. \quad (3.7)$$

where $[\alpha]_{\mathcal{A}_t^k}$ is the projection of α onto the feasible action space $\mathcal{A}_t^k$.

The next theorem characterizes the reward-optimal actions for the institutions.

Theorem 3.3 *Under Assumptions 3.1 and 3.2, the score-based reward, $R_k(s_t, a_t^k)$, is concave and the reward-optimal action for an institution k is given by*

$$a_{S,t}^k = \arg \max_{a_t^k \in \mathcal{A}_t^k} R_k(s_t, a_t^k) = \left[s_t + \frac{1}{c_k} \left(\sum_{j=1}^{k-1} c_j (s_t - \pi_t^j) \right) \right]_{\mathcal{A}_t^k}, \quad \forall k \in \{2, 3, \dots, K\} \quad (3.8)$$

and $a_{S,t}^1 = s_t$ where $[\cdot]_{\mathcal{A}_t^k}$ denotes the projection onto the feasible action space $\mathcal{A}_t^k$.

The detailed proof of this theorem can be found in Appendix B.1.

Remark 3.4 *Under the asymptotic regime of Assumption 3.1 (i.e., assuming that empirical distributions of applicant scores can be replaced by statistical distributions), the proof of Theorem 3.3 can rely on the following simplification. For institution k , the selection of applicants with top scores is equivalent to thresholding the group's score, and admitting applicants with scores within the group-specific lower and upper thresholds, denoted by $t_g^{k,S,low}$ and $t_g^{k,S,up}$ respectively. The upper threshold $t_g^{k,S,up}$ equals the lower threshold of institution $k - 1$ for group g . In Lemma B.1 we show that for reward optimality for any institution k , its group-specific lower thresholds should be equal; $t_0^{k,S,low} = t_1^{k,S,low}$, if possible. While this holds even if the group-specific score distributions are different, we can derive closed-form expressions for reward-optimal actions as in Theorem 3.3 when the distributions are identical across groups.*

Remark 3.5 *The optimal policy of institution k that maximizes its multi-agent fairness aware utility can be expressed as a convex combination of its pre-projected optimal score-*

based reward action and optimal fair-only actions, as

$$\pi_t^k = \gamma_{k,t} \left(s_t + \frac{1}{c_k} \left(\sum_{j=1}^{k-1} c_j (s_t - \pi_t^j) \right) \right) + (1 - \gamma_{k,t})\alpha,$$

where $\gamma_{k,t} \in [0, 1]$ (please refer to Appendix B.2 for details). The closed-form expression for $\gamma_{k,t}$ is not available in general, but it can be numerically computed due to the concavity of the fairness-aware utility function. It can be shown that the reward-optimal action can be expressed as:

$$a_{S,t}^k = s_t + \frac{(s_t - \alpha)}{c_k} \left(\sum_{j=1}^{k-1} c_j \prod_{i=j}^{k-1} (1 - \gamma_{i,t}) \right). \quad (3.9)$$

Thus, when the hyperparameter $\lambda > 0$, if $s_t < \alpha$, for $k \in \{2, 3, \dots, K\}$ we have $a_{S,t}^k < s_t$, and vice-versa. Note that, for a special case where there is a single institution, π_t^1 lies between the state s_t and long-term fairness target α . But that is not necessarily the case for π_t^k for $k > 1$, when there are multiple institutions.

In the following lemma, we characterize the weighted policy of the institutions, that ultimately governs the pool evolution (3.5).

Lemma 3.6 *Under Assumptions 3.1 and 3.2, the weighted MFG policy π_t^W can be expressed as the following under the pure positive reinforcement model:*

$$\pi_t^W = s_t + (\alpha - s_t) \frac{\sum_{j=1}^K c_j \prod_{i=j}^K (1 - \gamma_{i,t})}{\sum_{j=1}^K c_j} = s_t + (\alpha - s_t) \bar{\gamma}_t, \quad (3.10)$$

where $\bar{\gamma}_t \in [0, 1]$.

Please refer to Appendix B.2 for proof.

We utilize this result to prove the convergence of the applicant pool and admission proportions to the long-term fairness target. In the following theorem, we first show that the target proportion is a unique fixed-point of the pool evolution update. We then show that the weighted policy lies between the applicant and target proportions, as a result of which the mean state parameter and the state converge to the target. Although our analysis is for identical group-wise score distributions, we later provide empirical evidence of equilibrium in state and actions of all institutions when the score distributions are different across groups.

Theorem 3.7 *Under Assumptions 3.1, 3.2 and $\lambda \neq 0$, the weighted MFG policy is such that $\pi_t^W \in (s_t, \alpha)$ if $s_t < \alpha$, or $\pi_t^W \in (\alpha, s_t)$ if $s_t > \alpha$. Further, α is a unique fixed-point of the weighted policy π_t^W . In addition, the applicant pool proportion converges to the long-term fairness target α , if the step size parameter η_t is decaying with time and satisfies the assumptions that $\sum_t \eta_t = \infty$ and $\sum_t \eta_t^2 < \infty$. Further, the admission proportions of all institutions approach the long-term fairness target at equilibrium.*

The proof of this theorem can be found in Appendix B.2.

Remark 3.8 *For identical group-wise distributions as considered in Theorem 3.7, the convergence to the target α holds irrespective of the value of the hyperparameter λ weighing fairness in the utility function, as long as it is positive ($\lambda > 0$). For non-identical score distributions across groups, we provide empirical evidence for the existence of equilibria of the applicant pool proportion and the admission proportions in Section 3.5.1. However, the actual value of $\lambda > 0$ is now found to be important in determining how close to the fairness target we get.*

What happens if we ignore fairness? In the scenario of identical score distributions, if each institution completely disregards the fairness objective and optimizes only

its score-based reward, as seen from (3.9) where $\gamma_{k,t} = 1$, it follows that $\pi_t^k = a_{S,t}^k = s_t$ for all k . Consequently, the mean parameter θ_t would not experience any drift, and the composition of the applicant pool would remain unchanged. While each institution greedily acquires the best applicants by sacrificing diversity, there is no hope of influencing the participation of the underrepresented. However, biasing slightly in favor of the underrepresented group by introducing the fairness objective could lead to both θ_t and π_t^k converging to the long-term fairness target.

3.4 Variations on the Pool Evolution Model

Achieving long-term fairness relies on how the applicant pool evolves with institutional decisions over time. In this section, we introduce variations of the pure positive reinforcement model, and examine how the MFG policy fares under these new variants of the evolution model termed (i) order-based (ii) weighted (iii) role-model reinforcement. The bulk of our discussion in this section focuses on the motivation and study of role-model reinforcement and pitfalls under it, while we first briefly discuss the other two below:

Order-based positive reinforcement This simple variant allows control over the strength of positive reinforcement by raising the feedback in pure positive reinforcement to a power $\beta > 0$:

$$\theta_{t+1} = [\theta_t + \eta_t \text{sign}(\pi_t^W - s_t) |\pi_t^W - s_t|^\beta]_{\mathcal{C}}. \quad (3.11)$$

Pure positive reinforcement is a special case of the above model for $\beta = 1$. The feedback is amplified for $\beta < 1$ and is attenuated for $\beta > 1$, since $0 \leq |\pi_t^W - s_t| \leq 1$.

Weighted positive reinforcement The second simple variation allows customized weights $\{z_k > 0\}$ on the institutions' policies

$$\pi_t^z = \frac{\sum_{k=1}^K z_k \pi_t^k}{\sum_{k=1}^K z_k}, \quad (3.12)$$

with pool evolution

$$\theta_{t+1} = [\theta_t + \eta_t(\pi_t^z - s_t)]_{\mathcal{C}}. \quad (3.13)$$

Here $z_k = c_k$ reduces back to pure positive reinforcement, which gives greater influence to an agent with higher capacity. The more general models (3.12)-(3.13) here allow us more flexibility in assigning influence. For example, we can give equal influence to all institutions by setting $z_k = z > 0$.

Role model reinforcement Thus far, we have assumed that all selected applicants in an institution have equal ability to influence the reinforcement. However, applicants who do well post-selection in a given institution, who we term *role models*, could influence future applicant pools significantly more than other admitted applicants in the institution. The motivation for this model stems from the fact that in practice, success in an institution depends not just on qualifications at the time of admission, but on support mechanisms within the institution to aid in continued growth and success of the admitted applicants, as well as on circumstances that may be difficult to characterize. However, to develop analytical insight into what happens when positive reinforcement occurs due to the applicants who do well post-selection, we consider a score-based criterion for identifying these role models: among all applicants (from both groups) admitted to institution k , the role models are composed of a fraction $r \in [0, 1]$ of applicants with the highest scores. From among this group, the fraction of minority (group $g = 0$) applicants, denoted by r_t^k , drive the pool evolution. We now develop the notation required to define this model

precisely.

Let $\mathcal{X}_t^k$ denote the set of scores of all applicants admitted by the institution k at time t ,

$$\mathcal{X}_t^k = \left\{ X_{(m_{k,t}^0+1)}^0, X_{(m_{k,t}^0+2)}^0, \dots, X_{(m_{k,t}^0+A_{k,t}^0)}^0, X_{(m_{k,t}^1+1)}^1, X_{(m_{k,t}^1+2)}^1, \dots, X_{(m_{k,t}^1+A_{k,t}^1)}^1 \right\}. \quad (3.14)$$

The role models for each institution are selected as top r proportion of the admissions with the highest scores, $\mathcal{R}_t^k = \operatorname{argmax}_{\mathcal{X}' \subset \mathcal{X}_t^k, |\mathcal{X}'| = \lfloor r A_{k,t} \rfloor} \sum_{x \in \mathcal{X}'} x$. Then, let r_t^k denote the fraction of the minority group in the role models set $\mathcal{R}_t^k$ for the institution k at round t . It can be written as:

$$r_t^k = \frac{\left| \mathcal{R}_t^k \cap \left\{ X_{(m_{k,t}^0+1)}^0, X_{(m_{k,t}^0+2)}^0, \dots, X_{(m_{k,t}^0+A_{k,t}^0)}^0 \right\} \right|}{|\mathcal{R}_t^k|}. \quad (3.15)$$

Then, the pool evolution model depends on the fraction of the applicants from the minority group among the role models, r_t^k , instead of the fraction of the applicants from the minority in the admissions, π_t^k . The weighted role model parameter is defined as

$$\pi_t^r = \frac{\sum_{k=1}^K c_k r_t^k}{\sum_{k=1}^K c_k}. \quad (3.16)$$

If a group has a higher proportion of role models in comparison to its application proportion, it will provide positive reinforcement. The pool update is then governed by:

$$\theta_{t+1} = [\theta_t + \eta_t(\pi_t^r - s_t)]_C. \quad (3.17)$$

For $r = 1$, role model reinforcement reduces back to pure positive reinforcement. Setting $r = 0.5$ implies that minority applicants whose scores are above the median score for admitted applicants in their institution are role models driving pool evolution.

While we showed that convergence to long-term fairness target is assured under pure positive reinforcement, we shall now see that it is not the case, under this variant evolution model. First, let us focus first on a special case when $K = 1$. Under a limiting case of role-model reinforcement when $r = 1$, convergence to long-term fairness is guaranteed. In the other extreme case, when the bar for role models is extremely high, i.e., $r = \epsilon$ (small), $\pi_t^r = r_t^k \approx s_t$. Thus, although the MFG policy biases in favor of minority group, there could be no drift in the mean state parameter, leading to a stagnation in the composition of the applicant pool. However, when there are multiple institutions involved, this could lead to explicit negative feedback, causing the proportion of minority in the applicant pool to steadily reduce. We now show in Proposition 3.9 below that role model reinforcement can lead to negative feedback under the MFG policy, specifically when there are multiple institutions ($K > 1$). Assuming identical group-wise score distributions, we show that if the long-term fairness target is higher than the initial proportion of the minority group, the first institution will admit more minority applicants than their application proportion. This results in the subsequent institutions having a higher proportion of top admissions from the majority group, leading to a lower proportion of minority group role models. This can eventually cause the minority group to be removed from the applicant pool, as formally demonstrated next, and empirically supported by experiments in Section 3.5.1. Furthermore, we empirically demonstrate that coordinated behavior by the agents could help in alleviating this negative feedback.

Proposition 3.9 *Under Assumptions 3.1 and 3.2, for role model reinforcement there exists a role model parameter r that is small enough such that the MFG policy can cause negative feedback leading to the loss of representation of the under-represented group in the pool.*

Please see Appendix B.3 for proof.

Remark 3.10 *The evolution of the applicant pool under role model reinforcement is such that only admitted applicants from group $g = 0$ with scores larger than a group-independent threshold t_r^k , will contribute to reinforcing the pool. The threshold increases as we reduce the role model parameter r , raising the bar for being a role model. Let $t_g^{k, low}$ and $t_g^{k, up}$ denote the lower and upper thresholds of the MFG policy for group g and institution k . The key idea behind the proof is that when the initial mean parameter θ_t is small, there exists a small enough r such that for institution k , we have $t_r^k \geq t_1^{k, low} > t_0^{k, low}$. Using this, we show that the role model proportions are such that $r_t^k < s_t$ for $k \in \{2, 3, \dots, K\}$ and $r_t^1 = s_t$. Hence, the mean parameter obtains a negative drift, and the proportion of the minority group in the applicant pool approaches zero.*

In order to mitigate the effects of such negative feedback, we define a centralized version of the MFG policy in the next section.

3.4.1 Centralized Multi-agent Fair-Greedy Policy

The selection process under the original MFG policy, in which the institutions do not cooperate, is sequential based on the ranking order of the institutions. We now consider decision-making by a central coordinator which knows the utility of each institution, and maximizes the sum utility across institutions:

$$U(s_t, a_t^1, a_t^2, \dots, a_t^K) = \sum_{k=1}^K U_k(s_t, a_t^k). \quad (3.18)$$

Thus, the *Centralized Multi-agent Fair-Greedy (CMFG) policy* is defined as a set of policies $\pi_{CMFG}(s_t) = \{\pi_t^1, \pi_t^2, \dots, \pi_t^K\}$ maximizing the total utility over the space of joint

actions as:

$$\begin{aligned} \{\pi_t^1, \pi_t^2, \dots, \pi_t^K\} &= \arg \max_{a_t^1, a_t^2, \dots, a_t^K} U(s_t, a_t^1, a_t^2, \dots, a_t^K) \\ \text{subject to } 0 &\leq a_t^k \leq 1, \quad \sum_{k=1}^K a_t^k c_k \leq s_t, \quad \sum_{k=1}^K (1 - a_t^k) c_k \leq 1 - s_t. \end{aligned} \quad (3.19)$$

The main difference between the MFG and CMFG policies lies in the way that institutions work together. Intuitively, we can view the set of policies as the joint imposition of group-dependent thresholds corresponding to all institutions simultaneously in order to maximize the total utility, as opposed to the earlier decentralized version, where a higher ranked institution imposes its thresholds without consideration of downstream effects on the utilities of lower-ranked institutions. Thus, in CMFG policy, institutions collaborate with each other to maximize their total utility. This allows a central coordinator to distribute the cost of fairness among all institutions, potentially leading to a selection policy that is different from sequential decentralized selection. In particular, the effects of the negative feedback loop observed with the sequential selection under role model reinforcement can be mitigated, as shown in the experimental results in Section 3.5.1.

In summary, the discussion of the role model reinforcement variant model serves the following objectives within the context of this study. Firstly, it allows us to simulate a realistic and plausible scenario wherein only a subset of admitted applicants possesses the capacity to exert influence on future participation dynamics. To illustrate this concept, we designate top applicants from each institution as role models, albeit with an acknowledgment of the inherent complexity of truly characterizing successful applicants in real-world situations. This abstraction enables us to underscore the notion that a simplistic strategy of biasing in favor of minority applicants alone does not suffice as

a comprehensive solution for fostering future participation. Instead, it underscores the critical importance of institutional support for the admitted applicants to facilitate their growth and eventual success. The effectiveness of this support hinges upon the mechanisms and extent of backing provided by each institution. We show that as a growing number of admitted applicants receive the necessary support to attain a level of success that qualifies them as symbolic role models, the potential for positive reinforcement substantially increases. The precise means by which institutions implement and coordinate these support mechanisms constitutes an open question of paramount significance, particularly for policymakers tasked with shaping equitable participation.

3.5 Experimental Evaluation

We empirically evaluate the proposed MFG policy under different models for the evolution of the applicant pool. We begin with experiments on synthetic data, where the scores are sampled from group-specific Gaussian distributions, followed by learning the score distributions from real-world datasets. The MFG and centralized MFG policies are computed numerically and the optimal policies are obtained using the grid search over the space of policies. The number of applicants is kept finite in the experiments. The code for our experiments is available in the project repository.

3.5.1 Multi-agent framework evaluated on synthetic data

Identical score distributions We first consider the case when the score distributions across the two groups in the population are the same. The parameter setting employed for these experiments is listed next. The long-term fairness target is set at $\alpha = 0.4$. The number of institutions is $K = 3$ with capacities $c_1 = 0.1$, $c_2 = 0.05$ and $c_3 = 0.2$, resulting in a total capacity of 0.35. The score distributions are Gaussian, with means

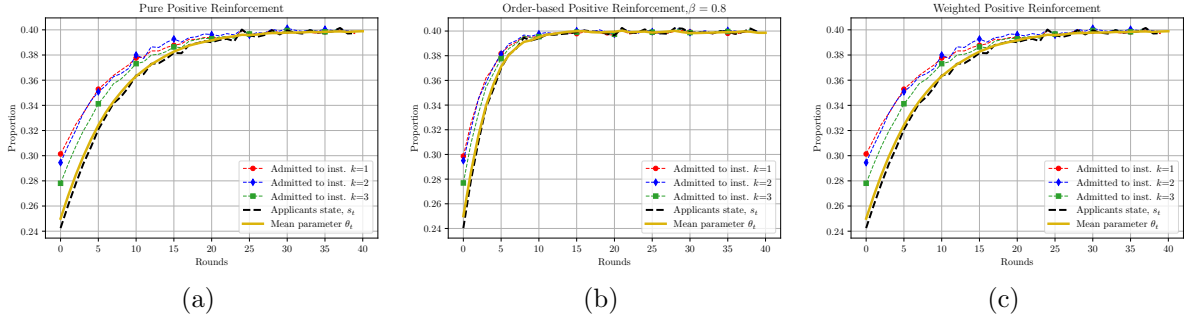


Figure 3.1: (a) With pure positive reinforcement, the MFG policy reaches long-term fairness when score distributions are identical for both groups. (b) The MFG policy attains long-term fairness more quickly with order-based positive reinforcement when $\beta = 0.8$. (c) Under weighted positive reinforcement, the MFG policy converges when institution weights are equal, for this setting.

$\mu_0 = \mu_1 = 5$ and variances $\sigma_0^2 = \sigma_1^2 = 1$. The initial mean parameter is $\theta_0 = 0.25$, giving the minority group lower representation in the resource pool. The range of the mean parameter is $[0.01, 0.99]$, and the mean parameter is projected to the range at each iteration. The state s_t is defined as $\frac{N_t^0}{N_t^0 + N_t^1}$, where N_t^0 follows a Poisson distribution with parameter $\theta_t N$, N_t^1 follows a Poisson distribution with parameter $(1 - \theta_t)N$, and $N = 400$. For computational efficiency, once the state is computed, the total number of applicants is fixed to 400, and the number of applicants from each group is adjusted according to the state and then rounded to the nearest integer for numerical convenience. Other parameters include $\lambda = 0.75$ and the step-size is fixed as $\eta = 0.5$. All the plots are averaged over 200 instances of the problem.

We demonstrate the evolution of the applicant and admission proportions, and the mean parameter, for the MFG policy under different reinforcement models. In Figure 3.1a, we can observe that under the pure positive reinforcement model, for each institution, the admission proportion is larger than the applicant proportion (state), and hence the weighted MFG policy being larger than the applicant proportion results in the positive reinforcement of the applicant pool (observed through the evolution of mean

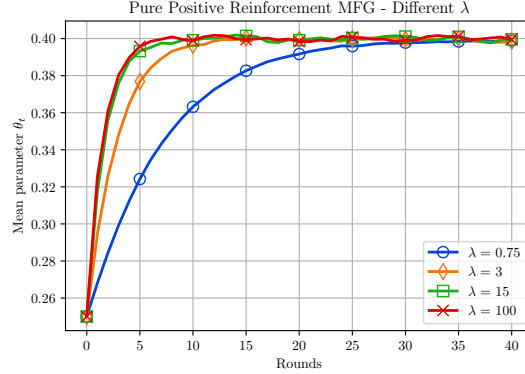


Figure 3.2: MFG policy under identical scores reaches long-term fairness target, independent of λ .

parameter θ_t) and long-term fairness in admissions as well. Next, we focus on the robustness of the MFG policy under different evolution models. The order-based reinforcement in Figure 3.1b uses $\beta = 0.8$, and shows that faster convergence can be achieved in comparison to the pure positive reinforcement model. The weighted positive reinforcement model in Figure 3.1c uses identical weights $w_k = 1$ for all institutions, showcasing the pool evolution when equal importance is assigned to every institution. In both these cases, for the setting considered, the MFG policy leads to the achievement of long-term fairness.

Further, we show the evolution of the mean applicant pool parameter under pure positive reinforcement model and the MFG policy for different weights, λ , allotted to fairness loss in Figure 3.2, where we observe that the mean parameter achieves the same long-term fairness target in equilibrium, independent of the hyperparameter λ , under identical score distributions, and long as $\lambda > 0$. However, the rate of convergence depends on λ .

Next, we gain valuable insights into the operational dynamics of the MFG policy. Our objective is to comprehend the trade-offs made, in terms of scores of admitted applicants of the majority and minority groups, to foster fairness. Figure 3.3a depicts the percentile

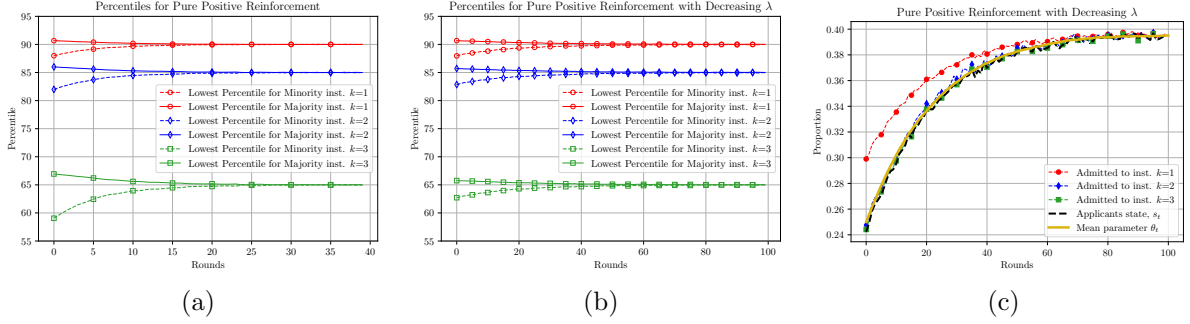


Figure 3.3: The score percentile of the admitted applicant with the lowest score, from each group and for each institution, under the setting where (a) $\lambda = 0.75$ for all institutions and (b) λ is in decreasing order for the three institutions. (c) The evolution of the mean parameter and admission proportions under the case of decreasing λ .

at which the admitted applicant with the lowest score is positioned, for both the groups and all institutions, when $\lambda = 0.75$ across all participating institutions. Next, in Figure 3.3b, we extend our examination to a scenario in which the λ values are decremented across institutions as $[0.75, 0.375, 0.1875]$, signifying that lower-ranked institutions temporarily de-prioritize diversity to minimize a significant decline in the quality of admitted applicants. Consequently, this approach illustrates how a judicious delay in the pursuit of immediate fairness objectives by lower-ranked institutions can assist in averting a large drop in their admission standards. However, this strategy of decreasing λ impacts the overall convergence rate (see Figure 3.3c), as all institutions experience a deferment in the applicant pool reaching the long-term fairness objective. This serves as an example of the delicate balance that institutions must navigate between the convergence to fairness targets and maximization of short-term rewards. Nonetheless, the precise strategies and mechanisms to be adopted by individual institutions in achieving this balance remains open.

Potential for negative feedback The evolution dynamics of role model reinforcement, under the MFG policy is examined here. We consider a scenario where role model

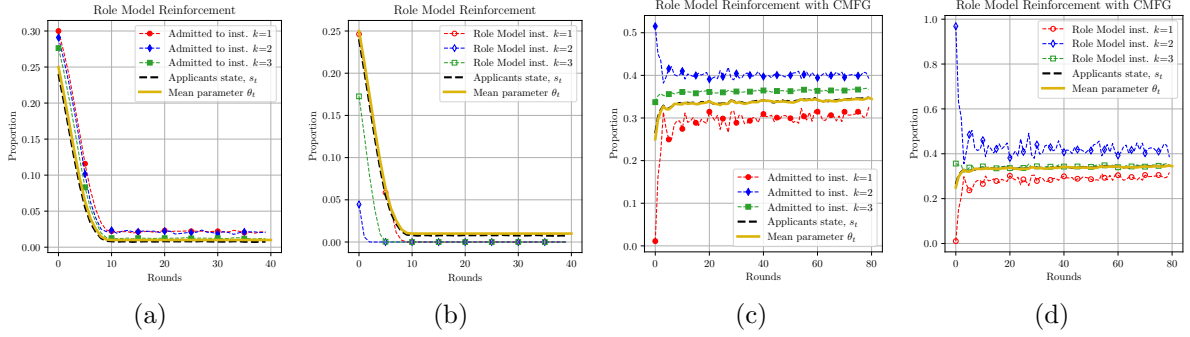


Figure 3.4: (a) MFG policy creates a negative feedback loop under the role-model reinforcement. (b) The evolution of the proportions of role models for each institution, under MFG policy. (c) CMFG policy could potentially alleviate negative feedback under role-model reinforcement. (d) The evolution of the proportions of role models for each institution, under CMFG policy.

reinforcement is applied with a parameter of $r = 0.5$, i.e., the admissions with scores above the median score in the respective institution are considered role models for the resource pool evolution. As seen in Figure 3.4a, the MFG policy creates a negative feedback loop, causing a significant decrease of the minority group in the applicant pool over time. This is because the role model proportions r_t^k are such that the weighted role model policy is consistently smaller than the state. In particular, as seen in Figure 3.4b, the second and the third institutions see a very small fraction of role models among the minority group, since the first institution admits the top minority applicants. These effects could potentially be alleviated by considering a centralized policy, such as the CMFG policy, whose evolution for the admitted applicants and the proportion of the role models under this setting are shown in Figure 3.4c and 3.4d. Although the initial dynamics between the institutions under CMFG policy are not desirable in the real-world and not well understood, it shows a potential of ultimately attaining an equilibrium close to the long-term fairness target. We also remark that the adverse effects under role model reinforcement in conjunction with the MFG policy could be avoided by increasing the r parameter, which essentially means that the institutions must design intervention mechanisms or remedies

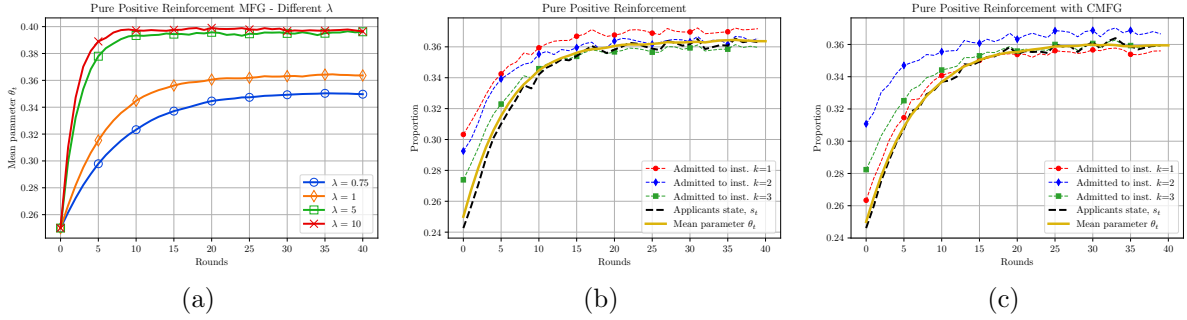


Figure 3.5: (a) Convergence of the mean parameter under the MFG policy with pure positive reinforcement is impacted by the fairness loss coefficient, λ , when score distributions are distinct. (b) The MFG policy reaches an equilibrium with pure positive reinforcement when score distributions are distinct. (c) CMFG also reaches an equilibrium, albeit with different decisions.

to support a large fraction of the admissions to eventually be successful in society, and influence others by being true role models.

For completeness, we show the evolution of applicant and admission proportions under the CMFG policy under pure positive reinforcement, order-based and weighted positive reinforcement models in Appendix B.4.1.

Distinct score distributions We now examine the scenario where the score distributions for the groups are distinct. We assume that the minority group has a slightly lower mean and higher variance than the majority group ($\mu_0 = 4.9, \mu_1 = 5$ and $\sigma_0^2 = 1.1, \sigma_1^2 = 1$). The other parameters remain the same as in the first scenario, with a long-term fairness target of $\alpha = 0.4$, capacities $c_1 = 0.1$, $c_2 = 0.05$ and $c_3 = 0.2$, an initial mean parameter $\theta_0 = 0.25$, and fixed step size $\eta = 0.5$.

Figure 3.5a shows the evolution of the mean parameter under different weights, λ , for fairness loss, under non-identical distributions. We observe that by altering λ , the equilibrium point can be varied. Thus, a higher value of λ is required to achieve the long-term fairness goal. We then show in Figures 3.5b and 3.5c, the evolution of the applicant pool, admission proportions, and their convergence under the pure positive reinforcement

model, under MFG and CMFG policies respectively, with $\lambda = 1$. They both converge to similar equilibrium although the admission decisions in initial rounds are quite distinct. We observe a behavior of negative feedback under the role-model reinforcement model under distinct score distributions as well. The results showing negative feedback and the alleviation of this effect through the CMFG policy are deferred to Appendix B.4.2. In the next section, we will examine cases where the score distributions are significantly different, obtained from a real-world dataset.

3.5.2 Multi-agent framework with semi-synthetic dataset

In this section, we report on experiments on the law school bar study dataset [76]. The dataset is used to construct the initial setup of the experiments as defined in the previous section, therefore we refer it to as a semi-synthetic dataset. This dataset contains information collected by the Law School Admission Council from law schools in the US, including information on whether an applicant passed the bar exam based on features such as LSAT scores, undergraduate GPA, law school GPA, race, gender, family income, age, and others. We take race as the protected attribute and simplify it to a binary classification problem by grouping all races except "white" into the "non-white" category, serving as the minority group (group 0) with only 25% representation in the dataset. The dataset used in our experiments contains around 1800 instances and we followed the same pre-processing steps at [77]. Next, we follow the procedure outlined in [72] to obtain the score distributions for each group. After pre-processing the data, we fit a logistic regression model to approximate the score distributions as Gaussians. This gives us the means $\mu_0 = -1.46$ and $\mu_1 = 0.79$, and the variances $\sigma_0^2 = 2.73$ and $\sigma_1^2 = 3.16$, as seen in Figure 3.6a. The long-term fairness target is set to 0.5, with three institutions having capacities of $c_1 = 0.15$, $c_2 = 0.10$ and $c_3 = 0.05$. The step size, η , is fixed at

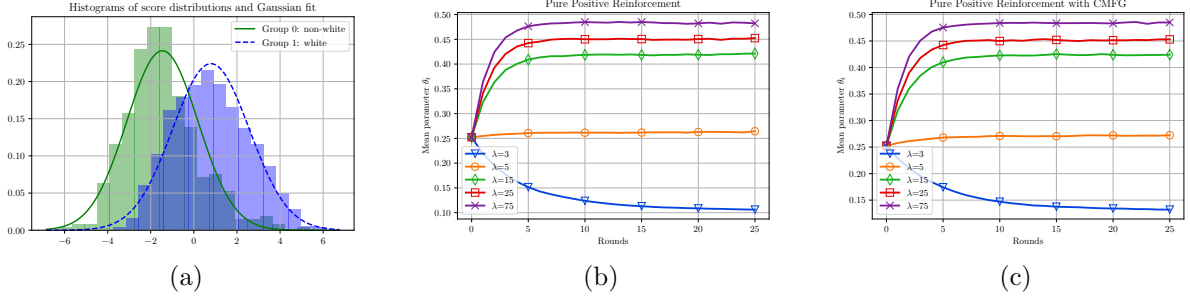


Figure 3.6: (a) Score distributions of the white and non-white groups in the law school dataset, along with their Gaussian approximations. (b), (c) Evolution of the mean parameter θ_t and its equilibrium with varying values of the fairness loss coefficient λ , under the MFG and centralized MFG policies, respectively, with pure positive reinforcement.

0.5. Our first examination is of the pure positive reinforcement model. Figures 3.6b and 3.6c depict the mean parameter, θ_t at equilibrium for the MFG and CMFG policies, respectively, with varying fairness loss coefficients, λ . A higher value of λ is needed to achieve the long-term fairness goal for both policies, with the CMFG showing a slightly better equilibrium point at a smaller value of λ (for instance, $\lambda = 3$). We defer the evaluation under role-model reinforcement model to Appendix B.4.3.

Our experiments with distinct distributions illustrate that the principles of reinforcement and the policies developed in this chapter hold even when the score distributions are non-identical.

3.6 Conclusion

This chapter studies the evolution of long-term fairness in a selection setting with multiple decision-makers choosing from a common pool. We have shown that the Multi-agent Fair-Greedy (MFG) policy does succeed in achieving long-term fairness targets under the model of pure positive reinforcement. However, when we set a higher bar for successful influence via the role-model reinforcement model, the minority group may ac-

tually experience negative feedback under MFG policy, and ultimately exit the selection process. Centralized coordination among the institutions could potentially alleviate this problem, raising the question of whether we can design mechanisms for competing institutions to collaborate (without laying themselves open to charges of collusion) in order to advance long-term fairness in society at large.

We hope that this work, despite the simplicity of the models considered, stimulates continuing discussion on the long-term societal impact of automated decision-making in a multi-agent setting, and how we can shape it. The sensitivity of our simple system to the model for evolution motivates a concerted effort to launch real-world experiments and data collection in which algorithm designers collaborate closely with social scientists and policymakers. An important complementary effort is to pursue analytical insights for more complex models that capture different aspects of the real world. For example, while our current concept of role model is based on the relative score upon admission, qualifications upon admission are often not a predictor of ultimate success; support mechanisms provided by the institution may be more important. Can we derive insights from a plausible model for such support mechanisms? Similarly, is there a qualitative difference in our conclusions if we relax the simplifying assumption of strict institutional rankings determining preferences for all applicants?

Chapter 4

Counterfactually Fair Conformal Prediction

4.1 Introduction

As machine learning models are being increasingly used in high-stakes applications like healthcare, criminal justice, test score estimation, etc.; it is crucial to quantify and mitigate the associated bias [78, 79]. This problem has been extensively explored in the past for both regression and classification problems [80, 81, 82]. The general idea is to ensure that subpopulations are treated equally (group fairness) or focus on individual fairness such as counterfactual fairness [11]. Although these fairness mitigation techniques are effective, most of them focus on point predictions and fail to capture the inherent uncertainty in ML models. Moreover, recent work [83] has shown that conformal prediction sets can improve decision quality in high-stakes tasks, underscoring the practical importance of uncertainty-aware prediction sets beyond point predictors.

An appealing methodology that promotes model fairness under uncertainty is the idea of *fair prediction sets* [84, 85]. These methods provide marginal coverage guarantees

while enforcing a group-level fairness criterion. Typically, they adapt classical group-fairness metrics for point predictors (e.g., demographic parity) to the set-valued setting via inclusion-based definitions [86]. To the best of our knowledge, all these past works on fair prediction sets have focused on group-level fairness. Furthermore, recent human-subject studies [87] suggest that enforcing certain group-fairness criteria for prediction sets, such as equalized coverage, may inadvertently increase disparate impact, highlighting the need for alternative fairness notions for conformal prediction beyond group-level criteria. Thus, although group-level fairness is a powerful tool, certain applications demand individual-level guarantees. Consider healthcare, for instance, where the intention could be to ensure a patient’s recommended treatment is not systematically influenced by protected attributes; such concerns have been widely discussed in the context of fair machine learning for laboratory medicine [88]. A method can satisfy group-level parity yet still produce biased recommendations for particular individuals, which could be undesirable for such an application. Furthermore, as noted by [11], the notion of counterfactual fairness better captures the causality of bias in ML algorithms.

The above motivates prediction sets with individual-level fairness. We extend counterfactual fairness notion to set-valued predictors: for a given individual, the prediction set should remain unchanged under counterfactual interventions on the protected attribute. For example, consider constructing a prediction set for a particular student’s SAT score; a counterfactually fair prediction set would be identical for that student whether we intervene to change their protected attribute (e.g., gender) or not.

We focus on conformal prediction (CP) and develop a framework called *Counterfactually Fair Conformal Prediction* (CF-CP). Concretely, we compute conformity scores under each intervention on the protected attribute and then aggregate these per-intervention scores using a symmetric function to obtain a counterfactually fair score. We then use these scores with the standard split CP procedure. We show that under an invertible

structural causal model (SCM) and the exchangeability assumptions, CF-CP guarantees both marginal coverage and set-level counterfactual fairness. We emphasize that the latter holds even when the underlying point predictor is not counterfactually fair. Empirically, we evaluate CF-CP on synthetic and real-world datasets, showing that it significantly reduces unfairness in the prediction sets compared to standard split CP, with only a modest increase in average set size.

We note that prediction sets generated by CP are not inherently fair. They can systematically differ between protected groups or change when we hypothetically alter an individual’s sensitive attributes [84, 86]. CF-CP alters the scoring function of standard CP such that counterfactual fairness and coverage is ensured.

The key contributions of this chapter are summarized below. This chapter is based in part on our published work [18].

- We extend the notion of counterfactual fairness to set-valued predictors and propose a post-training conformal prediction procedure (CF-CP) that symmetrizes conformity scores across protected attribute interventions.
- We prove that under an invertible SCM and exchangeability assumptions, CF-CP guarantees both marginal coverage and set-level counterfactual fairness.
- We empirically demonstrate that CF-CP effectively reduces counterfactual unfairness in prediction sets on synthetic and real-world datasets while maintaining desired coverage.

4.1.1 Related Work

Fairness in conformal prediction has primarily focused on group fairness notions [84]. One recent method adapts common group fairness notions (e.g., demographic parity,

equalized odds) to conformal prediction without requiring group membership during test time [86]. One line of work tailors CP-based regression to group-fairness goals, specifically Demographic Parity and Equal Opportunity [85, 89]. A separate line advances CP methods that seek equalized coverage across groups [90]. Our work differs in that we focus on individual-level counterfactual fairness for set-valued predictors. Moreover, recent human-subject studies show that enforcing equalized coverage across groups can inadvertently increase disparate impact in decision-making, further underscoring the need for alternative fairness notions for prediction sets [87]. There is also a growing body of work that combines causality with uncertainty quantification, though not in the context of fairness. Several recent works study conformal prediction for potential outcomes in causal inference [91, 92, 93]. However, these methods do not consider fairness. We next summarize counterfactual fairness methods for point predictors.

Counterfactual fairness (CF) is an individual-level fairness notion that requires a predictor’s output to remain invariant under counterfactual changes to protected attributes [11]. There are several strategies to design CF predictors. One approach is to learn a predictor only on the features that are not descendants of the protected attribute A in the causal graph [11]. Another approach achieves CF by mixing representations of factual and counterfactual instances before feeding them to a predictor [94]. A related approach combines both factual and counterfactual predictions weighted by the probability of the corresponding protected attribute value [95]. Other works study regularizers that penalize differences in predictions between factual and counterfactual instances [96, 97]. These methods focus on point predictors, while we extend CF to set-valued predictors and design a conformal prediction procedure that enforces this property.

4.2 Background: Conformal Prediction

Notation and Setup. We consider observed features $X \in \mathcal{X}$, a protected attribute $A \in \mathcal{A}$, and a label $Y \in \mathcal{Y}$. Throughout this chapter, we assume $\mathcal{A}$ is finite (e.g., binary or categorical). We write a (possibly randomized) predictor as $\hat{f} : \mathcal{X} \times \mathcal{A} \rightarrow \mathcal{S}$, where $\mathcal{S} = \mathbb{R}$ for regression or $\mathcal{S} = \Delta^{K-1}$ for K -class classification. A *prediction set* procedure maps (X, A) to a subset $\mathcal{C}(X, A) \subseteq \mathcal{Y}$.

For causal notation, we adopt structural causal models (SCMs) [98]. An SCM is a tuple (U, V, F) , where U are exogenous variables, $V = \{X, A, Y\}$ are endogenous variables, and F are structural equations that determine each variable in V as a function of its parents and some exogenous noise. Counterfactuals consider the following question: “What would the value of observed variables be, had we intervened to set A to a different value?” We denote by $X_{A \leftarrow a}$ the counterfactual version of X under the intervention $do(A = a)$, and similarly for $Y_{A \leftarrow a}$. For more details on structural causal models and counterfactuals, see [98].

