Towards Long-term Fairness in Sequential Decision Making

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy in Engineering with a concentration in Computer Science

by

Yaowei Hu
Sichuan University
Bachelor of Engineering in Network Engineering, 2015
Sichuan University
Master of Science in Computer Science and Technology, 2018

Dec 2023
University of Arkansas

This dissertation is approved for recommendation to the Graduate Council.

Lu Zhang, Ph.D.
Dissertation Director

Xintao Wu, Ph.D.
Committee Member

Xiao Liu, Ph.D.
Committee Member

Susan Gauch, Ph.D.
Committee Member

ABSTRACT

With the development of artificial intelligence, automated decision-making systems are increasingly integrated into various applications, such as hiring, loans, education, recommendation systems, and more. These machine learning algorithms are expected to facilitate faster, more accurate, and impartial decision-making compared to human judgments. Nevertheless, these expectations are not always met in practice due to biased training data, leading to discriminatory outcomes.

In contemporary society, countering discrimination has become a consensus among people, leading the EU and the US to enact laws and regulations that prohibit discrimination based on factors such as gender, age, race, and religion. Consequently, addressing algorithmic discrimination has garnered considerable attention, emerging as a crucial research area. To tackle this challenge, association-based fairness notions are proposed based on two legal doctrines of disparate treatment and disparate impact. Subsequently, several causality-based fairness notions are introduced to provide a more comprehensive understanding of how sensitive attributes influence decisions. Moreover, researchers have devised a range of pre-process, in-process, and post-process fairness algorithms to adhere to the above fairness metrics. However, much of the literature on fair machine learning focuses on static or one-shot scenarios, whereas real-world automated decision systems often make sequential decisions within dynamic environments. Consequently, current fairness algorithms cannot be directly applied to dynamic settings to achieve long-term fairness.

In this dissertation, we investigate how to achieve long-term fairness in sequential decision making by addressing the issue of distribution shift, defining appropriate long-term fairness notion, and designing different fairness algorithms. Leveraging Pearl's structural causal model, we view the deployment of each model as a soft intervention, enabling us to

infer the post-intervention distribution and approximate the actual data distribution, thereby mitigating the problem of distribution shift. Additionally, we propose to measure indirect causal effects in time-lagged causal graphs as the causality-based long-term fairness.

By integrating the aforementioned techniques, we introduce an algorithm that can concurrently learn multiple fairness models from a static dataset containing multi-step data. Furthermore, we convert traditional optimization into performative risk optimization, facilitating the training of a single model to achieve long-term fairness. Then, we design a three-phase deep generative framework where a single decision model is trained using high-fidelity generated time series data, significantly enhancing the performance of the decision model. Finally, we extend our focus to Markov decision processes, formulating a novel reinforcement learning algorithm that can effectively achieve both long-term and short-term fairness simultaneously.

ACKNOWLEDGEMENTS

During my Ph.D. stage in the past few years, I have received a lot of guidance and help from many people in my study and life. Here I want to express my sincere gratitude to them.

First and foremost, I would like to express my sincerest gratitude to my advisor, Dr. Lu Zhang, for his mentorship over the last five years. I appreciate all his time, guidance, support, and patience. His rigor, enthusiasm and persistence in academic research, as well as his patience and help to students inspired and motivated me, and taught me a lot. It's my luck to have a such good advisor.

I would like to thank Dr. Xintao Wu and Dr. Yongkai Wu from Clemson University. In the collaboration, their suggestions, experience and ideas have greatly contributed to my research work and my life.

I would like to thank my dissertation committee members, Dr. Xintao Wu, Dr. Susan Gauch, and Dr. Xiao Liu for serving in my dissertation committee and providing me constructive feedback and comments to improve the quality of this dissertations.

I also thank my friends and colleagues in the Causal Artificial Intelligence Lab at the University of Arkansas. I appreciate the collaboration with Jacob Lear. I also want to thank my other friends and colleagues with whom I had a great time: Saima Absar, Kshitiz Tiwari, Anidita Saha, Wen Huang, Dr. Wei Du, Dr. Depeng Xu, Dr. Panpan Zheng, Dr. Shuhan Yuan.