4.2.1 Split Conformal Prediction

Here, we briefly review conformal prediction (CP); see [13] for a recent survey. Conformal prediction (CP) constructs prediction sets with a pre-specified probability $1 - \alpha$ of containing the true label, without distributional assumptions beyond exchangeability [12, 99]. For simplicity, we focus on the popular *split* (or inductive) CP variant [100]. The protected attribute A is not needed for standard CP, but we include it in the notation for clarity and to make explicit the dependence used in later sections. Suppose we are given a pre-trained predictor $\hat{f}$ (e.g., a neural network) and a calibration dataset of size n , $\{(X_i, A_i, Y_i)\}_{i=1}^n$, independent of $\hat{f}$. The goal is to construct a prediction set $\mathcal{C}_\alpha(X, A)$ that covers the true label Y with probability of at least $1 - \alpha$ over the randomness of

(X, A, Y) and the calibration dataset. To this end, we define a *conformity score* function $s : \mathcal{X} \times \mathcal{A} \times \mathcal{Y} \rightarrow \mathbb{R}$ that measures how well a candidate label y conforms with the features (x, a) under the predictor $\hat{f}$. Without loss of generality, we use a negative-oriented score, where smaller values indicate better conformity. The score function inherently depends on the predictor $\hat{f}$ but for ease of notation we omit this dependence. Then, one computes conformity scores on the calibration set:

$$S_i = s(X_i, A_i, Y_i), \quad i \in \mathcal{I}_{\text{cal}}, \quad (4.1)$$

where $\mathcal{I}_{\text{cal}}$ denotes the index set of the calibration data. Let $\hat{q}_{1-\alpha}$ be the $(1 - \alpha)(1 + 1/|\mathcal{I}_{\text{cal}}|)$ -empirical quantile of $\{S_i\}_{i \in \mathcal{I}_{\text{cal}}}$. For a test point (X_{n+1}, A_{n+1}) , the split conformal prediction set is computed as follows:

$$\mathcal{C}_\alpha(X_{n+1}, A_{n+1}) = \{y \in \mathcal{Y} : s(X_{n+1}, A_{n+1}, y) \leq \hat{q}_{1-\alpha}\} \quad (4.2)$$

If (X_i, A_i, Y_i) and $(X_{n+1}, A_{n+1}, Y_{n+1})$ are exchangeable, then $\mathcal{C}_\alpha(X_{n+1}, A_{n+1})$ satisfies the *marginal coverage* guarantee:

$$\Pr \{Y_{n+1} \in \mathcal{C}_\alpha(X_{n+1}, A_{n+1})\} \geq 1 - \alpha. \quad (4.3)$$

If the score values are almost surely distinct, the probability above is upper bounded by $1 - \alpha + 1/(n + 1)$. A detailed proof of this result can be found in [101]. The coverage guarantee is marginal, and the randomness is over both the calibration and test points.

4.3 Counterfactual Fairness

In this section, we formalize counterfactual fairness for point predictors and extend this definition to prediction set procedures. We also briefly recall past works that ensure counterfactual fairness for point predictors.

4.3.1 Point predictors

Building on the causal framework of Pearl [98], counterfactual fairness (CF) was introduced as a notion requiring a predictor’s output to remain invariant under counterfactual interventions on protected attributes [11]. More formally:

Definition 4.1 (Counterfactual Fairness—point predictor) *A predictor $\hat{f} : \mathcal{X} \times \mathcal{A} \rightarrow \mathcal{S}$ is counterfactually fair (CF) if, for all $a, a' \in \mathcal{A}$ and $s \in \mathcal{S}$,*

$$\Pr \left\{ \hat{f}(X_{A \leftarrow a}, a) = s \mid A = a, U = u \right\} = \Pr \left\{ \hat{f}(X_{A \leftarrow a'}, a') = s \mid A = a, U = u \right\} \quad (4.4)$$

That is, after intervening on A while holding the latent U fixed, the distribution of predictions should not change. Equivalently, $\hat{f}(X_{A \leftarrow a}, a) \mid (A = a, U = u)$ and $\hat{f}(X_{A \leftarrow a'}, a') \mid (A = a, U = u)$ have the same conditional distribution.

Several strategies exist to design CF predictors $\hat{f}$ satisfying (4.4). One approach is to learn a predictor only on the features that are not descendants of A in the causal graph; then no causal path exists from A to $\hat{f}(X, A)$, ensuring CF [11]. This approach may be impractical if all features are descendants of A . Counterfactually Fair Representations (CFR) [94] satisfies fairness by learning a latent representation Z for factual and counterfactual instances, then combines them via a permutation-invariant operation (e.g., mean) before feeding to a predictor. Another method satisfies CF by combining both factual and counterfactual predictions weighted by the probability of the corresponding

protected attribute value, called Plug-in Counterfactual Fairness (PCF) [95].

4.3.2 Prediction Sets

We extend counterfactual fairness notion from *point predictors* to *prediction sets* by requiring that the set produced for an individual remain invariant under counterfactual interventions on the protected attribute. We call this property *set-level counterfactual fairness*.

Definition 4.2 (Counterfactual fairness—prediction sets) *A set-valued procedure $\mathcal{C}_\alpha$ is counterfactually fair if, for all $a, a' \in \mathcal{A}$ and $y \in \mathcal{Y}$,*

$$\Pr \{y \in \mathcal{C}_\alpha(X_{A \leftarrow a}, a) \mid A = a, U = u\} = \Pr \{y \in \mathcal{C}_\alpha(X_{A \leftarrow a'}, a') \mid A = a, U = u\}. \quad (4.5)$$

That is, for any individual with latent attributes $U = u$ and protected attribute $A = a$, the probability of including any label y in the prediction set is invariant to counterfactual changes in A .

4.4 Counterfactually Fair Prediction Sets

We aim to design a split-CP procedure that preserves marginal coverage while enforcing set-level counterfactual invariance. Precisely, given a pre-trained (possibly counterfactually unfair) predictor $\hat{f}$, a calibration set $\mathcal{I}_{\text{cal}}$, and a target level $1 - \alpha$, construct a conformal prediction procedure $\mathcal{C}_\alpha$ that (i) preserves the standard marginal coverage guarantee and (ii) satisfies set-level counterfactual fairness (Definition 4.2).

In this section, we propose an algorithm that satisfies these two properties. Then we compare CF-CP to baselines and discuss the differences. Finally, we demonstrate

that this procedure results in counterfactually fair prediction sets along with marginal coverage guarantees.

4.4.1 Score Symmetrization and CF-CP Sets

Let $s : \mathcal{X} \times \mathcal{A} \times \mathcal{Y} \rightarrow \mathbb{R}$ be any conformity score used by split CP with a base predictor $\hat{f}$. We define the *counterfactual score symmetrization* by aggregating scores over protected-attribute interventions:

$$s_{\text{cf}}(x, a, y) = \text{Agg} \{s(x_{A \leftarrow a'}, a', y) : a' \in \mathcal{A}\} \quad (4.6)$$

where Agg is a symmetric operator that is invariant to permutations of its inputs and $x_{A \leftarrow a'}$ denotes the counterfactual features obtained by setting $A := a'$. Examples of the operator Agg include mean, max and min operations—we elaborate more on this below.

We then run split CP *with* s_{cf} : on calibration points $\{(X_i, A_i, Y_i)\}_{i \in \mathcal{I}_{\text{cal}}}$ compute $S_i^{\text{cf}} = s_{\text{cf}}(X_i, A_i, Y_i)$, take the usual quantile $\hat{q}_{1-\alpha}$ with finite sample correction. At test time, for a new point (x, a) , the CF-CP set is

$$\mathcal{C}_\alpha^{\text{CF}}(x, a) = \{y \in \mathcal{Y} : s_{\text{cf}}(x, a, y) \leq \hat{q}_{1-\alpha}\}. \quad (4.7)$$

This procedure is summarized in Algorithm 1.

Aggregator choices. The only requirement on Agg is to be permutation-invariant—i.e., the output of the aggregator should not change with the order of the inputs. In this chapter, we consider three simple choices: (i) **mean**, (ii) **max**, and (iii) **min**. During the calibration threshold learning (step 3 of Algorithm 1), **max** leads to larger thresholds $\hat{q}_{1-\alpha}$, because it takes the maximum score across interventions for each sample in

Algorithm 1: Split CF-CP via score symmetrization

Input : Base predictor $\hat{f}$, score $s(\cdot)$, calibration set $\{(X_i, A_i, Y_i)\}_{i \in \mathcal{I}_{\text{cal}}}$, target $1 - \alpha$, aggregator Agg

Output: Prediction set $\mathcal{C}_\alpha^{\text{CF}}(x, a)$

- 1 Define $s_{\text{cf}}(x, a, y) \leftarrow \text{Agg}\{s(x_{A \leftarrow a'}, a', y) : a' \in \mathcal{A}\};$
- 2 **foreach** $i \in \mathcal{I}_{\text{cal}}$ **do**
- 3 $\lfloor$ Compute $S_i^{\text{cf}} \leftarrow s_{\text{cf}}(X_i, A_i, Y_i);$
- 4 Set $\hat{q}_{1-\alpha} \leftarrow \text{Quantile}_{\lceil (|\mathcal{I}_{\text{cal}}|+1)(1-\alpha) \rceil / |\mathcal{I}_{\text{cal}}|}(\{S_i^{\text{cf}}\});$
- 5 At test time, for (x, a) , form $\mathcal{C}_\alpha^{\text{CF}}(x, a) \leftarrow \{y : s_{\text{cf}}(x, a, y) \leq \hat{q}_{1-\alpha}\};$

calibration dataset. However, at test time (step 4 of Algorithm 1), the **max** aggregator leads to intersection of the per-intervention sets, which balances out the effect of a larger threshold. Conversely, **min** leads to smaller thresholds at the calibration phase, but the test-time sets are unions of per-intervention sets, which can lead to larger prediction sets. The **mean** aggregator is a compromise between these cases in both calibration and test phases. While we focus on these simple aggregators, more complex choices are possible as long as they are permutation-invariant.

Computational overhead. Compared to standard split-CP, CF-CP introduces two sources of additional overhead: (i) generating counterfactual interventions, and (ii) computing conformity scores for each intervention $a \in \mathcal{A}$. Overall, CF-CP incurs an additional $\mathcal{O}(|\mathcal{A}|)$ factor compared to split-CP. Thus, when the protected attribute has small cardinality, the computational cost of CF-CP remains close to that of split-CP.

4.4.2 Baselines and Comparisons

We compare CF-CP to two types of baselines that either retrain a fair point predictor or apply post-hoc set manipulations, and we note their limitations.

Retraining-based approaches. One option is to first obtain a point predictor that is counterfactually fair—*e.g.*, by excluding descendants of A in the SCM or via counterfactually fair representations (CFR) / plug-in counterfactual fairness (PCF)—and then apply standard split-CP on its score. This can yield set-level invariance by construction, but it typically trades predictive accuracy for fairness and thus inflates set sizes to maintain coverage; it also requires retraining, which may be infeasible. In addition, PCF variant relies on access to the probability distribution of the protected attribute, $\Pr(A)$, for post-training adjustments.

Post-hoc union over interventions. A second baseline constructs separate CP sets under each protected-attribute intervention and unions them:

$$\mathcal{C}_\alpha^\cup(x) := \bigcup_{a' \in \mathcal{A}} \mathcal{C}_\alpha(x_{A \leftarrow a'}, a') \quad (4.8)$$

This is again counterfactually fair by construction, but the union systematically enlarges sets and pushes effective coverage above the target coverage, $1 - \alpha$.

How CF-CP differs. CF-CP calibrates a *single, symmetrized* conformity score that aggregates per-intervention scores at the instance level. In contrast to the post-hoc union, CF-CP avoids the union-induced overcoverage while still ensuring set-level counterfactual invariance. Compared to retraining-based approaches (CFR/PCF or descendants-exclusion), CF-CP is training-free, does not require modifying features or imposing invariance penalties, and does not need access to $\Pr(A)$; all aggregation occurs on scores, not on model parameters or population distributions.

4.4.3 Theoretical Guarantees

In this section we show that CF-CP satisfies the set-level counterfactual fairness of Definition 4.2 while preserving the marginal coverage guarantee of split CP. First, we state the main assumption needed for our theoretical analysis.

Assumption 4.3 *The structural mapping between X and U is invertible for given A . Also, U and A are independent.*

The independence assumption is standard in the counterfactual fairness literature. Even though the invertibility assumption may seem strong, it is commonly used in the counterfactual estimation and fairness literature [102, 103, 95]. We also note that after relaxing the assumption, one can still achieve approximate counterfactual fairness, as we show empirically in Section 4.5.

Assumption 4.3 asserts that there exists a mapping $g : \mathcal{U} \times \mathcal{A} \rightarrow \mathcal{X}$ such that $X = g(U, A)$ and, for each $a \in \mathcal{A}$, the mapping $g_a(u) := g(u, a)$ is invertible with inverse $g_a^{-1} : \mathcal{X} \rightarrow \mathcal{U}$. For any target $a' \in \mathcal{A}$, the corresponding counterfactual features are therefore

$$x_{A \leftarrow a'} = g_{a'}(U) = g_{a'}(g_a^{-1}(x)). \quad (4.9)$$

Also, the symmetrized score can be written as:

$$s_{\text{cf}}(x, a, y) = \text{Agg} \{s(g_{a'}(g_a^{-1}(x)), a', y) : a' \in \mathcal{A}\} \quad (4.10)$$

This invertibility makes precise that the multiset $\{x_{A \leftarrow a'}\}_{a'}$ depends only on the shared latent U , not on the factual a . This results in invariance of the symmetrized score. We formalize this observation in the following lemma.

Lemma 4.4 (Invariance of the symmetrized score) *Let Assumption 4.3 hold. Fix any $y \in \mathcal{Y}$. Then for all $a, a' \in \mathcal{A}$,*

$$s_{\text{cf}}(X_{A \leftarrow a}, a, y) = s_{\text{cf}}(X_{A \leftarrow a'}, a', y). \quad (4.11)$$

The proof is given in the Appendix C.1.1. The key idea is that, under Assumption 4.3, the latent U can be uniquely recovered from the factual features X and protected attribute $A = a$ via the inverse function g_a^{-1} . This allows us to express all counterfactual features $X_{A \leftarrow a'}$ as functions of $X_{A \leftarrow a}$ and $A = a$, and vice versa. The symmetry of s_{cf} then follows from the permutation-invariance of Agg .

This score-level invariance leads to the desired set-level counterfactual fairness. Since the score value is the same across counterfactuals, the prediction sets formed by thresholding the score at $\hat{q}_{1-\alpha}$ must also coincide.

Corollary 4.5 (Set-level counterfactual invariance) *Assume that the conditions at Lemma 4.4 hold, then for all $a, a' \in \mathcal{A}$,*

$$\mathcal{C}_{\alpha}^{\text{CF}}(X_{A \leftarrow a}, a) = \mathcal{C}_{\alpha}^{\text{CF}}(X_{A \leftarrow a'}, a'). \quad (4.12)$$

The proof is immediate from Lemma 4.4 and the definition of CF-CP sets in (4.7).

The previous lemma and corollary establish that CF-CP enforces *set-level counterfactual invariance*. We now argue that replacing s with s_{cf} does *not* disturb the exchangeability required by split CP, so the standard marginal coverage guarantee continues to hold.

Theorem 4.6 (Exchangeability preservation) *Assume that Assumption 4.3 holds and that the calibration points $\{(X_i, A_i, Y_i)\}_{i \in \mathcal{I}_{\text{cal}}}$ and the test point $(X_{n+1}, A_{n+1}, Y_{n+1})$ are exchangeable. Suppose the score function s and the aggregator Agg are measurable,*

and Agg is permutation-invariant. Define the counterfactual symmetrization

$$s_{\text{cf}}(x, a, y) = \text{Agg} \{s(x_{A \leftarrow a'}, a', y) : a' \in \mathcal{A}\}.$$

Then, the scores

$$\{S_i^{\text{cf}} = s_{\text{cf}}(X_i, A_i, Y_i) : i \in \mathcal{I}_{\text{cal}} \cup \{n+1\}\} \quad (4.13)$$

are exchangeable.

The proof is given in the Appendix C.1.2. The main idea is that s_{cf} is a coordinate-wise measurable transformation of the exchangeable tuples (X_i, A_i, Y_i) , hence the resulting scores remain exchangeable.

Theorem 4.6 shows that the exchangeability structure required by split CP is preserved when using the symmetrized score s_{cf} . This leads to the following coverage guarantee for CF-CP.

Corollary 4.7 (Marginal coverage for CF-CP) *Under the same conditions as Theorem 4.6, the split conformal set constructed with s_{cf} satisfies the standard marginal coverage guarantee:*

$$\Pr \{Y_{n+1} \in \mathcal{C}_\alpha^{\text{CF}}(X_{n+1}, A_{n+1})\} \geq 1 - \alpha. \quad (4.14)$$

The proof directly follows the standard split CP coverage proof [12, 101] applied to the exchangeable scores S_i^{cf} .

Combining Corollaries 4.5 and 4.7, we conclude that CF-CP achieves both marginal coverage and set-level counterfactual invariance, as desired.

4.5 Experiments

In this section, we empirically evaluate CF-CP on synthetic and real datasets, spanning regression and classification tasks, and show that it achieves the desired counterfactual fairness while meeting the target coverage rate with minimal increase in the size of the prediction sets. The code is available in the project repository.

Methods. We compare the following methods: (i) **Split CP**: standard split conformal prediction using the original conformity score with the base predictor; (ii) **Post-hoc Union**: computes CP sets under each intervention and takes their union, specifically, $\mathcal{C}_\alpha^\cup(x) := \bigcup_{a' \in \mathcal{A}} \mathcal{C}_\alpha(x_{A \leftarrow a'}, a')$; (iii) **Counterfactual Fairness with U** [11]: train a counterfactually fair predictor by excluding descendants of A in the SCM, to simplify this we train base model only on U similar to the approach in [95] followed by split CP; (iv) **Counterfactual Fair Representations (CFR)** [94]: trains a predictor on the augmented features as the mean of factual and counterfactual features, specifically, $\tilde{X} = \frac{1}{|\mathcal{A}|} \sum_{a' \in \mathcal{A}} X_{A \leftarrow a'}$ and then applies split CP; (v) **Plug-in Counterfactual Fairness (PCF)** [95]: uses a weighted combination of the predictions for each intervention, specifically, $\hat{f}(X, A) = \sum_{a' \in \mathcal{A}} \Pr(A = a') \hat{f}(X_{A \leftarrow a'}, a')$ where $\hat{f}$ is the base predictor and then follows with split CP; and lastly our method (vi) **CF-CP**: our proposed method that uses the counterfactual symmetrized score in split CP, and we consider three aggregation functions: **mean**, **max**, and **min**. Following common practice, in the classification tasks if the prediction set is empty, we include the label with the highest predicted probability. The base predictor $\hat{f}$ is a linear model for both regression and classification tasks. More details on hyperparameters and training are in the Appendix C.2.1.

Score functions. For regression tasks, we use the absolute residual score $s(x, a, y) = |\hat{f}(x, a) - y|$. So, the less the absolute error, the better the conformity. For classification

tasks, we use Least Ambiguous Classifier (LAC) [104] score $s(x, a, y) = 1 - \hat{f}_y(x, a)$, where $\hat{f}_y(x, a)$ is the predicted probability of class y . The more probable the class, the better the conformity. We also provide additional results with different base score functions in Appendix C.3.

Evaluation metrics. For each dataset and method, we report the following metrics— $\uparrow$ and $\downarrow$ indicate that a larger and smaller value is desirable, respectively: (i) **MSE** ($\downarrow$)/**Accuracy** ($\uparrow$): the mean squared error for regression and accuracy for classification on the test set; (ii) **Total Effect(TE)** ($\downarrow$): measures the counterfactual fairness of the base predictor $\hat{f}$, for each test point, we compute the change in the model’s output when A is counterfactually flipped, and then average over the test set; (iii) **Coverage**: the empirical coverage of the prediction sets on the test set, $\frac{1}{|\mathcal{I}_{\text{test}}|} \sum_{i \in \mathcal{I}_{\text{test}}} \mathbb{I}[Y_i \in \mathcal{C}_\alpha(X_i, A_i)]$; (iv) **Avg. Set Size** ($\downarrow$): the average size of the prediction sets on the test set, for regression it is the average length of the intervals, and for classification it is the average cardinality of the sets, specifically, $\frac{1}{|\mathcal{I}_{\text{test}}|} \sum_{i \in \mathcal{I}_{\text{test}}} |\mathcal{C}_\alpha(X_i, A_i)|$; and (v) **Counterfactual Set Disparity (CSD)** ($\downarrow$): quantifies how much the set changes across protected-attribute interventions for each test point. The details of CSD are as follows: For binary A , we use the Jaccard distance between the factual and counterfactual sets:

$$\text{CSD}(x) = 1 - \frac{|\mathcal{C}_\alpha(x_{A \leftarrow a}, a) \cap \mathcal{C}_\alpha(x_{A \leftarrow a'}, a')|}{|\mathcal{C}_\alpha(x_{A \leftarrow a}, a) \cup \mathcal{C}_\alpha(x_{A \leftarrow a'}, a')|}. \quad (4.15)$$

For multi-valued A , it can be extended by averaging pairwise Jaccard distances across interventions relative to the factual value. A lower CSD indicates more similar sets across counterfactuals, with 0 indicating identical sets and 1 indicating non-overlapping sets. We report the average counterfactual set disparity on the test set, specifically, $\frac{1}{|\mathcal{I}_{\text{test}}|} \sum_{i \in \mathcal{I}_{\text{test}}} \text{CSD}(X_i)$. The metrics for each method are computed over 10 random splits

of the data into training, calibration, and test sets, and we report the mean and standard deviation in the tables and figures. For all tables in this chapter, we use a common highlighting convention. For the tables that use oracle counterfactuals, bold (underlined) values mark the best (second-best) MSE/Accuracy and Avg. Set Size among the methods that achieve the target coverage and perfect counterfactual fairness for prediction sets (CSD=0). For the tables with noisy counterfactuals, bold (underlined) values mark the best (second-best) CSD, since in this setting we are primarily interested in how fairness behaves under imperfect counterfactuals.

Datasets. We evaluate on synthetic and real datasets spanning regression and multiclass classification. Throughout our experiments, we assume access to an underlying structural causal model that generates the data; for the synthetic datasets, the SCM is specified by construction, and for the real datasets, the SCMs are obtained by fitting structural equations to the observed data; full details on data generation, preprocessing, SCM specification and fitting, and counterfactual construction are in Appendix C.2.2.

Synthetic Regression. A nonlinear SCM with binary protected attribute A generates (X, Y) ; the task is to predict Y from (X, A) adopted from [94]. Counterfactuals are formed by flipping A in the X -equation while holding exogenous noise fixed.

Synthetic Classification. A multi-class ($K=10$) SCM extends prior binary settings; the task is to predict Y from (X, A) . It is inspired by [95] but extended to multi-class targets. Counterfactuals replace A only in the structural equation for X (latent noise held fixed).

Law School. Predict first-year average (FYA) from LSAT, UGPA, race, and gender [76]; we take race as A (binary: white vs non-white). Counterfactuals are obtained by fitting a LiNGAM-based SCM to the observed data and intervening on A .

Bias in Bios. Predict profession (28 classes) from biography text; A is gender (bi-

Method	Regression					Classification				
	MSE ($\downarrow$)	TE ($\downarrow$)	Coverage	Avg. Set Size ($\downarrow$)	CSD ($\downarrow$)	Accuracy ($\uparrow$)	TE ($\downarrow$)	Coverage	Avg. Set Size ($\downarrow$)	CSD ($\downarrow$)
SplitCP	0.377 ± 0.012	1.219 ± 0.014	0.901 ± 0.008	2.032 ± 0.038	0.713 ± 0.009	0.730 ± 0.011	0.543 ± 0.037	0.905 ± 0.011	2.025 ± 0.209	0.642 ± 0.043
Post-hoc Union	0.377 ± 0.012	1.219 ± 0.014	0.944 ± 0.005	3.251 ± 0.043	0.000 ± 0.000	0.730 ± 0.011	0.543 ± 0.037	0.936 ± 0.010	3.123 ± 0.255	0.000 ± 0.000
CFU	1.112 ± 0.017	0.000 ± 0.000	0.901 ± 0.008	3.447 ± 0.088	0.000 ± 0.000	<u>0.589 ± 0.017</u>	0.000 ± 0.000	0.907 ± 0.009	2.700 ± 0.125	0.000 ± 0.000
CFR	0.832 ± 0.014	0.000 ± 0.000	0.899 ± 0.010	2.964 ± 0.055	0.000 ± 0.000	<u>0.589 ± 0.017</u>	0.000 ± 0.000	0.908 ± 0.009	2.700 ± 0.125	0.000 ± 0.000
PCF	<u>0.826 ± 0.013</u>	0.000 ± 0.000	0.898 ± 0.008	<u>2.951 ± 0.046</u>	0.000 ± 0.000	0.571 ± 0.017	0.000 ± 0.000	0.905 ± 0.012	2.526 ± 0.166	0.000 ± 0.000
CF-CP-mean	0.377 ± 0.012	1.219 ± 0.014	0.899 ± 0.010	2.971 ± 0.050	0.000 ± 0.000	0.730 ± 0.011	0.543 ± 0.037	0.904 ± 0.014	<u>2.542 ± 0.185</u>	0.000 ± 0.000
CF-CP-max	0.377 ± 0.012	1.219 ± 0.014	0.900 ± 0.009	3.275 ± 0.053	0.000 ± 0.000	0.730 ± 0.011	0.543 ± 0.037	0.935 ± 0.006	3.696 ± 0.168	0.038 ± 0.008
CF-CP-min	0.377 ± 0.012	1.219 ± 0.014	0.900 ± 0.010	2.897 ± 0.058	0.000 ± 0.000	0.730 ± 0.011	0.543 ± 0.037	0.906 ± 0.013	2.581 ± 0.180	0.000 ± 0.000

Table 4.1: Synthetic results at $\alpha = 0.1$ (mean $\pm$ std over 10 runs). Left block: regression; right block: classification. ($\downarrow$) and ($\uparrow$) indicate that a smaller and larger value is desirable, respectively. Highlighting follows the convention described in the evaluation metrics paragraph.

nary) [105]. Counterfactuals for this dataset are constructed by the method proposed in [106], which involves first obtaining text embeddings via a pre-trained BERT model [107], then decomposing the embedding into components aligned with and independent of A ; counterfactuals flip the A component while keeping the independent component fixed.

Except for Bios dataset, we use $\alpha = 0.1$; for Bios we use $\alpha = 0.025$ since $\alpha = 0.1$ leads to near-singletons due to high accuracy of the base predictor.

4.5.1 Oracle Counterfactuals

We first present results when the counterfactuals are generated from the true SCM (oracle counterfactuals)—later, we consider the case where noise is added to the exogenous variables. Table 4.1 shows the results on synthetic regression and classification datasets. Table 4.2 shows the results on real datasets, Law School for regression and Bios for classification. First, we observe that vanilla split CP achieves the target coverage but has high counterfactual set disparity (CSD) ranging from 0.19 to 0.71. As expected, Post-hoc Union achieves counterfactual invariance for prediction sets (CSD=0), but at the cost of a significant increase in the size of the prediction sets and over-coverage due to inflation of the prediction sets. Methods that train counterfactually fair predictors (CFU, CFR, PCF) achieve counterfactual fairness for both point predictions and prediction sets

Method	Law School Regression					Bios Classification				
	MSE ($\downarrow$)	TE ($\downarrow$)	Coverage	Avg. Set Size ($\downarrow$)	CSD ($\downarrow$)	Accuracy ($\uparrow$)	TE ($\downarrow$)	Coverage	Avg. Set Size ($\downarrow$)	CSD ($\downarrow$)
SplitCP	0.758 $\pm$ 0.005	0.723 $\pm$ 0.019	0.898 $\pm$ 0.009	2.847 $\pm$ 0.070	0.405 $\pm$ 0.010	0.801 $\pm$ 0.001	0.094 $\pm$ 0.003	0.975 $\pm$ 0.003	3.311 $\pm$ 0.199	0.199 $\pm$ 0.005
Post-hoc Union	0.758 $\pm$ 0.005	0.723 $\pm$ 0.019	0.944 $\pm$ 0.007	3.570 $\pm$ 0.078	0.000 $\pm$ 0.000	0.801 $\pm$ 0.001	0.094 $\pm$ 0.002	0.978 $\pm$ 0.002	3.884 $\pm$ 0.227	0.000 $\pm$ 0.000
CFU	0.829 $\pm$ 0.007	0.000 $\pm$ 0.000	0.898 $\pm$ 0.011	2.968 $\pm$ 0.078	0.000 $\pm$ 0.000	-	-	-	-	-
CFR	<u>0.827 $\pm$ 0.007</u>	0.000 $\pm$ 0.000	0.897 $\pm$ 0.012	<u>2.967 $\pm$ 0.084</u>	0.000 $\pm$ 0.000	<u>0.799 $\pm$ 0.000</u>	0.000 $\pm$ 0.000	0.974 $\pm$ 0.003	3.379 $\pm$ 0.183	0.000 $\pm$ 0.000
PCF	0.828 $\pm$ 0.007	0.000 $\pm$ 0.000	0.897 $\pm$ 0.011	2.958 $\pm$ 0.072	0.000 $\pm$ 0.000	0.795 $\pm$ 0.001	0.000 $\pm$ 0.000	0.974 $\pm$ 0.003	<u>3.471 $\pm$ 0.208</u>	0.000 $\pm$ 0.000
CF-CP-mean	0.758 $\pm$ 0.005	0.723 $\pm$ 0.019	0.900 $\pm$ 0.013	3.106 $\pm$ 0.100	0.000 $\pm$ 0.000	0.801 $\pm$ 0.001	0.093 $\pm$ 0.003	0.974 $\pm$ 0.003	3.486 $\pm$ 0.191	0.000 $\pm$ 0.000
CF-CP-max	0.758 $\pm$ 0.005	0.723 $\pm$ 0.019	0.900 $\pm$ 0.013	3.106 $\pm$ 0.100	0.000 $\pm$ 0.000	0.801 $\pm$ 0.001	0.094 $\pm$ 0.003	0.975 $\pm$ 0.003	3.700 $\pm$ 0.208	0.000 $\pm$ 0.000
CF-CP-min	0.758 $\pm$ 0.005	0.723 $\pm$ 0.019	0.900 $\pm$ 0.013	3.106 $\pm$ 0.100	0.000 $\pm$ 0.000	0.801 $\pm$ 0.001	0.093 $\pm$ 0.003	0.974 $\pm$ 0.003	3.508 $\pm$ 0.199	0.000 $\pm$ 0.000

Table 4.2: Real data results. Left block: regression on Law School dataset with $\alpha = 0.1$; right block: classification on Bios dataset with $\alpha = 0.025$ (mean $\pm$ std over 10 runs). Highlighting follows the convention described in the evaluation metrics paragraph.

(CSD=0) but at the cost of lower accuracy/MSE and prediction sets that are moderately larger than split CP. Our proposed CF-CP method achieves counterfactual fairness for prediction sets (CSD=0) while maintaining the accuracy/MSE of the base predictor, and it produces prediction sets that are similar in size to counterfactually fair baselines and moderately larger than split CP. Only **max** aggregation leads to a nonzero CSD in synthetic classification, because of non-empty prediction set rule. Also, CF-CP achieves comparable or better average set size compared to other counterfactual fairness methods (CFU, CFR, PCF) without requiring retraining or access to the probability distribution of A . Among the three aggregation functions, the results are quite similar, with **mean** achieving the smallest average set size in most cases.

4.5.2 Noisy Counterfactuals

In this section, we examine the robustness of CF-CP to the quality of the available counterfactuals, since its performance necessarily depends on how accurate these counterfactuals are. Specifically, we study how fairness and performance change as we introduce noise into the counterfactuals. For the datasets where we have access to the exogenous variables U (synthetic datasets and Law School), we add Gaussian noise to the true exogenous variables before generating counterfactuals. For the Bios dataset, we add

Method	Synth. Regression		Synth. Classification		Law School Regression		Bios Classification	
	Avg. Set Size ($\downarrow$)	CSD ($\downarrow$)	Avg. Set Size ($\downarrow$)	CSD ($\downarrow$)	Avg. Set Size ($\downarrow$)	CSD ($\downarrow$)	Avg. Set Size ($\downarrow$)	CSD ($\downarrow$)
SplitCP	2.032 ± 0.038	0.713 ± 0.009	2.025 ± 0.209	0.642 ± 0.043	2.847 ± 0.070	0.405 ± 0.010	3.311 ± 0.199	0.199 ± 0.005
Post-hoc Union	3.259 ± 0.044	0.152 ± 0.001	3.115 ± 0.254	0.260 ± 0.006	3.571 ± 0.078	<u>0.036 ± 0.001</u>	3.991 ± 0.231	0.130 ± 0.002
CFU	3.551 ± 0.088	0.135 ± 0.004	3.462 ± 0.239	0.345 ± 0.008	2.998 ± 0.063	0.055 ± 0.002	-	-
CFR	3.013 ± 0.078	0.115 ± 0.003	2.913 ± 0.209	0.209 ± 0.004	2.984 ± 0.064	0.029 ± 0.001	3.450 ± 0.226	0.099 ± 0.003
PCF	2.983 ± 0.061	0.115 ± 0.003	2.600 ± 0.210	0.263 ± 0.006	2.961 ± 0.071	0.037 ± 0.001	3.560 ± 0.197	<u>0.109 ± 0.002</u>
CF-CP-mean	3.016 ± 0.054	<u>0.118 ± 0.002</u>	2.609 ± 0.203	0.259 ± 0.006	3.108 ± 0.108	0.029 ± 0.001	3.581 ± 0.204	<u>0.109 ± 0.002</u>
CF-CP-max	3.374 ± 0.057	0.145 ± 0.003	4.271 ± 0.185	<u>0.230 ± 0.004</u>	3.114 ± 0.112	0.041 ± 0.001	3.813 ± 0.210	0.123 ± 0.002
CF-CP-min	2.935 ± 0.050	0.168 ± 0.003	2.657 ± 0.205	0.271 ± 0.005	3.107 ± 0.093	0.041 ± 0.001	3.599 ± 0.165	0.128 ± 0.003

Table 4.3: Results with noisy estimations of counterfactuals (mean $\pm$ std over 10 runs). Highlighting follows the convention described in the evaluation metrics paragraph.

Gaussian noise to the counterfactual embeddings. The results are shown in Table 4.3. Here we only report average set size and CSD due to space constraints and the other metrics and noise variances can be found in Appendix C.3. The noisy counterfactuals lead to a slight increase in CSD for all methods, but CF-CP still achieves a significant reduction in CSD compared to split CP, with only a modest increase in the size of the prediction sets. CF-CP achieves comparable average set size and CSD trade-off compared to other counterfactual fairness methods (CFU, CFR, PCF) even with noisy counterfactuals.

To further illustrate the robustness of CF-CP to noisy counterfactuals, we plot CSD against different noise levels in Figure 4.1. We vary the noise level from 0 (oracle counterfactuals) to 1 for synthetic datasets and Law School, and from 0 to 0.16 for Bios, as higher noise levels lead to poor performance. We observe that as the noise level increases, CSD also increases for all methods as expected. However, CF-CP methods show similar robustness to noise as other counterfactual fair models (CFU, CFR, PCF) and post-hoc union, and they consistently achieve lower CSD compared to split CP across all noise levels. This shows that CF-CP is effective in reducing counterfactual unfairness even when counterfactuals are noisy.

We emphasize two key takeaways: (i) CF-CP does not require retraining the model or access to the probability distribution of A , making it a more practical choice in real-world scenarios where counterfactuals may be noisy or imperfect; and (ii) as stated earlier, CF-

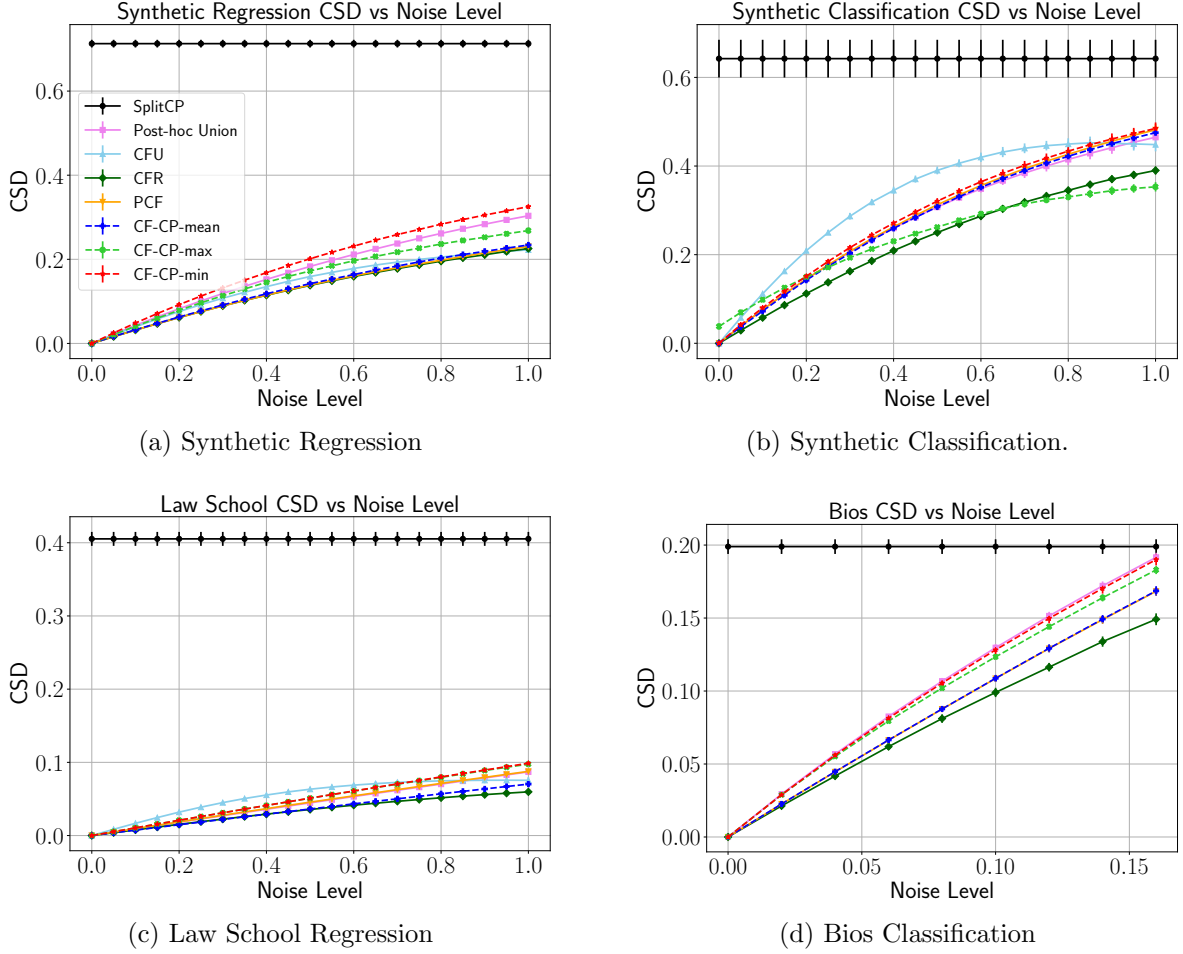


Figure 4.1: Counterfactual Set Disparity (CSD) versus the noise level in counterfactual generation. CF-CP methods exhibit similar robustness to noise as other counterfactually fair models.

CP does not reduce the accuracy of the baseline predictor. Compared to CFU, CFR and PCF, the baseline predictor for CF-CP is more accurate—see Appendix C.3.

4.6 Conclusion

In this chapter, we extended the notion of counterfactual fairness to set-valued predictors and then proposed a novel post-training conformal prediction method that enforces set-level counterfactual fairness by symmetrizing the conformity scores across protected

attribute interventions. Under an invertible SCM, we showed that our method guarantees both marginal coverage and counterfactual invariance for prediction sets. Empirically, across synthetic and real-world datasets, including Law School and Bios, CF-CP substantially reduces counterfactual set disparity compared to standard split CP, with only a modest increase in average set size, and without requiring retraining or access to the probability distribution of the protected attribute. Our results suggest that enforcing counterfactual fairness at the set level is both feasible and effective. Two promising directions for future work are: (i) extending this fairness notion to path-dependent counterfactual fairness [108], where only certain causal pathways from the protected attribute to the outcome are considered unfair; (ii) generalizing CF-CP to dynamic settings where covariates and protected attributes evolve over time.

Chapter 5

Multi-Bin Batching for LLM Inference

5.1 Introduction

Large Language Model (LLM) inference systems are becoming increasingly popular due to their various abilities, such as text generation [109], coding assistance [110], and question answering [111]. As the demand for LLM inference systems grows, so does the need to optimize their efficiency.

Several techniques have been proposed to improve the efficiency of LLM inference systems, and *batched inference* [112, 15, 113] is one of the most promising techniques among them. With batched inference, multiple requests are processed simultaneously, using the underlying hardware’s parallelism to improve throughput. Figure 5.1a shows the measured throughput of the Phi-3.5 Mini Instruct model [114] for various batch sizes on an NVIDIA A100 80G GPU. Throughput is calculated as the number of total tokens generated across all requests divided by time.

However, batched inference comes with some critical drawbacks. The execution time

of each request depends on the number of tokens generated, which varies across requests. In standard batched inference systems, a computing unit remains locked until all requests in the batch are completed, leading to resource underutilization when requests within a batch have widely differing execution times. This inefficiency offsets the throughput gains achieved through parallelism¹.

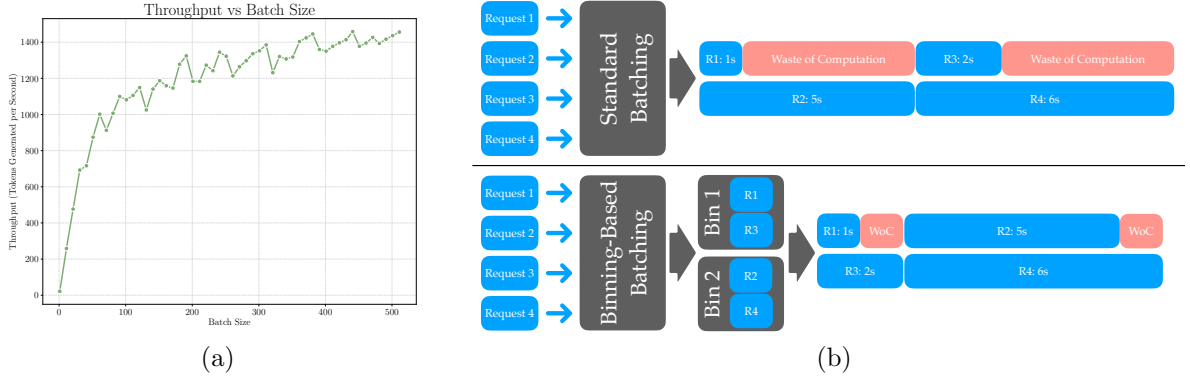


Figure 5.1: (a) Batch serving improves the throughput for LLM inference systems. (b) Standard batching causes underutilization of resources due to varying answer sizes.

Inspired by this, a natural question arises: can we achieve near-optimal throughput from batched inference?

We propose a novel control policy to optimize batched inference by introducing multi-bin batching that can *provably improve LLM inference throughput in the static-batching model* by grouping requests based on their predicted output lengths. Instead of placing all requests into a single queue, we create multiple “bins”, each serving as a waiting area for requests with similar (predicted) output lengths. This binning policy ensures that requests with comparable execution times are grouped together, minimizing inefficiencies due to varying execution times. Batches are then formed within each bin and dispatched to a central queue to be processed.

¹With some low-level system engineering, one could potentially allow for dispatch of waiting requests in the queue to a *partially busy* computing node [14]. We assume such fine-grained request dispatching is not available in this chapter. See Section 5.2 for more details.

To see why this approach is beneficial, consider the example illustrated in Figure 5.1b, where four requests arrive at nearly the same time, with execution times of 1, 5, 2, and 6 seconds, respectively. In a standard batching system with a batch size of $B = 2$, the requests would be grouped based on their arrival time, forming two batches: (Request 1: 1s, Request 2: 5s) and (Request 3: 2s, Request 4: 6s). The total execution time would be 11 seconds (5 seconds for the first batch and 6 seconds for the second). On the other hand, two-bin batching (one bin for service time between 1 to 3 seconds and another for those between 4 to 6 seconds) will result in different batches: (Request 1: 1s, Request 3: 2s) and (Request 2: 5s, Request 4: 6s). This will reduce the total execution time to 8 seconds, demonstrating how multi-bin batching can improve LLM inference throughput.

Our analysis shows that, within the static-batching framework, increasing the number of bins makes the throughput approach the idealized static-batching capacity by reducing inefficiencies caused by heterogeneous execution times within batches.

On the Dolly dataset workload, estimated-length multi-bin batching improves token throughput by 150.1% over static batching at $k = 8$, but remains 43.0% below the continuous-batching baseline. The corresponding oracle-length result improves over static batching by 253.1%, while remaining 19.5% below continuous batching. These results show that length-aware grouping substantially mitigates static-batching inefficiency, but also that native continuous batching remains a strong practical baseline on this workload.

Although our presentation is motivated by LLM inference, the underlying queueing formulation also applies more broadly to autoregressive batch-serving systems in which heterogeneous completion times create straggler effects within fixed batches.

The main contributions of this chapter are summarized below. This chapter is based in part on our work [19].

- We propose a novel binning-based scheduling strategy that dynamically groups

requests into bins based on output lengths, minimizing inefficiencies from varying execution times and optimizing throughput for LLM inference systems.

- We use queueing-theoretical analysis to show that our multi-bin batching strategy achieves asymptotic throughput optimality under the static batching model as the number of bins grows. Additionally, we characterize the number of bins required to achieve a desired throughput improvement.
- Our experiments on real LLM inference workloads show that estimated-length multi-bin batching substantially improves generated-token throughput over static batching, reaching 150.1% improvement on the Dolly dataset workload, while remaining below the continuous-batching baseline.

5.2 Related Work

LLM Inference and Scheduling Recent research has focused on optimizing large language model (LLM) inference through scheduling techniques and queueing-theoretic tools. FastServe [115] utilizes a preemptive scheduling algorithm, skip-join Multi-Level Feedback Queue, to improve job completion time in LLM inference systems. Prior work on dynamic batching [116] studies a setting where the system serves up to B jobs depending on queue occupancy and derives closed-form upper bounds for the mean latency. Slice-level scheduling [117] splits the maximum output length into slices and serves batches slice by slice, improving memory utilization and response time. Llumnix [118] addresses heterogeneous and unpredictable LLM inference requests through runtime rescheduling across multiple model instances, improving tail latency and resource utilization. Recent systems further exploit the two-phase structure of LLM inference: Sarathi-Serve [119] uses chunked prefills and stall-free scheduling to improve the throughput–latency trade-

off, while Splitwise [120] separates prompt computation and token generation across different resources to exploit the distinct computational characteristics of prefill and decode. A queueing-theoretic analysis of low-latency LLM inference [121] studies an M/G/1 model and shows that maximum-output-token limits and batch-size optimization can significantly reduce latency under token-length variability. Queue management for SLO-oriented LLM serving [122] introduces grouping strategies based on request length to improve throughput. Our work is complementary to these studies: rather than modeling general queueing delay in an M/G/1 system or optimizing prefill/decode scheduling, we study a length-aware static-batching control policy, characterize bin boundaries in tractable service-time models, and quantify how finite-bin discretization and prediction errors affect the resulting effective batched service rate.

LLM Serving and Answer Length Estimation Several studies have improved LLM inference throughput and latency using response-length or execution-time estimation. Response-time prediction via prompting [123] asks the model an additional question to estimate response time. Learning-to-rank based scheduling [124] predicts the relative ordering of requests by execution time and uses this ranking in a shortest-job-first scheduling algorithm to reduce head-of-line blocking. Proxy-model-based prediction [125] uses a lightweight proxy model to predict execution time and then applies speculative shortest-job-first scheduling to improve serving throughput. S³ [113] estimates answer length and uses this estimate to improve memory efficiency, thereby increasing the effective batch size of the system. Generation-length prediction for LMaaS [126] uses input length to predict response length and optimize batch size, improving throughput and reducing response time. Similarly, SyncIntellects [127] enhances response-length prediction using a transformer-based model and implements QoS-friendly length control, improving both throughput and latency. These works show that length information can be a useful

scheduling signal in LLM serving. Our method uses this signal differently: instead of selecting a job order or batch size, it partitions requests into length-aware bins before batch formation.

LLM Inference Optimization Recent studies have focused on optimizing the inference efficiency of LLM models through various techniques. Quantization has emerged as a key approach to reduce the memory footprint and improve the inference efficiency of LLM models. Methods like LLM.Int8() [128], GPTQ [129], SmoothQuant [130], and AWQ [131] have demonstrated effective weight quantization techniques for LLM models, while QLoRA [132] combines quantization with parameter-efficient fine-tuning. Memory management innovations such as PagedAttention [15] have significantly improved the serving throughput. KV cache optimizations, including compression techniques like Gear [133] have further improved the memory efficiency of LLM inference systems. Systems like FastServe [115], and FlexGen [112] have integrated these techniques to create comprehensive LLM serving solutions.

Continuous Batching Continuous batching, also known as iteration-level scheduling, allows completed requests to leave a batch and new requests to enter during decoding, rather than requiring a fixed batch to remain locked until all requests finish [14]. Modern serving systems such as Orca [14], vLLM/PagedAttention [15], and SGLang [134] use this style of scheduling or related mechanisms to improve utilization and reduce the head-of-line blocking present in static batching. At the same time, continuous-batching systems must manage heterogeneous prefill and decode phases, KV-cache constraints, and scheduling hyperparameters that affect the balance between throughput and latency. This distinction is especially important because prefill and decode have different computational and memory characteristics, and recent systems such as Sarathi-Serve [119] and

Splitwise [120] explicitly exploit this two-phase structure. Our theoretical analysis does not attempt to characterize iteration-level continuous batching. Instead, it provides a tractable static-batching framework that isolates the effect of output-length heterogeneity. In the experiments, we therefore include the backend’s native continuous-batching behavior as a strong practical reference baseline. The resulting interpretation is that multi-bin batching is a length-aware grouping principle that may complement modern serving systems, for example through queue classes, or admission-control classes rather than a replacement for continuous batching.

5.3 Problem Setup and the Multi-Bin Batching Algorithm

We begin by introducing the system model and key assumptions that form the foundation of our analysis. This model represents a typical LLM inference system as a queueing system with specific characteristics. Following this, we will propose our novel batching algorithm, which leverages a multi-binning approach to optimize request processing. The model studied in this section is a stylized static-batching framework: it is designed to isolate the effect of service-time, and in particular decode/output-length, heterogeneity on batch efficiency rather than to capture all costs in a production LLM serving system.

Assumption 5.1 *The LLM inference system is a single-server queueing system with an infinite queue length capacity. The system receives requests from a Poisson process with rate λ .*

Assumption 5.1 is a standard assumption in queueing theory, and it is well-suited for LLM inference systems since requests are typically generated by users in a random manner. Although this analysis assumes a single-server system, extension to multi-server systems

is rather straightforward.

The system forms batches of size B , and the serving time of a batch is determined by the maximum of the serving times of its requests. For the purposes of this chapter, we assume a “first formed batch, first served” policy, where batches are processed in the order they are fully formed, regardless of when its constituent requests arrived. Alternative policies, such as “shortest batch, first served”, could also be considered depending on specific system requirements.

Assumption 5.2 *The service time of each request is independent and identically distributed (i.i.d.) with a uniform distribution in the range $[l_{\min}, l_{\max}]$, i.e., $l \sim \mathcal{U}(l_{\min}, l_{\max})$.*

We make this assumption to simplify the analysis, and it is justified since LLM answer lengths typically fall within a specific range due to maximum token length limitations. We also extend our analysis to the case where the service time is exponentially distributed, which can be found in the Section 5.6. The uniform and exponential assumptions should be interpreted as analytically tractable service-time models, not as claims that real LLM inference times exactly follow these distributions. In autoregressive generation, decode time is often closely related to the number of generated tokens, so the service-time variable can be viewed as an abstraction of output-length-driven decode cost. The analysis is therefore most directly applicable to decode-dominant workloads, or to systems in which prefill and decode are separated. It does not explicitly model prefill time, input-length heterogeneity, or the interaction between prefill and decode costs; these factors are included only in the end-to-end empirical measurements in Section 5.7.

In our theoretical analysis, we assume that the system always forms batches of size B and then starts processing them. However, in real systems, there could be a parameter that specifies the maximum time a batch can wait before it is processed. This way the system can ensure that the latency of a request does not exceed a certain threshold and

the quality of service is maintained.

5.3.1 Multi-Bin Batching Algorithm

We propose a novel scheduling policy that aims to improve the throughput of LLM inference systems. The key idea is to group requests into k bins based on their service times before forming batches within each bin. This binning procedure reduces inefficiencies by ensuring that requests with similar execution times are batched together.

Algorithm 2: Multi-Bin Batching with k -bins

Input: Bin boundaries, $[l_{i-1}, l_i]$, $i = 1, \dots, k$ -bins, batch size B , and serving policy

- 1 **for** *each incoming request* **do**
- 2 Estimate its service time l ;
- 3 Assign the request to bin i where $l_{i-1} \leq l < l_i$;
- 4 **for** *each bin* **do**
- 5 Form batches of size B when available;
- 6 Add completed batches to the service queue;
- 7 Serve batches from the service queue based on the serving policy provided;
- 8 The serving time of a batch is the maximum of the serving times of the requests in the batch;

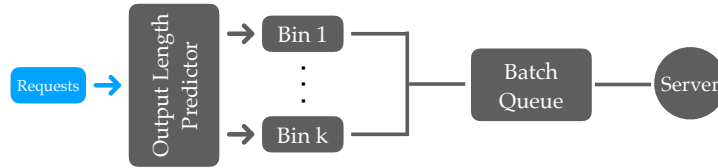


Figure 5.2: k-Bin Batching

The multi-bin batching algorithm is illustrated in Figure 5.2. This algorithm works by first dividing the range of possible service times into k bins with boundaries determined by prior knowledge or system profiling. Incoming requests are assigned to bins based on their estimated service times and grouped into batches of size B . Completed batches are added to a service queue and processed using the “first completed batch, first served” policy, the

default for this chapter. By batching requests with similar service times together, this method reduces the serving time of each batch, and ensures efficient processing across all bins.

5.4 Throughput Analysis

In this section, we present a queueing-theoretical analysis to show that our multi-bin batching policy improves the throughput of LLM inference systems within the static-batching framework introduced above. To analyze the throughput of the system, we first derive the optimal decision boundaries for each bin in our batching system. Using these boundaries, we calculate the expected service time of a batch of requests processed under the multi-bin batching policy. Throughout this section, “throughput” refers to the saturated request processing rate of the static-batching, i.e., the number of requests completed per unit service time. This is a service-rate quantity for the batched server and should be distinguished from the maximum stable arrival rate of a full queueing system with finite waiting effects.

Definition 5.3 (Throughput under static batching) *For a saturated static batching server, the throughput is the ratio of the batch size to the expected service time. Specifically, it can be written as*

$$\text{Throughput} = \frac{B}{\mathbb{E}[t_{\text{service}}]}, \quad (5.1)$$

where B is the batch size and $\mathbb{E}[t_{\text{service}}]$ is the expected service time of a batch of B requests.

Remark 5.4 *Within static batching framework, the idealized upper bound on throughput is achieved when the service time of a batch, t_{service} , equals the average service time of a single request rather than the maximum service time within the batch. In this case, the*

expected throughput becomes, $\text{Throughput}_{\max} = \frac{B}{\mathbb{E}[t]}$, where $\mathbb{E}[t]$ is the mean service time of a single request. This scenario represents the idealized case of perfect parallelism without any delays due to batching overhead. This bound is specific to the fixed-batch model analyzed here and is not intended as a universal upper bound for continuous-batching systems, which can pipeline request admission and completion at a finer granularity.

For our multi-bin batching system, we can derive the expected service time of a batch of B requests as follows:

$$\mathbb{E}[t_{\text{service}, k}] = \sum_{i=1}^k \Pr(\text{bin} = i) \mathbb{E} \left[\max_{j \in [B]} x_j | \text{bin} = i \right], \quad (5.2)$$

where $\Pr(\text{bin} = i)$ is the probability that a batch served by the system is in bin i , and the expected service time of a batch of B requests from bin i is denoted by $\mathbb{E} [\max_{j \in [B]} x_j | \text{bin} = i]$. We also denote it as $\mathbb{E}[t_{\text{service}, k}]$ to emphasize that it is the expected service time of multi-bin batching system with k bins. Then, the first step is to determine the optimal decision boundaries for each bin in the multi-bin batching system for a fixed number of bins k , as provided by the following lemma.

Lemma 5.5 *Under Assumption 5.2 and a fixed number of bins k , the throughput is maximized when each bin has equal probability mass, and the decision boundaries are determined as follows:*

$$l_i = l_{\min} + \frac{i}{k}(l_{\max} - l_{\min}), \quad i \in [k]. \quad (5.3)$$

and where $l_0 = l_{\min}$.

Proof. We begin by defining the expected service time of a request in bin i as follows,

$$\mathbb{E} \left[\max_{j \in [B]} x_j | \text{bin} = i \right] = \frac{1}{B+1} l_{i-1} + \frac{B}{B+1} l_i. \quad (5.4)$$

Since, the service time of a request in bin i is uniformly distributed in the range $[l_{i-1}, l_i]$, the expected value of the maximum of B uniform random variables in the range $[l_{i-1}, l_i]$ is well-known and can be computed easily. Then, the expected service time of the system is given by,

$$\mathbb{E}[t_{\text{service}, k}] = \sum_{i=1}^k \Pr(\text{bin} = i) \mathbb{E} \left[\max_{j \in [B]} x_j | \text{bin} = i \right] \quad (5.5)$$

It can be written as,

$$\mathbb{E}[t_{\text{service}, k}] = \sum_{i=1}^k \frac{l_i - l_{i-1}}{l_{\max} - l_{\min}} \left(\frac{1}{B+1} l_{i-1} + \frac{B}{B+1} l_i \right). \quad (5.6)$$

For k bins, we have $k-1$ decision boundaries, and $l_0 = l_{\min}$ and $l_k = l_{\max}$. We can denote the expected service time of the system as a function of $\mathbf{l} = [l_1, l_2, \dots, l_{k-1}]$ as follows,

$$f_k(\mathbf{l}) = \sum_{i=1}^k \frac{l_i - l_{i-1}}{l_{\max} - l_{\min}} \left(\frac{1}{B+1} l_{i-1} + \frac{B}{B+1} l_i \right). \quad (5.7)$$

We can compute the partial derivative of $f_k(\mathbf{l})$ with respect to $l_i, i \in [k-1]$ as follows,

$$\frac{\partial f_k(\mathbf{l})}{\partial l_i} = -\frac{B-1}{B+1} \frac{l_{i+1}}{l_{\max} - l_{\min}} + \frac{2(B-1)}{B+1} \frac{l_i}{l_{\max} - l_{\min}} - \frac{B-1}{B+1} \frac{l_{i-1}}{l_{\max} - l_{\min}}. \quad (5.8)$$

Then, the second-order partial derivative of $f_k(\mathbf{l})$ with respect to l_i and $l_j, i, j \in [k-1]$ is given by,

$$\frac{\partial^2 f_k(\mathbf{l})}{\partial l_i \partial l_j} = \begin{cases} \frac{2(B-1)}{B+1} \frac{1}{l_{\max} - l_{\min}} & \text{if } i = j, \\ -\frac{B-1}{B+1} \frac{1}{l_{\max} - l_{\min}} & \text{if } |i - j| = 1, \\ 0 & \text{otherwise.} \end{cases} \quad (5.9)$$

The Hessian matrix of $f_k(\mathbf{l})$ is a tridiagonal matrix in the form of,

$$\nabla^2 f_k(\mathbf{l}) = c \begin{bmatrix} 2 & -1 & 0 & \cdots & 0 & 0 \\ -1 & 2 & -1 & \cdots & 0 & 0 \\ 0 & -1 & 2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 2 & -1 \\ 0 & 0 & 0 & \cdots & -1 & 2 \end{bmatrix}, \quad (5.10)$$

where

$$c = \frac{B-1}{B+1} \frac{1}{l_{\max} - l_{\min}}. \quad (5.11)$$

The determinant of the Hessian matrix can be computed via the recursive formula for the determinant of a tridiagonal matrix as follows,

$$\det(\nabla^2 f_k(\mathbf{l})) = k \left(\frac{(B-1)}{(B+1)(l_{\max} - l_{\min})} \right)^{k-1} > 0 \quad (5.12)$$

Since $k > 1$ and $B > 1$, and since the tridiagonal matrix with diagonal entries 2 and off-diagonal entries -1 is positive definite, the Hessian matrix is positive definite. Therefore, the function $f_k(\mathbf{l})$ is convex with respect to $l_1, l_2, \dots, l_{k-1}$. Then, one can solve Equation (5.8) for l_i by setting the partial derivative to zero, i.e., $\frac{\partial f_k(\mathbf{l})}{\partial l_i} = 0$. It can be seen that the optimal decision boundaries are given by,

$$l_i = l_{\min} + \frac{i}{k}(l_{\max} - l_{\min}) \quad \forall i \in [k-1]. \quad (5.13)$$

This completes the proof. ■

Given the optimal decision boundaries in Lemma 5.5, for a fixed number of bins k , we can have the following theorem for the expected throughput of the system.

Theorem 5.6 *Under Assumption 5.2, the expected throughput of the k -bin batching is*

$$\text{Throughput}_k = \frac{B}{\mathbb{E}[t_{\text{service}, k}]} = \frac{B}{\frac{l_{\max} + l_{\min}}{2} + \frac{1}{k}l_B}, \quad (5.14)$$

where

$$l_B = \frac{B}{B+1}l_{\max} + \frac{1}{B+1}l_{\min} - \frac{l_{\max} + l_{\min}}{2}$$

and it is an increasing function of the number of bins k .

Proof. The expected service time of a batch of B requests is,

$$\mathbb{E}[t_{\text{service}, k}] = \sum_{i=1}^k \Pr(\text{bin} = i) \mathbb{E} \left[\max_{j \in [B]} x_j | \text{bin} = i \right] \quad (5.15)$$

$$= \sum_{i=1}^k \frac{1}{k} \left(\frac{B}{B+1}l_i + \frac{1}{B+1}l_{i-1} \right), \quad (5.16)$$

because each bin has equal probability mass, and the service time of a batch of requests follows from the uniform distribution in the range $[l_{i-1}, l_i]$. If we substitute the optimal decision boundaries in Equation 5.3 into Equation 5.15, we can derive the expected service time of a batch of B requests with multi-bin batching as follows,

$$\begin{aligned} \mathbb{E}[t_{\text{service}, k}] &= \frac{1}{k} \sum_{i=1}^k \frac{B}{B+1} \left(l_{\min} + \frac{i}{k}(l_{\max} - l_{\min}) \right) + \frac{1}{B+1} \left(l_{\min} + \frac{i-1}{k}(l_{\max} - l_{\min}) \right) \\ &= \frac{1}{k} \frac{B}{B+1} \frac{k+1}{2} (l_{\max} - l_{\min}) + \frac{1}{k} \frac{1}{B+1} \frac{k-1}{2} (l_{\max} - l_{\min}) + l_{\min} \\ &= \frac{l_{\max} + l_{\min}}{2} + \frac{1}{k} \left(\frac{B}{B+1}l_{\max} + \frac{1}{B+1}l_{\min} - \frac{l_{\max} + l_{\min}}{2} \right) \end{aligned}$$

Then, the expected throughput of the system with multi-bin batching with k bins is,

$$\text{Throughput}_k = \frac{B}{\mathbb{E}[t_{\text{service}, k}]} = \frac{B}{\frac{l_{\max} + l_{\min}}{2} + \frac{1}{k}l_B}, \quad (5.17)$$

and it can be observed that it is an increasing function of the number of bins k since the denominator is decreasing with respect to k . ■

Remark 5.7 *The standard batching system is a special case of the multi-bin batching system with $k = 1$. If we substitute $k = 1$ into Equation 5.14, we can derive the expected throughput of the system with standard batching. Since the expected throughput of the system with the multi-bin batching is an increasing function of the number of bins k , the throughput of the system with our multi-bin batching system is higher than the standard batching system. Hence, the multi-bin batching system can improve the throughput of the system.*

Remark 5.8 *The expected throughput of the system with multi-bin batching increases with the number of bins k and converges to the maximum capacity of the system as $k \rightarrow \infty$:*

$$c_{\max} = \lim_{k \rightarrow \infty} \text{Throughput}_k = \frac{B}{\frac{l_{\max} + l_{\min}}{2}}. \quad (5.18)$$

With infinitely many bins, the binning becomes so fine-grained that the batch service time equals that of a single request, achieving the optimal throughput of static batching framework asymptotically.

In the following theorem, we derive the minimum number of bins k required to achieve a target throughput below the maximum capacity.

Theorem 5.9 *Under Assumptions 5.1, and 5.2, for any $\epsilon > 0$, the desired throughput of the system $c_{\max} - \epsilon$ can be achieved by the multi-bin batching system with k bins, where k is the smallest integer satisfying the condition*

$$k \geq \left\lceil \frac{(c_{\max} - \epsilon)l_B}{\epsilon \frac{l_{\max} + l_{\min}}{2}} \right\rceil = O\left(\frac{1}{\epsilon}\right). \quad (5.19)$$

Proof. The desired throughput of the system is $c_{\max} - \epsilon$. From Theorem 5.6, the expected throughput of the system with multi-bin batching with k bins is,

$$\text{Throughput}_k = \frac{B}{\frac{l_{\max} + l_{\min}}{2} + \frac{1}{k}l_B}. \quad (5.20)$$

where $l_B = \frac{B}{B+1}l_{\max} + \frac{1}{B+1}l_{\min} - \frac{l_{\max} + l_{\min}}{2}$. Then, we can find the smallest integer k that satisfies the following condition,

$$c_{\max} - \epsilon \leq \frac{B}{\frac{l_{\max} + l_{\min}}{2} + \frac{1}{k}l_B}. \quad (5.21)$$

We can solve the above inequality for k to find the smallest integer k that satisfies the desired throughput of the system,

$$(c_{\max} - \epsilon) \left[\left(\frac{l_{\max} + l_{\min}}{2} \right) + \frac{1}{k}l_B \right] \leq B. \quad (5.22)$$

It can be simplified as follows,

$$B - \epsilon \left(\frac{l_{\max} + l_{\min}}{2} \right) + (c_{\max} - \epsilon) \frac{1}{k}l_B \leq B. \quad (5.23)$$

This implies,

$$k \geq \frac{(c_{\max} - \epsilon)l_B}{\epsilon \frac{l_{\max} + l_{\min}}{2}}. \quad (5.24)$$

Therefore, the smallest integer k that satisfies the desired throughput of the system is given as in the statement of the theorem. ■

Figure 5.3 shows the average throughput of the system with varying numbers of bins k as a function of the arrival rate λ . In this figure, we assumed that the batch size $B = 128$, the minimum service time $l_{\min} = 1$, and the maximum service time $l_{\max} = 20$,

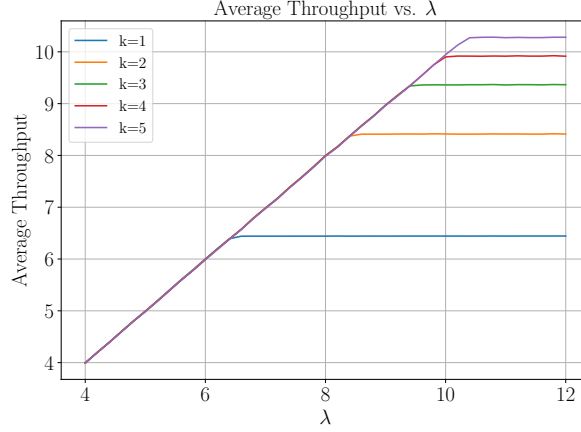


Figure 5.3: Average throughput of the system with varying numbers of bins k as a function of the arrival rate λ .

resulting a maximum theoretical capacity of $c_{\max} = \frac{128}{\frac{20+1}{2}} \approx 12.3$. We submit 128000 requests to the system and measure the time taken to process all the requests. We run the simulations for 10 times and report the average throughput of the system. Then, we calculate the average throughput of the system as the number of requests processed per unit time. The results show that throughput increases with the number of bins k , approaching the maximum capacity when $k = 5$. However, increasing the number of bins introduces a trade-off: as k grows, the time required to construct batches also increases, potentially leading to higher latency. In the next section, we evaluate the latency of the system with multi-bin batching and compare it with the standard batching system.

5.5 Latency Analysis

The latency of a request is defined as the total time from when it enters the system until it is completely served. The latency of a request consists of two components: the queuing time and the service time. In the previous section, we analyzed the expected service time of a request for our multi-bin batching scheduling policy. In this section, we focus on the queuing time, which can be further decomposed into two components: the

time spent waiting to form the current batch and the time spent waiting for the current batch to start processing. In this analysis, we focus on the time spent waiting to form the current batch, which provides a lower bound for the latency of a request. Thus, the analysis in this section should be interpreted as an idealized batch-formation latency calculation, rather than a complete end-to-end latency model for a loaded single-server system.

To derive this theoretical lower bound for latency, we assume ideal conditions where batches are processed immediately upon formation.

Assumption 5.10 *The number of servers in the system is infinite. Therefore, any batch that is ready for processing begins immediately without delay.*

This assumption allows us to approximate latency in an idealized setting, where the waiting time for batches to start processing is negligible. Equivalently, Assumption 5.10 intentionally removes server-side waiting after a batch has formed, so the resulting expression isolates only the batch-formation component and the service-time component. While this lower bound holds across all load conditions, it closely approximates the latency in underloaded systems. We denote the expected latency of a request as $\mathbb{E}[t_{\text{latency}}]$. The following lemma provides the expected latency of a request in our system with the assumption of infinite servers.

Lemma 5.11 *Under Assumptions 5.2, and 5.10, and given the arrival rate λ and k -bins with equal probability mass, the expected latency of a request is given by*

$$\mathbb{E}[t_{\text{latency}}] = \frac{l_{\max} + l_{\min}}{2} + \frac{B-1}{2\lambda}k + \frac{1}{k}l_B.$$

The proof of Lemma 5.11 is given in Appendix D.1.1. The proof utilizes the fact that the arrival process is Poisson and the effective arrival rate of each bin is λ/k . The inverse

dependence on λ in Lemma 5.11 reflects the fact that higher arrival rates reduce the time needed to collect enough requests within each bin. It should not be interpreted as a claim that end-to-end latency necessarily decreases with traffic in a practical single-server system, where server-side queueing delay can dominate.

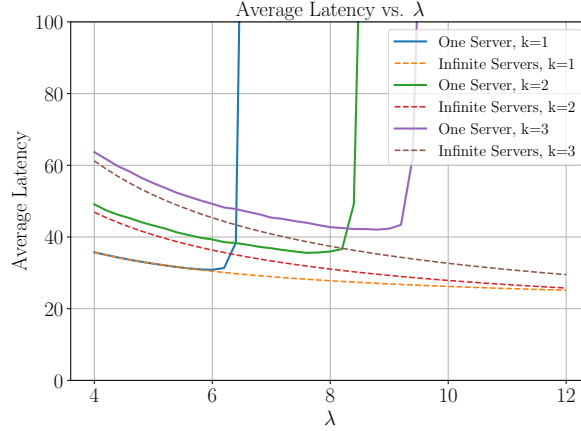


Figure 5.4: The latency of a request vs the arrival rate λ for different k values.

In Figure 5.4, we plot the average latency of a request as a function of the arrival rate λ for different k values. We use the parameters $l_{\min} = 1$, $l_{\max} = 20$, $B = 128$, and $k = 1, 2, 3$. We submit 128000 requests to the system and measure the latency of each request. We run the simulations for 10 times and report the average. It can be seen that our Lemma 5.11 provides a close approximation of the latency of a request in our system when the arrival rate is low. In the idealized batch-formation component captured by Lemma 5.11, increasing the arrival rate reduces the time needed to form a batch. In a one-server system, however, this effect can be offset by server-side queueing as the system approaches its service-rate limit. Overall, these results illustrate that multi-bin batching can improve throughput relative to standard static batching, but the number of bins should be selected with the associated batch-formation latency in mind. More generally, the number of bins k controls a tradeoff between latency and throughput: increasing k reduces service-time waste within a batch, but also decreases the arrival rate

into each bin and can increase the time required to form a full batch. Hence, k should be viewed as a workload and latency dependent tuning parameter. Larger values of k are most attractive in high-load or throughput-oriented regimes, while smaller values may be preferable in low-load or latency-sensitive regimes. A full end-to-end latency-throughput Pareto analysis is an important direction for future work.

5.6 Analysis under Exponential Distribution

In this section, we provide the expected service time of a batch of B requests when the service time of each request is exponentially distributed with rate μ . As in the uniform case, the exponential model is used as an analytically tractable service-time distribution for studying the effect of length heterogeneity under static batching; it is not intended as an empirical claim about the exact distribution of LLM inference times. Hence, here we make the following assumption:

Assumption 5.12 *The service time of each request is independent and identically distributed (i.i.d.) with an exponential distribution with rate μ , i.e., $l \sim \text{Exp}(\mu)$.*

Then, for our multi-bin batching system, we need to decide the optimal decision boundaries to minimize the expected service time of a batch of B requests. One can utilize the order statistics of the truncated exponential distribution [135] to derive the expected service time of a batch of B requests with k bins. However, the exact values of truncated exponential order statistics are not easy to compute. Therefore, we use a simpler approach to derive an upper bound on the expected service time of a batch of B requests with k bins. For the bins before the last bin, we can upper bound the expected service time of a batch of B requests as the decision boundary of that bin, i.e., for bin i , the expected service time of a batch of B requests is upper bounded by l_i . For the last

bin, the exact expected service time of a batch of B requests is known and it is $l_{k-1} + \frac{H_B}{\mu}$, where H_B is the B -th harmonic number. Then, we have the following lemma.

Lemma 5.13 *Under Assumption 5.12, the expected service time of a batch of B requests with k bins is upper bounded by,*

$$\mathbb{E}[t_{\text{service}}, k] \leq \sum_{i=1}^{k-1} \Pr(\text{bin} = i) l_i + \Pr(\text{bin} = k) \left(l_{k-1} + \frac{H_B}{\mu} \right). \quad (5.25)$$

and this upper bound is minimized when the decision boundaries are set as,

$$l_i = \frac{1}{\mu} \sum_{j=1}^i \log(L_{k-j}) \quad \forall i \in [k-1] \quad (5.26)$$

where L_m is defined recursively as:

$$L_m = \begin{cases} H_B & \text{if } m = 1 \\ 1 + \log(L_{m-1}) & \text{if } m > 1 \end{cases} \quad (5.27)$$

The proof of Lemma 5.13 is given in Appendix D.1.2. The proof bounds the expected service time by breaking it into contributions from each bin and reformulating the problem using a logarithmic transformation. It then shows the reformulated upper bound is convex via its Hessian so that setting its gradient to zero yields the optimal decision boundaries.

Given the optimal decision boundaries, the service-time upper bound in Lemma 5.13 yields the following lower bound on throughput.

Corollary 5.14 *Under Assumption 5.12 and the optimal decision boundaries given in*

Lemma 5.13, the expected service time satisfies

$$\mathbb{E}[t_{\text{service}, k}] \leq \frac{1}{\mu} f(\mathbf{q}), \quad (5.28)$$

and therefore the throughput satisfies

$$\text{Throughput}_k = \frac{B}{\mathbb{E}[t_{\text{service}, k}]} \geq \frac{B\mu}{f(\mathbf{q})}, \quad (5.29)$$

where

$$f(\mathbf{q}) = \sum_{i=1}^{k-1} \frac{L_{k-i} - 1}{\prod_{j=1}^i L_{k-j}} \cdot \sum_{j=1}^i \log(L_{k-j}) + \frac{1}{\prod_{j=1}^{k-1} L_{k-j}} \left(\sum_{j=1}^{k-1} \log(L_{k-j}) + H_B \right), \quad (5.30)$$

and L_m is defined recursively as in Lemma 5.13.

The proof of corollary follows from the optimal decision boundaries in Lemma 5.13 and the expected service time of a batch of B requests with k bins.

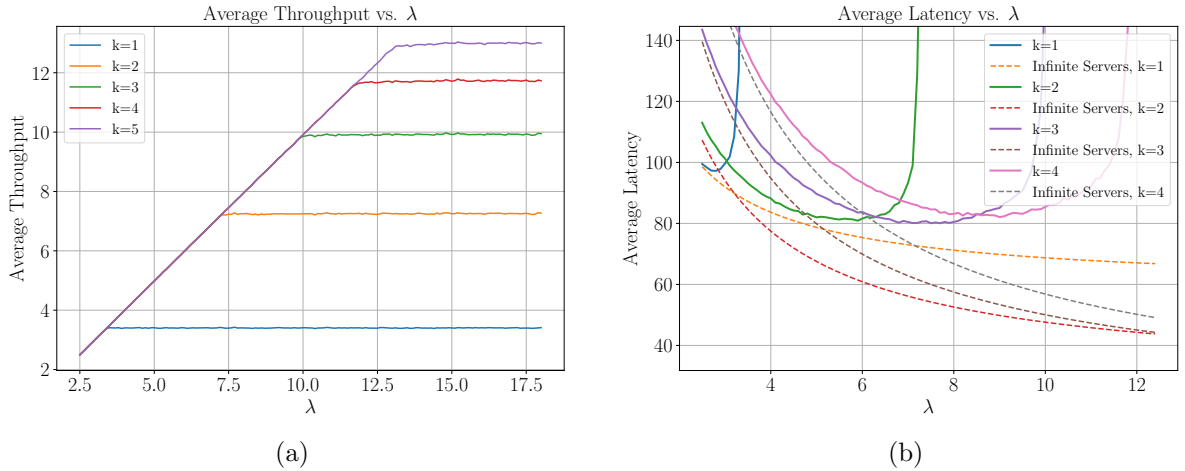


Figure 5.5: Performance of the system with respect to the arrival rate λ for different values of k . (a) Throughput. (b) Expected latency of a request.

Similarly to the uniform distribution case, we provide numerical results for the expo-

nentially distributed service time case. In Figure 5.5a and 5.5b, we provide the throughput and expected latency of the system with respect to the arrival rate λ for different values of k . We assume that the service time of each request is exponentially distributed with rate $\mu = 0.1$, the batch size is $B = 200$, and the total number of requests is $N = 200000$. We run the simulations for 10 different seeds and provide the average throughput and expected latency of the system. It can be observed that the throughput of the system increases with the number of bins k with the multi-bin batching policy in Figure 5.5a. The average latency of the system depicted in Figure 5.5b decreases with the number of bins k . It can be seen that with the increasing number of bins, the system first achieves a lower latency but after a certain point, the latency starts to increase. This is different from the results in the uniform distribution case, where the latency increases with the number of bins.

5.7 Experiments

We evaluate multi-bin batching in four complementary settings. First, we run a controlled real-inference workload using `llama.cpp`, where output lengths are selected from a balanced uniform-bin distribution and bin assignments use oracle output lengths. Second, we evaluate a more realistic Dolly dataset [136] workload, again using oracle output lengths for bin assignment. Third, we repeat the Dolly workload with bin assignments obtained from an estimated-length predictor. Finally, we retain the simulation-based symmetric error study from the original evaluation to isolate the effect of controlled bin-assignment noise.

Throughout this section, throughput denotes generated tokens per second unless otherwise stated. We use *oracle length* to mean that bin assignment is based on the true generated output length. We use *estimated length* to mean that bin assignment is based

on a length predictor; the inference run itself still generates the true target number of tokens. This distinction allows us to separate the scheduling benefit available with perfect length information from the benefit attainable with a practical length estimator.

5.7.1 Experimental Setup

The real-inference experiments evaluate scheduling policies using `llama.cpp` as the serving backend. All real-inference experiments use `Llama-3.2-3B-Instruct` [137] served through `llama.cpp`. For each workload, requests are issued to a local `llama.cpp` server according to the scheduling policy under evaluation, and throughput is measured as the total number of generated tokens divided by the wall-clock completion time of the workload. We compare static batching, native continuous batching, and multi-bin static batching.

We compare three scheduling policies. *Static batching* forms fixed batches in request order and does not refill a batch until all requests in that batch complete. *Continuous batching* uses the backend’s native continuous-batching scheduler, which can admit new requests as running requests finish. This is a substantially stronger and more practically relevant baseline than fixed static batching. *Multi-bin static batching* assigns requests to length bins, forms fixed batches within each bin, and dispatches ready batches in earliest-request-id order. Thus, static batching isolates the straggler effect targeted by our method, while the native continuous-batching baseline provides a strong practical reference point for assessing whether length-aware grouping remains useful beyond fixed-batch comparisons.

The controlled oracle-length experiment uses 1024 requests with batch size 32. Target generation lengths are sampled from a balanced uniform-bin distribution over 500–2000 tokens. We evaluate $k \in \{2, 4, 8, 16\}$ bins, with a total assigned output length of 1,282,127

tokens. The Dolly experiments use 1478 test requests from a Dolly-derived labeled workload with batch size 64. The true target lengths range from 3 to 978 tokens, for a total of 408,352 generated tokens. All throughput numbers below are reported in generated tokens per second.

All real-inference experiments were run on a single NVIDIA A100 80GB GPU. The model was served in FP16 precision with a context size of 4096. Before each measured run, we performed a warm-up run using the same serving configuration. For the controlled real-inference workload, we use batch size $B = 32$. For the Dolly workload, we use batch size $B = 64$.

5.7.2 Controlled Oracle-Length Inference

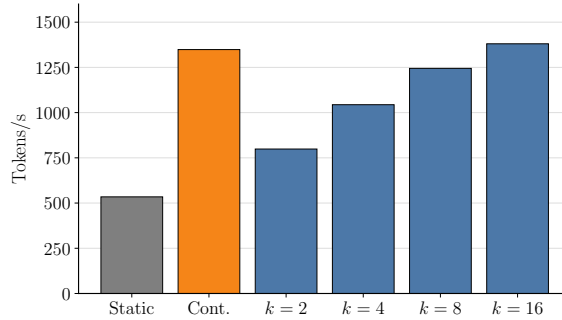


Figure 5.6: Controlled oracle-length `llama.cpp` throughput.

Figure 5.6 reports the controlled oracle-length experiment. Static batching achieves 534.23 tokens/s. Multi-bin batching improves monotonically with the number of bins, reaching 1380.48 tokens/s at $k = 16$. This corresponds to a 158.4% improvement over static batching, showing that length-aware grouping can substantially reduce the inefficiency caused by heterogeneous output lengths in fixed batches.

Continuous batching is much stronger than static batching, achieving 1348.49 tokens/s. In this controlled workload, $k = 16$ multi-bin batching is 2.4% higher than

continuous batching, while smaller values of k remain below the continuous baseline. We view this result as evidence that sufficiently fine oracle-length grouping can approach, and in this case slightly exceed, the throughput of the native continuous-batching baseline under controlled length heterogeneity.

5.7.3 Dolly Dataset Inference with Oracle Lengths

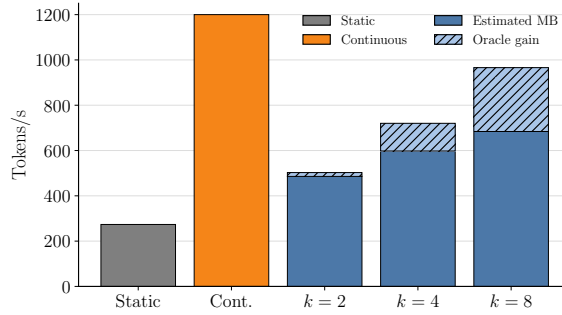


Figure 5.7: Dolly dataset [136] llama.cpp throughput with estimated and oracle output lengths. For multi-bin bars, the solid portion shows estimated-length batching, and the hatched segment shows the additional throughput obtained with oracle lengths.

Figure 5.7 reports the Dolly real-inference experiment with oracle bin assignments. Because bins are assigned using the true output lengths, this experiment isolates the scheduling benefit from any prediction error. Accordingly, these oracle results should be interpreted as an upper bound on the scheduling gains available from accurate length information, rather than as directly achievable without a sufficiently accurate estimator. Static batching achieves 273.53 tokens/s, whereas continuous batching achieves 1200.07 tokens/s. Multi-bin batching improves over static batching for all tested values of k : $k = 2$ achieves 502.65 tokens/s, $k = 4$ achieves 720.17 tokens/s, and $k = 8$ achieves 965.97 tokens/s.