Lastly, I would also like to thank my family for their love and support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 Introduction

This chapter starts by introducing the background and the major challenges of our research. Next, we summarize the contributions of the entire research, and end with an introduction to the structure of this dissertation.

1.1 Background

With the development of artificial intelligence, automated decision-making systems are increasingly used in various applications, such as hiring [1, 2], loans [3, 4], education [5, 6], recommendation systems [7, 8], etc. Based on the powerful computing ability of computers and huge amount of data, it is expected that automated decision-making systems integrated by various machine learning algorithms can generate fast, accurate and fair decisions. However, such expectation cannot be met. Traditional machine learning algorithms usually regard predictive performance as the main training goal, e.g., accuracy, which ignore or damage the performance of the model in terms of fairness. As a result, the trained models inherit and even amplify biases in the data and make discriminatory decisions. For example, researchers have found that face recognition algorithms have a higher accuracy rate for white people, but a lower recognition rate for black people [9]. Another well-known example is COMPAS [10, 11], a software used by many US courts to assess risk of recidivism. The software has been found to make discriminatory decisions against African-Americans, leading to their lower chances of bail. Many people's lives are significantly impacted by those automated decisions.

Since discriminatory decisions not only harm the interests of individuals, but also affect the ethics of society, anti-discrimination has become a consensus among people, thus the EU and the US have enacted laws and regulations to prohibit discrimination. Discrimination refers to the unfair decisions towards individuals based on the memberships grouped by their ages, races or genders, etc. In legal domain, two formal definitions of discrimination have been introduced known as disparate treatment and disparate impact [12]. We say the decision making process suffers from disparate treatment if membership information (i.e., sensitive attribute) are directly used to make decisions. If the outcomes are indirectly affected by the membership information, it is the disparate impact. Compared with disparate treatment, disparate impact is a more covert discrimination, which may exist even in seemingly neural policy. According to these two legal doctrines, researchers have proposed many fairness notions which can quantitatively measure discrimination, e.g., fairness through unawareness [13], demographic parity [13, 14], equalized odds [15], equal opportunity [15], etc. These notions rely only on statistics to measure the fairness, whereas correlation does not imply causation. Therefore, some researchers propose various causality-based fairness notions with the help of causality frameworks, e.g., counterfactual fairness [16], path-specific fairness [17], PC-fairness [18]. Causality-based fairness notions take into account the causal structure of the problem, which reflects the causal relationship between variables and the process of data generation. The causal structure can visually show us how membership affects decision-making, which greatly improves the interpretability of models [19].

The proposal of the above fairness notions has greatly promoted researchers' attention and research on fairness. Consequently, various algorithms are proposed to improve the fairness of existing algorithms by eliminating discrimination within them. According to the stage at which they are applied, these algorithms can generally be divided into three categories: pre-processing, in-processing algorithms, post-processing algorithms [8, 20, 21]. As the bias mainly exists in the data, the idea of pre-processing algorithms is to remove the bias in the training data directly through pre-processing, which can be used to various downstream algorithms without any modification, but may have lower accuracy and residual unfairness. The in-processing algorithms usually incorporate various fairness constraints into the objective function and train them together, which enables models to perform well in both good accuracy and fairness. Nevertheless, such combined objective functions lead to non-convex optimization and suboptimal models. The post-processing algorithms are model-agnostic since they usually promote the fairness by reassigning the predicted labels. Without having access to the sensitive attributes, post-processing algorithms can not be used which limits their field of application. These algorithms are mainly applied to static or one-shot datasets and only guarantee the fairness of the algorithm in the current data.

Unlike static or one-shot settings, sequential decision-making system is a dynamic environment where data distributions keep changing along the time and decisions made in the past will affect the subsequent data distributions and then affect the future decisions. In addition, applications in real life are dynamic environments rather than static environments, so considering dynamic characteristics is more in line with real life. For example, when a person applies for a bank loan, whether the bank grants the loan will affect the person's financial situation, which in turn will affect whether he can get a loan next time; If a system makes a decision to assign police to a certain place to patrol, then that place is more likely to have more criminal records reported, causing the system to assign police here more often, and this circular effect is continuously amplified. The study of fairness in dynamic systems is more complex, which makes it less studied than in static environments. Recently, some work has begun to focus on studying fairness in dynamic environments [22, 23, 24]. Different analysis frameworks, such as Poly Urn model [25], one-step feedback model [24] and reinforcement learning [26], are used to study how fairness changes according to different decision-making policies in various specific application scenarios, e.g., labor market, hiring, resource allocation in predictive policing, bank loan. Much literature has widely demonstrated that fairness in static settings does not provide insight into how long-term fairness changes, and often static fairness does not guarantee long-term fairness, and may even exacerbate discrimination [27]. Therefore, it is an important topic to study fairness in a dynamic setting and explore how to design algorithms to achieve long-term fairness.

1.2 Challenges

Fair machine learning in static settings has been studied extensively, while long-term fairness in sequential decision problems has only recently been concerned and some preliminary exploratory work has been done. Therefore, there are still some significant challenges to study long-term fairness, which motivate us to do relative research.

The first challenge is that there is no appropriate definition of causality-based long-term fairness. The existing work mainly adopts definitions of fairness proposed in static fair machine learning, e.g., demographic parity [28], equal opportunity [15], equalized odds [15]. However, those definitions are not very proper metrics to measure the long-term fairness. Moreover, discrimination is the result of a causal relationship between a sensitive attribute and an outcome, which is widely known and legally adopted. Statistical definitions of fairness do not provide all the implications of causality because association does not imply causation. A proper causality-based long-term fairness is urgently needed to guide us how to design algorithms to achieve this goal.

The second one is distribution shift. When a model is deployed, its decisions can cause data distribution shift, making the original training data inconsistent with the newly generated data. For example, in the lending scenario banks decide whether to grant loans to the applicants and the decisions will affect applicants' financial situation. The distribution shift problem makes direct application of traditional supervised algorithms no longer suitable and ignoring the distribution shift will critically affect the achievement of long-term fairness, as long-term fairness is affected by all decisions made by the model along the time. Further, the interaction between the user and the decision system creates a feedback loop. In a dynamic settings, previous decisions will indirectly affect subsequent decisions by affecting subsequent data distributions of population. Without knowing how the population would be reshaped by decisions, enforcing any fairness constraint may create negative feedback loops and eventually harm fairness in the long run.

Compared with supervised learning and fairness in static settings, studying fairness in dynamic settings needs to consider multiple goals: utility, short-term fairness, and long-term fairness. In static settings, the trade-off between fairness and utility has been pointed out by many works. When we need to consider more objectives in more complex settings, the relationship between multiple objectives will also become more complicated, and there may be multiple trade-offs between them, which will make it difficult to choose hyperparameters and evaluate the performance of models.

1.3 Contributions

Inspired by current research and analysis of above challenges, our goal in this dissertation is to propose an appropriate definition of long-term fairness and design algorithms to achieve long-term fairness for the sequential decision-making problems. As a mathematical framework for analyzing causality between variables, structural causal model [29] is leveraged by us as the main tool to study long-term fairness. Within the framework of the structural causal model, we use causal graphs to explicitly describe the dependencies between variables.

To overcome the challenge of distribution shift, we use soft interventions to simulate the deployment of decision models and infer the post-interventional distributions. We first study how to train multiple fair classifiers simultaneously in the few-shot classification settings. Then, we propose a causality-based long-term fairness and design a algorithm to achieve both short-term and long-term fairness in sequential decision making. To address limitations of last work, we further propose a three-phase deep generative framework where the decision model is trained on high-fidelity generated time series data to achieve long-term fairness.

The contributions of this dissertation are summarized as follows.