Relative to static batching, the $k = 8$ oracle result gives a 253.1% throughput improvement. However, unlike the controlled oracle-length workload, the Dolly oracle result

remains below the native continuous-batching baseline: $k = 8$ oracle multi-bin batching is 19.5% below continuous batching. These results emphasize two points. First, accurate length information can recover a large amount of the static-batching inefficiency. Second, the remaining gap to continuous batching suggests that more fine-grained binning, adaptive bin boundaries, or integration with continuous-batching mechanisms may be needed to fully exploit length information.

5.7.4 Dolly Dataset Inference with Estimated Lengths

Bins	Exact acc.	Adjacent acc.
$k = 2$	0.943	1.000
$k = 4$	0.791	0.998
$k = 8$	0.567	0.949

Table 5.1: Estimator bin-assignment quality on the Dolly dataset.

We next evaluate the same Dolly workload when bin assignments are obtained from estimated rather than oracle output lengths. The estimator is a lightweight length regressor built on prompt-prefill hidden-state features from Llama-3.2-3B-Instruct, motivated by recent work showing that the served LLM’s internal representations can provide useful signal for output-length prediction [138]. Each prompt is processed with the same Llama-3 style chat template used for label generation, but no answer tokens are generated for bin assignment. This design aligns the predictor with the model used in the serving experiments, while keeping bin assignment limited to prompt-prefill computation.

Table 5.1 summarizes the resulting bin-assignment quality. The realized exact bin accuracies are 94.28%, 79.09%, and 56.69% for $k = 2$, $k = 4$, and $k = 8$, respectively. The corresponding adjacent-bin accuracies are 100.0%, 99.80%, and 94.89%. Thus, although exact bin accuracy decreases as the number of bins increases, most errors remain localized near the correct bin, especially for $k = 8$, where fewer than 5.2% of requests are assigned

more than one bin away from their oracle bin. Figure 5.8 shows the corresponding row-normalized confusion matrices; for readability, the $k = 8$ panel is annotated using whole-number percentages.

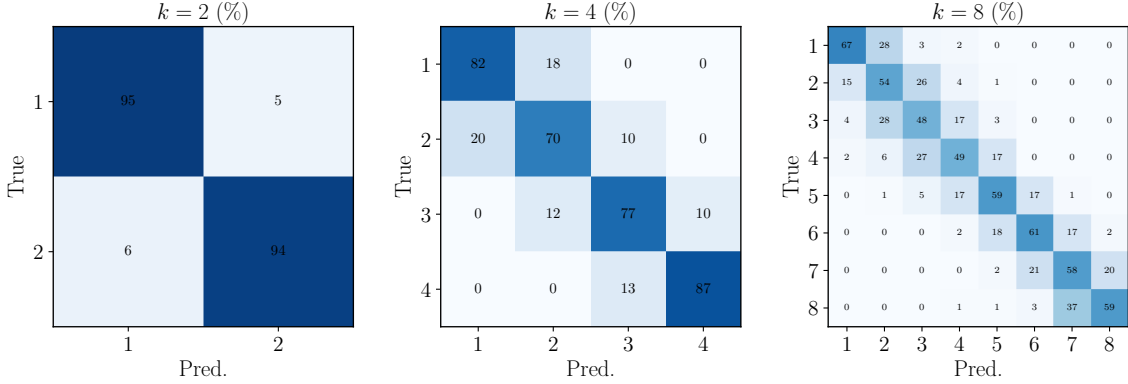


Figure 5.8: Row-normalized confusion matrices for the estimated Dolly dataset bin assignments.

As shown in Figure 5.7, estimated-length binning still improves substantially over static batching. Throughput increases from 273.53 tokens/s under static batching to 485.70, 597.91, and 684.02 tokens/s for $k = 2$, $k = 4$, and $k = 8$, respectively. The best estimated-length result, obtained at $k = 8$, is 150.1% higher than static batching. However, it remains 43.0% below the continuous-batching baseline.

These results indicate that practical length estimation preserves a large fraction of the gain over static batching, while leaving substantial room for further improvement. The gap between the estimated and oracle bars in Figure 5.7 quantifies the remaining opportunity from more accurate length information: at $k = 8$, oracle assignments increase throughput from 684.02 to 965.97 tokens/s. Although the Dolly oracle result remains below the continuous-batching baseline in this experiment, it closes much of the gap and suggests that finer binning, improved lightweight length estimators, and bin boundaries adapted to workload-specific prediction errors can further strengthen multi-bin batching. These results support our view that length-aware grouping is a useful scheduling principle

that can complement modern continuous-batching systems.

5.7.5 Effect of Symmetrical Prediction Errors

Finally, we retain the controlled robustness study with symmetric binning errors. This experiment is a simulation rather than a measured `llama.cpp` inference run. Its purpose is to isolate how multi-bin batching degrades as bin assignments become noisier under a simple and controlled error model.

For a true interior bin i , the predicted bin is $i - 1$ with probability p_e , $i + 1$ with probability p_e , and i with probability $1 - 2p_e$. For edge bins, the prediction moves to the only neighboring bin with probability p_e and remains correct with probability $1 - p_e$. This error model captures adjacent-bin mistakes, which are also the dominant type of error observed in the estimated Dolly experiments. It should not be interpreted as a full model of real predictor errors, which may be asymmetric, workload-dependent, or span multiple bins.

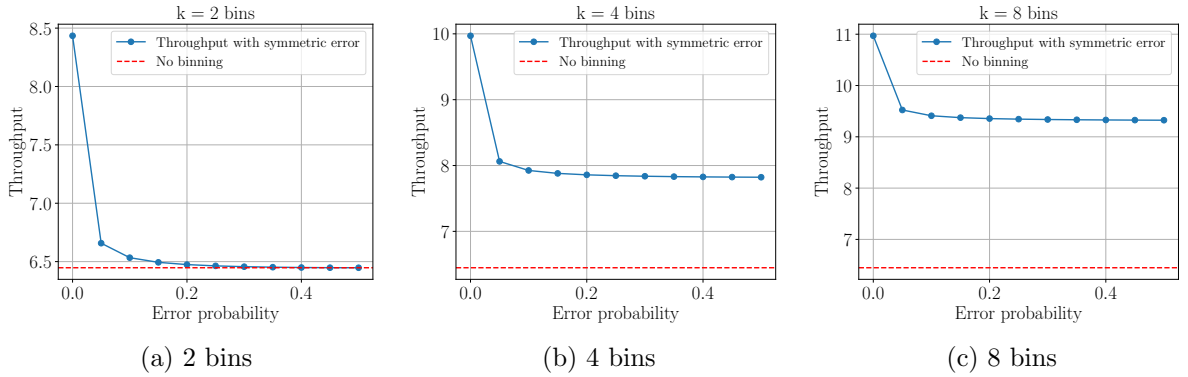


Figure 5.9: Throughput as a function of the symmetric adjacent-bin error probability p_e for different values of k .

Figure 5.9 reports throughput as p_e varies for $k = 2$, $k = 4$, and $k = 8$. The simulation uses batch size $B = 128$, service times between $l_{\min} = 1$ and $l_{\max} = 20$, and 128000 submitted requests. Results are averaged over 10 simulation runs. As expected,

throughput decreases as the error probability increases. Nevertheless, the multi-bin policy remains robust to moderate adjacent-bin errors. This simulation complements the estimated Dolly experiment by showing, in a controlled setting, how throughput changes as the binning signal becomes progressively noisier.

5.8 Conclusion

This chapter presented multi-bin batching, a queueing-based control policy that groups requests with similar output lengths to improve batching efficiency. Our theoretical results show that length-aware binning improves the effective batched service rate and approaches the idealized static-batching capacity as the number of bins increases. The analysis isolates the effect of output-length heterogeneity and should be interpreted as complementary to, rather than a complete model of, modern continuous-batching systems. Our empirical studies support this interpretation. The real-inference experiments show significant gains over static batching, and they also compare directly with the backend’s native continuous-batching behavior. On the Dolly workload, both oracle and estimated-length multi-bin batching substantially improve over static batching. Although they remain below the backend’s native continuous-batching baseline in this experiment, the oracle $k = 8$ result narrows the gap considerably, and the estimated-length result demonstrates that practical length prediction still preserves significant throughput gains. These results suggest that accurate length information can provide a useful scheduling signal, while also showing that predictor quality is central to the practical value of the method. Future work could focus on improving lightweight length predictors, incorporating uncertainty-aware bin assignment, developing adaptive binning strategies, and studying the full latency–throughput Pareto tradeoff in end-to-end serving systems.

Appendix A

Supplementary Materials for Chapter 2

A.1 Theoretical Results

A.1.1 Connections between EI, DP, and BE

In this section, we provide the proof of Theorem 2.5 and Corollary 2.6.

Proof of Theorem 2.5. All we need to prove are three statements:

1. Prove that EI and BE imply DP
2. Prove that DP and EI imply BE
3. Prove that BE and DP imply EI

Below we prove each statement.

1. EI, BE $\Rightarrow$ DP Suppose a classifier f achieves EI and BE. Recall that a classifier achieves EI if

$$\mathbb{P} \left(\max_{\mu(\Delta \mathbf{x}_I) \leq \delta} f(\mathbf{x} + \Delta \mathbf{x}) \geq 0.5 \mid f(\mathbf{x}) < 0.5, z = z \right) = \mathbb{P} \left(\max_{\mu(\Delta \mathbf{x}_I) \leq \delta} f(\mathbf{x} + \Delta \mathbf{x}) \geq 0.5 \mid f(\mathbf{x}) < 0.5 \right) \quad (\text{A.1})$$

and a classifier achieves BE if

$$\mathbb{P} \left(\max_{\mu(\Delta \mathbf{x}_I) \leq \delta} f(\mathbf{x} + \Delta \mathbf{x}) \geq 0.5, f(\mathbf{x}) < 0.5 \mid z = z \right) = \mathbb{P} \left(\max_{\mu(\Delta \mathbf{x}_I) \leq \delta} f(\mathbf{x} + \Delta \mathbf{x}) \geq 0.5, f(\mathbf{x}) < 0.5 \right) \quad (\text{A.2})$$

By dividing both sides of A.2 by the corresponding sides of A.1, we have

$$\frac{\mathbb{P} \left(\max_{\mu(\Delta \mathbf{x}_I) \leq \delta} f(\mathbf{x} + \Delta \mathbf{x}) \geq 0.5, f(\mathbf{x}) < 0.5 \mid z = z \right)}{\mathbb{P} \left(\max_{\mu(\Delta \mathbf{x}_I) \leq \delta} f(\mathbf{x} + \Delta \mathbf{x}) \geq 0.5 \mid f(\mathbf{x}) < 0.5, z = z \right)} = \frac{\mathbb{P} \left(\max_{\mu(\Delta \mathbf{x}_I) \leq \delta} f(\mathbf{x} + \Delta \mathbf{x}) \geq 0.5, f(\mathbf{x}) < 0.5 \right)}{\mathbb{P} \left(\max_{\mu(\Delta \mathbf{x}_I) \leq \delta} f(\mathbf{x} + \Delta \mathbf{x}) \geq 0.5 \mid f(\mathbf{x}) < 0.5 \right)} \quad (\text{A.3})$$

Then, it can be simplified as

$$\mathbb{P}(f(\mathbf{x}) < 0.5 \mid z = z) = \mathbb{P}(f(\mathbf{x}) < 0.5),$$

which implies that the classifier achieves demographic parity,

$$\mathbb{P}(f(\mathbf{x}) \geq 0.5 \mid z = z) = \mathbb{P}(f(\mathbf{x}) \geq 0.5)$$

2. DP, EI $\Rightarrow$ BE Suppose a classifier f achieves DP and EI. Recall that a classifier achieves DP if

$$\mathbb{P}(f(\mathbf{x}) \geq 0.5 \mid z = z) = \mathbb{P}(f(\mathbf{x}) \geq 0.5), \quad (\text{A.4})$$

which implies

$$\mathbb{P}(f(\mathbf{x}) < 0.5 \mid z = z) = \mathbb{P}(f(\mathbf{x}) < 0.5). \quad (\text{A.5})$$

Recall that a classifier achieves EI if

$$\mathbb{P}\left(\max_{\mu(\Delta\mathbf{x}_I) \leq \delta} f(\mathbf{x} + \Delta\mathbf{x}) \geq 0.5 \mid f(\mathbf{x}) < 0.5, z = z\right) = \mathbb{P}\left(\max_{\mu(\Delta\mathbf{x}_I) \leq \delta} f(\mathbf{x} + \Delta\mathbf{x}) \geq 0.5 \mid f(\mathbf{x}) < 0.5\right) \quad (\text{A.6})$$

By multiplying both sides of A.5 and A.6, we have

$$\begin{aligned} & \mathbb{P}\left(\max_{\mu(\Delta\mathbf{x}_I) \leq \delta} f(\mathbf{x} + \Delta\mathbf{x}) \geq 0.5 \mid f(\mathbf{x}) < 0.5, z = z\right) \mathbb{P}(f(\mathbf{x}) < 0.5 \mid z = z) \\ &= \mathbb{P}\left(\max_{\mu(\Delta\mathbf{x}_I) \leq \delta} f(\mathbf{x} + \Delta\mathbf{x}) \geq 0.5 \mid f(\mathbf{x}) < 0.5\right) \mathbb{P}(f(\mathbf{x}) < 0.5) \end{aligned} \quad (\text{A.7})$$

Then, it can be simplified as

$$\mathbb{P}\left(\max_{\mu(\Delta\mathbf{x}_I) \leq \delta} f(\mathbf{x} + \Delta\mathbf{x}) \geq 0.5, f(\mathbf{x}) < 0.5 \mid z = z\right) = \mathbb{P}\left(\max_{\mu(\Delta\mathbf{x}_I) \leq \delta} f(\mathbf{x} + \Delta\mathbf{x}) \geq 0.5, f(\mathbf{x}) < 0.5\right), \quad (\text{A.8})$$

which implies that the classifier f achieves BE.

3. BE, DP $\Rightarrow$ EI Suppose a classifier f achieves BE and DP. Recall that a classifier achieves DP if

$$\mathbb{P}(f(\mathbf{x}) \geq 0.5 \mid z = z) = \mathbb{P}(f(\mathbf{x}) \geq 0.5), \quad (\text{A.9})$$

which implies

$$\mathbb{P}(f(\mathbf{x}) < 0.5 \mid z = z) = \mathbb{P}(f(\mathbf{x}) < 0.5) \quad (\text{A.10})$$

Recall that a classifier achieves BE if

$$\mathbb{P} \left(\max_{\mu(\Delta \mathbf{x}_I) \leq \delta} f(\mathbf{x} + \Delta \mathbf{x}) \geq 0.5, f(\mathbf{x}) < 0.5 \mid z = z \right) = \mathbb{P} \left(\max_{\mu(\Delta \mathbf{x}_I) \leq \delta} f(\mathbf{x} + \Delta \mathbf{x}) \geq 0.5, f(\mathbf{x}) < 0.5 \right). \quad (\text{A.11})$$

By dividing both sides of A.11 by the corresponding sides of A.10, we have

$$\frac{\mathbb{P} \left(\max_{\mu(\Delta \mathbf{x}_I) \leq \delta} f(\mathbf{x} + \Delta \mathbf{x}) \geq 0.5, f(\mathbf{x}) < 0.5 \mid z = z \right)}{\mathbb{P}(f(\mathbf{x}) < 0.5 \mid z = z)} = \frac{\mathbb{P} \left(\max_{\mu(\Delta \mathbf{x}_I) \leq \delta} f(\mathbf{x} + \Delta \mathbf{x}) \geq 0.5, f(\mathbf{x}) < 0.5 \right)}{\mathbb{P}(f(\mathbf{x}) < 0.5)} \quad (\text{A.12})$$

Then, it can be simplified as

$$\mathbb{P} \left(\max_{\mu(\Delta \mathbf{x}_I) \leq \delta} f(\mathbf{x} + \Delta \mathbf{x}) \geq 0.5 \mid f(\mathbf{x}) < 0.5, z = z \right) = \mathbb{P} \left(\max_{\mu(\Delta \mathbf{x}_I) \leq \delta} f(\mathbf{x} + \Delta \mathbf{x}) \geq 0.5 \mid f(\mathbf{x}) < 0.5 \right), \quad (\text{A.13})$$

which implies that the classifier f achieves EI. ■

Proof of Corollary 2.6. Corollary 2.6 can be proved directly from Theorem 2.5. ■

A.1.2 Connections between EI and ER

In this part, we investigate the connections between EI and ER.

Lemma A.1 *Consider $\mathbf{x} \mid z = z \sim \mathcal{N}(\mu_z, \sigma^2)$ for $z \in \{0, 1\}$, $\mu_z, \sigma \in \mathbb{R}$, and classifiers characterized by two accepting thresholds (τ_0, τ_1) , where $\tau_0, \tau_1 \in \mathbb{R}$. If a classifier satisfies EI, then it satisfies ER.*

Proof. Here we use $\Phi = 1 - Q$ and ϕ to denote the CDF and PDF of standard Gaussian distribution, respectively. We consider the cost function $\mu = |\cdot|$.

Recall the definition of EI disparity and ER disparity:

$$\begin{aligned} \text{EI Disparity} = & \left| \mathbb{P} \left(\underbrace{\max_{\mu(\Delta x) < \delta} f(x + \Delta x) > 0.5}_{x > \tau_0 - \delta} \mid \underbrace{f(x) < 0.5}_{x \leq \tau_0}, z = 0 \right) \right. \\ & \left. - \mathbb{P} \left(\max_{\mu(\Delta x) < \delta} f(x + \Delta x) > 0.5 \mid f(x) < 0.5, z = 1 \right) \right|. \end{aligned} \quad (\text{A.14})$$

$$\begin{aligned} \text{ER Disparity} = & \left| \mathbb{E} \left[\underbrace{\min_{f(x+\Delta x) \geq 0.5} \mu(\Delta x)}_{\tau_0 - x} \mid \underbrace{f(x) < 0.5}_{x \leq \tau_0}, z = 0 \right] \right. \\ & \left. - \mathbb{E} \left[\min_{f(x+\Delta x) \geq 0.5} \mu(\Delta x) \mid f(x) < 0.5, z = 1 \right] \right|. \end{aligned} \quad (\text{A.15})$$

Consequently, the EI constraint and ER constraint can be written as

$$\text{EI Disparity } (\tau_0, \tau_1) = \left| \frac{\Phi \left(\frac{\tau_0 - \delta - \mu_0}{\sigma} \right)}{\Phi \left(\frac{\tau_0 - \mu_0}{\sigma} \right)} - \frac{\Phi \left(\frac{\tau_1 - \delta - \mu_1}{\sigma} \right)}{\Phi \left(\frac{\tau_1 - \mu_1}{\sigma} \right)} \right| = 0. \quad (\text{A.16})$$

$$\begin{aligned} \text{ER Disparity } (\tau_0, \tau_1) = & \left| \frac{1}{\Phi((\tau_0 - \mu_0)/\sigma)} \int_{-\infty}^{\tau_0} (\tau_0 - t) \phi \left(\frac{t - \mu_0}{\sigma} \right) dt \right. \\ & \left. - \frac{1}{\Phi((\tau_1 - \mu_1)/\sigma)} \int_{-\infty}^{\tau_1} (\tau_1 - t) \phi \left(\frac{t - \mu_1}{\sigma} \right) dt \right| = 0. \end{aligned} \quad (\text{A.17})$$

In this proof, we will first show that achieving EI is equivalent to $\tau_0 - \mu_0 = \tau_1 - \mu_1$, and then show that the classifier with $\tau_0 - \mu_0 = \tau_1 - \mu_1$ satisfies the ER constraint.

1. EI constraint.

Let $\varphi(x) = \Phi(\frac{x-\delta}{\sigma})/\Phi(\frac{x}{\sigma})$. First, we show that φ is a strictly increasing function.

Note that

$$\varphi'(x) = \frac{1}{\sigma \Phi\left(\frac{x}{\sigma}\right)^2} \left(\phi\left(\frac{x-\delta}{\sigma}\right) \Phi\left(\frac{x}{\sigma}\right) - \Phi\left(\frac{x-\delta}{\sigma}\right) \phi\left(\frac{x}{\sigma}\right) \right). \quad (\text{A.18})$$

Therefore, to show that φ is strictly increasing, it is sufficient to show that

$$\phi\left(\frac{x-\delta}{\sigma}\right) \Phi\left(\frac{x}{\sigma}\right) > \Phi\left(\frac{x-\delta}{\sigma}\right) \phi\left(\frac{x}{\sigma}\right). \quad (\text{A.19})$$

We show that (A.19) is equivalent as the following inequality by dividing both the left-hand side and right-hand side by $\phi\left(\frac{x-\delta}{\sigma}\right)\phi\left(\frac{x}{\sigma}\right)$:

$$\Phi\left(\frac{x}{\sigma}\right) / \phi\left(\frac{x}{\sigma}\right) > \Phi\left(\frac{x-\delta}{\sigma}\right) / \phi\left(\frac{x-\delta}{\sigma}\right). \quad (\text{A.20})$$

Note that $\frac{1-\Phi(\cdot)}{\phi(\cdot)}$ is known in literatures as Mills' ratio [139], which is strictly decreasing on $\mathbb{R}$. Therefore, $\frac{\Phi(\cdot)}{\phi(\cdot)}$ is strictly increasing on $\mathbb{R}$. Since $\frac{x}{\sigma} > \frac{x-\delta}{\sigma}$, (A.20) holds, thereby (A.19) holds and φ is strictly increasing.

Given that $\varphi(x) = \Phi\left(\frac{x-\delta}{\sigma}\right) / \Phi\left(\frac{x}{\sigma}\right)$ is a strictly increasing function on $\mathbb{R}$,

$$(A.16) = |\varphi(\tau_0 - \mu_0) - \varphi(\tau_1 - \mu_1)| = 0 \quad (\text{A.21})$$

yields that

$$\tau_0 - \mu_0 = \tau_1 - \mu_1. \quad (\text{A.22})$$

2. ER constraint.

We first note that

$$\int_{-\infty}^{\tau_0} (\tau_0 - t) \phi\left(\frac{t - \mu_0}{\sigma}\right) dt \stackrel{t' = t - \mu_0}{=} \int_{-\infty}^{\tau_0 - \mu_0} (\tau_0 - \mu_0 - t') \phi\left(\frac{t'}{\sigma}\right) dt' \quad (\text{A.23})$$

Let $\psi(x) = \frac{1}{\Phi(x/\sigma)} \int_{-\infty}^x (x-t)\phi(\frac{t}{\sigma})dt$. It is clear that ER constraint is equivalent to

$$(A.17) = |\psi(\tau_0 - \mu_0) - \psi(\tau_1 - \mu_1)|. \quad (A.24)$$

Therefore, the classifier with $\tau_0 - \mu_0 = \tau_1 - \mu_1$ clearly satisfies the ER constraint.

Combining all the discussion above completes the proof. ■

A.2 Supplementary Materials on the Fairness Notions

Recall that in Section 2.2 and Section 2.3, this work proposes a new fairness notion called equal improvability (EI), compares it with other existing effort-based fairness notions, and finds a classifier by solving a EI-constrained optimization which is formulated as a minimax problem. In Section A.2.1, we first explain what each term in the definition of EI means. Then, we provide more details of reward function selection of bounded effort (BE) [23] and discuss the vulnerability of equal recourse (ER) [22] in Section A.2.2 and A.2.3, respectively. Then in Section A.2.4, we provide how we solved the inner maximization problem in the EI-constrained optimization. Finally, in Section A.2.5, we provide numerical methods for finding the optimal solution for the EI-constrained problem, under simple synthetic dataset setting.

A.2.1 Meaning of each term in the definition of EI

To help readers better understand our EI definition, here we explain what each term means in EI definition means. Let the data sample $(\mathbf{x}, y, z)$ follows the distribution

$\mathcal{P}_{(\mathbf{x}, y, z)}$. The meaning of each term of EI defined in Definition 2.1 is detailed below.

$$\mathbb{P}_{\underbrace{\mathbf{x}, y, z \sim \mathcal{P}_{\mathbf{x}, y, z}}_{\text{Randomness is over the data distribution}}} \left(\underbrace{\max_{\mu(\Delta \mathbf{x}_I) \leq \delta} f(\mathbf{x} + \Delta \mathbf{x}) \geq 0.5}_{\substack{\text{Maximum score} \\ \text{after improvement}}} \mid \underbrace{f(\mathbf{x}) < 0.5, z = z}_{\substack{\text{Event in which the sample} \\ \text{comes from group } z \\ \text{and gets rejected}}} \right) \quad (\text{A.25})$$

Event in which the sample can be accepted after improvement

$$= \mathbb{P}_{\underbrace{\mathbf{x}, y, z \sim \mathcal{P}_{\mathbf{x}, y, z}}_{\text{Randomness is over the data distribution}}} \left(\underbrace{\max_{\mu(\Delta \mathbf{x}_I) \leq \delta} f(\mathbf{x} + \Delta \mathbf{x}) \geq 0.5}_{\substack{\text{Maximum score} \\ \text{after improvement}}} \mid \underbrace{f(\mathbf{x}) < 0.5}_{\substack{\text{Event in which the} \\ \text{sample gets rejected}}} \right), \quad (\text{A.26})$$

Event in which the sample can be accepted after improvement

A.2.2 Specifying the reward function for BE

In this part, we introduce how we set the reward function so that we can connect BE with EI. Recall that the reward function in [23] is defined as the benefit gained by changing an individual's characteristics from $w = (\mathbf{x}, y)$ to $w' = (\mathbf{x}', y')$:

$$\mathcal{R}(w, w') = b(f(\mathbf{x}'), y') - b(f(\mathbf{x}), y) \quad (\text{A.27})$$

where b is the benefit function, and $\mathbf{x}' = \mathbf{x} + \Delta \mathbf{x}$ is the updated feature. If we set the benefit function as $b(f(\mathbf{x}), y) = \mathbf{1}\{f(\mathbf{x}) \geq 0.5\}$, then the reward function becomes:

$$\mathcal{R}(w, w') = \mathbf{1}\{f(\mathbf{x} + \Delta \mathbf{x}) \geq 0.5\} - \mathbf{1}\{f(\mathbf{x}) \geq 0.5\} \quad (\text{A.28})$$

Then, the bounded effort fairness is defined as:

$$\mathbb{E} \left[\max_{\Delta \mathbf{x}} \mathcal{R}(w, w') \text{ s.t. } \mu(\Delta \mathbf{x}) < \delta \mid z = z \right] = \mathbb{E} \left[\max_{\Delta \mathbf{x}} \mathcal{R}(w, w') \text{ s.t. } \mu(\Delta \mathbf{x}) < \delta \right] \text{ for all } z \quad (\text{A.29})$$

Consequently,

$$\left(\max_{\Delta \mathbf{x}} \mathcal{R}(w, w') \text{ s.t. } \mu(\Delta \mathbf{x}) < \delta \right) = \begin{cases} 1 & \text{if } \max_{\mu(\Delta \mathbf{x}) < \delta} f(\mathbf{x} + \Delta \mathbf{x}) \geq 0.5 \text{ and } f(\mathbf{x}) < 0.5 \\ 0 & \text{otherwise} \end{cases} \quad (\text{A.30})$$

Therefore, we can write expectation as probability:

$$\begin{aligned} P \left(\max_{\mu(\Delta \mathbf{x}) \leq \delta} f(\mathbf{x} + \Delta \mathbf{x}) \geq 0.5, f(\mathbf{x}) < 0.5 \mid z = z \right) \\ = P \left(\max_{\mu(\Delta \mathbf{x}) \leq \delta} f(\mathbf{x} + \Delta \mathbf{x}) \geq 0.5, f(\mathbf{x}) < 0.5 \right). \end{aligned} \quad (\text{A.31})$$

which is in Table 2.1.

A.2.3 Vulnerability of ER to outliers

As we mentioned in Table 2.1, ER is vulnerable to outliers. Here we provide a simple analysis. Suppose an outlier, having feature-attribute pair $(\mathbf{x}, z)$, is added to the dataset with n samples. Let the outlier is misqualified $f(\mathbf{x}) < 0.5$, and requires effort $\mu(\Delta \mathbf{x}) = M$ to achieve $f(\mathbf{x} + \Delta \mathbf{x}) \geq 0.5$. In such case, the EI disparity increases at most $\frac{1}{n}$ since EI measures the *portion* of samples with improved outcomes after making efforts. On the other hand, the ER disparity increases by $\frac{M}{n}$, since ER measures the minimum required efforts *averaged* over all samples. Note that a single outlier with a large M can significantly increase the ER disparity, which does not happen for EI. Thus, one can observe that EI is much more robust to outliers, compared with ER. We conduct a numerical experiment in Section A.3.4 to further highlight the robustness of EI to outliers.

A.2.4 Solving the inner maximization problem for EI

As explained in Section 2.3, finding a EI classifier can be formulated as a minimax problem (2.3), where solving the inner maximization problem $\max_{\|\Delta \mathbf{x}_I\| \leq \delta} f(\mathbf{x} + \Delta \mathbf{x})$ is required to compute U_δ in the regularization term, and the outer problem is the regularized-loss minimization finding the optimal model parameter $\mathbf{w}$ for the classifier $f = f_{\mathbf{w}}$. In this section, we provide two ways of solving the inner maximization problem. In particular, in Section A.2.4 we give the explicit expression of the optimizer $\Delta \mathbf{x}_I$ when generalized linear model is considered. In Section A.2.4, we solve the problem under a more general setting via adversarial training.

Closed-form solution for generalized linear model

Consider a Generalized Linear Model (GLM) written as $f(\mathbf{x}) = g^{-1}(\mathbf{w}^\top \mathbf{x})$, where $g : [0, 1] \rightarrow \mathbb{R}$ is a strictly increasing link function, and $\mathbf{w}$ is the model parameter. Denote the weights corresponding to $\mathbf{x}_I$ as $\mathbf{w}_I$. Then, the inner maximization problem can be written as:

$$\max_{\|\Delta \mathbf{x}_I\| \leq \delta} f(\mathbf{x} + \Delta \mathbf{x}) = \max_{\|\Delta \mathbf{x}_I\| \leq \delta} g^{-1}(\mathbf{w}^\top (\mathbf{x} + \Delta \mathbf{x})) \quad (\text{A.32})$$

$$= \max_{\|\Delta \mathbf{x}_I\| \leq \delta} g^{-1}(\mathbf{w}^\top \mathbf{x} + \mathbf{w}_I^\top \Delta \mathbf{x}_I) \quad (\because \Delta \mathbf{x} = (\Delta \mathbf{x}_I, 0, 0)) \quad (\text{A.33})$$

$$= g^{-1} \left(\mathbf{w}^\top \mathbf{x} + \max_{\|\Delta \mathbf{x}_I\| \leq \delta} \mathbf{w}_I^\top \Delta \mathbf{x}_I \right) \quad (\because g \text{ is strictly increasing}) \quad (\text{A.34})$$

When $\|\cdot\| = \|\cdot\|_\infty$, the maximum is achieved by letting $\Delta \mathbf{x}_I = \delta \text{sign}(\mathbf{w}_I)$, which results in $\mathbf{w}^\top \Delta \mathbf{x}_I = \delta \|\mathbf{w}_I\|_1$. When $\|\cdot\| = \|\cdot\|_2$, the maximum can be achieved by letting $\Delta \mathbf{x}_I = \delta \mathbf{w}_I / \|\mathbf{w}_I\|_2$ which results in $\mathbf{w}^\top \Delta \mathbf{x}_I = \delta \|\mathbf{w}_I\|_2$.

Adversarial training based approach for general setup

Here we discuss how we solve the inner maximization problem under a more general setting. Following popular adversarial training methods, we apply projected gradient descent (PGD) for multiple times to update $\Delta \mathbf{x}_I$, *i.e.*, set

$$\Delta \mathbf{x}_I = \mathcal{P}(\Delta \mathbf{x}_I + \gamma \nabla_{\Delta \mathbf{x}_I} f(\mathbf{x} + \Delta \mathbf{x})), \quad (\text{A.35})$$

where $\gamma > 0$ is the step size, and $\mathcal{P}$ is the projection onto the constrained space $\|\Delta \mathbf{x}_I\| \leq \delta$. For instance, $\mathcal{P}$ is equivalent to the clipping process when we use ℓ_∞ norm. Denote the maximizer of the inner maximization problem as $\Delta \mathbf{x}^* = (\Delta \mathbf{x}_I^*, \mathbf{0}, \mathbf{0})$. Then, from Danskin's theorem [30], we have $\nabla_{\mathbf{w}} \max_{\|\Delta \mathbf{x}_I\| \leq \delta} f_{\mathbf{w}}(\mathbf{x} + \Delta \mathbf{x}) = \nabla_{\mathbf{w}} f_{\mathbf{w}}(\mathbf{x} + \Delta \mathbf{x}^*)$. We can use this derivative to update $\mathbf{w}$ in the outer loss minimization problem. The pseudocode of this adversarial training based method is shown in Algorithm 3.

Algorithm 3: Pseudocode for achieving EI

Input : Dataset $\mathcal{D}$
Output: Model parameter $\mathbf{w}$ for the classifier f .

- 1 Initialize $\mathbf{w}$;
- 2 **foreach** *iteration* **do**
- 3 **foreach** $(\mathbf{x}_i, y_i) \in \mathcal{D}$ **do**
- 4 Initialize $\Delta \mathbf{x}_i^*$;
- 5 **foreach** *PGD iteration* **do**
- 6 Update $\Delta \mathbf{x}_i^*$ according to (A.35);
- 7 Update $\mathbf{w}$ according to the regularized loss function defined in (2.3);

A.2.5 Derivation of optimal EI classifier for synthetic dataset

This section shows how we obtain the optimal EI classifier (that minimizes the cost function in (2.3)) for a synthetic dataset having two features $\mathbf{x} = [x_1, x_2]$ sampled from a Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_{z,y}, \boldsymbol{\Sigma}_{z,y})$ where the mean $\boldsymbol{\mu}_{z,y}$ and the standard deviation $\boldsymbol{\Sigma}_{z,y}$ depends on the label $y \in \{0, 1\}$ and the group attribute $z \in \{0, 1\}$. Note that the

performance curve of the optimal EI classifier obtained in this section is provided in the yellow line in Figure 2.3.

The optimal EI classifier is obtained in the following steps: (i) define mathematical notations used for analysis (Section A.2.5), (ii) compute the error probability (Section A.2.5), (iii) compute EI disparity (Section A.2.5), and (iv) solve the EI-regularized optimization problem and find the optimal EI classifier (Section A.2.5).

Notations

We consider finding a z -aware linear classifier which predicts the label y from two features x_1, x_2 and one sensitive attribute z . In other words, given $\mathbf{x}$ and z , the output of a model is represented as $f(\mathbf{x}) = w_1x_1 + w_2x_2 + w_3z + b$ ¹ where $[w_1, w_2, w_3]$ is the weight vector, b is the bias.

For group $z = 0$,

$$\hat{y} = \begin{cases} 1 & \text{if } w_1x_1 + w_2x_2 > -b \\ 0 & \text{else} \end{cases}$$

For group $z = 1$,

$$\hat{y} = \begin{cases} 1 & \text{if } w_1x_1 + w_2x_2 > -w_3 - b \\ 0 & \text{else} \end{cases}$$

Without the loss of generality, let $\sqrt{w_1^2 + w_2^2} = 1$, and parameterize them as $w_1 = \sin \theta, w_2 = \cos \theta$. Then, for group $z = 0$,

$$\hat{y} = \begin{cases} 1 & \text{if } (\sin \theta)x_1 + (\cos \theta)x_2 > b_0 \\ 0 & \text{else} \end{cases}$$

¹In this case, the decision boundary is $\{\mathbf{x} : f(\mathbf{x}) = 0\}$ instead of $\{\mathbf{x} : f(\mathbf{x}) = 0.5\}$ used in the main chapter.

and for group $z = 1$,

$$\hat{y} = \begin{cases} 1 & \text{if } (\sin \theta)x_1 + (\cos \theta)x_2 > b_1 \\ 0 & \text{else} \end{cases}$$

where $b_0 = -b$ and $b_1 = -w_3 - b$. Since the linear combination of multivariate Gaussian is a univariate Gaussian, we have

$$\mathbf{w}_\theta^\top \mathbf{x} \sim \mathcal{N}(\mathbf{w}_\theta^\top \boldsymbol{\mu}_{z,y}, \mathbf{w}_\theta^\top \boldsymbol{\Sigma}_{z,y} \mathbf{w}_\theta) \quad (\text{A.36})$$

where $\mathbf{w}_\theta = [\sin \theta, \cos \theta]$. The decision rules can be written in terms of the $\mathbf{w}_\theta$. For group $z = 0$,

$$\hat{y} = \begin{cases} 1 & \text{if } \mathbf{w}_\theta^\top \mathbf{x} > b_0 \\ 0 & \text{else} \end{cases}$$

For group $z = 1$,

$$\hat{y} = \begin{cases} 1 & \text{if } \mathbf{w}_\theta^\top \mathbf{x} > b_1 \\ 0 & \text{else} \end{cases}$$

Now, the question is, what is the optimal parameters θ, b_0, b_1 that solve the optimization problem in (2.3). In order to answer this question, we need to understand how the equal improvability condition is represented in terms of the model parameters. Suppose we use 0-1 loss function l , and use l_∞ norm $\mu(\mathbf{x}) = \|\mathbf{x}\|_\infty$. From the result in Section A.2.4, given the effort budget δ , the maximum score improvement $(\max_{\mu(\Delta \mathbf{x}_I) \leq \delta} f(\mathbf{x} + \Delta \mathbf{x}) - f(\mathbf{x})) = \delta \|\mathbf{w}_I\|_1 = \delta(|\sin \theta| + |\cos \theta|)$ where $\mathbf{w}_I$ is the weights for improvable features x_1, x_2 . Thus, if we denote the $\hat{y}^{\max}$ as the estimated label after

the improvement, we have

$$\hat{y}^{\max} = \begin{cases} 1 & \text{if } (\sin \theta)x_1 + (\cos \theta)x_2 > b_0 - \delta(|\sin \theta| + |\cos \theta|) = b'_0 \\ 0 & \text{else} \end{cases}$$

for group $z = 0$ and

$$\hat{y}^{\max} = \begin{cases} 1 & \text{if } (\sin \theta)x_1 + (\cos \theta)x_2 > b_1 - \delta(|\sin \theta| + |\cos \theta|) = b'_1 \\ 0 & \text{else} \end{cases}$$

for group $z = 1$.

Compute Error Probability

The error probability can be written as,

$$\Pr(\hat{y} \neq y) = \sum_{i=0}^1 \Pr(z = i) \Pr(\hat{y} \neq y | z = i)$$

We can derive the term $\Pr(\hat{y} \neq y | z = 0)$ as below:

$$\begin{aligned} \Pr(\hat{y} \neq y | z = 0) &= \Pr(y = 0 | z = 0) \Pr(\hat{y} = 1 | y = 0, z = 0) \\ &\quad + \Pr(y = 1 | z = 0) \Pr(\hat{y} = 0 | y = 1, z = 0). \end{aligned} \tag{A.37}$$

We can look each term $\Pr(\hat{y} = 1 | y = 0, z = 0)$, $\Pr(\hat{y} = 0 | y = 1, z = 0)$ and write those terms in terms of Q-functions because,

$$\Pr(\hat{y} = 1 | y = 0, z = 0) = \Pr(\mathbf{w}_\theta^\top \mathbf{x} > b_0 | y = 0, z = 0)$$

$$\Pr(\hat{y} = 0 | y = 1, z = 0) = \Pr(\mathbf{w}_\theta^\top \mathbf{x} < b_0 | y = 1, z = 0)$$

From (A.36), we have

$$\Pr(\hat{y} = 1|y = 0, z = 0) = \Pr(\mathbf{w}_\theta^\top \mathbf{x} > b_0|y = 0, z = 0) = Q\left(\frac{b_0 - \mathbf{w}_\theta^\top \boldsymbol{\mu}_{0,0}}{\sqrt{\mathbf{w}_\theta^\top \boldsymbol{\Sigma}_{0,0} \mathbf{w}_\theta}}\right)$$

$$\Pr(\hat{y} = 0|y = 1, z = 0) = \Pr(\mathbf{w}_\theta^\top \mathbf{x} < b_0|y = 1, z = 0) = Q\left(\frac{\mathbf{w}_\theta^\top \boldsymbol{\mu}_{1,0} - b_0}{\sqrt{\mathbf{w}_\theta^\top \boldsymbol{\Sigma}_{1,0} \mathbf{w}_\theta}}\right)$$

One can derive the error rates for group $z = 1$ similarly. So, the total error rate can be written as

$$\begin{aligned} \Pr(\hat{y} \neq y) = & \Pr(z = 0) \left[\Pr(y = 0 | z = 0) Q\left(\frac{b_0 - \mathbf{w}_\theta^\top \boldsymbol{\mu}_{0,0}}{\sqrt{\mathbf{w}_\theta^\top \boldsymbol{\Sigma}_{0,0} \mathbf{w}_\theta}}\right) \right. \\ & \left. + \Pr(y = 1 | z = 0) Q\left(\frac{\mathbf{w}_\theta^\top \boldsymbol{\mu}_{1,0} - b_0}{\sqrt{\mathbf{w}_\theta^\top \boldsymbol{\Sigma}_{1,0} \mathbf{w}_\theta}}\right) \right] \\ & + \Pr(z = 1) \left[\Pr(y = 0 | z = 1) Q\left(\frac{b_1 - \mathbf{w}_\theta^\top \boldsymbol{\mu}_{1,0}}{\sqrt{\mathbf{w}_\theta^\top \boldsymbol{\Sigma}_{1,0} \mathbf{w}_\theta}}\right) \right. \\ & \left. + \Pr(y = 1 | z = 1) Q\left(\frac{\mathbf{w}_\theta^\top \boldsymbol{\mu}_{1,1} - b_1}{\sqrt{\mathbf{w}_\theta^\top \boldsymbol{\Sigma}_{1,1} \mathbf{w}_\theta}}\right) \right]. \end{aligned} \quad (\text{A.38})$$

We have three parameters to optimize the error rate θ, b_0, b_1 . All the other terms are known.

Compute EI Disparity

To compute EI disparity, we start with computing

$$\Pr(\hat{y}^{\max} = 1|\hat{y} = 0, z = 0) = \frac{\Pr(\hat{y}^{\max} = 1, \hat{y} = 0|z = 0)}{\Pr(\hat{y} = 0|z = 0)} \quad (\text{A.39})$$

The denominator of (A.39) can be expanded as

$$\begin{aligned} \Pr(\hat{y} = 0 \mid z = 0) &= \Pr(y = 0 \mid z = 0) \Pr(\hat{y} = 0 \mid y = 0, z = 0) \\ &\quad + \Pr(y = 1 \mid z = 0) \Pr(\hat{y} = 0 \mid y = 1, z = 0). \end{aligned} \quad (\text{A.40})$$

We can look each term $\Pr(\hat{y} = 0 \mid y = 0, z = 0)$, $\Pr(\hat{y} = 0 \mid y = 1, z = 0)$ and write those terms in terms of Q-function because,

$$\Pr(\hat{y} = 0 \mid y = 0, z = 0) = \Pr(\mathbf{w}_\theta^\top \mathbf{x} < b_0 \mid y = 0, z = 0)$$

$$\Pr(\hat{y} = 0 \mid y = 1, z = 0) = \Pr(\mathbf{w}_\theta^\top \mathbf{x} < b_0 \mid y = 1, z = 0)$$

From (A.36), we have

$$\Pr(\hat{y} = 0 \mid y = 0, z = 0) = \Pr(\mathbf{w}_\theta^\top \mathbf{x} < b_0 \mid y = 0, z = 0) = Q\left(\frac{\mathbf{w}_\theta^\top \boldsymbol{\mu}_{0,0} - b_0}{\sqrt{\mathbf{w}_\theta^\top \boldsymbol{\Sigma}_{0,0} \mathbf{w}_\theta}}\right)$$

$$\Pr(\hat{y} = 0 \mid y = 1, z = 0) = \Pr(\mathbf{w}_\theta^\top \mathbf{x} > b_0 \mid y = 1, z = 0) = Q\left(\frac{\mathbf{w}_\theta^\top \boldsymbol{\mu}_{1,0} - b_0}{\sqrt{\mathbf{w}_\theta^\top \boldsymbol{\Sigma}_{1,0} \mathbf{w}_\theta}}\right)$$

Then,

$$\begin{aligned} \Pr(\hat{y} = 0 \mid z = 0) &= \Pr(y = 0 \mid z = 0) Q\left(\frac{\mathbf{w}_\theta^\top \boldsymbol{\mu}_{0,0} - b_0}{\sqrt{\mathbf{w}_\theta^\top \boldsymbol{\Sigma}_{0,0} \mathbf{w}_\theta}}\right) \\ &\quad + \Pr(y = 1 \mid z = 0) Q\left(\frac{\mathbf{w}_\theta^\top \boldsymbol{\mu}_{1,0} - b_0}{\sqrt{\mathbf{w}_\theta^\top \boldsymbol{\Sigma}_{1,0} \mathbf{w}_\theta}}\right). \end{aligned} \quad (\text{A.41})$$

The numerator of (A.39) can be expanded as

$$\Pr(\hat{y}^{\max} = 1, \hat{y} = 0 \mid z = 0) = \Pr(y = 0 \mid z = 0) \Pr(\hat{y}^{\max} = 1, \hat{y} = 0 \mid y = 0, z = 0) \quad (\text{A.42})$$

$$+ \Pr(y = 1 \mid z = 0) \Pr(\hat{y}^{\max} = 1, \hat{y} = 0 \mid y = 1, z = 0). \quad (\text{A.43})$$

We can look each term $\Pr(\hat{y}^{\max} = 1, \hat{y} = 0 | y = 0, z = 0)$, $\Pr(\hat{y}^{\max} = 1, \hat{y} = 0 | y = 1, z = 0)$ and write those terms in terms of Q-function because,

$$\Pr(\hat{y}^{\max} = 1, \hat{y} = 0 | y = 0, z = 0) = \Pr(b'_0 < \mathbf{w}_\theta^\top \mathbf{x} < b_0 | y = 0, z = 0)$$

$$\Pr(\hat{y}^{\max} = 1, \hat{y} = 0 | y = 1, z = 0) = \Pr(b'_0 < \mathbf{w}_\theta^\top \mathbf{x} < b_0 | y = 1, z = 0)$$

From (A.36), we have

$$\begin{aligned} \Pr(\hat{y}^{\max} = 1, \hat{y} = 0 | y = 0, z = 0) &= \Pr(b'_0 < \mathbf{w}_\theta^\top \mathbf{x} < b_0 | y = 0, z = 0) \\ &= Q\left(\frac{\mathbf{w}_\theta^\top \boldsymbol{\mu}_{0,0} - b_0}{\sqrt{\mathbf{w}_\theta^\top \boldsymbol{\Sigma}_{0,0} \mathbf{w}_\theta}}\right) - Q\left(\frac{\mathbf{w}_\theta^\top \boldsymbol{\mu}_{0,0} - b'_0}{\sqrt{\mathbf{w}_\theta^\top \boldsymbol{\Sigma}_{0,0} \mathbf{w}_\theta}}\right), \quad (\text{A.44}) \end{aligned}$$

$$\begin{aligned} \Pr(\hat{y}^{\max} = 1, \hat{y} = 0 | y = 1, z = 0) &= \Pr(b'_0 < \mathbf{w}_\theta^\top \mathbf{x} < b_0 | y = 1, z = 0) \\ &= Q\left(\frac{\mathbf{w}_\theta^\top \boldsymbol{\mu}_{1,0} - b_0}{\sqrt{\mathbf{w}_\theta^\top \boldsymbol{\Sigma}_{1,0} \mathbf{w}_\theta}}\right) - Q\left(\frac{\mathbf{w}_\theta^\top \boldsymbol{\mu}_{1,0} - b'_0}{\sqrt{\mathbf{w}_\theta^\top \boldsymbol{\Sigma}_{1,0} \mathbf{w}_\theta}}\right). \quad (\text{A.45}) \end{aligned}$$

Then,

$$\begin{aligned} \Pr(\hat{y}^{\max} = 1, \hat{y} = 0 | z = 0) &= \Pr(y = 0 | z = 0) \left[Q\left(\frac{\mathbf{w}_\theta^\top \boldsymbol{\mu}_{0,0} - b_0}{\sqrt{\mathbf{w}_\theta^\top \boldsymbol{\Sigma}_{0,0} \mathbf{w}_\theta}}\right) - Q\left(\frac{\mathbf{w}_\theta^\top \boldsymbol{\mu}_{0,0} - b'_0}{\sqrt{\mathbf{w}_\theta^\top \boldsymbol{\Sigma}_{0,0} \mathbf{w}_\theta}}\right) \right] \\ &\quad + \Pr(y = 1 | z = 0) \left[Q\left(\frac{\mathbf{w}_\theta^\top \boldsymbol{\mu}_{1,0} - b_0}{\sqrt{\mathbf{w}_\theta^\top \boldsymbol{\Sigma}_{1,0} \mathbf{w}_\theta}}\right) - Q\left(\frac{\mathbf{w}_\theta^\top \boldsymbol{\mu}_{1,0} - b'_0}{\sqrt{\mathbf{w}_\theta^\top \boldsymbol{\Sigma}_{1,0} \mathbf{w}_\theta}}\right) \right]. \quad (\text{A.46}) \end{aligned}$$

The corresponding expression for group $z = 1$ is derived similarly. Thus, the EI disparity can be expressed in terms of θ , b_0 , and b_1 .

Solve the Optimization Problem

In the previous two sections, we derived the error rate and EI disparity in terms of Q-functions containing parameters θ, b_0, b_1 . Therefore, we can construct an EI-constrained optimization problem (which is essentially same as (2.3)):

$$\begin{aligned} \min_{\theta, b_0, b_1} \quad & \Pr(\hat{y} \neq y) \\ \text{s.t.} \quad & \max_{i \in \{0,1\}} |\Pr(\hat{y}^{\max} = 1 \mid \hat{y} = 0, z = i) - \Pr(\hat{y}^{\max} = 1 \mid \hat{y} = 0)| < c. \end{aligned} \tag{A.47}$$

where c is a hyperparameter we can choose to balance error rate and EI disparity.

After writing error rate and EI disparity in terms of Q-functions (using the derivations in Section A.2.5 and Section A.2.5), we numerically solve the constrained optimization problems above with a popular python module `scipy.optimize`. To get the experimental results in Figure 2.3, we numerically solved the above problem for 20 different c values, where the maximum c is picked as the EI disparity of the unconstrained optimization problem.

A.3 Supplementary Materials on Experiments

In this section, we provide the details of the experiment setup and additional experimental results.

A.3.1 Synthetic dataset generation

We define y, z as $z \sim \text{Bern}(0.4)$, $(y \mid z = 0) \sim \text{Bern}(0.3)$, and $(y \mid z = 1) \sim \text{Bern}(0.5)$. The feature $\mathbf{x}$ follows the conditional distribution $(\mathbf{x} \mid y = y, z = z) \sim \mathcal{N}(\boldsymbol{\mu}_{y,z}, \boldsymbol{\Sigma}_{y,z})$

where the mean of each cluster is

$$\boldsymbol{\mu}_{0,0} = [-0.1, -0.2], \boldsymbol{\mu}_{0,1} = [-0.2, -0.3], \boldsymbol{\mu}_{1,0} = [0.1, 0.4], \boldsymbol{\mu}_{1,1} = [0.4, 0.3]$$

and the covariance matrix of each cluster is

$$\boldsymbol{\Sigma}_{0,0} = \begin{bmatrix} 0.4 & 0.0 \\ 0.0 & 0.4 \end{bmatrix}, \boldsymbol{\Sigma}_{1,0} = \boldsymbol{\Sigma}_{0,1} = \begin{bmatrix} 0.2 & 0.0 \\ 0.0 & 0.2 \end{bmatrix}, \boldsymbol{\Sigma}_{1,1} = \begin{bmatrix} 0.1 & 0.0 \\ 0.0 & 0.1 \end{bmatrix}.$$

A.3.2 Hyperparameter selection

In all our experiments, we perform cross-validation to select the learning rate from $\{0.0001, 0.001, 0.01, 0.1\}$. In addition, for each penalty term we did two-step cross-validation to choose λ . In the first step, we used a set $\lambda \in \{0, 0.2, 0.4, 0.6, 0.8, 0.9\}$. In the second step, we generate a second set around the best λ^* found in the first step, *i.e.*, the second set is $\{\max\{\lambda^* + \varepsilon, 0\} : \varepsilon \in \{-0.1, -0.05, 0, 0.05, 0.1\}\}$. For example, if $\lambda^* = 0.4$ is the best at the first step, then at the second step we use the set $\lambda \in \{0.3, 0.35, 0.4, 0.45, 0.5\}$.

A.3.3 Additional experimental results on algorithm evaluation

Table A.1 shows the performance of baseline and our three approaches (covariance-based, KDE-based, loss-based) introduced in Section 2.3, for a multi-layer perceptron (MLP) network having one hidden layer with four neurons. Similar to the result in Table 2.2 for the logistic regression model, our methods can reduce the EI disparity without losing the classification accuracy (*i.e.*, increasing the error rate) much.

Meanwhile, according to a previous work [140], large deep learning models are observed to overfit to fairness constraints during training and therefore produce unfair

Dataset	Metric	Methods			
		ERM	Covariance-based	KDE-based	Loss-based
Synthetic	Error Rate ($\downarrow$)	$.205 \pm .003$	$.242 \pm .006$	$.227 \pm .008$	$.229 \pm .012$
	EI Disp. ($\downarrow$)	$.141 \pm .036$	$.004 \pm .002$	$.011 \pm .006$	$.018 \pm .009$
German Stat.	Error Rate ($\downarrow$)	$.221 \pm .010$	$.299 \pm .012$	$.232 \pm .018$	$.238 \pm .035$
	EI Disp. ($\downarrow$)	$.059 \pm .045$	$.013 \pm .025$	$.041 \pm .025$	$.013 \pm .019$
ACSIncome-CA	Error Rate ($\downarrow$)	$.181 \pm .002$	$.202 \pm .002$	$.182 \pm .002$	$.185 \pm .001$
	EI Disp. ($\downarrow$)	$.042 \pm .002$	$.010 \pm .006$	$.010 \pm .002$	$.006 \pm .003$

Table A.1: Comparison of error rate and EI disparities of ERM baseline and proposed methods on the synthetic, German Statlog Credit, and ACSIncome-CA datasets using a Multi-Layer Perceptron (MLP). For each dataset, the lowest EI disparity value is in boldface. Compared with ERM, all three methods proposed in this chapter achieve much lower EI disparity without substantial loss in accuracy. All reported numbers are evaluated on the test set.

predictions on the test data. To confirm whether our method is also having such limitations, we investigate the performance of our algorithms on over-parameterized models. Specifically, we conduct experiments on a five-layer ReLU network with 200 hidden neurons per layer, which is over-parameterized for German dataset. Table A.2 reports the performance of EI-constrained classifiers on such over-parameterized setting. One can confirm that our methods (covariance-based, KDE-based, loss-based) perform well in both training and test dataset, and we do not observe the overfitting problem.

Dataset	Metric	Methods			
		ERM	Covariance-based	KDE-based	Loss-based
German Stat.	Train Err. ($\downarrow$)	$.117 \pm .004$	$.133 \pm .003$	$.125 \pm .008$	$.132 \pm .011$
	Test Err. ($\downarrow$)	$.117 \pm .010$	$.118 \pm .010$	$.121 \pm .010$	$.130 \pm .009$
	Train EI Disp. ($\downarrow$)	$.022 \pm .017$	$.018 \pm .011$	$.018 \pm .009$	$.015 \pm .013$
	Test EI Disp. ($\downarrow$)	$.060 \pm .032$	$.049 \pm .024$	$.057 \pm .028$	$.047 \pm .025$

Table A.2: Error rate and EI disparities for ERM baseline and proposed methods, for an over-parameterized neural network on the German Statlog Credit dataset. Performances on train/test datasets are presented. Note that we do not observe overfitting in the over-parameterized setting.

In addition, in Table A.3, we include ER and BE as baselines for the synthetic dataset experiment provided in Table 2.2. We leverage the algorithm suggested by [22] for mitigating ER disparity. We extend the loss-based approach to reduce BE disparity, by redefining the BE loss of group z as

$$\tilde{L}_z^{\text{BE}} \triangleq \frac{1}{\text{number of samples in group } z} \sum_{i \in I_{-,z}} \ell(1, \max_{\|\Delta \mathbf{x}_i\| \leq \delta} f(\mathbf{x}_i + \Delta \mathbf{x}_i)),$$

where

$$\tilde{L}_z^{\text{EI}} \triangleq \frac{1}{\text{number of rejected samples in group } z} \sum_{i \in I_{-,z}} \ell(1, \max_{\|\Delta \mathbf{x}_i\| \leq \delta} f(\mathbf{x}_i + \Delta \mathbf{x}_i)),$$

and $I_{-,z}$ is the set of rejected samples in group z for $z \in [Z]$.

Table A.3 shows that the minimum EI disparity is achieved by our methods. Hence, if the EI fairness needs to be achieved, then it cannot be replaced with the existing other fairness notions for some datasets.

Dataset	Metric	Methods					
		ERM	ER [22]	BE Loss-based	Covariance-based	KDE-based	Loss-based
Synthetic	Error Rate ($\downarrow$)	.221 $\pm$.001	.235 $\pm$.009	.252 $\pm$.006	.253 $\pm$.003	.250 $\pm$.001	.246 $\pm$.001
	EI Disp. ($\downarrow$)	.117 $\pm$.007	.036 $\pm$.018	.006 $\pm$.004	.003 $\pm$.001	.005 $\pm$.003	.002 $\pm$.001

Table A.3: Comparison of error rate and EI disparities of ERM, ER, BE baseline, and proposed methods on the synthetic dataset. The lowest EI disparity value is in boldface. The three EI-regularized approaches achieve the lowest EI disparity while maintaining low error rates. All reported numbers are evaluated on the test set.

A.3.4 Evaluating Robustness of EI

In this section, we highlight the advantages of EI over BE and ER in terms of robustness to outliers and imbalanced negative prediction rates. In doing so, we consider

certain data distributions and follow the method discussed in Section A.2.5 for solving the classifiers.

EI versus ER: robustness to outliers

As we discussed in Section A.2.3, ER is vulnerable to outliers. In this experiment, we systematically study the robustness of EI and ER to outliers.

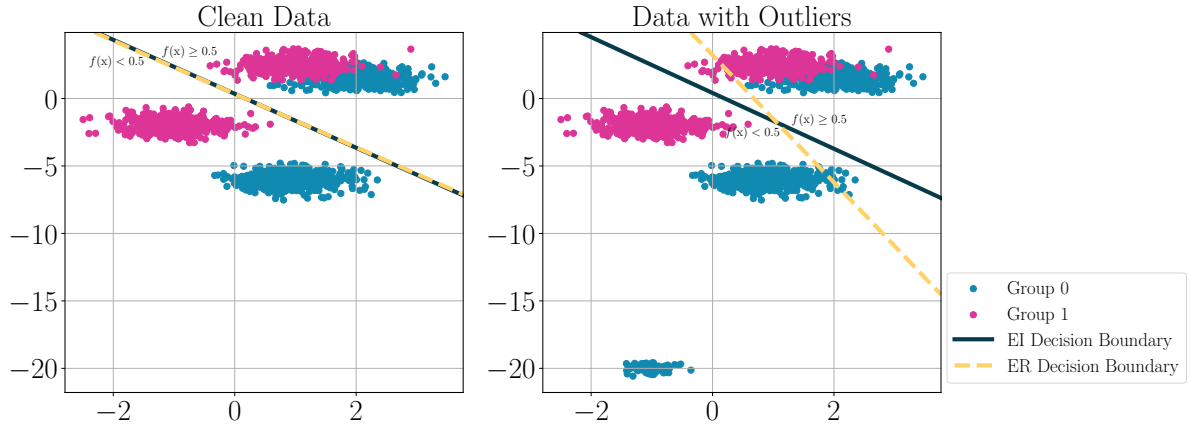


Figure A.1: Visualizations of the EI and ER decision boundaries without and with the presence of outliers. We observe that the decision boundary of ER changes a lot in the presence of outliers, while the decision boundary of EI is not affected. This phenomenon implies the robustness of EI to outliers.

Data distributions (Clean) Let sensitive attribute $z \sim \text{Bern}(0.5)$ and label $y \sim \text{Bern}(0.5)$ be independent of sensitive attribute z . Given the sensitive attribute z and label y , feature $\mathbf{x}$ follows the conditional distribution $\mathbf{x} \mid y = y, z = z \sim \mathcal{N}(\boldsymbol{\mu}_{y,z}, \boldsymbol{\Sigma}_{y,z})$, where the mean and covariance of the four Gaussian clusters are

$$\boldsymbol{\mu}_{0,0} = [1, -6], \boldsymbol{\mu}_{0,1} = [-1, -2], \boldsymbol{\mu}_{1,0} = [2, 1.5], \boldsymbol{\mu}_{1,1} = [1, 2.5],$$

and

$$\Sigma_{0,0} = \Sigma_{1,0} = \Sigma_{0,1} = \Sigma_{1,1} = \begin{bmatrix} 0.25 & 0.0 \\ 0.0 & 0.25 \end{bmatrix}.$$

(Contaminated) We contaminate the distribution by introducing additional 5% outliers to group $z = 0$. The outliers follow Gaussian distribution with mean and covariance matrix

$$\mu_{\text{outlier},0} = [-1, -20], \Sigma_{\text{outlier},0} = \begin{bmatrix} 0.05 & 0.0 \\ 0.0 & 0.05 \end{bmatrix}.$$

Results The decision boundaries of EI and ER for both the clean dataset and contaminated dataset are depicted in Figure A.1. These decision boundaries are the optimal linear decision boundaries based on the distributional information, we followed the same procedure as we mentioned in A.2.5. The δ for the EI classifier is picked as 1.5. We observe that the introduction of outliers makes the ER decision boundary rotate a lot, leading to a drop in classification accuracy and ER disparity *w.r.t.* the clean data distribution. Moreover, we note that the newly added outliers fail to destroy EI classifier, which implies the robustness of EI to outliers. Table A.4 also presents the accuracy, EI, and ER disparity of each classifier. These metrics are computed after excluding the outliers. It shows that the EI and ER disparities have higher values under the dataset with the outliers for the ER classifier.

EI versus BE: robustness to imbalanced group negative rate

In this experiment, we investigate the robustness of EI and BE to an imbalanced negative rate.

Data Distributions (Same Negative Rate) We consider the distribution with balanced subgroups. In other words, let sensitive attribute $z \sim \text{Bern}(0.5)$, and label $y \sim$

Dataset	Metric	Methods	
		EI Classifier	ER Classifier
Dataset without Outliers	Error Rate ($\downarrow$)	.001 $\pm$.001	.001 $\pm$.001
	EI Disp. ($\downarrow$)	.001 $\pm$.001	.005 $\pm$.001
	ER Disp. ($\downarrow$)	.004 $\pm$.001	.001 $\pm$.001
Dataset with Outliers	Error Rate ($\downarrow$)	.001 $\pm$.001	.020 $\pm$.001
	EI Disp. ($\downarrow$)	.017 $\pm$.001	.348 $\pm$.001
	ER Disp. ($\downarrow$)	.020 $\pm$.001	.291 $\pm$.001

Table A.4: Error rate and EI and ER disparities of linear EI and ER classifiers.

Bern(0.5) be independent of sensitive attribute z . Given the sensitive attribute z and label y , feature, feature $\mathbf{x}$ follows the Gaussian distribution ($\mathbf{x} \mid y = y, z = z$) $\sim \mathcal{N}(\boldsymbol{\mu}_{y,z}, \boldsymbol{\Sigma}_{y,z})$ where the mean of each cluster is

$$\boldsymbol{\mu}_{0,0} = [-2, -1], \boldsymbol{\mu}_{0,1} = [-1, -2], \boldsymbol{\mu}_{1,0} = [1, 2], \boldsymbol{\mu}_{1,1} = [2, 1]$$

and the covariance matrix of each cluster is

$$\boldsymbol{\Sigma}_{0,0} = \boldsymbol{\Sigma}_{1,0} = \boldsymbol{\Sigma}_{0,1} = \boldsymbol{\Sigma}_{1,1} = \begin{bmatrix} 0.25 & 0.0 \\ 0.0 & 0.25 \end{bmatrix}.$$

(Different Negative Rate) We manipulate the distribution of label y for constructing data distribution with different negative rates. To be more specific, we let $y \mid z = 0 \sim \text{Bern}(0.7)$, and $y \mid z = 1 \sim \text{Bern}(0.3)$.

Results Figure A.2 shows the decision boundaries of EI and BE when (i) the dataset has the same negative rate, and (ii) the dataset has a different negative rate. These decision boundaries are the optimal linear decision boundaries based on the distributional information, we followed the same procedure as we mentioned in A.2.5. The δ for the EI

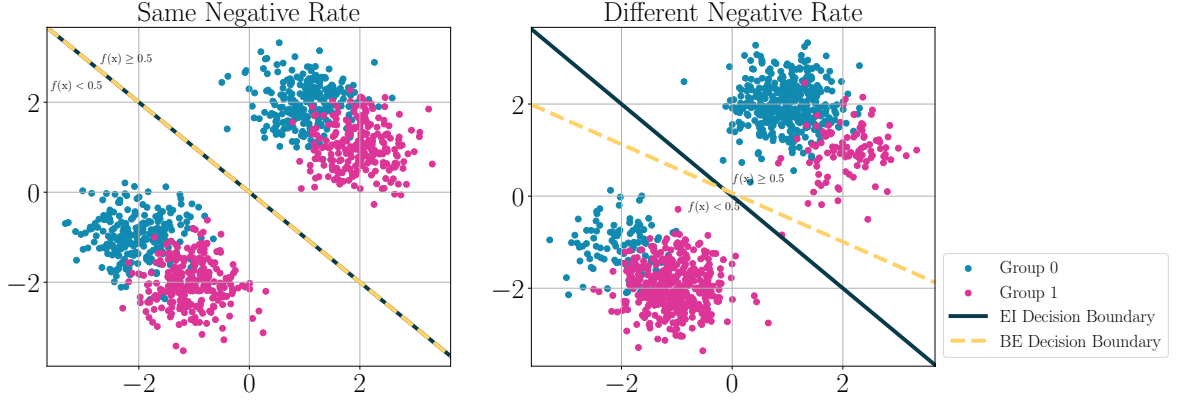


Figure A.2: Visualizations of the EI and BE decision boundaries given the data distribution with the same negative rates and different negative rates. The decision boundary of BE rotates a lot when the negative rate of the dataset becomes different, implying the sensitivity of BE to imbalanced negative rates. In contrast, the consistency of EI decision boundaries showcases the robustness of EI *w.r.t.* imbalanced negative rates.

and BE classifiers is picked as 1.5. The huge difference in BE decision boundaries under the two cases verifies our claim that BE cannot handle imbalanced negative prediction rates. In contrast, the decision boundaries of EI learned with two datasets with different group proportions are consistent.

A.3.5 Dynamic modeling

Here we justify the dynamic model we used in our experiments in Section 2.4.2, for updating individual's features and for updating the classifier. Specifically, Section A.3.5 explains the choice of $\epsilon(x)$, the amount of effort a sample (having feature x), while Section A.3.5 explains the choice of $(\tau_t^{(0)}, \tau_t^{(1)})$ defining the classifier at round t .

Updating individual's features (choice of $\epsilon(x)$)

At first, we designed a model by assuming that each individual makes an effort, where the amount of the effort he/she makes is (1) proportional to the reward (the improvement

on the outcome) it will get by making such effort, and (2) inversely proportional to the required amount of efforts to improve the outcome, as shown in the below equation:

$$\text{Realized effort } \epsilon(x) \triangleq \frac{\text{improvement on the outcome}}{\text{required efforts for improving the outcome}^2} \quad (\text{A.48})$$

$$= \frac{\mathbf{1}\{x < \tau_t^{(z)}\}}{(\tau_t^{(z)} - x)^2}, \quad (\text{A.49})$$

where $\tau_t^{(z)}$ is the accepting threshold for the individuals from group z at round t , and x is the feature. This equation can be interpreted as follows: an individual that is unqualified ($x < \tau_t^{(z)}$) is willing to make positive effort, but he/she is less motivated if the distance $\tau_t^{(z)} - x$ to the decision boundary is too large.

In order to upper bound the amount of reward, we added a small constant $\beta > 0$ in the numerator, thus having the final form:

$$\epsilon(x) = \frac{\mathbf{1}\{x < \tau_t^{(z)}\}}{(\tau_t^{(z)} - x + \beta)^2}. \quad (\text{A.50})$$

Updating the classifier

Note that at each round t , each individual's features are updated. Accordingly, we also update the classifier (having parameters $\tau_t^{(0)}$ and $\tau_t^{(1)}$) in the same round. We update the EI classifier based on the following constrained optimization problem:

$$\min (\text{EI Disparity}) \quad \text{s.t.} \quad \text{error rate} \leq \text{threshold}, \quad (\text{A.51})$$

which implies that we aim to find a classifier that guarantees good accuracy and small EI disparity. This optimization can be rewritten as

$$\begin{aligned}
 \min_{\tau_t^{(0)}, \tau_t^{(1)}} & \left[\overbrace{\mathbb{P}(\underbrace{x_t + \delta_t > \tau_t^{(0)}}_{\text{the outcome can be improved}} \mid z = 0, \underbrace{x_t < \tau_t^{(0)}}_{\text{rejected}}) - \mathbb{P}(x_t + \delta_t > \tau_t^{(1)} \mid z = 1, x_t < \tau_t^{(1)})}_{\text{EI Disparity}} \right] \\
 \text{s.t. } & \underbrace{\mathbb{P}(\hat{y}_t \neq y_t)}_{\text{error rate}} \leq c.
 \end{aligned} \tag{A.52}$$

Given the data distribution at each round t , we numerically solve the constrained optimization problems using a popular python module `scipy.optimize`.

A.3.6 Obtaining baseline classifiers for dynamic scenarios

Continued from Section 2.4.2, this section describes how we compute the classifiers that satisfy demographic parity (DP), bounded effort (BE) [23] equal recourse (ER) [22] and individual-level fairness causal recourse (ILFCR) [24], respectively. Similar to EI classifier, we obtain the best DP classifier by considering the following constrained optimization problem:

$$\min |\mathbb{P}(\hat{y}_t = 1 \mid z = 0) - \mathbb{P}(\hat{y}_t = 1 \mid z = 1)| \quad \text{s.t. } \mathbb{P}(\hat{y}_t \neq y_t) \leq c. \tag{A.53}$$

The best BE classifier can be obtained by solving the following problem:

$$\begin{aligned}
 \min_{\tau_t^{(0)}, \tau_t^{(1)}} & \left| \mathbb{P}(\tau_t^{(0)} - \delta_t < x_t < \tau_t^{(0)} \mid z = 0) - \mathbb{P}(\tau_t^{(1)} - \delta_t < x_t < \tau_t^{(1)} \mid z = 1) \right| \\
 \text{s.t. } & \mathbb{P}(\hat{y}_t \neq y_t) \leq c.
 \end{aligned} \tag{A.54}$$

Similarly, the optimization problem for obtaining the best ER classifier is written as:

$$\begin{aligned} \min_{\tau_t^{(0)}, \tau_t^{(1)}} \quad & \left| \mathbb{E} \left[\tau_t^{(0)} - \mathbf{x}_t \mid \mathbf{x}_t < \tau_t^{(0)}, z = 0 \right] - \mathbb{E} \left[\tau_t^{(1)} - \mathbf{x}_t \mid \mathbf{x}_t < \tau_t^{(1)}, z = 1 \right] \right| \\ \text{s.t.} \quad & \mathbb{P}(\hat{y}_t \neq y_t) \leq c. \end{aligned} \quad (\text{A.55})$$

While the computation of ILFCR classifier is less straightforward, we first describe the setting considered in [24]. This chapter assumes that each observed variable x_i is determined by (i) its direct causes (causal parents) which include the sensitive attribute z and other observed variables x_j , and (ii) an unobserved variable u_i . Since our dynamics experiment considers only one-dimensional variable x , it is determined by the sensitive attribute z and latent variable u . Therefore, we write x as a function of z and u , i.e., $x(z, u)$.

Before we describe how we compute ILFCR in our experiments, we introduce the definition of causal recourse and ILFCR in more detail. Given a model and a sample with feature x , the causal recourse $C(x)$ of the sample is the minimum cost of changing the features, in order to alter the decision of the model. Let $\tau^{(z)} \in \mathbb{R}$ be the acceptance threshold of group $z \in \{0, 1\}$ and $\mu(\Delta x) = |\Delta x|$ be the cost of improving the feature by Δx . Let x' denote the improved feature. In our one-dimensional dynamic setting, the causal recourse is defined as

$$C(x(z, u)) = \min_{x' \geq \tau^{(z)}} \mu(x' - x(z, u)) = \max(\tau^{(z)} - x(z, u), 0), \quad (\text{A.56})$$

for $z = 0, 1$. ILFCR requires different groups have the same causal recourse for all realizations of the latent variable u :

$$\max_u |C(x(0, u)) - C(x(1, u))| = 0. \quad (\text{A.57})$$

Now we describe how we compute ILFCR disparity. Recall that for dynamic experiments, we assumed the feature x follows a Gaussian distribution for each group $z \in \{0, 1\}$, i.e., $x \mid z \sim \mathcal{N}(\mu_z, \sigma_z^2)$. This can be represented using the notation in [24]: the latent variable is $u \sim \mathcal{N}(0, 1)$ and the feature is represented as $x(z, u) = \mu_z + \sigma_z u$. We measure the ILFCR disparity of a decision boundary pair $(\tau^{(0)}, \tau^{(1)})$ by

$$\begin{aligned}
 \text{ILFCR Disparity}(\tau^{(0)}, \tau^{(1)}) &= \max_u |C(x(0, u)) - C(x(1, u))| \\
 &= \max_u \left| \max(\tau^{(0)} - x(0, u), 0) - \max(\tau^{(1)} - x(1, u), 0) \right| \\
 &= \max_u \left| \max(\tau^{(0)} - \mu_0 - \sigma_0 u, 0) - \max(\tau^{(1)} - \mu_1 - \sigma_1 u, 0) \right|.
 \end{aligned} \tag{A.58}$$

For u sufficiently negative, the last expression grows as

$$|\tau^{(0)} - \tau^{(1)} - \mu_0 + \mu_1 - (\sigma_0 - \sigma_1)u|, \tag{A.59}$$

which diverges as $u \rightarrow -\infty$ when $\sigma_0 \neq \sigma_1$.

Note that the ILFCR disparity is infinite if we take the maximum over all u . Therefore, we instead ignore the tail distribution and focus on the samples with $x \in [\mu_z - 3\sigma_z, \mu_z + 3\sigma_z]$ for group $z = 0, 1$. In order to find a classifier with small ILFCR disparity, we numerically solve the following constrained optimization problem:

$$\begin{aligned}
 \min_{\tau^{(0)}, \tau^{(1)}} \quad & \text{ILFCR Disparity}(\tau^{(0)}, \tau^{(1)}) \\
 \text{s.t.} \quad & \text{error rate} \leq \alpha/2.
 \end{aligned} \tag{A.60}$$

Given the data distribution at each round t , we numerically solve all the constrained optimization problems above using the Python module `scipy.optimize`.

A.3.7 Performance comparison between fairness notions under other dynamic models

Here we provide one example in which EI improves long-term fairness, while some other methods (EO, BE, and ER) harm long-term fairness.

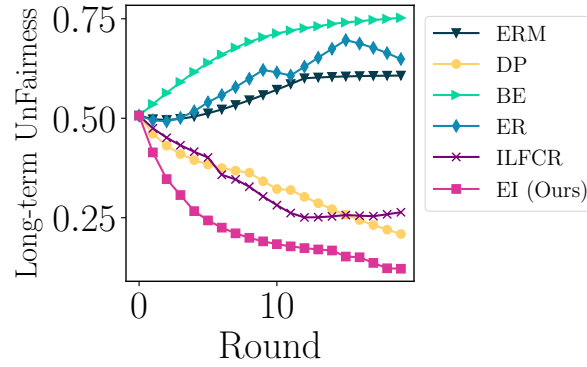


Figure A.3: Long-term unfairness $d_{TV}(\mathcal{P}_t^{(0)}, \mathcal{P}_t^{(1)})$ at each round t for various algorithms in the setup described in Section A.3.7.

In this example, we modified the effort function from $\epsilon(x) = \frac{1}{(\tau_t^{(z)} - x + \beta)^2} \mathbf{1}\{x < \tau_t^{(z)}\}$ to $\epsilon(x) = \log\left(\max\left(\frac{1}{(\tau_t^{(z)} - x + \beta)^2}, 1\right)\right) \mathbf{1}\{x < \tau_t^{(z)}\}$ so that the rejected individuals make less improvement than the setting in the original manuscript. Here $\tau_t^{(z)}$ is the acceptance threshold for group z at iteration t , and $\beta > 0$ is a small constant to avoid zero denominators. We set $\beta = 0.2$ and the true acceptance rate $\alpha = 0.5$. Figure A.3 reports how long-term unfairness changes when the initial distribution is $x \mid z = 0 \sim \mathcal{N}(0.0, 3.0^2)$ for group 0 and $x \mid z = 1 \sim \mathcal{N}(1.0, 1.0^2)$ for group 1.

A.4 Examples: Investigating the One-Step Impact of Fairness Notions

In this section, we conduct mathematical analysis on two examples to better understand how EI improves long-term fairness compared to existing fairness notions. We first introduce a basic setup for the mathematical analysis and then apply it to two examples to compare EI with BE, ERM, and ER in terms of their one-step impact on data distributions.

A.4.1 Basic setup and preliminaries

Classifiers

We consider the z -aware classifier, having

$$f(x) = \begin{cases} \mathbf{1}\{x \geq \tau_0\}, & z = 0, \\ \mathbf{1}\{x \geq \tau_1\}, & z = 1, \end{cases} \quad (\text{A.61})$$

which is parameterized by the threshold pair (τ_0, τ_1) . We assume the effort budget is $\delta = \frac{m}{2}$ where m is defined in the dataset.

Recall the zero equal improvability (EI) disparity condition is:

$$\mathbb{P} \left(\max_{\mu(\Delta x) \leq \delta} f(x + \Delta x) \geq 0.5 \mid f(x) < 0.5, z = 0 \right) = \mathbb{P} \left(\max_{\mu(\Delta x) \leq \delta} f(x + \Delta x) \geq 0.5 \mid f(x) < 0.5, z = 1 \right) \quad (\text{A.62})$$

where $\mu(\Delta x) = |\Delta x|$. This condition is equivalent to

$$\frac{\int_{\tau_0 - \delta}^{\tau_0} p_0(x) dx}{\int_{-\infty}^{\tau_0} p_0(x) dx} = \frac{\int_{\tau_1 - \delta}^{\tau_1} p_1(x) dx}{\int_{-\infty}^{\tau_1} p_1(x) dx}. \quad (\text{A.63})$$

We denote the improbability ratio of each group as

$$r_0(\tau_0) = \frac{\int_{\tau_0-\delta}^{\tau_0} p_0(x) dx}{\int_{-\infty}^{\tau_0} p_0(x) dx}, \quad (\text{A.64})$$

$$r_1(\tau_1) = \frac{\int_{\tau_1-\delta}^{\tau_1} p_1(x) dx}{\int_{-\infty}^{\tau_1} p_1(x) dx} \quad (\text{A.65})$$

The classifier that satisfies this zero EI disparity condition and minimizes the error rate is denoted as the optimal EI classifier.

Recall that the bounded effort (BE) fairness constraint is:

$$\mathbb{P} \left(\max_{\mu(\Delta x) \leq \delta} f(x + \Delta x) \geq 0.5, f(x) < 0.5 \mid z = 0 \right) = \mathbb{P} \left(\max_{\mu(\Delta x) \leq \delta} f(x + \Delta x) \geq 0.5, f(x) < 0.5 \mid z = 1 \right) \quad (\text{A.66})$$

where $\mu(\Delta x) = |\Delta x|$. Meanwhile, the Equal Recourse (ER) constraint is defined as

$$\mathbb{E} \left[\min_{f(x+\Delta x) \geq 0.5} \mu(\Delta x) \mid f(x) < 0.5, z = 0 \right] = \mathbb{E} \left[\min_{f(x+\Delta x) \geq 0.5} \mu(\Delta x) \mid f(x) < 0.5, z = 1 \right] \quad (\text{A.67})$$

Dynamic scenario

Suppose each rejected sample improves its feature by

$$\varepsilon(x) = \begin{cases} \delta \cdot \mathbf{1}\{x \in [\tau_0 - \delta, \tau_0)\}, & \text{if } z = 0, \\ \delta \cdot \mathbf{1}\{x \in [\tau_1 - \delta, \tau_1)\}, & \text{if } z = 1 \end{cases} \quad (\text{A.68})$$

Note that depending on the classifier we are using, the threshold pair (τ_0, τ_1) changes, thus the formulation for $\varepsilon(x)$ also changes. Here, we use $\varepsilon^{\text{ERM}}(x)$ to denote the improvement of features given ERM classifier, and similarly define $\varepsilon^{\text{EI}}(x)$, $\varepsilon^{\text{BE}}(x)$ and $\varepsilon^{\text{ER}}(x)$.

Let $p_z^{\text{ERM}}(x) = p_z(x + \varepsilon^{\text{ERM}}(x))$ for $z \in \{0, 1\}$, which represent the data distribution

after the features are improved based on ERM classifier. Similarly, we define $p_z^{\text{EI}}(x)$, $p_z^{\text{BE}}(x)$ and $p_z^{\text{ER}}(x)$ for EI/BE/ER classifiers, respectively. In the upcoming sections, we measure the total-variation (TV) distance

$$d_{TV}(p_0, p_1) = \frac{1}{2} \int_{\mathbb{R}} |p_0(x) - p_1(x)| dx \quad (\text{A.69})$$

between two groups after a single step of feature improvement, and provide how this measurement differs for various classifiers.

A.4.2 EI versus BE & ERM

Finding the optimal classifier for each fairness criterion

Setup Let $p_z(x)$ be the data distribution of each group $z \in \{0, 1\}$, shown in Figure A.4. We consider the case of each sample having one feature x , and the label is assigned as

$$y = \begin{cases} \mathbf{1}\{x \geq m/2\}, & \text{if } z = 0 \\ \mathbf{1}\{x \geq 0\}, & \text{if } z = 1 \end{cases} \quad (\text{A.70})$$

Note that we have $\mathbb{P}(y = 0|z = 0) = \frac{3}{4}$, $\mathbb{P}(y = 1|z = 0) = \frac{1}{4}$, $\mathbb{P}(y = 0|z = 1) = \frac{1}{2}$, and $\mathbb{P}(y = 1|z = 1) = \frac{1}{2}$. We set $\mathbb{P}(z = 0) = \frac{1}{4}$ and $\mathbb{P}(z = 1) = \frac{3}{4}$.

ERM classifier We have

$$(\tau_0^{\text{ERM}}, \tau_1^{\text{ERM}}) = \left(\frac{m}{2}, 0\right) \quad (\text{A.71})$$

since this threshold pair has zero classification error.

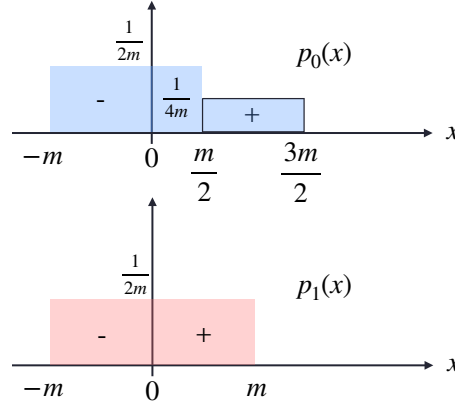


Figure A.4: The data distribution $p_z(x)$ for each group $z \in \{0, 1\}$. Samples with (+) sign have the true label $y = 1$, while samples with (-) sign have the true label $y = 0$.