The existing work focuses on fair machine learning in static settings where a single fair decision model is trained. However, our study considers that a system can include multiple decision models, each of which is required to satisfy fairness constraints. As our first exploration of long-term fairness in dynamic settings, we have to address the challenge of correctly estimating the data distribution affected by the deployed models. Leveraging structural causal model, we treat each decision model as a soft intervention. Based on the theory of causal inference and do-calculus [29], post-interventional distributions resulting from model deployment can be derived. After knowing the post-interventional distribution, the loss function and fair constraints can be derived as well. By optimizing the loss function with fairness constraints, we can obtain multiple fair models simultaneously. To the best of our knowledge, this is the first work to study fair multiple decision making where the feature distribution may change due to the deployment of decision models.

As stated in the challenge above, although there are many definitions of fairness in static fair machine learning, there is no definition of long-term fairness in sequential decision making. In this work, we first define the causality-based long-term fairness as the path-specific effect on the time-lagged causal graph. Then the problem is formulated as a constrained optimization problem that can achieve trade-off between long-term fairness, short-term fairness and model utility. We find that if the post-interventional distribution is calculated according to the conventional method, the result will become more complicated with more time steps. To make the problem tractable, the original optimization problem is converted to a performative risk optimization problem and we propose an optimization algorithm by leveraging the repeated risk minimization. As far as we know, this is the first work to propose a long-term fairness notion.

Previous work [30] proposed an approach to reduce the discrimination and bias up to a certain time step, however a critical limitation is that to achieve fairness at certain time step it requires a time series training dataset whose time length is greater than that time step. We address the above limitation by developing a deep generative model that can predictively generate data following both observational and interventional distributions, and integrating the prediction and training into a collaborative training framework so that the predicted data could be used as reliable data for training the decision model. The above methods are integrated into a three-phase framework through which we can obtain a generative model and a fair decision model. Besides, the optimization problem is formulated as a performative risk minimization and solved by using the repeated gradient descent algorithm. Moreover, we design and generate two datasets of synthetic data and semi-synthetic data, on which we conduct various experiments to verify the effectiveness of proposed method.

In addition to supervised settings, we extend the study of long-term fairness to the field of reinforcement learning. We propose an algorithm to learn fair policies for considering both short-term and long-term fairness in sequential decision-making scenarios. To model the dynamics of sequence decisions, this problem is formulated as a constrained Markov decision process (MDP) with different fairness notions as constraints, so that we can readily adopt off-the-shelf policy optimization algorithms. As in the previous work, first we use statistical fairness notions as short-term fairness, in accordance with some laws and regulations to ensure that every decision is fair. By actively altering some low-confidence actions in sequence generation to generate fairer sequences, we train the model on these data to make fairer decisions, which is a model-agnostic approach. Furthermore, since sequential decisions affect feature distributions, we argue that the gap between feature distributions of different groups can be continuously narrowed in the long run by carefully choosing different decisions. Inspired by [31], we impose the long-term fairness in a policy optimization algorithm by adding a distribution distance penalty to the advantage function. We apply the proposed method on three cases, bank loans, attention allocation and epidemic control, and achieve good performance compared to other baselines.

1.4 Organization

The remaining of this dissertation is organized as follows.

In Chapter 2, we present a comprehensive related work on fairness-aware machine learning. Specifically, we divide the related research into two parts according to the research settings, namely, fair classification in static setting and fair classification in dynamic setting. In Chapter 3, we introduce the uniform notations and some preliminary knowledge of structural causal models and fairness notions which play an important role in understanding and facilitating follow-up chapters. Moreover, short and highly related work and preliminaries are also introduced in subsequent chapters.

We present our main research work in Chapter 4 - Chapter 7. Chapter 4 proposes an approach that learns multiple classifiers and achieves fairness for all of them simultaneously, by treating each decision model as a soft intervention and inferring the post-intervention distributions to formulate the loss function as well as the fairness constraints. In Chapter 5, we propose a framework for achieving long-term fair sequential decision making, which formulates as a constrained optimization problem with the utility as the objective and the long-term and short-term fairness as constraints. We propose a framework based on deep generative models to achieve long-term fairness by learning the mechanism of feature distribution shift and narrowing the data distributions of two groups through minimizing the Wasserstein distance in Chapter 6. In Chapter 7, we study long-term fairness in reinforcement learning. In the proposed algorithm, not only long-term fairness is achieved by regularizing the advantage function, but also short-term fairness is achieved by modifying the training data.

In Chapter 8, we conclude this dissertation and discuss the future work.

2 Related Work

According to the fairness problem setting, we divide fairness-aware machine learning into two categories: (1) fairness in static settings; (2) fairness in dynamic settings. Specifically, we focus mainly on the fairness of classification problem. In this chapter, we review the related literature on these two categories, and comprehensively introduce the related concepts and technical routes of the existing work. Finally, we summarize the content of this chapter.

2.1 Fairness in Static Settings

In the literature, the algorithms for eliminating discrimination in static settings can be divided into three types according to their roles in different stages of model training: pre-processing, in-processing and post-processing.

For pre-processing algorithms [32, 33, 34, 35, 36], they process or transform raw data to eliminate bias so that any model learned from those unbiased data will be fair. These pre-processing methods can be applied by modifying non-sensitive features or labels, generating new unbiased data, or learning fair representations. In [34, 35], they processed data through adversarial learning to eliminate discrimination in the data while maximizing the preservation of information for future classification tasks. Specifically, Xu et al. [34] built a fairness-aware generative adversarial networks with two discriminators, one to ensure the similarity between the generated data and the original data, and another to ensure that the generated data is discrimination-free. Different from the previous work, Madras et al. [35] proposed a method that utilizes adversarial learning to learn a fair representation instead of generating new data. To modify the raw data, Kamiran and Calders [37] designed two methods: massaging, which changes labels of subjects from difference sensitive groups, and re-weighing, which assigns different weights to subjects, while Calmon et al. [36] formulated the data transformation problem as an optimization problem to simultaneously achieve three objectives: controlling discrimination, limiting distortion, and preserving utility. By solving this optimization problem, a fair transformed dataset can be obtained. Pre-processing methods have the advantages of flexibility and model-agnosticism, which can be used for any downstream models, without considering certain assumptions.

For in-processing algorithms [28, 38, 39, 40], they usually adjust the training procedure of models, often by adding fairness constraints or training with advarsarial learning. Wu et al [38] proposed a general framework using surrogate functions to integrate various fairness metrics into the classical classification model to form a convex optimization problem with fairness constraints. Zafar et al. [28] proposed a flexible approach to train fair margin-based classifiers by minimizing the covariance between the users' sensitive attribute and the signed distance from the users' feature vectors to the decision boundary. Adversarial learning can be used not only in pre-processing methods, but also in in-processing methods. In [40], an adversary network is added and takes the predictions of a classifier as input, which attempts to predict the sensitive attributes. The gradient of the adversary is used to update the classifier in order to reduce the sensitive information passed through the predicted labels. Compared to other methods, in-processing methods can achieve good performance in both accuracy and fairness, while obtaining a good trade-off between them by adjusting hyperparameters.

For post-processing algorithms [15, 41, 42, 43, 44], they meet the fairness constraint by modifying the output scores. For example, Hardt et al. [15] suggested flipping certain predicted labels to meet the fairness criteria of equal odds or equal opportunity. Kamiran et al. [41] proposed two solutions, called Reject Option based Classifier (ROC) and Discrimination-Aware Ensemble (DAE), to modify the predictive labels close to the decision boundary. Williamson et al [44] showed that selecting instance-dependent thresholds can obtain a good trade-off between accuracy and fairness. Mehrabi et al. [43] proposed using attention mechanisms to learn the relationship between features and decisions, and reducing unfairness by decreasing the attention weights of some features.

2.2 Fairness in Dynamic Settings

In the last decade, fairness in static settings has been extensively studied in the literature. However, in reality most decision-making systems work in a dynamic environment, and the distribution of data will change over time, which will affect the accuracy and fairness of the decision-making system. Therefore, traditional methods in static settings are no longer applicable in dynamic settings, which has inspired researchers to extend the fairness problem to dynamic settings.