BE classifier We have

$$(\tau_0^{\text{BE}}, \tau_1^{\text{BE}}) = \left(\frac{m}{2}, 0\right) \quad (\text{A.72})$$

since this threshold pair has zero classification error and satisfy the BE condition given in (A.66).

EI classifier The optimal EI classifier for dataset in Figure A.4 is

$$(\tau_0^{\text{EI}}, \tau_1^{\text{EI}}) = (0, 0). \quad (\text{A.73})$$

Proof. Note that $(\tau_0^{\text{EI}}, \tau_1^{\text{EI}}) = (0, 0)$ has error rate $\mathbb{P}(\text{error}) = \mathbb{P}(z = 0)\mathbb{P}(\text{error}|z = 0) + \mathbb{P}(z = 1)\mathbb{P}(\text{error}|z = 1) = \frac{1}{4} \cdot \frac{1}{4} + \frac{3}{4} \cdot 0 = \frac{1}{16}$. We prove that no other classifier satisfying EI condition in (A.63) is having error rate less than $\frac{1}{16}$. Note that when $|\tau_1| > \frac{m}{6}$, the error rate is larger than $\frac{1}{16}$. Thus it is sufficient to consider cases when $|\tau_1| \leq \frac{m}{6}$. Combining this with the fact that

- $r_0(\tau_0)$ in (A.64) and $r_1(\tau_1)$ in (A.65) are monotonically decreasing,
- EI condition in (A.63) is satisfied when $r_0(\tau_0) = r_1(\tau_1)$ holds,

- $r_0(\tau_0) = r_1(\tau_1)$ holds for $\tau_0 = \tau_1 \leq \frac{m}{2}$,

we can see that the optimal EI classifier satisfies $\tau_0 = \tau_1$ and $|\tau_1| \leq \frac{m}{6}$. The error rate for these classifiers is represented as $\mathbb{P}(\text{error}) = \mathbb{P}(z = 0)\mathbb{P}(\text{error}|z = 0) + \mathbb{P}(z = 1)\mathbb{P}(\text{error}|z = 1) = \frac{1}{4} \cdot (\frac{m}{2} - \tau_0) \cdot \frac{1}{2m} + \frac{3}{4} \cdot |\tau_1| \cdot \frac{1}{2m}$. Plugging in $\tau_0 = \tau_1$ and optimizing the error probability over $|\tau_1| \leq \frac{m}{6}$ completes the proof. ■

Total-variation distance between two groups

For the dataset given in Figure A.4, the total-variation distance between two groups for each classifier is:

$$d_{TV}(p_0^{\text{ERM}}, p_1^{\text{ERM}}) = 0.5, \quad (\text{A.74})$$

$$d_{TV}(p_0^{\text{BE}}, p_1^{\text{BE}}) = 0.5, \quad (\text{A.75})$$

$$d_{TV}(p_0^{\text{EI}}, p_1^{\text{EI}}) = 0.125. \quad (\text{A.76})$$

Proof. Since ERM solution is identical to BE solution, proving the above equation for BE and EI is sufficient. Recall that the expression of BE/EI classifiers are in (A.72) and (A.73). Using this expression, we can derive the distribution of each group:

$$p_0^{\text{BE}}(x) = \begin{cases} \frac{3}{4m}, & \text{if } x \in [\frac{m}{2}, m] \\ \frac{1}{2m}, & \text{if } x \in [-m, 0] \\ \frac{1}{4m}, & \text{if } x \in [m, \frac{3m}{2}] \\ 0, & \text{if } x \in [0, \frac{m}{2}] \text{ or } x \notin [-m, \frac{3m}{2}] \end{cases}, \quad (\text{A.77})$$

$$p_1^{\text{BE}}(x) = \begin{cases} \frac{1}{m}, & \text{if } x \in [0, \frac{m}{2}] \\ \frac{1}{2m}, & \text{if } x \in [-m, -\frac{m}{2}] \cup [\frac{m}{2}, m] \\ 0, & \text{if } x \in [-\frac{m}{2}, 0] \text{ or } x \notin [-m, m] \end{cases}, \quad (\text{A.78})$$

$$p_0^{\text{EI}}(x) = \begin{cases} \frac{1}{m}, & \text{if } x \in [0, \frac{m}{2}] \\ \frac{1}{2m}, & \text{if } x \in [-m, -\frac{m}{2}] \\ \frac{1}{4m}, & \text{if } x \in [\frac{m}{2}, \frac{3m}{2}] \\ 0, & \text{if } x \in [-\frac{m}{2}, 0] \text{ or } x \notin [-m, \frac{3m}{2}] \end{cases}, \quad (\text{A.79})$$

$$p_1^{\text{EI}}(x) = p_1^{\text{BE}}(x) \quad \forall x \quad (\text{A.80})$$

From this expression, we can derive the total-variation distance, which completes the proof. ■

A.4.3 EI versus ER

Finding the optimal classifier for each fairness criterion

Setup Let $p_z(x)$ be the data distribution of each group $z \in \{0, 1\}$, show in Figure A.5.

We consider the case of each sample having one feature x , and the label is assigned as

$$y = \begin{cases} \mathbf{1}\{x \geq 0\}, & \text{if } z = 0 \\ \mathbf{1}\{x \geq 0\}, & \text{if } z = 1 \end{cases}. \quad (\text{A.81})$$

Note that we have $\mathbb{P}(y = 0 \mid z = 0) = \mathbb{P}(y = 0 \mid z = 1) = \mathbb{P}(y = 1 \mid z = 0) = \mathbb{P}(y = 1 \mid z = 1) = \frac{1}{2}$. We set $\mathbb{P}(z = 0) = \mathbb{P}(z = 1) = \frac{1}{2}$.

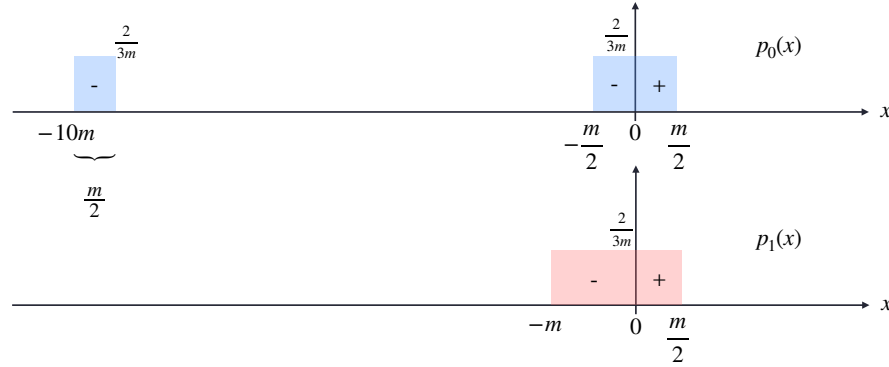


Figure A.5: The data distribution $p_z(x)$ for each group $z \in \{0, 1\}$. Samples with (+) sign have the true label $y = 1$, while samples with (-) sign have the true label $y = 0$.

ER classifier The optimal ER classifier for the dataset in Figure A.5 is

$$(\tau_0^{\text{ER}}, \tau_1^{\text{ER}}) = (-9m, 0). \quad (\text{A.82})$$

Proof. Note that the accepting thresholds $(-9m, 0)$ has error rate $\mathbb{P}(\text{error}) = \mathbb{P}(\text{error} \mid z = 0)\mathbb{P}(z = 0) + \mathbb{P}(\text{error} \mid z = 1)\mathbb{P}(z = 1) = \frac{1}{3} \cdot \frac{1}{2} + 0 \cdot \frac{1}{2} = \frac{1}{6}$. We prove that no other classifier satisfying ER constraint (A.67) is having error rate less than $\frac{1}{6}$.

The necessary condition for the classifier to have an error rate less than $\frac{1}{6}$ is $\tau_0, \tau_1 \in (-\frac{m}{2}, \frac{m}{2})$. Note that when $\tau_0 \in (-\frac{m}{2}, \frac{m}{2})$, we have the recourse of group 0:

$$\begin{aligned} \mathbb{E} \left[\min_{f(x+\Delta x) \geq 0.5} \mu(\Delta x) \mid f(x) < 0.5, z = 0 \right] &= \mathbb{E} [\tau_0 - x \mid x < \tau_0, z = 0] \\ &= \tau_0 - \underbrace{\mathbb{E} [x \mid x < \tau_0, z = 0]}_{\varphi(\tau_0)}. \end{aligned} \quad (\text{A.83})$$

where

$$\begin{aligned}
\varphi(\tau_0) &= \mathbb{E}[x \mid x < \tau_0, z = 0] \\
&= \underbrace{\mathbb{P}\left(x \in \left[-10m, -\frac{19}{2}m\right] \mid x < \tau_0, z = 0\right)}_{\triangleq \lambda} \mathbb{E}\left[x \mid x \in \left[-10m, -\frac{19}{2}m\right], z = 0\right] \\
&\quad + \underbrace{\mathbb{P}\left(x \in \left[-\frac{m}{2}, \tau_0\right] \mid x < \tau_0, z = 0\right)}_{1-\lambda} \mathbb{E}\left[x \mid x \in \left[-\frac{m}{2}, \tau_0\right], z = 0\right].
\end{aligned} \tag{A.84}$$

for all $\tau_0 \in [-\frac{m}{2}, \frac{m}{2}]$, where $\lambda \in [\frac{1}{3}, 1]$. Since

$$\mathbb{E}\left[x \mid x \in \left[-10m, -\frac{19}{2}m\right], z = 0\right] < \mathbb{E}\left[x \mid x \in \left[-\frac{m}{2}, \tau_0\right], z = 0\right],$$

$\varphi(\tau_0)$ is a decreasing function. Consequently,

$$\begin{aligned}
\varphi(\tau_0) &\leq \frac{1}{3} \mathbb{E}\left[x \mid x \in \left[-10m, -\frac{19}{2}m\right], z = 0\right] + \frac{2}{3} \mathbb{E}\left[x \mid x \in \left[-\frac{m}{2}, \tau_0\right], z = 0\right] \\
&= -\frac{39}{12}m + \frac{\tau_0}{3} - \frac{m}{6} = \frac{\tau_0}{3} - \frac{41}{12}m.
\end{aligned} \tag{A.85}$$

Thus, the recourse of group 0 is

$$(A.83) \geq \frac{2\tau_0}{3} + \frac{41}{12}m \geq \frac{37}{12}m. \tag{A.86}$$

Meanwhile, when $\tau_1 \in (-\frac{m}{2}, \frac{m}{2})$, we have the recourse of group 1:

$$\begin{aligned}
\mathbb{E} \left[\min_{f(x+\Delta x) \geq 0.5} \mu(\Delta x) \mid f(x) < 0.5, z = 1 \right] &= \mathbb{E} [\tau_1 - x \mid x < \tau_1, z = 1] \\
&= \tau_1 - \mathbb{E} [x \mid x < \tau_1, z = 1] \\
&= \tau_1 - \frac{\tau_1 - m}{2} \\
&= \frac{\tau_1 + m}{2} \in \left(\frac{m}{4}, \frac{3}{4}m \right). \tag{A.87}
\end{aligned}$$

Therefore, combining (A.86) and (A.87) implies that when $\tau_0, \tau_1 \in (-\frac{m}{2}, \frac{m}{2})$, the ER constraint (A.67) cannot be satisfied. Thus, the way for achieving error rate $< \frac{1}{6}$ is $\tau_0 < -\frac{m}{2}$ and $\tau_1 = 0$. Note that the recourse of group 0 which is written in (A.83) is a strictly increasing function when $\tau_0 < -\frac{m}{2}$. Therefore, there exists a unique classifier that achieves both EI while maintaining error rate $< \frac{1}{6}$. One can easily verify that $(-9m, 0)$ is the optimal ER classifier. ■

EI classifier We have

$$(\tau_0^{\text{EI}}, \tau_1^{\text{EI}}) = (0, 0) \tag{A.88}$$

since this threshold pair has zero classification error and satisfies the EI constraint (A.63).

Total-variation distance between two groups

For the dataset given in Figure A.5, the total variation distance between two groups for EI and ER classifier is

$$d_{TV}(p_0^{\text{ER}}, p_1^{\text{ER}}) = \frac{2}{3}, \tag{A.89}$$

$$d_{TV}(p_0^{\text{EI}}, p_1^{\text{EI}}) = \frac{1}{3}. \tag{A.90}$$

Proof. By (A.88) and (A.82),

$$p_0^{\text{ER}}(x) = \begin{cases} \frac{2}{3m}, & \text{if } x \in [-\frac{19m}{2}, -8m] \\ \frac{2}{3m}, & \text{if } x \in [-\frac{m}{2}, \frac{m}{2}] \\ 0, & \text{o.w.} \end{cases}, \quad (\text{A.91})$$

$$p_1^{\text{ER}}(x) = \begin{cases} \frac{2}{3m}, & \text{if } x \in [-m, -\frac{m}{2}] \\ \frac{4}{3m}, & \text{if } x \in [0, \frac{m}{2}] \\ 0, & \text{o.w.} \end{cases}, \quad (\text{A.92})$$

$$p_0^{\text{EI}}(x) = \begin{cases} \frac{2}{3m}, & \text{if } x \in [-10m, -\frac{19m}{2}] \\ \frac{4}{3m}, & \text{if } x \in [0, \frac{m}{2}] \\ 0, & \text{o.w.} \end{cases}, \quad (\text{A.93})$$

$$p_1^{\text{EI}}(x) = \begin{cases} \frac{2}{3m}, & \text{if } x \in [-m, -\frac{m}{2}] \\ \frac{4}{3m}, & \text{if } x \in [0, \frac{m}{2}] \\ 0, & \text{o.w.} \end{cases}. \quad (\text{A.94})$$

$$(\text{A.95})$$

For this expression, we can derive the total-variation distance, which completes it. ■

Appendix B

Supplementary Materials for Chapter 3

B.1 Score-Based Reward Optimization

We provide below a detailed proof of Theorem 3.3 for the optimal action that maximizes the score-based reward of each institution.

Proof. Assuming the score distribution's CDF is denoted by $\mathcal{F}$. By Lemma B.1, the score-based reward function of institution k is concave and is maximized when the absolute difference between the lower thresholds of both groups, $|t_0^{k,S,\text{low}} - t_1^{k,S,\text{low}}|$, is minimized. For each $k \in \{2, \dots, K\}$, this is equivalent to the following minimization:

$$\arg \min_{a_t^k \in \mathcal{A}_t^k} \left| \mathcal{F}^{-1} \left(1 - \frac{\bar{c}_t^{0,k} + a_t^k c_k}{s_t} \right) - \mathcal{F}^{-1} \left(1 - \frac{\bar{c}_t^{1,k} + (1 - a_t^k) c_k}{1 - s_t} \right) \right|, \quad (\text{B.1})$$

where $\bar{c}_t^{0,k} = \sum_{j=1}^{k-1} \pi_t^j c_j$ and $\bar{c}_t^{1,k} = \sum_{j=1}^{k-1} (1 - \pi_t^j) c_j$. Due to the monotonicity of the inverse CDF, the optimal action that maximizes the score-based reward is equivalent to

minimizing the following:

$$\arg \min_{a_t^k \in \mathcal{A}_t^k} \left| \left(1 - \frac{\bar{c}_t^{0,k} + a_t^k c_k}{s_t} \right) - \left(1 - \frac{\bar{c}_t^{1,k} + (1 - a_t^k) c_k}{1 - s_t} \right) \right|. \quad (\text{B.2})$$

This can be simplified to the following:

$$\arg \min_{a_t^k \in \mathcal{A}_t^k} \left| \frac{\bar{c}_t^{0,k} + a_t^k c_k}{s_t} - \frac{\bar{c}_t^{1,k} + (1 - a_t^k) c_k}{1 - s_t} \right|. \quad (\text{B.3})$$

The argument of the minimum can be multiplied by $s_t(1 - s_t)$ to simplify the expression as follows:

$$\arg \min_{a_t^k \in \mathcal{A}_t^k} \left| a_t^k c_k (1 - s_t) - (1 - a_t^k) c_k s_t - \sum_{j=1}^{k-1} (1 - \pi_t^j) c_j s_t + \sum_{j=1}^{k-1} \pi_t^j c_j (1 - s_t) \right|. \quad (\text{B.4})$$

The terms involving the action a_t^k and the terms independent of the action a_t^k can be grouped as follows:

$$\arg \min_{a_t^k \in \mathcal{A}_t^k} \left| a_t^k - s_t - \frac{1}{c_k} \left(\sum_{j=1}^{k-1} c_j (s_t - \pi_t^j) \right) \right| = \left[s_t + \frac{1}{c_k} \left(\sum_{j=1}^{k-1} c_j (s_t - \pi_t^j) \right) \right]_{\mathcal{A}_t^k}. \quad (\text{B.5})$$

Therefore, the optimal action that maximizes the score-based reward is as stated in the theorem. It can also be seen from Lemma B.1 that the score-based reward function of institution 1 is maximized when the lower thresholds of both groups are equal to each other, $t_0^{1,\text{S},\text{low}} = t_1^{1,\text{S},\text{low}}$. This is satisfied when $a_t^1 = s_t$. ■

B.1.1 Technical Lemmas

In this section, we show that the score-based reward function of each institution is concave and has a unique action that maximizes its reward.

Lemma B.1 *Under Assumption 3.1, the score-based reward function, $R_k(s_t, a_t)$, is concave and it is maximized when the absolute difference between the lower thresholds of both groups is minimized, $|t_0^{k,S,low} - t_1^{k,S,low}|$. This is equivalent to the following minimizing;*

$$\min_{a_t^k \in \mathcal{A}_t^k} \left| \mathcal{F}_0^{-1} \left(1 - \frac{\bar{c}_t^{0,k} + a_t c_k}{s_t} \right) - \mathcal{F}_1^{-1} \left(1 - \frac{\bar{c}_t^{1,k} + (1 - a_t) c_k}{1 - s_t} \right) \right|, \forall k \in \{2, \dots, K\} \quad (\text{B.6})$$

where $\bar{c}_t^{0,k} = \sum_{j=1}^{k-1} \pi_t^j c_j$ and $\bar{c}_t^{1,k} = \sum_{j=1}^{k-1} (1 - \pi_t^j) c_j$ and for $k = 1$, the action maximizing the score-based reward is equivalent to minimizing $\left| \mathcal{F}_0^{-1} \left(1 - \frac{a_t c_1}{s_t} \right) - \mathcal{F}_1^{-1} \left(1 - \frac{(1 - a_t) c_1}{1 - s_t} \right) \right|$.

Proof. Dropping the superscript k in the notation of the action for brevity, the score-based reward of the k^{th} institution is:

$$R_k(s_t, a_t) = a_t \frac{\mathbb{E} \left[\sum_{i=m_{k,t}^0+1}^{m_{k,t}^0+a_t A_{k,t}} X_{(i)}^0 \right]}{a_t A_{k,t}} + (1 - a_t) \frac{\mathbb{E} \left[\sum_{i=m_{k,t}^1+1}^{m_{k,t}^1+(1-a_t) A_{k,t}} X_{(i)}^1 \right]}{(1 - a_t) A_{k,t}} \quad (\text{B.7})$$

The number of admitted applicants is approximated as $A_{k,t}^0 = \lfloor a_t A_{k,t} \rfloor \approx a_t A_{k,t}$, as we are considering that the number of applicants is large. Under this regime, the collection of scores is in accordance with their respective group-specific score distributions $\mathcal{P}^g$. Following the idea in [72], the average scores of selected applicants from each group can be represented by the following conditional expectations:

$$\lim_{N \rightarrow \infty} \frac{\sum_{i=m_{k,t}^0+1}^{m_{k,t}^0+a_t A_{k,t}} X_{(i)}^0}{a_t A_{k,t}} = \mathbb{E} \left[X^0 \mid t_0^{k,S,low} \leq X^0 \leq t_0^{k,S,up} \right] \quad (\text{B.8})$$

$$\lim_{N \rightarrow \infty} \frac{\sum_{i=m_{k,t}^1+1}^{m_{k,t}^1+(1-a_t) A_{k,t}} X_{(i)}^1}{(1 - a_t) A_{k,t}} = \mathbb{E} \left[X^1 \mid t_1^{k,S,low} \leq X^1 \leq t_1^{k,S,up} \right] \quad (\text{B.9})$$

Then, the score-based reward function of the institution k can be written as follows:

$$R_g^k(s_t, a_t) = a_t \mathbb{E} \left[X^0 \mid t_0^{k,S,\text{low}} \leq X^0 \leq t_0^{k,S,\text{up}} \right] + (1 - a_t) \mathbb{E} \left[X^1 \mid t_1^{k,S,\text{low}} \leq X^1 \leq t_1^{k,S,\text{up}} \right] \quad (\text{B.10})$$

We denote the cumulative distribution function (CDF) of the score-distributions by $\mathcal{F}_0$ and $\mathcal{F}_1$. The tail of the distributions beyond the upper thresholds represents the proportion of the applicants from the particular group who have been already admitted by the better-ranked institutions.

$$1 - \mathcal{F}_0(t_0^{k,S,\text{up}}) = \frac{\sum_{j=1}^{k-1} c_j \pi_t^j}{s_t} \implies t_0^{k,S,\text{up}} = \mathcal{F}_0^{-1} \left(1 - \frac{\bar{c}_t^{0,k}}{s_t} \right) \quad (\text{B.11})$$

$$1 - \mathcal{F}_1(t_1^{k,S,\text{up}}) = \frac{\sum_{j=1}^{k-1} c_j (1 - \pi_t^j)}{(1 - s_t)} \implies t_1^{k,S,\text{up}} = \mathcal{F}_1^{-1} \left(1 - \frac{\bar{c}_t^{1,k}}{1 - s_t} \right) \quad (\text{B.12})$$

Moreover, the area between the lower and upper thresholds in the distributions signifies the proportion of applicants admitted by the specific group from the total number of applicants belonging to that group at the k^{th} institution. Thus, we have:

$$\mathcal{F}_0(t_0^{k,S,\text{up}}) - \mathcal{F}_0(t_0^{k,S,\text{low}}) = \frac{a_t c_k}{s_t} \quad \text{and} \quad \mathcal{F}_1(t_1^{k,S,\text{up}}) - \mathcal{F}_1(t_1^{k,S,\text{low}}) = \frac{(1 - a_t) c_k}{1 - s_t} \quad (\text{B.13})$$

Then using the upper thresholds, we can write the lower thresholds as follows:

$$1 - \mathcal{F}_0(t_0^{k,S,\text{low}}) = \frac{\bar{c}_t^{0,k}}{s_t} + \frac{a_t c_k}{s_t} \quad \text{and} \quad 1 - \mathcal{F}_1(t_1^{k,S,\text{low}}) = \frac{\bar{c}_t^{1,k}}{1 - s_t} + \frac{(1 - a_t) c_k}{1 - s_t} \quad (\text{B.14})$$

Then, we can write the lower thresholds as follows:

$$t_0^{k,S,\text{low}} = \mathcal{F}_0^{-1} \left(1 - \frac{\bar{c}_t^{0,k} + a_t c_k}{s_t} \right) \quad \text{and} \quad t_1^{k,S,\text{low}} = \mathcal{F}_1^{-1} \left(1 - \frac{\bar{c}_t^{1,k} + (1 - a_t) c_k}{1 - s_t} \right) \quad (\text{B.15})$$

We can write the score reward function of the institution k as follows:

$$R_g^k(s_t, a_t) = a_t \int_{t_0^{k,S,\text{low}}}^{t_0^{k,S,\text{up}}} x_0 \frac{f_0(x_0)}{\int_{t_0^{k,S,\text{low}}}^{t_0^{k,S,\text{up}}} f_0(x_0) dx_0} dx_0 + (1 - a_t) \int_{t_1^{k,S,\text{low}}}^{t_1^{k,S,\text{up}}} x_1 \frac{f_1(x_1)}{\int_{t_1^{k,S,\text{low}}}^{t_1^{k,S,\text{up}}} f_1(x_1) dx_1} dx_1 \quad (\text{B.16})$$

We can move the common denominator out of the integral since it is a constant:

$$R_g^k(s_t, a_t) = \frac{a_t}{\int_{t_0^{k,S,\text{low}}}^{t_0^{k,S,\text{up}}} f_0(x_0) dx_0} \int_{t_0^{k,S,\text{low}}}^{t_0^{k,S,\text{up}}} x_0 f_0(x_0) dx_0 + \frac{(1 - a_t)}{\int_{t_1^{k,S,\text{low}}}^{t_1^{k,S,\text{up}}} f_1(x_1) dx_1} \int_{t_1^{k,S,\text{low}}}^{t_1^{k,S,\text{up}}} x_1 f_1(x_1) dx_1 \quad (\text{B.17})$$

Then, the denominator can be written in terms of the CDF as follows:

$$R_g^k(s_t, a_t) = \frac{a_t}{\mathcal{F}_0(t_0^{k,S,\text{up}}) - \mathcal{F}_0(t_0^{k,S,\text{low}})} \int_{t_0^{k,S,\text{low}}}^{t_0^{k,S,\text{up}}} x_0 f_0(x_0) dx_0 + \frac{(1 - a_t)}{\mathcal{F}_1(t_1^{k,S,\text{up}}) - \mathcal{F}_1(t_1^{k,S,\text{low}})} \int_{t_1^{k,S,\text{low}}}^{t_1^{k,S,\text{up}}} x_1 f_1(x_1) dx_1 \quad (\text{B.18})$$

Using the equations (B.13), we can write the score-based reward function of the institution k as follows:

$$R_g^k(s_t, a_t) = \frac{s_t}{c_k} \int_{t_0^{k,S,\text{low}}}^{t_0^{k,S,\text{up}}} x_0 f_0(x_0) dx_0 + \frac{(1 - s_t)}{c_k} \int_{t_1^{k,S,\text{low}}}^{t_1^{k,S,\text{up}}} x_1 f_1(x_1) dx_1 \quad (\text{B.19})$$

The upper thresholds are not dependent on the current institution's action. Then, we can use the definitions of lower thresholds from equations (B.15) to write the score-based reward function of the institution k as follows:

$$R_g^k(s_t, a_t) = \frac{s_t}{c_k} \int_{\mathcal{F}_0^{-1}\left(1 - \frac{\bar{c}_t^{0,k} + a_t c_k}{s_t}\right)}^{t_0^{k,S,\text{up}}} x_0 f_0(x_0) dx_0 + \frac{(1 - s_t)}{c_k} \int_{\mathcal{F}_1^{-1}\left(1 - \frac{\bar{c}_t^{1,k} + (1 - a_t) c_k}{1 - s_t}\right)}^{t_1^{k,S,\text{up}}} x_1 f_1(x_1) dx_1 \quad (\text{B.20})$$

Using the fundamental theorem of calculus, we can write the derivative of the score-based

reward function of the institution k as follows:

$$\frac{dR_g^k(s_t, a_t)}{da_t} = \mathcal{F}_0^{-1} \left(1 - \frac{\bar{c}_t^{0,k} + a_t c_k}{s_t} \right) - \mathcal{F}_1^{-1} \left(1 - \frac{\bar{c}_t^{1,k} + (1 - a_t) c_k}{1 - s_t} \right) = t_0^{k,S,\text{low}} - t_1^{k,S,\text{low}} \quad (\text{B.21})$$

Then, we can write the second derivative as follows:

$$\frac{d^2 R_g^k(s_t, a_t)}{da_t^2} = -\frac{c_k}{s_t} \frac{1}{f_0 \left(\mathcal{F}_0^{-1} \left(1 - \frac{\bar{c}_t^{0,k} + a_t c_k}{s_t} \right) \right)} - \frac{c_k}{(1 - s_t)} \frac{1}{f_1 \left(\mathcal{F}_1^{-1} \left(1 - \frac{\bar{c}_t^{1,k} + (1 - a_t) c_k}{1 - s_t} \right) \right)} \quad (\text{B.22})$$

Since the PDF functions are non-negative, $c_k \in (0, 1]$, and $s_t \in (0, 1)$, the second derivative is non-positive. This means the score-based reward function of the institution k is concave.

The score-based reward function of the institution k is concave and its first derivative is monotone and defined on the interval $[\max(0, 1 - \frac{1 - s_t - \bar{c}_t^{1,k}}{c_k}), \min(1, \frac{s_t - \bar{c}_t^{0,k}}{c_k})]$. Then, the score-based reward function of the institution k is maximized when the absolute difference between the lower thresholds of both groups is minimized, $|t_0^{k,S,\text{low}} - t_1^{k,S,\text{low}}|$. This is equivalent to the following minimization:

$$\min_{a_t^k \in \mathcal{A}_t^k} \left| \mathcal{F}_0^{-1} \left(1 - \frac{\bar{c}_t^{0,k} + a_t c_k}{s_t} \right) - \mathcal{F}_1^{-1} \left(1 - \frac{\bar{c}_t^{1,k} + (1 - a_t) c_k}{1 - s_t} \right) \right| \quad (\text{B.23})$$

where $\bar{c}_t^{0,k} = \sum_{j=1}^{k-1} \pi_t^j c_j$ and $\bar{c}_t^{1,k} = \sum_{j=1}^{k-1} (1 - \pi_t^j) c_j$. For $k = 1$, the action that maximizes the score-based reward is equivalent to minimizing $\left| \mathcal{F}_0^{-1} \left(1 - \frac{a_t c_1}{s_t} \right) - \mathcal{F}_1^{-1} \left(1 - \frac{(1 - a_t) c_1}{1 - s_t} \right) \right|$. ■

B.2 Convergence Under MFG Policy

We first provide a proof of Lemma 3.6 below.

Proof. Firstly, we observe that the optimal policy of institution k can be expressed as

a convex combination of its optimal score-based reward action and optimal fair-only action as $\pi_t^k = \omega_{k,t} a_{S,t}^k + (1 - \omega_{k,t}) a_{F,t}^k$, where $\omega_{k,t} \in [0, 1]$. This follows from the fact that the fairness-aware utility function is a concave function of the action, and the score-based reward function and the fairness loss functions are monotone. Then, we claim that the optimal policy of institution k can be expressed as a convex combination of its pre-projection optimal score-based reward action and pre-projection optimal fairness-aware action (before being projected to the feasible action space $\mathcal{A}_t^k$), as $\pi_t^k = \gamma_{k,t} \left(s_t + \frac{1}{c_k} \left(\sum_{j=1}^{k-1} c_j (s_t - \pi_t^j) \right) \right) + (1 - \gamma_{k,t}) \alpha$, where $\gamma_{k,t} \in [0, 1]$.

Let $\bar{a}_{S,t}^k = \left(s_t + \frac{1}{c_k} \left(\sum_{j=1}^{k-1} c_j (s_t - \pi_t^j) \right) \right)$ and $\bar{a}_{F,t}^k = \alpha$, which represent the pre-projected optimal actions. If both $\bar{a}_{S,t}^k \in \mathcal{A}_t^k$ and $\bar{a}_{F,t}^k \in \mathcal{A}_t^k$, then the claim is straightforward. Similarly, if only one of them is not in the feasible action space, then the claim is straightforward. The only case that needs to be considered is when both pre-projected optimal actions are not in the feasible action space. If one of them lies on the left of the feasible action interval and the other lies on the right of the feasible action interval, then the claim is straightforward. We will show that pre-projected optimal actions cannot simultaneously lie on the left/right of the feasible action interval. The first case is when both pre-projected optimal actions are on the left of the feasible action interval. Suppose that $\bar{a}_{F,t}^k < \max(0, 1 - \frac{1-s_t-\sum_{j=1}^{k-1}(1-\pi_t^j)c_j}{c_k})$ and $\bar{a}_{S,t}^k < \max(0, 1 - \frac{1-s_t-\sum_{j=1}^{k-1}(1-\pi_t^j)c_j}{c_k})$. If the maximum is equal to 0, then there is a contradiction since $\bar{a}_{F,t}^k \in [0, 1]$. Similarly, if the maximum is equal to $1 - \frac{1-s_t-\sum_{j=1}^{k-1}(1-\pi_t^j)c_j}{c_k}$, then there is a contradiction since $\bar{a}_{S,t}^k > 1 - \frac{1-s_t-\sum_{j=1}^{k-1}(1-\pi_t^j)c_j}{c_k}$, due to the condition that $\sum_{j=1}^K c_j < 1$. Thus, pre-projected optimal actions cannot lie on the left of the feasible action interval at the same time. The second case is when both pre-projected optimal actions are on the right of the feasible action interval. Suppose that $\bar{a}_{F,t}^k > \min(1, \frac{s_t-\sum_{j=1}^{k-1}\pi_t^j c_j}{c_k})$ and $\bar{a}_{S,t}^k > \min(1, \frac{s_t-\sum_{j=1}^{k-1}\pi_t^j c_j}{c_k})$. If the minimum is equal to 1, then there is a contradiction since $\bar{a}_{F,t}^k \in [0, 1]$. Similarly, if the minimum is equal to $\frac{s_t-\sum_{j=1}^{k-1}\pi_t^j c_j}{c_k}$, then there is a contradiction since $\bar{a}_{S,t}^k < \frac{s_t-\sum_{j=1}^{k-1}\pi_t^j c_j}{c_k}$,

due to the condition that $\sum_{j=1}^K c_j < 1$. Thus, the pre-projected optimal actions cannot lie on the right of the feasible action interval at the same time. Therefore, our claim holds.

By utilizing this relation, equation (3.8) can be expressed as the following by iteratively writing expressions for the optimal score-based reward action in the increasing order of the institutions:

$$a_{S,t}^k = \left[\frac{s_t}{c_k} \left(c_k + \sum_{j=1}^{k-1} c_j \prod_{i=j}^{k-1} (1 - \gamma_{i,t}) \right) - \frac{\alpha}{c_k} \left(\sum_{j=1}^{k-1} c_j \prod_{i=j}^{k-1} (1 - \gamma_{i,t}) \right) \right]_{\mathcal{A}_t^k} = \left[s_t + \frac{(s_t - \alpha)}{c_k} \left(\sum_{j=1}^{k-1} c_j \prod_{i=j}^{k-1} (1 - \gamma_{i,t}) \right) \right]_{\mathcal{A}_t^k} \quad (\text{B.24})$$

We use the above expression and now prove the lemma. The optimal action of the institution k can be expressed as follows:

$$\pi_t^k = \gamma_{k,t} \left(s_t + \frac{(s_t - \alpha)}{c_k} \left(\sum_{j=1}^{k-1} c_j \prod_{i=j}^{k-1} (1 - \gamma_{i,t}) \right) \right) + (1 - \gamma_{k,t})\alpha \quad (\text{B.25})$$

We can compute the weighted action as follows;

$$c_k \pi_t^k = c_k \gamma_{k,t} s_t + \gamma_{k,t} (s_t - \alpha) \sum_{j=1}^{k-1} c_j \prod_{i=j}^{k-1} (1 - \gamma_{i,t}) + c_k (1 - \gamma_{k,t})\alpha \quad (\text{B.26})$$

We can add and subtract $c_k s_t$;

$$c_k \pi_t^k = c_k s_t + c_k (1 - \gamma_{k,t})(\alpha - s_t) - \gamma_{k,t}(\alpha - s_t) \sum_{j=1}^{k-1} c_j \prod_{i=j}^{k-1} (1 - \gamma_{i,t}) \quad (\text{B.27})$$

We can add and subtract $(\alpha - s_t) \sum_{j=1}^{k-1} c_j \prod_{i=j}^{k-1} (1 - \gamma_{i,t})$;

$$c_k \pi_t^k = c_k s_t + c_k (1 - \gamma_{k,t}) (\alpha - s_t) + (1 - \gamma_{k,t}) (\alpha - s_t) \sum_{j=1}^{k-1} c_j \prod_{i=j}^{k-1} (1 - \gamma_{i,t}) - (\alpha - s_t) \sum_{j=1}^{k-1} c_j \prod_{i=j}^{k-1} (1 - \gamma_{i,t}) \quad (\text{B.28})$$

It can be grouped as follows:

$$c_k \pi_t^k = c_k s_t + (\alpha - s_t) \sum_{j=1}^k c_j \prod_{i=j}^k (1 - \gamma_{i,t}) - (\alpha - s_t) \sum_{j=1}^{k-1} c_j \prod_{i=j}^{k-1} (1 - \gamma_{i,t}) \quad (\text{B.29})$$

It can be seen that the second and third terms are telescoping, and the summation of the weighted actions of all institutions can be written as follows:

$$\sum_{k=1}^K c_k \pi_t^k = s_t \sum_{k=1}^K c_k + (\alpha - s_t) \sum_{k=1}^K c_k \prod_{i=k}^K (1 - \gamma_{i,t}) \quad (\text{B.30})$$

After dividing by the total capacity, we obtain the desired expression. ■

We will now utilize the result of the above lemma in showing the proof of Theorem 3.7, where we state that the applicant pool proportion, and the admission proportions of the institutions converge to the long-term fairness target set by the agents, under the MFG policy with pure positive reinforcement model.

Proof. Suppose that the state of the MDP equals the fairness target α . It follows that the top institution maximizes its utility by selecting α proportion of group 0 among all its admitted applicants $\pi_t^1 = \alpha$, as its optimal score-based reward action itself is $a_{S,t}^1 = s_t = \alpha$. Further, $a_{S,t}^2 = \alpha$ by equation (3.8), which implies that $\pi_t^2 = \alpha$. It can be seen that $\pi_t^k = \alpha, \forall k$, due to which the weighted average $\pi_t^W = \alpha$, i.e., it is a fixed-point of the weighted policy. We will show the uniqueness of the fixed point by showing $\gamma_{i,t} \neq 1, \forall i \in [K]$. Assume that $s_t < \alpha$. Then, the optimal score-based

reward action of the top institution is $a_{S,t}^1 = s_t$, since the top institution maximizes its score-based reward when it sets same threshold for both groups. The optimal fair-only action is $a_{F,t}^1 = [\alpha]_{\mathcal{A}_t^1}$. Since $a_{F,t}^1 > s_t$ the MFG policy for the top institution will accept more from the minority group than the applicant proportion due to the structure of the weighted MFG policy. Then, $\pi_t^1 > s_t$ and $\gamma_{1,t} \neq 1$. The score-based reward of the second institution is maximized when it sets the same threshold for both groups, this implies that $a_{S,t}^2 < s_t$ because the first institution admitted more from the minority group than the applicant proportion. The optimal fair-only action is $a_{F,t}^2 = [\alpha]_{\mathcal{A}_t^2}$. Then, $\pi_t^2 > s_t$ and $\gamma_{2,t} \neq 1$. The same argument can be made for the subsequent institutions, since the optimal score-based reward action of the institution k is $a_{S,t}^k < s_t$ because the previous institutions admitted more from the minority group than the applicant proportion. The optimal fair-only action is $a_{F,t}^k = [\alpha]_{\mathcal{A}_t^k}$. Then, $\pi_t^k > s_t$ and $\gamma_{k,t} \neq 1$. The same argument can be used when $s_t > \alpha$.

Next, since from the update for pure positive reinforcement (3.5), it follows that the mean parameter θ_t drifts towards the fairness target α , irrespective of the state, due to the structure of the weighted MFG policy. In addition, since α is a unique fixed-point, as the step size is decaying with time, it can be shown that the mean parameter converges to the fairness target. Let $d_t = \frac{1}{2}(\theta_t - \alpha)^2$. Fix an $\epsilon > 0$. Then, we need to show that there exists some $t_0(\epsilon)$ such that for all $t > t_0(\epsilon)$,

$$d_{t+1} \leq d_t - \zeta_t, \text{ if } d_t \geq \epsilon \quad (\text{B.31})$$

$$d_{t+1} \leq c\epsilon \text{ if } d_t < \epsilon \quad (\text{B.32})$$

where c is a positive constant. Moreover $\zeta_t > 0$ and $\sum_{t=0}^{\infty} \zeta_t = \infty$. If the above conditions are satisfied, then eventually for some $t_1(\epsilon) \geq t_0(\epsilon)$, we have $d_t < \epsilon$. But due to equations (B.31) and (B.32), we have $d_{t+1} \leq c\epsilon$ for all $t > t_1(\epsilon)$. Since ϵ is arbitrary, we

have $\theta_t \rightarrow \alpha$ as $t \rightarrow \infty$.

$$\begin{aligned}
d_{t+1} &= \frac{1}{2}(\theta_{t+1} - \alpha)^2 \\
&= \frac{1}{2}([\theta_t + \eta_t(\pi_t^W - s_t)]_c - \alpha)^2 \\
&\leq \frac{1}{2}(\theta_t + \eta_t(\pi_t^W - s_t) - \alpha)^2 \\
&= d_t + \eta_t(\theta_t - \alpha)(\pi_t^W - s_t) + \frac{1}{2}\eta_t^2(\pi_t^W - s_t)^2 \\
&\leq d_t + \eta_t(\theta_t - \alpha)(\pi_t^W - s_t) + \frac{1}{2}\eta_t^2 \tag{B.33}
\end{aligned}$$

$$\leq d_t + \frac{\eta_t}{2}((\theta_t - \alpha)^2 + 1) + \frac{1}{2}\eta_t^2 \tag{B.34}$$

Since $\eta_t \rightarrow 0$, if $d_t < \epsilon$, then $d_{t+1} < c\epsilon$ for some $c > 0$.

When $d_t \geq \epsilon$, first we will account for the stochasticity of s_t . We have $\pi_t^W - s_t = \pi_t^W - \theta_t + (\theta_t - s_t)$. Denoting $z_t = \theta_t - s_t$, using equation (B.33), we have

$$d_{t+1} \leq d_t + \eta_t(\theta_t - \alpha)(\pi_t^W - \theta_t + z_t) + \frac{1}{2}\eta_t^2$$

where z_t is a zero-mean random variable. Also $\mathbb{E}[z_t^2] = \text{var}(s_t) < \infty$, which is bounded. Therefore, $\nu_t := \sum_{i=0}^t \eta_i z_i$ is a martingale and $\mathbb{E}[\nu_t^2]$ is also bounded. This implies by the martingale convergence theorem that ν_t converges to a finite random variable. Therefore, we have $\sum_{i=t}^{\infty} \eta_i z_i \rightarrow 0$. Since, $|\theta_t - \alpha|$ is bounded, the effect of the noise z_t is negligible. Therefore we have

$$d_{t+1} \leq d_t + \eta_t(\theta_t - \alpha)(\pi_t^W - \theta_t) + \frac{1}{2}\eta_t^2$$

We want to show that

$$(\theta_t - \alpha)(\pi_t^W - \theta_t) \leq -\delta(\epsilon) \tag{B.35}$$

for some $\delta(\epsilon) > 0$. If this holds, we have

$$d_{t+1} \leq d_t - \eta_t \delta(\epsilon) + \frac{\eta_t^2}{2} \quad (\text{B.36})$$

Let us denote $\zeta_t = \eta_t \delta(\epsilon) - \frac{\eta_t^2}{2}$. Since, $\eta_t \rightarrow 0$, there exists $t_2(\epsilon)$ such that $\zeta_t > 0$ for all $t > t_2(\epsilon)$. Moreover, due to assumptions on the step size, we have $\sum_{t=0}^{\infty} \zeta_t = \infty$.

What remains to show is equation (B.35). In the regime where the number of applicants N is large, we can see that the state s_t is equal to its mean θ_t with probability approaching 1 through Chebyshev inequality. When $d_t \geq \epsilon$, since s_t is equal to its mean θ_t , we need to consider only cases (i) $s_t < \alpha$ and (ii) $s_t > \alpha$. Under both cases, we have $(\theta_t - \alpha)(\pi_t^W - \theta_t) < 0$ due to the structure of the weighted MFG policy when the score distributions are identical. ■

B.3 Role-Model Negative Feedback

Here, we provide a proof for Proposition 3.9, and show that MFG policy can cause negative feedback under role model reinforcement and drive the under-represented group out of the system.

Proof. We assume that the MFG policy assigns non-zero weight to the long-term fairness objective, i.e., $\lambda > 0$. Since we are interested in the regime where the number of applicants N_t is large, we assume that the histograms of the scores of applicants approach the distribution. Without loss of generality, we denote group 0 to be the under-represented group, and assume that the initial state is less than the fairness target, $s_t < \alpha$. However, the same procedure applies when group 0 has a higher proportion than the long-term fairness target. We will also assume that the initial mean parameter θ_t is small.

We recall that for the top institution, the action maximizing its score-based reward

$a_{S,t}^1 = s_t$, the current state, by Theorem 3.3 and the action minimizing its fairness loss $a_{F,t}^1 = [\alpha]_{\mathcal{A}_t^1}$, the long-term fairness target. We also know the MFG policy for the top institution will be the convex combination $\pi_t^1 = \gamma_1 s_t + (1 - \gamma_k)\alpha$. Then, $\pi_t^1 > s_t$ because $\alpha > s_t$ by our assumption. Viewing the first institution's MFG policy as applying group-dependent lower and upper thresholds on the score distributions as in the proof of Theorem 3.3, and admitting all applicants between the group-specific thresholds, we can infer that the lower threshold for group 0 (denoted by $t_0^{1, \text{low}}$) is strictly less than the lower threshold for group 1 (denoted by $t_1^{1, \text{low}}$), due to the fact that π^1 admits more from the minority group than the applicant proportion. Implicitly, the upper thresholds for the first institution are infinity. Thus we have:

$$t_0^{1, \text{low}} < t_1^{1, \text{low}} \quad (\text{B.37})$$

Note that these thresholds are different from the thresholds optimizing only for the score-based rewards, which have also been described by the same notation in the proof of Theorem 3.3.

Let us assume that the fraction of the role models $r = \epsilon$. The evolution of the applicant pool is such that only those admitted applicants with scores larger than a certain threshold determined by parameter r will contribute to reinforcing the pool. Let us denote this threshold for institution k as t_r^k . As the parameter r decreases, the threshold t_r^k increases. It follows that there exists r small enough, such that the role model threshold for the first institution is $t_r^1 \geq t_1^{1, \text{low}}$. We remark that the role model threshold is independent of the group membership.

Now, note that r_t^1 is equivalent to the ratio of the number of group 0 applicants with scores higher than t_r^1 , to the number of applicants with scores larger than t_r^1 . Thus we can write an expression for r_t^1 in terms of the CDF of the score distribution, denoted by

$\mathcal{F}$, as

$$r_t^1 = \frac{s_t (1 - \mathcal{F}(t_r^1))}{s_t (1 - \mathcal{F}(t_r^1)) + (1 - s_t) (1 - \mathcal{F}(t_r^1))} = s_t. \quad (\text{B.38})$$

For the subsequent institutions, it is known from (B.24) that for all $k \geq 2$, if $s_t < \alpha$, the action optimizing the score-based reward is $a_{S,t}^k < s_t$. Therefore, the policy for the k^{th} institution ($k \geq 2$) is

$$\pi_t^k = (1 - \gamma_{k,t})\alpha + \gamma_{k,t} \left(s_t + \frac{1}{c_k} \left(\sum_{j=1}^{k-1} c_j (s_t - \pi_t^j) \right) \right) > a_{S,t}^k \quad (\text{B.39})$$

since we have $\alpha > s_t > a_{S,t}^k$. Note that for the MFG policies of the institutions, the lower threshold of institution $k - 1$ will be equal to the upper threshold of institution k , as it would admit all applicants with scores in between the lower and upper thresholds.

If the institutions with $k \geq 2$ were optimizing only the score-based reward their lower thresholds would be equal across the groups, as argued in the equation (B.23). But here, since the MFG policy admits a higher proportion from group 0, i.e., (B.39), the thresholds are such that the lower threshold of group 0 is always less than the lower threshold of group 1, $\forall k \in [K]$

$$t_0^{k,\text{low}} < t_1^{k,\text{low}}. \quad (\text{B.40})$$

Furthermore, if r is small enough, the role model thresholds for all subsequent institutions are large enough to satisfy $t_1^{k,\text{low}} \leq t_r^k$. Then, we can express the proportion of the role models from group 0 for the institution k as

$$r_t^k = \frac{s_t \left(\mathcal{F}(t_0^{k,\text{up}}) - \mathcal{F}(t_r^k) \right)}{s_t \left(\mathcal{F}(t_0^{k,\text{up}}) - \mathcal{F}(t_r^k) \right) + (1 - s_t) \left(\mathcal{F}(t_1^{k,\text{up}}) - \mathcal{F}(t_r^k) \right)}. \quad (\text{B.41})$$

Using $t_1^{k,\text{up}} > t_0^{k,\text{up}}$, we can upper bound r_t^k as

$$r_t^k < \frac{s_t \left(\mathcal{F}(t_0^{k,\text{up}}) - \mathcal{F}(t_r^k) \right)}{s_t \left(\mathcal{F}(t_0^{k,\text{up}}) - \mathcal{F}(t_r^k) \right) + (1 - s_t) \left(\mathcal{F}(t_0^{k,\text{up}}) - \mathcal{F}(t_r^k) \right)} \implies r_t^k < s_t \quad (\text{B.42})$$

for all $k \in \{2, 3, \dots, K\}$ and $r_t^1 = s_t$.

Under the role model reinforcement, group 0's applicant proportion receives a positive drift only if the weighted parameter π_t^r in equation (3.16) is larger than s_t . Due to (B.38) and (B.42), the weighted proportion of the role models is

$$\pi_t^r = \frac{\sum_{k=1}^K c_k r_t^k}{\sum_{k=1}^K c_k} < s_t \implies \pi_t^r - s_t < 0. \quad (\text{B.43})$$

Hence, the pool update parameter $\theta_{t+1} = \theta_t + \eta_t(\pi_t^r - s_t) < \theta_t$. If the initial mean parameter θ_t is small enough, i.e., group 0 is heavily under-represented in the initial pool, with a large probability the future state $s_{t+1} < \alpha$. Hence we can *approximately* see that the mean parameter of the proportion of minority group in the applicant pool decreases to zero. Therefore, we show that MFG policy can cause a negative feedback loop under role model reinforcement if the role model proportion is small enough, driving the under-represented group out of the system. ■

B.4 Additional Experiments

B.4.1 Identical score distributions

We show the evolution of applicant and admission proportions of the CMFG policy under the pure positive reinforcement model, order-based reinforcement model, and weighted positive reinforcement model in Figures B.1a, B.1b, and B.1c. The parameter

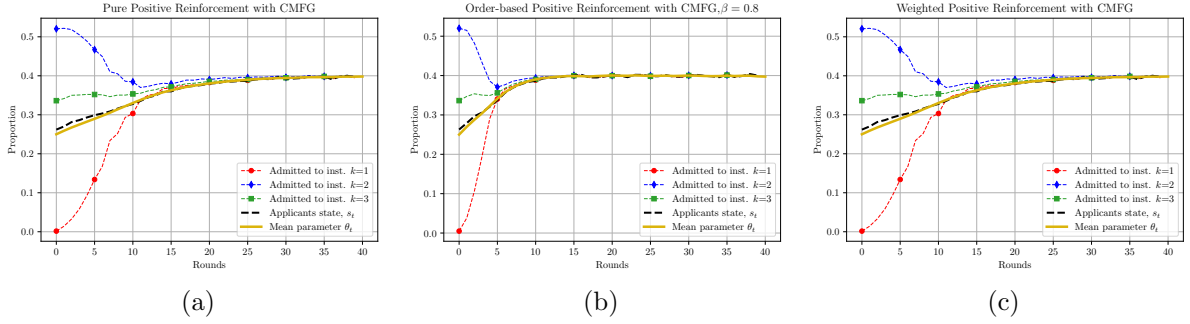


Figure B.1: The centralized MFG policy achieves long-term fairness under (a) pure positive reinforcement, (b) order-based reinforcement, and (c) weighted positive reinforcement.

settings are exactly the same as in Section 3.5.1 for the case when the score distributions are identical. In these figures, we observe that an institution's admission proportion is larger than the applicant proportion, and since the weighted average of the admission proportion governs the evolution, the pool gets positive feedback and approaches the fairness target.

B.4.2 Distinct score distributions

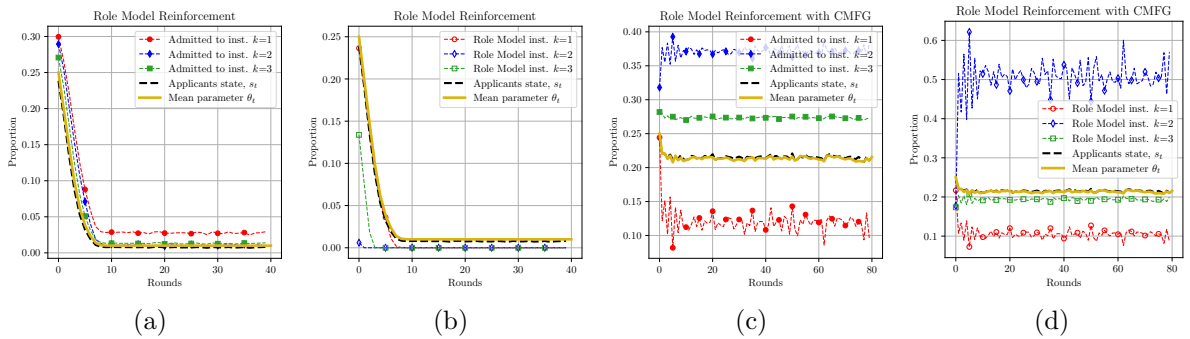


Figure B.2: (a) MFG policy creates negative feedback under role-model reinforcement, under distinct score distributions. (b) The evolution of the proportions of role models for each institution, under MFG policy. (c) CMFG policy avoids negative feedback under distinct score distributions. (d) The evolution of the proportions of role models for each institution, under CMFG policy.

Here, we consider the role-model reinforcement with $r = 0.5$, under distinct score distributions, with all other parameters as described in Section 3.5.1. As seen in Figure B.2a, the MFG policy again leads to a negative feedback loop with the distinct score distributions, resulting in the reduction of the minority group in the applicants pool. As seen in Figure B.2b, the second and the third institutions have a very small fraction of role models as a result of the first institution admitting the top minority applicants. However, the CMFG policy could alleviate this, resulting in an equilibrium point lower than the initial mean parameter as shown in Figure B.2c. In Figure B.2d, it is evident that all institutions feature role models from the minority group, leading to an equilibrium point within the applicant pool.

B.4.3 Semi-synthetic dataset

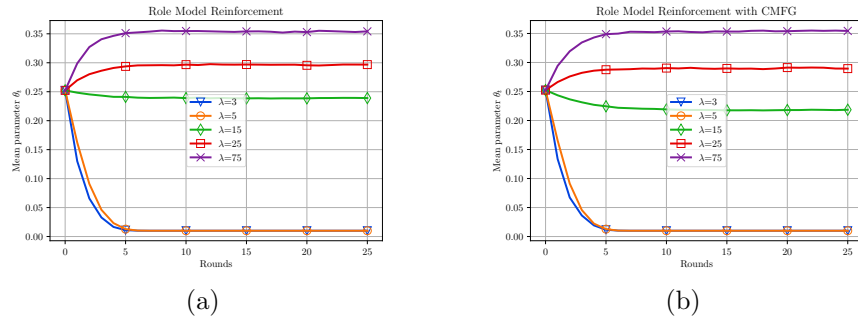


Figure B.3: Evolution of the mean parameter θ_t and its equilibrium with varying values of λ under (a) the MFG policy and (b) the CMFG policy, under role-model reinforcement.

We consider the role-model reinforcement with $r = 0.8$, where the top 80% of the admitted applicants are considered role models for the resource pool. Figures B.3a and B.3b show the mean parameter θ_t at equilibrium for the MFG and CMFG policies, respectively, with varying values of λ . Both policies perform similarly on the law school dataset, possibly due to a larger difference between the score distributions of the groups.

Appendix C

Supplementary Materials for Chapter 4

C.1 Proofs

We provide the proofs for CF-CP.

C.1.1 Proof of Lemma 4.4

Proof. Fix $y \in \mathcal{Y}$ and $a, a' \in \mathcal{A}$. Under Assumption 4.3, for the factual pair $(X_{A \leftarrow a}, a)$, there exists a unique latent variable

$$U = g_a^{-1}(X_{A \leftarrow a}), \tag{C.1}$$

where g_a^{-1} is the inverse of the forward map g_a in the SCM that generates X from U under the intervention $A \leftarrow a$. Likewise, for $(X_{A \leftarrow a'}, a')$, we have the same latent variable $U = g_{a'}^{-1}(X_{A \leftarrow a'})$, since both counterfactuals arise from the same structural model applied to the same exogenous noise.

For any target intervention $a'' \in \mathcal{A}$, the corresponding counterfactual feature vector is

$$X_{A \leftarrow a''} = g_{a''}(U), \quad (\text{C.2})$$

regardless of whether we start from $(X_{A \leftarrow a}, a)$ or $(X_{A \leftarrow a'}, a')$. Hence, the multiset of per-intervention scores

$$\{s(g_{a''}(U), a'', y) : a'' \in \mathcal{A}\} \quad (\text{C.3})$$

is identical in both cases, up to permutation. Since Agg in (4.6) is permutation-invariant, applying it to these equal multisets gives

$$s_{\text{cf}}(X_{A \leftarrow a}, a, y) = s_{\text{cf}}(X_{A \leftarrow a'}, a', y). \quad (\text{C.4})$$

This proves the claim. ■

C.1.2 Proof of Theorem 4.6

Proof. Let π be any permutation of $\mathcal{I}_{\text{cal}} \cup \{n+1\}$. By assumption,

$$\{(X_{\pi(i)}, A_{\pi(i)}, Y_{\pi(i)})\}_i \stackrel{d}{=} \{(X_i, A_i, Y_i)\}_i. \quad (\text{C.5})$$

Define the coordinate-wise transformation

$$T(x, a, y) := \text{Agg} \{s(g_{a'}(g_a^{-1}(x)), a', y) : a' \in \mathcal{A}\}. \quad (\text{C.6})$$

Under Assumption 4.3, the maps g_a^{-1} and $g_{a'}$ are deterministic and measurable for all $a, a' \in \mathcal{A}$. Since s and Agg are also measurable, T is measurable. Moreover, the same transformation T is applied to each coordinate.

Set

$$S_i^{\text{cf}} := T(X_i, A_i, Y_i). \quad (\text{C.7})$$

Exchangeability is preserved under identical measurable coordinate-wise transformations [141, Proposition 4]. Therefore,

$$\{S_i^{\text{cf}}\}_i \stackrel{d}{=} \{s_{\text{cf}}(X_{\pi(i)}, A_{\pi(i)}, Y_{\pi(i)})\}_i. \quad (\text{C.8})$$

Thus, $\{S_i^{\text{cf}}\}_{i \in \mathcal{I}_{\text{cal}} \cup \{n+1\}}$ are exchangeable. ■

C.2 Experimental Details

C.2.1 Implementation Details

Here we provide additional implementation details. All GPU related experiments were run on GTX 1080 Ti. Remaining experiments were run on Apple M1 CPU.

Base predictor. For all datasets, the base predictor $\hat{f}$ is a linear model for both regression and classification tasks. For synthetic and Law School datasets, we use `scikit-learn`’s `LinearRegression` and `LogisticRegression` implementations [142]. The default hyperparameters are used except for the maximum number of iterations which is set to 1000 for `LogisticRegression`. For Bias in Bios, we use `PyTorch` [143] to implement the logistic regression and utilize GPU to speed up optimization. We use LBFGS optimizer with a learning rate of 1 and a maximum of 100 iterations.

Conformal prediction sets. When constructing conformal prediction sets, we include at least one label in the prediction set for classification tasks if the score of the most likely label is above the threshold. This is to avoid empty prediction sets which can happen

when the model is very uncertain about the prediction. It is a common practice in conformal prediction to ensure non-empty prediction sets.

C.2.2 Datasets

Here we introduce the datasets used in our experiments.

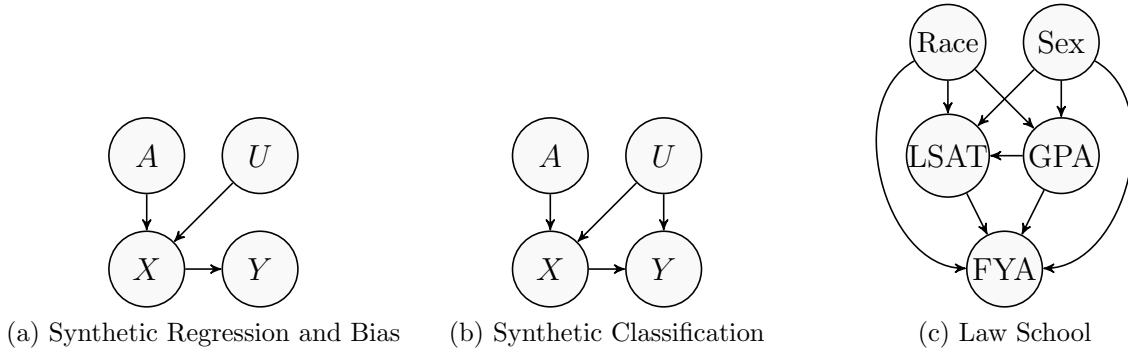


Figure C.1: Causal Graphs of the Datasets Used in Experiments

Synthetic Regression. We follow the data generation process in [94] to generate a synthetic regression dataset with a binary protected attribute A . The data is generated according to the following structural equations:

$$\begin{aligned}
 U_i &\sim \mathcal{N}(0, 1) \quad i \in 1, 2, \quad A \sim \text{Bernoulli}(0.4), \\
 X &\leftarrow \sin(U_1) + \cos(AU_2) + A + 0.1, \\
 Y &\leftarrow 0.2X^2 + 1.2X + 0.2 + \epsilon_Y, \quad \epsilon_Y \sim \mathcal{N}(0, 0.6).
 \end{aligned}$$

The causal graph is shown in Figure C.1a. Counterfactuals change only A in the X -equation (holding U_1, U_2, ϵ_Y fixed). For noisy experiments, we add i.i.d. Gaussian noise to U when constructing counterfactuals, i.e., we use $\tilde{U} = U + \epsilon_U$ where $\epsilon_U \sim \mathcal{N}(0, \sigma_U^2 I)$ and $\sigma_U = 0.4$ for the results in Table 4.3. We use $\alpha = 0.1$ and the split of the data is

as follows: 5000 samples for training, 1000 samples for calibration, and 5000 samples for testing.

Synthetic Classification. We extend the binary classification setting in [95] to a multi-class classification setting with $K = 10$ classes. The SCM is

$$\begin{aligned} U &\sim \mathcal{N}(\mathbf{0}, I_d), \quad A \sim \text{Bernoulli}(0.5), \\ X &\leftarrow (A - 0.5)w_A + UD_U, \\ Y &\sim \text{Categorical}(\text{softmax}(X^{\odot 3}W_X + UW_U + E)), \quad E \sim \mathcal{N}(\mathbf{0}, \sigma^2 I_K). \end{aligned}$$

The causal graph is shown in Figure C.1b. Here $d = 10$ is the feature dimension and $K = 10$ the number of classes; $w_A \in \mathbb{R}^{1 \times d}$ encodes the linear effect of A on X ; $D_U \in \mathbb{R}^{d \times d}$ mixes the latent U into X (assumed full rank); $W_X, W_U \in \mathbb{R}^{d \times K}$ map the (nonlinear) features and latent into class logits; E is i.i.d. Gaussian *logit* noise with variance $\sigma = 0.2$ and $X^{\odot 3}$ denotes elementwise cubing of X . All the matrices are generated as identity matrix plus small uniform noise between $[0, 0.2]$ added to each entry, w_A has first $r = 3$ entries sampled from $\text{Uniform}([2, 2.2])$ and the rest are zeros. Counterfactuals change only A in the X -equation (holding U and E fixed). Similarly, for noisy experiments, we add i.i.d. Gaussian noise to U when constructing counterfactuals, i.e., we use $\tilde{U} = U + \epsilon_U$ where $\epsilon_U \sim \mathcal{N}(0, \sigma_U^2 I)$ and $\sigma_U = 0.4$ for the results in Table 4.3. We use $\alpha = 0.1$ and the split of the data is as follows: 5000 samples for training, 1000 samples for calibration, and 5000 samples for testing.

Law School. We use the Law School dataset from [76], which has been used in prior work on counterfactual fairness [11, 94]. The task is to predict the first-year average grade (FYA) of law students based on their LSAT score, undergraduate GPA, race and

gender. We consider race as the protected attribute A (binary: white vs non-white). We preprocess the LSAT and GPA features by standardizing them to have zero mean and unit variance. To construct counterfactuals, we fit a LiNGAM model to the data using the `causal-learn` package [144]. The fitted causal graph is shown in Figure C.1c. We assume that race and gender are root nodes in the causal graph and there are no children of FYA node. For noisy experiments, we add i.i.d. Gaussian noise to U when constructing counterfactuals, and we assume U only effects LSAT and GPA, i.e., we use $\tilde{U} = U + \epsilon_U$ where $\epsilon_U \sim \mathcal{N}(0, \sigma_U^2 I)$ and $\sigma_U = 0.4$ for the results in Table 4.3. We use $\alpha = 0.1$ and the total number of samples is 21791, with a split of 1000 for calibration, 10000 for testing, and the rest for training.

Bias in Bios. We use the Bios dataset from [105] which contains biographies of individuals in various professions. The task is to predict the profession of an individual based on their biography text, it has gender labels which we consider as the protected attribute A (binary: male vs female), there are 28 professions (classes) in total. We get the embeddings of the factual and counterfactual biographies via the method proposed by [106], using their code is publicly available.¹ They assume a causal graph similar to Figure C.1a where X is the text embedding and Y is the profession, and U is some latent variable that is not observed. We use the version of the dataset introduced in [145]. First we obtain text embeddings using a pre-trained BERT model [107], then we split this embedding into two orthogonal components, one that is aligned with the protected attribute A and one that is independent of A . To create counterfactual embeddings, we flip the component aligned with A while keeping the independent component fixed. In this dataset, we do not have access to U , so we cannot apply CFU method. For noisy experiments, we add i.i.d. Gaussian noise to the embedding when constructing counter-

¹<https://github.com/ToineSayan/counterfactual-representations-for-explanation>

Method	Regression					Classification				
	MSE ($\downarrow$)	TE ($\downarrow$)	Coverage	Avg. Set Size ($\downarrow$)	CSD ($\downarrow$)	Accuracy ($\uparrow$)	TE ($\downarrow$)	Coverage	Avg. Set Size ($\downarrow$)	CSD ($\downarrow$)
SplitCP	0.377 $\pm$ 0.012	1.219 $\pm$ 0.014	0.901 $\pm$ 0.008	2.032 $\pm$ 0.038	0.713 $\pm$ 0.009	0.730 $\pm$ 0.011	0.543 $\pm$ 0.037	0.905 $\pm$ 0.011	2.025 $\pm$ 0.209	0.642 $\pm$ 0.043
Post-hoc Union	0.377 $\pm$ 0.012	1.219 $\pm$ 0.014	0.944 $\pm$ 0.006	3.259 $\pm$ 0.044	0.152 $\pm$ 0.001	0.730 $\pm$ 0.011	0.543 $\pm$ 0.037	0.935 $\pm$ 0.009	3.115 $\pm$ 0.254	0.260 $\pm$ 0.006
CFU	1.170 $\pm$ 0.020	0.267 $\pm$ 0.007	0.902 $\pm$ 0.008	3.551 $\pm$ 0.088	0.135 $\pm$ 0.004	0.512 $\pm$ 0.011	0.235 $\pm$ 0.004	0.908 $\pm$ 0.014	3.462 $\pm$ 0.239	0.345 $\pm$ 0.008
CFR	0.852 $\pm$ 0.014	0.191 $\pm$ 0.002	0.900 $\pm$ 0.012	3.013 $\pm$ 0.078	0.115 $\pm$ 0.003	0.563 $\pm$ 0.012	0.140 $\pm$ 0.002	0.907 $\pm$ 0.014	2.913 $\pm$ 0.209	0.209 $\pm$ 0.004
PCF	0.847 $\pm$ 0.014	0.189 $\pm$ 0.002	0.897 $\pm$ 0.010	2.983 $\pm$ 0.061	0.115 $\pm$ 0.003	0.550 $\pm$ 0.015	0.165 $\pm$ 0.004	0.908 $\pm$ 0.012	2.600 $\pm$ 0.210	0.263 $\pm$ 0.006
CF-CP-mean	0.377 $\pm$ 0.012	1.219 $\pm$ 0.014	0.900 $\pm$ 0.008	3.016 $\pm$ 0.054	<u>0.118 $\pm$ 0.002</u>	0.730 $\pm$ 0.011	0.543 $\pm$ 0.037	0.908 $\pm$ 0.013	2.609 $\pm$ 0.203	0.259 $\pm$ 0.006
CF-CP-max	0.377 $\pm$ 0.012	1.219 $\pm$ 0.014	0.895 $\pm$ 0.009	3.374 $\pm$ 0.057	0.145 $\pm$ 0.003	0.730 $\pm$ 0.011	0.543 $\pm$ 0.037	0.945 $\pm$ 0.004	4.271 $\pm$ 0.185	<u>0.230 $\pm$ 0.004</u>
CF-CP-min	0.377 $\pm$ 0.012	1.219 $\pm$ 0.014	0.905 $\pm$ 0.008	2.935 $\pm$ 0.050	0.168 $\pm$ 0.003	0.730 $\pm$ 0.011	0.543 $\pm$ 0.037	0.910 $\pm$ 0.012	2.657 $\pm$ 0.205	0.271 $\pm$ 0.005

Table C.1: Synthetic results at $\alpha = 0.1$ (mean $\pm$ std over 10 runs) with noisy counterfactuals. Left block: regression; right block: classification. Highlighting follows the convention described in the evaluation metrics paragraph.

factuials, i.e., we use $\tilde{X}_{A \leftarrow a'} = X_{A \leftarrow a'} + \epsilon_X$ where $\epsilon_X \sim \mathcal{N}(0, \sigma_X^2 I)$ and $\sigma_X = 0.1$ for the results in Table 4.3. We use $\alpha = 0.025$ since the task is fairly easy, if we use $\alpha = 0.1$ all methods return singletons as the prediction sets most of the time. The total number of samples is 393423. We use 5000 samples for calibration, half of the dataset for testing, and the rest for training.

C.3 Additional Results

In this section, we provide additional experimental results.

C.3.1 Results on Noisy Counterfactuals

Here we present additional results on the experiments with noisy counterfactuals. In Table 4.3, we report the results on all datasets with noisy counterfactuals. We only report average set size and CSD due to space constraints, the other metrics are reported here in Table C.1 and Table C.2. It can be seen that due to noisy counterfactuals, MSE or accuracy of counterfactually fair methods (CFU, CFR, PCF) are slightly worse than oracle counterfactuals. This also leads to a slight increase in average set size for these methods. However, counterfactually fair methods and CF-CP still achieve a significant reduction in CSD compared to split CP, with only a modest increase in the size of the

Method	Law School Regression					Bios Classification				
	MSE ($\downarrow$)	TE ($\downarrow$)	Coverage	Avg. Set Size ($\downarrow$)	CSD ($\downarrow$)	Accuracy ($\uparrow$)	TE ($\downarrow$)	Coverage	Avg. Set Size ($\downarrow$)	CSD ($\downarrow$)
SplitCP	0.758 $\pm$ 0.005	0.723 $\pm$ 0.019	0.898 $\pm$ 0.009	2.847 $\pm$ 0.070	0.405 $\pm$ 0.010	0.801 $\pm$ 0.001	0.094 $\pm$ 0.003	0.975 $\pm$ 0.003	3.311 $\pm$ 0.199	0.199 $\pm$ 0.005
Post-hoc Union	0.758 $\pm$ 0.005	0.723 $\pm$ 0.019	0.943 $\pm$ 0.007	3.571 $\pm$ 0.078	<u>0.036 $\pm$ 0.001</u>	0.801 $\pm$ 0.001	0.094 $\pm$ 0.002	0.979 $\pm$ 0.002	3.991 $\pm$ 0.231	0.130 $\pm$ 0.002
CFU	0.835 $\pm$ 0.007	0.087 $\pm$ 0.003	0.900 $\pm$ 0.009	2.998 $\pm$ 0.063	0.055 $\pm$ 0.002	-	-	-	-	-
CFR	0.829 $\pm$ 0.007	0.045 $\pm$ 0.001	0.899 $\pm$ 0.009	2.984 $\pm$ 0.064	0.029 $\pm$ 0.001	0.796 $\pm$ 0.001	0.045 $\pm$ 0.001	0.974 $\pm$ 0.003	3.450 $\pm$ 0.226	0.099 $\pm$ 0.003
PCF	0.829 $\pm$ 0.007	0.056 $\pm$ 0.001	0.897 $\pm$ 0.010	2.961 $\pm$ 0.071	0.037 $\pm$ 0.001	0.792 $\pm$ 0.001	0.049 $\pm$ 0.001	0.975 $\pm$ 0.002	3.560 $\pm$ 0.197	<u>0.109 $\pm$ 0.002</u>
CF-CP-mean	0.758 $\pm$ 0.005	0.723 $\pm$ 0.019	0.899 $\pm$ 0.014	3.108 $\pm$ 0.108	0.029 $\pm$ 0.001	0.801 $\pm$ 0.001	0.093 $\pm$ 0.003	0.975 $\pm$ 0.003	3.581 $\pm$ 0.204	<u>0.109 $\pm$ 0.002</u>
CF-CP-max	0.758 $\pm$ 0.005	0.723 $\pm$ 0.019	0.900 $\pm$ 0.014	3.114 $\pm$ 0.112	0.041 $\pm$ 0.001	0.801 $\pm$ 0.000	0.094 $\pm$ 0.003	0.975 $\pm$ 0.003	3.813 $\pm$ 0.210	0.123 $\pm$ 0.002
CF-CP-min	0.758 $\pm$ 0.005	0.723 $\pm$ 0.019	0.899 $\pm$ 0.012	3.107 $\pm$ 0.093	0.041 $\pm$ 0.001	0.801 $\pm$ 0.001	0.093 $\pm$ 0.003	0.975 $\pm$ 0.002	3.599 $\pm$ 0.165	0.128 $\pm$ 0.003

Table C.2: Real data results with noisy counterfactuals. Left block: regression on Law School dataset with $\alpha = 0.1$; right block: classification on Bios dataset with $\alpha = 0.025$ (mean $\pm$ std over 10 runs). Highlighting follows the convention described in the evaluation metrics paragraph.

prediction sets. The noise level for synthetic and Law School datasets is $\sigma_U = 0.4$, and for Bias in Bios dataset it is $\sigma_X = 0.1$.

C.3.2 Results on Synthetic Classification with Different Base Score Functions

Here we provide additional results on synthetic classification with different base score functions. In addition to LAC (used in the main text), we also consider the following two base score functions. The first one is Adaptive Prediction Sets (APS) score [146]:

$$s_{\text{APS}}(x, a, y) = \sum_{y' \in \mathcal{Y}, \hat{f}_{y'}(x, a) > \hat{f}_y(x, a)} \hat{f}_{y'}(x, a). \quad (\text{C.9})$$

We use APS score without additional randomness for simplicity. The second one is the regularized APS (RAPs) score, which adds a regularization term to the APS score to penalize large prediction sets [147]:

$$s_{\text{RAPs}}(x, a, y) = s_{\text{APS}}(x, a, y) + \lambda(\text{rank}(\hat{f}_y(x, a)) - k_{\text{reg}})_+, \quad (\text{C.10})$$

where $\lambda > 0$ is the regularization parameter, k_{reg} is a threshold parameter, and $\text{rank}(\cdot)$ is the rank of the true label y in the sorted list of predicted probabilities. We set $\lambda = 0.5$

Method	APS					RAPS				
	Accuracy ($\uparrow$)	TE ($\downarrow$)	Coverage	Avg. Set Size ($\downarrow$)	CSD ($\downarrow$)	Accuracy ($\uparrow$)	TE ($\downarrow$)	Coverage	Avg. Set Size ($\downarrow$)	CSD ($\downarrow$)
SplitCP	0.730 ± 0.011	0.543 ± 0.037	0.962 ± 0.006	3.517 ± 0.169	0.583 ± 0.032	0.730 ± 0.011	0.543 ± 0.037	0.905 ± 0.011	2.393 ± 0.280	0.604 ± 0.048
Post-hoc Union	0.730 ± 0.011	0.543 ± 0.037	0.979 ± 0.004	5.021 ± 0.238	0.000 ± 0.000	0.730 ± 0.011	0.543 ± 0.037	0.940 ± 0.010	3.559 ± 0.316	0.000 ± 0.000
CFU	<u>0.589 ± 0.017</u>	0.000 ± 0.000	0.932 ± 0.006	3.695 ± 0.126	0.000 ± 0.000	<u>0.589 ± 0.017</u>	0.000 ± 0.000	0.907 ± 0.014	2.973 ± 0.183	0.000 ± 0.000
CFR	<u>0.589 ± 0.017</u>	0.000 ± 0.000	0.931 ± 0.006	3.685 ± 0.134	0.000 ± 0.000	<u>0.589 ± 0.017</u>	0.000 ± 0.000	0.907 ± 0.014	2.973 ± 0.182	0.000 ± 0.000
PCF	0.571 ± 0.017	0.000 ± 0.000	0.928 ± 0.008	3.487 ± 0.145	0.000 ± 0.000	0.571 ± 0.017	0.000 ± 0.000	0.906 ± 0.014	2.859 ± 0.241	0.000 ± 0.000
CF-CP-mean	0.730 ± 0.011	0.543 ± 0.037	0.941 ± 0.004	<u>3.540 ± 0.119</u>	0.017 ± 0.005	0.730 ± 0.011	0.543 ± 0.037	0.927 ± 0.009	3.298 ± 0.160	0.025 ± 0.006
CF-CP-max	0.730 ± 0.011	0.543 ± 0.037	0.947 ± 0.005	4.130 ± 0.127	0.028 ± 0.005	0.730 ± 0.011	0.543 ± 0.037	0.942 ± 0.008	3.986 ± 0.147	0.034 ± 0.007
CF-CP-min	0.730 ± 0.011	0.543 ± 0.037	0.944 ± 0.004	3.690 ± 0.139	0.018 ± 0.004	0.730 ± 0.011	0.543 ± 0.037	0.912 ± 0.009	<u>2.948 ± 0.126</u>	0.003 ± 0.004

Table C.3: Synthetic classification results at $\alpha = 0.1$ (mean $\pm$ std over 10 runs) with APS and RAPS base scores with noiseless counterfactuals. Highlighting follows the convention described in the evaluation metrics paragraph. Since we do not use random sets for APS and RAPS, we relax the condition of achieving target coverage.

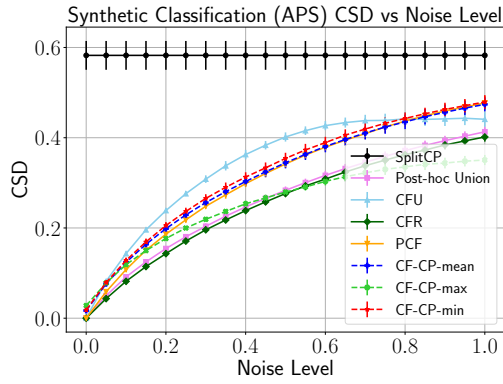
Method	APS					RAPS				
	Accuracy ($\uparrow$)	TE ($\downarrow$)	Coverage	Avg. Set Size ($\downarrow$)	CSD ($\downarrow$)	Accuracy ($\uparrow$)	TE ($\downarrow$)	Coverage	Avg. Set Size ($\downarrow$)	CSD ($\downarrow$)
SplitCP	0.730 ± 0.011	0.543 ± 0.037	0.962 ± 0.006	3.517 ± 0.169	0.583 ± 0.032	0.730 ± 0.011	0.543 ± 0.037	0.905 ± 0.011	2.393 ± 0.280	0.604 ± 0.048
Post-hoc Union	0.730 ± 0.011	0.543 ± 0.037	0.979 ± 0.004	4.945 ± 0.232	<u>0.247 ± 0.008</u>	0.730 ± 0.011	0.543 ± 0.037	0.938 ± 0.011	3.580 ± 0.322	0.258 ± 0.009
CFU	0.512 ± 0.011	0.235 ± 0.004	0.916 ± 0.012	4.007 ± 0.207	0.363 ± 0.010	0.512 ± 0.011	0.235 ± 0.004	0.904 ± 0.011	3.635 ± 0.239	0.343 ± 0.012
CFR	0.563 ± 0.012	0.140 ± 0.002	0.928 ± 0.008	3.790 ± 0.153	0.238 ± 0.005	0.563 ± 0.012	0.140 ± 0.002	0.909 ± 0.012	3.219 ± 0.173	0.212 ± 0.004
PCF	0.550 ± 0.015	0.165 ± 0.004	0.923 ± 0.010	3.484 ± 0.154	0.298 ± 0.008	0.550 ± 0.015	0.165 ± 0.004	0.907 ± 0.014	2.913 ± 0.244	0.278 ± 0.012
CF-CP-mean	0.730 ± 0.011	0.543 ± 0.037	0.945 ± 0.005	3.634 ± 0.118	0.303 ± 0.011	0.730 ± 0.011	0.543 ± 0.037	0.932 ± 0.007	3.511 ± 0.112	<u>0.228 ± 0.006</u>
CF-CP-max	0.730 ± 0.011	0.543 ± 0.037	0.952 ± 0.005	4.661 ± 0.203	0.254 ± 0.007	0.730 ± 0.011	0.543 ± 0.037	0.948 ± 0.006	4.410 ± 0.174	0.231 ± 0.008
CF-CP-min	0.730 ± 0.011	0.543 ± 0.037	0.947 ± 0.005	3.751 ± 0.126	0.313 ± 0.011	0.730 ± 0.011	0.543 ± 0.037	0.911 ± 0.010	2.991 ± 0.151	0.285 ± 0.017

Table C.4: Synthetic classification results at $\alpha = 0.1$ (mean $\pm$ std over 10 runs) with APS and RAPS base scores with noisy counterfactuals, where the noise level is $\sigma_U = 0.4$. Highlighting follows the convention described in the evaluation metrics paragraph.

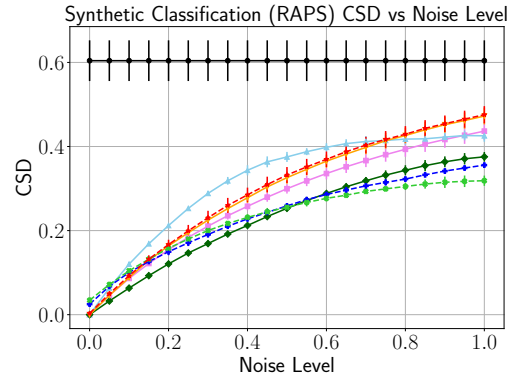
and $k_{\text{reg}} = 2$ in our experiments.

The results are shown in Table C.3 for oracle counterfactuals and Table C.4 for noisy counterfactuals. The results are consistent with those using LAC score in the main text. APS score tends to produce larger prediction sets compared to LAC score, while RAPS score produces smaller prediction sets thanks to the regularization. Since we use deterministic version of APS, the average coverage is slightly larger than the target coverage value for all the methods. Here again counterfactually fair methods (CFU, CFR, PCF) and CF-CP methods achieve a significant reduction in CSD compared to split CP, with only a modest increase in the size of the prediction sets. In the noiseless case, CF-CP methods has CSD slightly larger than 0, which is due to the constraint that the prediction sets must include at least one label. For completeness, we also report how CSD changes with different noise levels in Figure C.2 for APS and RAPS scores. The

results are consistent with those using LAC score in the main text. When the noise level is higher, all methods have higher CSD, but counterfactually fair methods and CF-CP methods still achieve a significant reduction in CSD compared to split CP.



(a) Synthetic Classification with APS



(b) Synthetic Classification with RAPS

Figure C.2: Counterfactual Set Disparity (CSD) versus the noise level in counterfactual generation for APS (left) and RAPS (right) base scores on the synthetic classification dataset, averaged over 10 runs.

Appendix D

Supplementary Materials for Chapter 5

D.1 Proofs

D.1.1 Proof of Lemma 5.11

In this section, we provide the proof of Lemma 5.11.

Proof. Under the assumption of infinite servers, the latency consists of the time spent waiting to complete the current batch and the service time. Therefore, the expected latency of a request is given by

$$\mathbb{E}[t_{\text{latency}}] = \mathbb{E}[t_{\text{batch}}] + \mathbb{E}[t_{\text{service}}]. \quad (\text{D.1})$$

The expected time spent waiting to complete the current batch is given by

$$\mathbb{E}[t_{\text{batch}}] = \sum_{i=1}^k \mathbb{P}(\text{bin} = i) \mathbb{E}[t_{\text{batch}} | \text{bin} = i]. \quad (\text{D.2})$$

Since the bins are equally likely, the arrival rate for each bin is λ/k . Then, for each request in the batch, the expected time spent waiting to complete the current batch is given by

$$\mathbb{E}[t_{\text{batch}}|\text{bin} = i] = \frac{1}{B} \sum_{j=1}^B \frac{(B-j)k}{\lambda} = \frac{B-1}{2\lambda}k. \quad (\text{D.3})$$

Then, the expected time spent waiting to complete the current batch is given by

$$\mathbb{E}[t_{\text{batch}}] = \frac{B-1}{2\lambda}k. \quad (\text{D.4})$$

The expected service time of a request is given by Theorem 5.6. Therefore, the expected latency of a request can be derived as in the statement of the lemma. $\blacksquare$

D.1.2 Exponentially Distributed Service Time

In this section, we provide the proof of Lemma 5.13.

Proof. The upper bound on the expected service time of a batch of B requests with k bins is derived based on the following observation.

$$\mathbb{E}[t_{\text{service}, k}|\text{bin} = i] \leq l_i \quad \forall i \in [k-1], \quad (\text{D.5})$$

and the exact expected service time of a batch of B requests with k bins is given by $l_{k-1} + \frac{H_B}{\mu}$. This is well-known in the literature, it is the maximum of shifted exponential random variables. Then, the expected service time of a batch of B requests with k bins is upper bounded by the sum of the expected service time of each bin. The upper bound could be written as follows,

$$\mathbb{E}[t_{\text{service}, k}] \leq \sum_{i=1}^{k-1} \Pr(\text{bin} = i)l_i + \Pr(\text{bin} = k) \left(l_{k-1} + \frac{H_B}{\mu} \right). \quad (\text{D.6})$$

The probability of each bin is given by $\Pr(\text{bin} = i) = \exp(-\mu l_{i-1}) - \exp(-\mu l_i)$. Then, we apply the following change of variables before minimizing the upper bound. Let $q_i = \exp(-\mu l_i)$ and $q_0 = 1$, then the upper bound can be written as,

$$\mathbb{E}[t_{\text{service}, k}] \leq \sum_{i=1}^{k-1} (q_{i-1} - q_i) \frac{\log(1/q_i)}{\mu} + q_{k-1} \left(\frac{\log(1/q_{k-1})}{\mu} + \frac{H_B}{\mu} \right) = f(\mathbf{q}). \quad (\text{D.7})$$

It can be seen the upper bound function can be decomposed as a function of $\mathbf{q} = [q_1, q_2, \dots, q_{k-1}]$ and a multiplicative factor of $1/\mu$. Therefore, we will assume that $\mu = 1$ for simplicity. Then, we can write the upper bound function as,

$$f(\mathbf{q}) = \sum_{i=1}^{k-1} (q_{i-1} - q_i) \log(1/q_i) + q_{k-1} (\log(1/q_{k-1}) + H_B) \quad (\text{D.8})$$

$$= \sum_{i=1}^{k-1} (q_i - q_{i-1}) \log(q_i) + q_{k-1} (H_B - \log(q_{k-1})). \quad (\text{D.9})$$

We can compute the partial derivative of $f(\mathbf{q})$ with respect to $q_i, i \in [k-2]$ as follows,

$$\frac{\partial f(\mathbf{q})}{\partial q_i} = \log(q_i) + \frac{q_i - q_{i-1}}{q_i} - \log(q_{i+1}) \quad \forall i \in [k-2]. \quad (\text{D.10})$$

Then, the partial derivative with respect to q_{k-1} is given by,

$$\frac{\partial f(\mathbf{q})}{\partial q_{k-1}} = H_B - \frac{q_{k-2}}{q_{k-1}}. \quad (\text{D.11})$$

Then, the second-order partial derivative of $f(\mathbf{q})$ with respect to $q_i q_j, i, j \in [k-1]$ is

given by,

$$\frac{\partial^2 f(\mathbf{q})}{\partial q_i \partial q_j} = \begin{cases} \frac{1}{q_i} + \frac{q_{i-1}}{q_i^2} & \text{if } i = j \text{ and } i \in [k-2], \\ \frac{q_{i-1}}{q_i^2} & \text{if } i = j = k-1, \\ -\frac{1}{q_{\max(i,j)}} & \text{if } |i-j| = 1, \\ 0 & \text{otherwise.} \end{cases} \quad (\text{D.12})$$

Then, the Hessian matrix of $f(\mathbf{q})$ is a tridiagonal matrix in the form of,

$$\nabla^2 f(\mathbf{q}) = \begin{bmatrix} \frac{q_1+q_0}{q_1^2} & -\frac{1}{q_2} & \dots & 0 & 0 \\ -\frac{1}{q_2} & \frac{q_2+q_1}{q_2^2} & \dots & 0 & 0 \\ 0 & -\frac{1}{q_3} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \frac{q_{k-1}+q_{k-2}}{q_{k-1}^2} & -\frac{1}{q_{k-1}} \\ 0 & 0 & \dots & -\frac{1}{q_{k-1}} & \frac{q_{k-2}}{q_{k-1}^2} \end{bmatrix}. \quad (\text{D.13})$$

The determinant of the matrix can be found using the following recursive formula:

$$f_n = A_{n,n}f_{n-1} - A_{n,n-1}A_{n-1,n}f_{n-2} \quad \forall n \in [2, k-1] \quad (\text{D.14})$$

where $f_1 = A_{1,1}$ and $f_0 = 1$. For all $n \in [2, k-2]$, it can be seen that:

$$f_n = \frac{q_{n-1} + q_n}{q_n^2} f_{n-1} - \frac{1}{q_n^2} f_{n-2} \quad (\text{D.15})$$

Our claim is that:

$$f_n = \frac{1}{q_1 q_2 \dots q_{n-1} q_n^2} + \frac{1}{q_n} f_{n-1} \quad \forall n \in [2, k-2] \quad (\text{D.16})$$

It holds for $n = 1$. Then, we can prove it by induction. Assume that it holds for $n - 1$.

Then, we can write the following:

$$f_n = \frac{q_{n-1} + q_n}{q_n^2} f_{n-1} - \frac{1}{q_n^2} f_{n-2} \quad (\text{D.17})$$

$$= \frac{q_{n-1} + q_n}{q_n^2} \left(\frac{1}{q_1 q_2 \cdots q_{n-2} q_{n-1}^2} + \frac{1}{q_{n-1}} f_{n-2} \right) - \frac{1}{q_n^2} f_{n-2} \quad (\text{D.18})$$

$$= \frac{1}{q_1 q_2 \cdots q_{n-1} q_n^2} + \frac{1}{q_n} f_{n-1} \quad (\text{D.19})$$

Hence, it is proven by induction.

Then, we can compute the determinant of the Hessian as:

$$\det(\nabla^2 f) = f_{k-1} = \frac{q_{k-2}}{q_{k-1}^2} f_{k-2} - \frac{1}{q_{k-1}^2} f_{k-3} \quad (\text{D.20})$$

We can replace the f_{k-2} with the formula:

$$f_{k-2} = \frac{1}{q_1 q_2 \cdots q_{k-3} q_{k-2}^2} + \frac{1}{q_{k-2}} f_{k-3} \quad (\text{D.21})$$

Then, we can compute the determinant of the matrix as:

$$\det(\nabla^2 f) = \frac{q_{k-2}}{q_{k-1}^2} \left(\frac{1}{q_1 q_2 \cdots q_{k-3} q_{k-2}^2} + \frac{1}{q_{k-2}} f_{k-3} \right) - \frac{1}{q_{k-1}^2} f_{k-3} \quad (\text{D.22})$$

$$= \frac{1}{q_1 q_2 \cdots q_{k-2} q_{k-1}^2} \quad (\text{D.23})$$

Therefore, the determinant of the Hessian matrix is as follows:

$$\det(\nabla^2 f) = \frac{1}{q_1 q_2 \cdots q_{k-2} q_{k-1}^2} > 0 \quad (\text{D.24})$$

Since all q_i are positive, the recursion above shows that all leading principal minors of the Hessian are positive. Therefore, by Sylvester's criterion, the Hessian is positive definite,

and the upper bound for the total service time is a convex function of the transformed decision variables $\mathbf{q}$. Then, the optimal decision boundaries can be found by setting the partial derivative of the upper bound function to zero. We can start from the partial derivative with respect to q_{k-1} as follows:

$$\frac{\partial f}{\partial q_{k-1}} = H_B - \frac{q_{k-2}}{q_{k-1}} = 0 \implies q_{k-2} = q_{k-1} H_B \quad (\text{D.25})$$

Then, we can compute the partial derivative with respect to $q_i, i \in [k-2]$ as follows:

$$\frac{\partial f}{\partial q_i} = \log(q_i) + \frac{q_i - q_{i-1}}{q_i} - \log(q_{i+1}) = 0 \implies \frac{q_{i-1}}{q_i} = 1 + \log\left(\frac{q_i}{q_{i+1}}\right) \quad (\text{D.26})$$

Utilizing the above equation and $q_0 = 1$, we can derive,

$$\frac{1}{q_1} = 1 + \log\left(\frac{q_1}{q_2}\right) \quad (\text{D.27})$$

$$= 1 + \log\left(1 + \log\left(\frac{q_2}{q_3}\right)\right) \quad (\text{D.28})$$

$$= 1 + \log\left(1 + \log\left(1 + \dots + \log\left(\frac{q_{k-2}}{q_{k-1}}\right)\right)\right) \quad (\text{D.29})$$

$$= 1 + \log(1 + \log(1 + \dots + \log(H_B))) = L_{k-1} \implies q_1 = \frac{1}{L_{k-1}} \quad (\text{D.30})$$

where L_{k-1} is defined recursively as:

$$L_m = \begin{cases} H_B & \text{if } m = 1 \\ 1 + \log(L_{m-1}) & \text{if } m > 1 \end{cases} \quad (\text{D.31})$$

Similarly, we can derive q_i for $i \in [k-1]$ as follows:

$$q_i = \frac{1}{\prod_{j=1}^i L_{k-j}} \quad \forall i \in [k-1] \quad (\text{D.32})$$

Then, the optimal decision boundaries are given by,

$$l_i = -\frac{1}{\mu} \log(q_i) = \frac{1}{\mu} \sum_{j=1}^i \log(L_{k-j}) \quad \forall i \in [k-1]. \quad (\text{D.33})$$

This completes the proof. ■

Bibliography

- [1] Y. LeCun, Y. Bengio, and G. E. Hinton, *Deep learning*, *Nature* **521** (2015), no. 7553 436–444.
- [2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, *Attention is all you need*, in *Advances in Neural Information Processing Systems*, vol. 30, pp. 5998–6008, 2017.
- [3] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, *Language models are few-shot learners*, in *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2020.
- [4] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, *et. al.*, *On the opportunities and risks of foundation models*, *arXiv preprint arXiv:2108.07258* (2021).
- [5] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. S. Zemel, *Fairness through awareness*, in *Innovations in Theoretical Computer Science (ITCS)*, pp. 214–226, 2012.
- [6] M. B. Zafar, I. Valera, M. Gomez-Rodriguez, and K. P. Gummadi, *Fairness constraints: Mechanisms for fair classification*, in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 962–970, 2017.
- [7] M. Hardt, E. Price, and N. Srebro, *Equality of opportunity in supervised learning*, in *Advances in Neural Information Processing Systems*, vol. 29, pp. 3315–3323, 2016.
- [8] L. T. Liu, S. Dean, E. Rolf, M. Simchowitz, and M. Hardt, *Delayed impact of fair machine learning*, in *International Conference on Machine Learning (ICML)*, pp. 3150–3158, 2018.

- [9] T. B. Hashimoto, M. Srivastava, H. Namkoong, and P. Liang, *Fairness without demographics in repeated loss minimization*, in *International Conference on Machine Learning (ICML)*, pp. 1929–1938, 2018.
- [10] S. Kannan, A. Roth, and J. Ziani, *Downstream effects of affirmative action*, in *Proceedings of the Conference on Fairness, Accountability, and Transparency (FAT*)*, pp. 240–248, 2019.
- [11] M. J. Kusner, J. R. Loftus, C. Russell, and R. Silva, *Counterfactual fairness*, in *Advances in Neural Information Processing Systems*, vol. 30, pp. 4066–4076, 2017.
- [12] V. Vovk, A. Gammerman, and G. Shafer, *Algorithmic Learning in a Random World*. Springer, 2005.
- [13] A. N. Angelopoulos and S. Bates, *Conformal prediction: A gentle introduction*, *Foundations and Trends in Machine Learning* **16** (2023), no. 4 494–591.
- [14] G. Yu, J. S. Jeong, G. Kim, S. Kim, and B. Chun, *Orca: A distributed serving system for transformer-based generative models*, in *USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, pp. 521–538, 2022.
- [15] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. Gonzalez, H. Zhang, and I. Stoica, *Efficient memory management for large language model serving with PagedAttention*, in *Proceedings of the Symposium on Operating Systems Principles (SOSP)*, pp. 611–626, 2023.
- [16] O. Guldogan, Y. Zeng, J. Sohn, R. Pedarsani, and K. Lee, *Equal improvability: A new fairness notion considering the long-term impact*, in *International Conference on Learning Representations (ICLR)*, 2023.
- [17] B. Puranik, O. Guldogan, U. Madhow, and R. Pedarsani, *Long-term fairness in sequential multi-agent selection with positive reinforcement*, *IEEE Journal on Selected Areas in Information Theory* **5** (2024) 424–441.
- [18] O. Guldogan, N. Sarna, Y. Li, and M. Berger, *Counterfactually fair conformal prediction*, in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2026.
- [19] O. Guldogan, J. Kunde, K. Lee, and R. Pedarsani, *Multi-bin batching for increasing LLM inference throughput*, *arXiv preprint arXiv:2412.04504* (2024).
- [20] M. B. Zafar, I. Valera, M. Gomez-Rodriguez, K. P. Gummadi, and A. Weller, *From parity to preference-based notions of fairness in classification*, in *Advances in Neural Information Processing Systems*, vol. 30, pp. 229–239, 2017.

- [21] M. B. Zafar, I. Valera, M. Gomez-Rodriguez, and K. P. Gummadi, *Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment*, in *Proceedings of the International Conference on World Wide Web (WWW)*, pp. 1171–1180, 2017.
- [22] V. Gupta, P. Nokhiz, C. D. Roy, and S. Venkatasubramanian, *Equalizing recourse across groups*, *arXiv preprint arXiv:1909.03166* (2019).
- [23] H. Heidari, V. Nanda, and K. P. Gummadi, *On the long-term impact of algorithmic decision policies: Effort unfairness and feature segregation through social learning*, in *International Conference on Machine Learning (ICML)*, pp. 2692–2701, 2019.
- [24] J. von Kügelgen, A. Karimi, U. Bhatt, I. Valera, A. Weller, and B. Schölkopf, *On the fairness of causal algorithmic recourse*, in *AAAI Conference on Artificial Intelligence*, pp. 9584–9594, 2022.
- [25] Y. Chen, J. Wang, and Y. Liu, *Linear classifiers that encourage constructive adaptation*, *arXiv preprint arXiv:2011.00355* (2020).
- [26] W. Huang, Y. Wu, L. Zhang, and X. Wu, *Fairness through equality of effort*, in *Companion Proceedings of The Web Conference*, pp. 743–751, 2020.
- [27] Y. Wu, L. Zhang, and X. Wu, *Counterfactual fairness: Unidentification, bound and algorithm*, in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1438–1444, 2019.
- [28] M. Donini, L. Oneto, S. Ben-David, J. Shawe-Taylor, and M. Pontil, *Empirical risk minimization under fairness constraints*, in *Advances in Neural Information Processing Systems*, vol. 31, pp. 2796–2806, 2018.
- [29] J. Cho, G. Hwang, and C. Suh, *A fair classifier using kernel density estimation*, in *Advances in Neural Information Processing Systems*, vol. 33, pp. 15088–15099, 2020.
- [30] J. M. Danskin, *The Theory of Max-Min and its Application to Weapons Allocation Problems*. Springer, 1967.
- [31] Y. Roh, K. Lee, S. E. Whang, and C. Suh, *Fairbatch: Batch selection for model fairness*, in *International Conference on Learning Representations (ICLR)*, 2021.
- [32] A. Shen, X. Han, T. Cohn, T. Baldwin, and L. Frermann, *Optimising equal opportunity fairness in model training*, in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pp. 4073–4084, 2022.

- [33] H. Hofmann, “Statlog (German Credit Data).” UCI Machine Learning Repository, 1994. DOI: <https://doi.org/10.24432/C5NC77>.
- [34] F. Ding, M. Hardt, J. Miller, and L. Schmidt, *Retiring adult: New datasets for fair machine learning*, in *Advances in Neural Information Processing Systems*, vol. 34, pp. 6478–6490, 2021.
- [35] H. Jiang and O. Nachum, *Identifying and correcting label bias in machine learning*, in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 702–712, 2020.
- [36] F. Kamiran and T. Calders, *Data preprocessing techniques for classification without discrimination*, *Knowledge and Information Systems* **33** (2012), no. 1 1–33.
- [37] F. Kamiran and T. Calders, *Classification with no discrimination by preferential sampling*, in *Proceedings of the Annual Machine Learning Conference of Belgium and The Netherlands*, pp. 1–6, 2010.
- [38] P. Gordaliza, E. del Barrio, F. Gamboa, and J. Loubes, *Obtaining fairness using optimal transport theory*, in *International Conference on Machine Learning (ICML)*, pp. 2357–2365, 2019.
- [39] K. Fukuchi, J. Sakuma, and T. Kamishima, *Prediction with model-based neutrality*, in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD)*, Lecture Notes in Computer Science, pp. 499–514, Springer, 2013.
- [40] T. Kamishima, S. Akaho, H. Asoh, and J. Sakuma, *Fairness-aware classifier with prejudice remover regularizer*, in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD)*, Lecture Notes in Computer Science, pp. 35–50, Springer, 2012.
- [41] T. Calders and S. Verwer, *Three naive bayes approaches for discrimination-free classification*, *Data Mining and Knowledge Discovery* **21** (2010), no. 2 277–292.
- [42] B. H. Zhang, B. Lemoine, and M. Mitchell, *Mitigating unwanted biases with adversarial learning*, in *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society (AIES)*, pp. 335–340, 2018.
- [43] Y. Roh, K. Lee, S. Whang, and C. Suh, *Fr-train: A mutual information-based approach to fair and robust training*, in *International Conference on Machine Learning (ICML)*, pp. 8147–8157, 2020.

- [44] W. Alghamdi, S. Asodeh, H. Wang, F. P. Calmon, D. Wei, and K. N. Ramamurthy, *Model projection: Theory and applications to fair machine learning*, in *IEEE International Symposium on Information Theory (ISIT)*, pp. 2711–2716, 2020.
- [45] D. Wei, K. N. Ramamurthy, and F. P. Calmon, *Optimized score transformation for fair classification*, in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 1673–1683, 2020.
- [46] B. E. Woodworth, S. Gunasekar, M. I. Ohannessian, and N. Srebro, *Learning non-discriminatory predictors*, in *Conference on Learning Theory (COLT)*, pp. 1920–1953, 2017.
- [47] X. Zhang, R. Tu, Y. Liu, M. Liu, H. Kjellstrom, K. Zhang, and C. Zhang, *How do fair decisions fare in long-term qualification?*, in *Advances in Neural Information Processing Systems*, vol. 33, pp. 18457–18469, 2020.
- [48] L. Li, T. Lassiter, J. Oh, and M. K. Lee, *Algorithmic hiring in practice: Recruiter and HR professional’s perspectives on AI use in hiring*, in *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society (AIES)*, pp. 166–176, 2021.
- [49] J. Dressel and H. Farid, *The accuracy, fairness, and limits of predicting recidivism*, *Science Advances* **4** (2018), no. 1 eaao5580.
- [50] J. A. Berkovec, G. B. Canner, S. A. Gabriel, and T. H. Hannan, *Mortgage discrimination and FHA loan performance*, *Cityscape: A Journal of Policy Development and Research* **2** (1996), no. 1 9–24.
- [51] R. S. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork, *Learning fair representations*, in *International Conference on Machine Learning (ICML)*, pp. 325–333, 2013.
- [52] T. Morgenroth, M. K. Ryan, and K. Peters, *The motivational theory of role modeling: How role models influence role aspirants’ goals*, *Review of General Psychology* **19** (2015), no. 4 465–483.
- [53] E. P. Bettinger and B. T. Long, *Do faculty serve as role models? the impact of instructor gender on female students*, *American Economic Review* **95** (2005), no. 2 152–157.
- [54] J. Cho, G. Hwang, and C. Suh, *A fair classifier using mutual information*, in *IEEE International Symposium on Information Theory (ISIT)*, pp. 2521–2526, 2020.
- [55] F. Calmon, D. Wei, B. Vinzamuri, K. Natesan Ramamurthy, and K. R. Varshney, *Optimized pre-processing for discrimination prevention*, in *Advances in Neural Information Processing Systems*, vol. 30, pp. 3992–4001, 2017.

- [56] A. Ghassami, S. Khodadadian, and N. Kiyavash, *Fairness in supervised learning: An information theoretic approach*, in *IEEE International Symposium on Information Theory (ISIT)*, pp. 176–180, 2018.
- [57] P. J. Kenfack, A. R. Rivera, A. M. Khan, and M. Mazzara, *Learning fair representations through uniformly distributed sensitive attributes*, in *IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pp. 58–67, 2023.
- [58] P. Kairouz, J. Liao, C. Huang, M. Vyas, M. Welfert, and L. Sankar, *Generating fair universal representations using adversarial models*, *IEEE Transactions on Information Forensics and Security* **17** (2022) 1970–1985.
- [59] X. Shen, Y. Wong, and M. S. Kankanhalli, *Fair representation: Guaranteeing approximate multiple group fairness for unknown tasks*, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **45** (2023), no. 1 525–538.
- [60] S. Jabbari, M. Joseph, M. J. Kearns, J. Morgenstern, and A. Roth, *Fairness in reinforcement learning*, in *International Conference on Machine Learning (ICML)*, pp. 1617–1626, 2017.
- [61] M. Wen, O. Bastani, and U. Topcu, *Algorithms for fairness in sequential decision making*, in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 1144–1152, 2021.
- [62] H. Heidari and A. Krause, *Preventing disparate treatment in sequential decision making*, in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2248–2254, 2018.
- [63] T. Yin, R. Raab, M. Liu, and Y. Liu, *Long-term fairness with unknown dynamics*, in *Advances in Neural Information Processing Systems*, vol. 36, pp. 55110–55139, 2023.
- [64] M. Joseph, M. J. Kearns, J. Morgenstern, S. Neel, and A. Roth, *Meritocratic fairness for infinite and contextual bandits*, in *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society (AIES)*, pp. 158–163, 2018.
- [65] Y. Chen, A. Cuellar, H. Luo, J. Modi, H. Nemlekar, and S. Nikolaidis, *Fair contextual multi-armed bandits: Theory and experiments*, in *Conference on Uncertainty in Artificial Intelligence, (UAI)*, pp. 181–190, 2020.
- [66] S. Gillen, C. Jung, M. J. Kearns, and A. Roth, *Online learning with an unknown fairness metric*, in *Advances in Neural Information Processing Systems*, vol. 31, pp. 2605–2614, 2018.

- [67] V. Patil, G. Ghalme, V. Nair, and Y. Narahari, *Achieving fairness in the stochastic multi-armed bandit problem*, *Journal of Machine Learning Research* **22** (2021), no. 174 1–31.
- [68] G. Ghalme, V. Nair, V. Patil, and Y. Zhou, *Long-term resource allocation fairness in average markov decision process (AMDP) environment*, in *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 525–533, 2022.
- [69] H. Mouzannar, M. I. Ohannessian, and N. Srebro, *From fair decision making to social equality*, in *Proceedings of the Conference on Fairness, Accountability, and Transparency, (FAT*)*, pp. 359–368, 2019.
- [70] J. Williams and J. Z. Kolter, *Dynamic modeling and equilibria in fair decision making*, *arXiv preprint arXiv:1911.06837* (2019).
- [71] X. Zhang, M. Khaliligarekani, C. Tekin, and M. Liu, *Group retention when using machine learning in sequential decision making: the interplay between user dynamics and fairness*, in *Advances in Neural Information Processing Systems*, vol. 32, pp. 15243–15252, 2019.
- [72] B. Puranik, U. Madhow, and R. Pedarsani, *A dynamic decision-making framework promoting long-term fairness*, in *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society (AIES)*, pp. 547–556, 2022.
- [73] X. Dai and M. I. Jordan, *Learning in multi-stage decentralized matching markets*, in *Advances in Neural Information Processing Systems*, vol. 34, pp. 12798–12809, 2021.
- [74] S. Cho, T. Todo, and M. Yokoo, *Two-sided matching over social networks*, in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 186–193, 2022.
- [75] University of California Office of the President, “California master plan for higher education.” <https://web.archive.org/web/20240318074731/https://www.ucop.edu/institutional-research-academic-planning/content-analysis/academic-planning/california-master-plan.html>, 2024. Accessed: March 2024.
- [76] L. F. Wightman, *LSAC national longitudinal bar passage study*, tech. rep., Law School Admission Council, Newtown, PA, 1998.
- [77] M. J. Kearns, S. Neel, A. Roth, and Z. S. Wu, *Preventing fairness gerrymandering: Auditing and learning for subgroup fairness*, in *International Conference on Machine Learning (ICML)*, pp. 2564–2572, 2018.

- [78] S. Barocas and A. D. Selbst, *Big data’s disparate impact*, *California Law Review* **104** (2016), no. 3 671–732.
- [79] S. Corbett-Davies, J. D. Gaebler, H. Nilforoshan, R. Shroff, and S. Goel, *The measure and mismeasure of fairness*, *Journal of Machine Learning Research* **24** (2023), no. 312 1–117.
- [80] R. Berk, H. Heidari, S. Jabbari, M. Kearns, and A. Roth, *Fairness in criminal justice risk assessments: The state of the art*, *Sociological Methods & Research* **50** (2021), no. 1 3–44.
- [81] A. Chouldechova, *Fair prediction with disparate impact: A study of bias in recidivism prediction instruments*, *Big Data* **5** (2017), no. 2 153–163.
- [82] A. Agarwal, M. Dudík, and Z. S. Wu, *Fair regression: Quantitative definitions and reduction-based algorithms*, in *International Conference on Machine Learning (ICML)*, pp. 120–129, 2019.
- [83] J. C. Cresswell, Y. Sui, B. Kumar, and N. Vouitsis, *Conformal prediction sets improve human decision making*, in *International Conference on Machine Learning (ICML)*, pp. 9439–9457, 2024.
- [84] Y. Romano, R. F. Barber, C. Sabatti, and E. Candès, *With malice toward none: Assessing uncertainty via equalized coverage*, *Harvard Data Science Review* **2** (2020), no. 2.
- [85] M. Liu, L. Ding, D. Yu, W. Liu, L. Kong, and B. Jiang, *Conformalized fairness via quantile regression*, in *Advances in Neural Information Processing Systems*, vol. 35, pp. 11561–11572, 2022.
- [86] A. T. Vadlamani, A. Srinivasan, P. Maneriker, A. Payani, and S. Parthasarathy, *A generic framework for conformal fairness*, in *International Conference on Learning Representations (ICLR)*, 2025.
- [87] J. C. Cresswell, B. Kumar, Y. Sui, and M. Belbahri, *Conformal prediction sets can cause disparate impact*, in *International Conference on Learning Representations (ICLR)*, 2025.
- [88] V. Azimi and M. A. Zaydman, *Optimizing equity: Working towards fair machine learning algorithms in laboratory medicine*, *The Journal of Applied Laboratory Medicine* **8** (2023), no. 1 113–128.
- [89] F. Wang, L. Cheng, R. Guo, K. Liu, and P. S. Yu, *Equal opportunity of coverage in fair regression*, in *Advances in Neural Information Processing Systems*, vol. 36, pp. 7743–7755, 2023.

- [90] C. Lu, A. Lemay, K. Chang, K. Höbel, and J. Kalpathy-Cramer, *Fair conformal predictors for applications in medical imaging*, in *AAAI Conference on Artificial Intelligence*, pp. 12008–12016, 2022.
- [91] L. Lei and E. J. Candès, *Conformal inference of counterfactuals and individual treatment effects*, *Journal of the Royal Statistical Society Series B: Statistical Methodology* **83** (2021), no. 5 911–938.
- [92] M. Schröder, D. Frauen, J. Schweisthal, K. Heß, V. Melnychuk, and S. Feuerriegel, *Conformal prediction for causal effects of continuous treatments*, in *Advances in Neural Information Processing Systems*, vol. 38, pp. 1500–1539, 2025.
- [93] Z. Chen, R. Guo, J. Ton, and Y. Liu, *Conformal counterfactual inference under hidden confounding*, in *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 397–408, 2024.
- [94] Z. Zuo, M. Khalili, and X. Zhang, *Counterfactually fair representation*, in *Advances in Neural Information Processing Systems*, vol. 36, pp. 12124–12140, 2023.
- [95] Z. Zhou, T. Liu, R. Bai, J. Gao, M. Kocaoglu, and D. I. Inouye, *Counterfactual fairness by combining factual and counterfactual predictions*, in *Advances in Neural Information Processing Systems*, vol. 37, pp. 47876–47907, 2024.
- [96] P. G. Di Stefano, J. M. Hickey, and V. Vasileiou, *Counterfactual fairness: removing direct effects through regularization*, *arXiv preprint arXiv:2002.10774* (2020).
- [97] S. Garg, V. Perot, N. Limtiaco, A. Taly, E. H. Chi, and A. Beutel, *Counterfactual fairness in text classification through robustness*, in *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society (AIES)*, pp. 219–226, 2019.
- [98] J. Pearl, *Causality*. Cambridge University Press, 2 ed., 2009.
- [99] G. Shafer and V. Vovk, *A tutorial on conformal prediction*, *Journal of Machine Learning Research* **9** (2008), no. 12 371–421.
- [100] H. Papadopoulos, K. Proedrou, V. Vovk, and A. Gammerman, *Inductive confidence machines for regression*, in *European Conference on Machine Learning (ECML)*, Lecture Notes in Computer Science, pp. 345–356, 2002.
- [101] J. Lei, M. G’Sell, A. Rinaldo, R. J. Tibshirani, and L. Wasserman, *Distribution-free predictive inference for regression*, *Journal of the American Statistical Association* **113** (2018), no. 523 1094–1111.

- [102] A. Nasr-Esfahany, M. Alizadeh, and D. Shah, *Counterfactual identifiability of bijective causal models*, in *International Conference on Machine Learning (ICML)*, pp. 25733–25754, 2023.
- [103] Z. Zhou, R. Bai, S. Kulinski, M. Kocaoglu, and D. I. Inouye, *Towards characterizing domain counterfactuals for invertible latent causal models*, in *International Conference on Learning Representations (ICLR)*, 2024.
- [104] M. Sadinle, J. Lei, and L. Wasserman, *Least ambiguous set-valued classifiers with bounded error levels*, *Journal of the American Statistical Association* **114** (2019), no. 525 223–234.
- [105] M. De-Arteaga, A. Romanov, H. M. Wallach, J. T. Chayes, C. Borgs, A. Chouldechova, S. C. Geyik, K. Kenthapadi, and A. T. Kalai, *Bias in bios: A case study of semantic representation bias in a high-stakes setting*, in *Proceedings of the Conference on Fairness, Accountability, and Transparency (FAT*)*, pp. 120–128, 2019.
- [106] P. Lemberger and A. Saillenfest, *Explaining text classifiers with counterfactual representations*, in *European Conference on Artificial Intelligence (ECAI)*, Frontiers in Artificial Intelligence and Applications, pp. 890–897, IOS Press, 2024.
- [107] J. Devlin, M. Chang, K. Lee, and K. Toutanova, *BERT: pre-training of deep bidirectional transformers for language understanding*, in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 4171–4186, 2019.
- [108] S. Chiappa, *Path-specific counterfactual fairness*, in *AAAI Conference on Artificial Intelligence*, pp. 7801–7808, 2019.
- [109] J. Li, T. Tang, W. X. Zhao, J. Nie, and J. Wen, *Pre-trained language models for text generation: A survey*, *ACM Computing Surveys* **56** (2024), no. 9 230:1–230:39.
- [110] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. D. O. Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, *et. al.*, *Evaluating large language models trained on code*, *arXiv preprint arXiv:2107.03374* (2021).
- [111] Z. Jiang, J. Araki, H. Ding, and G. Neubig, *How can we know When language models know? on the calibration of language models for question answering*, *Transactions of the Association for Computational Linguistics* **9** (2021) 962–977.
- [112] Y. Sheng, L. Zheng, B. Yuan, Z. Li, M. Ryabinin, B. Chen, P. Liang, C. Ré, I. Stoica, and C. Zhang, *Flexgen: High-throughput generative inference of large language models with a single GPU*, in *International Conference on Machine Learning (ICML)*, pp. 31094–31116, 2023.

- [113] Y. Jin, C. Wu, D. Brooks, and G. Wei, S^3 : *Increasing GPU utilization during generative inference for higher throughput*, in *Advances in Neural Information Processing Systems*, vol. 36, pp. 18015–18027, 2023.
- [114] M. Abdin, J. Aneja, H. Awadallah, A. Awadallah, A. A. Awan, *et. al.*, *Phi-3 technical report: A highly capable language model locally on your phone*, *arXiv preprint arXiv:2404.14219* (2024).
- [115] B. Wu, Y. Zhong, Z. Zhang, S. Liu, F. Liu, Y. Sun, G. Huang, X. Liu, and X. Jin, *Fast distributed inference serving for large language models*, *arXiv preprint arXiv:2305.05920* (2023).
- [116] Y. Inoue, *Queueing analysis of gpu-based inference servers with dynamic batching: A closed-form characterization*, *Performance Evaluation* **147** (2021) 102183.
- [117] K. Cheng, W. Hu, Z. Wang, H. Peng, J. Li, and S. Zhang, *Slice-level scheduling for high throughput and load balanced LLM serving*, *arXiv preprint arXiv:2406.13511* (2024).
- [118] B. Sun, Z. Huang, H. Zhao, W. Xiao, X. Zhang, Y. Li, and W. Lin, *Llumnix: Dynamic scheduling for large language model serving*, in *USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, pp. 173–191, 2024.
- [119] A. Agrawal, N. Kedia, A. Panwar, J. Mohan, N. Kwatra, B. S. Gulavani, A. Tumanov, and R. Ramjee, *Taming throughput-latency tradeoff in LLM inference with sarathi-serve*, in *USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, pp. 117–134, 2024.
- [120] P. Patel, E. Choukse, C. Zhang, A. Shah, Í. Goiri, S. Maleki, and R. Bianchini, *Splitwise: Efficient generative LLM inference using phase splitting*, in *ACM/IEEE Annual International Symposium on Computer Architecture (ISCA)*, pp. 118–132, 2024.
- [121] Y. Yang, L. Jiao, and Y. Xu, *A queueing theoretic perspective on low-latency LLM inference with variable token length*, in *International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, pp. 273–280, 2024.
- [122] A. Patke, D. Reddy, S. Jha, H. Qiu, C. Pinto, C. Narayanaswami, Z. Kalbarczyk, and R. K. Iyer, *Queue management for SLO-oriented large language model serving*, in *Proceedings of the ACM Symposium on Cloud Computing (SoCC)*, pp. 18–35, 2024.
- [123] Z. Zheng, X. Ren, F. Xue, Y. Luo, X. Jiang, and Y. You, *Response length perception and sequence scheduling: An LLM-empowered LLM inference pipeline*,

- in *Advances in Neural Information Processing Systems*, vol. 36, pp. 65517–65530, 2023.
- [124] Y. Fu, S. Zhu, R. Su, A. Qiao, I. Stoica, and H. Zhang, *Efficient LLM scheduling by learning to rank*, in *Advances in Neural Information Processing Systems*, vol. 37, pp. 59006–59029, 2024.
 - [125] H. Qiu, W. Mao, A. Patke, S. Cui, S. Jha, C. Wang, H. Franke, Z. T. Kalbarczyk, T. Başar, and R. K. Iyer, *Efficient interactive LLM serving with proxy model-based sequence length prediction*, in *International Workshop on Cloud Intelligence / AIOps at ASPLOS*, pp. 1–7, 2024.
 - [126] K. Cheng, W. Hu, Z. Wang, P. Du, J. Li, and S. Zhang, *Enabling efficient batch serving for LMaaS via generation length prediction*, in *IEEE International Conference on Web Services, (ICWS)*, pp. 853–864, 2024.
 - [127] X. Lin, Z. Zhang, P. Yue, H. Li, J. Zhang, B. Fan, H. Su, and X. Gong, *SyncIntellects: Orchestrating LLM inference with progressive prediction and QoS-friendly control*, in *IEEE/ACM International Symposium on Quality of Service (IWQoS)*, pp. 1–10, 2024.
 - [128] T. Dettmers, M. Lewis, Y. Belkada, and L. Zettlemoyer, *GPT3.int8(): 8-bit matrix multiplication for transformers at scale*, in *Advances in Neural Information Processing Systems*, vol. 35, pp. 30318–30332, 2022.
 - [129] E. Frantar, S. Ashkboos, T. Hoefler, and D. Alistarh, *OPTQ: accurate quantization for generative pre-trained transformers*, in *International Conference on Learning Representations (ICLR)*, 2023.
 - [130] G. Xiao, J. Lin, M. Seznec, H. Wu, J. Demouth, and S. Han, *SmoothQuant: Accurate and efficient post-training quantization for large language models*, in *International Conference on Machine Learning (ICML)*, pp. 38087–38099, 2023.
 - [131] J. Lin, J. Tang, H. Tang, S. Yang, W.-M. Chen, W.-C. Wang, G. Xiao, X. Dang, C. Gan, and S. Han, *AWQ: Activation-aware weight quantization for on-device LLM compression and acceleration*, in *Proceedings of Machine Learning and Systems*, vol. 6, pp. 87–100, 2024.
 - [132] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, *QLoRA: Efficient finetuning of quantized LLMs*, in *Advances in Neural Information Processing Systems*, vol. 36, pp. 10088–10115, 2023.
 - [133] H. Kang, Q. Zhang, S. Kundu, G. Jeong, Z. Liu, T. Krishna, and T. Zhao, *GEAR: An efficient KV cache compression recipe for near-lossless generative inference of LLM*, *arXiv preprint arXiv:2403.05527* (2024).

- [134] L. Zheng, L. Yin, Z. Xie, C. Sun, J. Huang, C. H. Yu, S. Cao, C. Kozyrakis, I. Stoica, J. E. Gonzalez, C. W. Barrett, and Y. Sheng, *SGLang: Efficient execution of structured language model programs*, in *Advances in Neural Information Processing Systems*, vol. 37, pp. 62557–62583, 2024.
- [135] P. C. Joshi, *Recurrence relations between moments of order statistics from exponential and truncated exponential distributions*, *Sankhyā: The Indian Journal of Statistics, Series B* **39** (1978), no. 4 362–371.
- [136] M. Conover, M. Hayes, A. Mathur, J. Xie, J. Wan, S. Shah, A. Ghodsi, P. Wendell, M. Zaharia, and R. Xin, “Free dolly: Introducing the world’s first truly open instruction-tuned LLM.” <https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm>, 2023. Accessed: May, 2026.
- [137] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan, *et. al.*, *The Llama 3 herd of models*, *arXiv preprint arXiv:2407.21783* (2024).
- [138] H. Xie, Y. Chen, L. Wang, L. Hu, and D. Wang, *Predicting LLM output length via entropy-guided representations*, in *International Conference on Learning Representations (ICLR)*, 2026.
- [139] D. S. Mitrinovic and P. M. Vasic, *Analytic inequalities*. Springer, 1970.
- [140] V. Cherepanova, V. Nanda, M. Goldblum, J. P. Dickerson, and T. Goldstein, *Technical challenges for training fair neural networks*, *arXiv preprint arXiv:2102.06764* (2021).
- [141] A. K. Kuchibhotla, *Exchangeability, conformal prediction, and rank tests*, *arXiv preprint arXiv:2005.06095* (2020).
- [142] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, *Scikit-learn: Machine learning in python*, *Journal of Machine Learning Research* **12** (2011), no. 85 2825–2830.
- [143] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Z. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, *PyTorch: An imperative style, high-performance deep learning library*, in *Advances in Neural Information Processing Systems*, vol. 32, pp. 8024–8035, 2019.

- [144] Y. Zheng, B. Huang, W. Chen, J. Ramsey, M. Gong, R. Cai, S. Shimizu, P. Spirtes, and K. Zhang, *Causal-learn: Causal discovery in python*, *Journal of Machine Learning Research* **25** (2024), no. 60 1–8.
- [145] S. Ravfogel, Y. Elazar, H. Gonen, M. Twiton, and Y. Goldberg, *Null it out: Guarding protected attributes by iterative nullspace projection*, in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 7237–7256, 2020.
- [146] Y. Romano, M. Sesia, and E. J. Candès, *Classification with valid and adaptive coverage*, in *Advances in Neural Information Processing Systems*, vol. 33, pp. 3581–3591, 2020.
- [147] A. N. Angelopoulos, S. Bates, M. I. Jordan, and J. Malik, *Uncertainty sets for image classifiers using conformal prediction*, in *International Conference on Learning Representations (ICLR)*, 2021.