It has been first studied in a compound decision-making process called pipeline [45, 46]. In pipelines, individuals may drop out at any stage, and classification in subsequent stages depends on the remaining cohort of individuals. For instance, hiring is at least a two-stage model: deciding whom to be interviewed from the applicant pool and then deciding whom to be hired from the interview pool. In addition to the pipeline, a more practical and challenging dynamic setting considers that decisions made in the past can reshape the data population and subsequently influence future decisions [47]. In this setting, several studies have demonstrated the inadequacy of static fairness approaches in various application scenarios, including credit lending [24], college admission [48], labor market [22]. In [49], the authors propose to use causal directed acyclic graphs (DAGs) as a unifying framework to study fairness in dynamical systems but have not reached any approach to achieve long-term fairness.

As another line of work, some research [48, 23, 50, 31, 51] studies long-term fairness in the context of reinforcement learning, which is a natural framework to describe sequential decision making problems and feedback dynamics. To understand the long-term impact of deployed models, D'Amour et al. [48] first proposed to study the fairness of some algorithms via simulation in which three cases, bank loans, attention of allocation and college admissions, are formulated as Markov Decision Processes (MDPs), and found static or one-step analyses do not give complete conclusions. Their public library builds an easy-to-used environment for future research. As with in-processing methods, training reinforcement learning algorithms with fairness constraints is an intuitive approach. Wen et al. [50] formulated the problem as constrained MDPs and tried to achieve fairness by developing two algorithms, one of which is a model-based algorithm solved by linear programming and the other one is model-free algorithm based on cross-entropy method to train. Instead of adding explicit fairness constraints, Yu et al. [31] recently designed a new method to impose fairness requirements in policy gradient algorithms, which regularizes the advantage function by including fairness penalties. And three case studies demonstrated the effectiveness of the proposed method. In addition, Zhang el al. [23] studied the dynamics of group qualification rates which is converted into a partially observed Markov decision problem. Their work show that static fairness constraints can either benefit or damage the fairness.

2.3 Summary

A large body of literature has emerged in the past decade to address the issue of fairness in static settings. In these works, many different definitions of fairness based on statistics or causality are proposed. At the same time, according to these definitions of fairness, a large number of algorithms are designed to eliminate discrimination and achieve fairness. These algorithms are generally divided into three types: pre-processing, in-processing, and post-processing. Compared with fairness in static settings, the problem of fairness in dynamic settings is more difficult and more challenging, which has led to it not get attention until recent several years. Some preliminary work has been proposed to consider the dynamics of sequential decision-making problems and effects of static fairness constraints on long-term fairness. Some of the work is done within the framework of supervised learning, while others focus on reinforcement learning. However, existing work does not yet have a suitable definition for measuring long-term fairness. Moreover, there is no benchmark datasets or methods to generate data for sequential decision-making problems and no consensus on how to evaluate an algorithm.

3 Preliminaries

In this chapter, we introduce the mathematical notations and some of the associated background preliminaries for the whole dissertation. Firstly, some notations are presented to describe variables, datasets, and distributions. Then we describe Peal's structural causal model framework which is the basis of our proposed algorithms. Finally, association-based and causality-based fairness notions are provided as a basis for subsequent chapters.

3.1 Notations

For the unification of notations, we use the various notations summarized here throughout the dissertation. We denote variables or features by uppercase letters and their values by lowercase letters respectively, i.e., X and x. Similarly, the sets of variables or vectors and their values are denoted by bold letters, i.e., $\mathbf{X}$ and $\mathbf{x}$. Variables and vectors will be used interchangeably unless otherwise specified. We represent datasets by $\mathcal{D} = \{(\mathbf{X}_i, Y_i)\}_{i=1}^{n}$ randomly sampling from the joint distribution $P(\mathbf{X}, Y)$, where $\mathbf{X}_i$ is the feature vector of i-th instance and Y_i is the output of i-th instance. When time is considered, the temporal datasets are represented by $\mathcal{D} = \{(\mathbf{X}^i, Y^i)\}_{i=1}^{t}$, where $\mathbf{X}^i$ is the i-th step feature vector and Y^i is the i-th step outputs.

3.2 Structural Causal Models

In our research, structural causal model (SCM) [29] is often used to analyze causality between variables and to define causality based fairness. SCM is a mathematical framework developed by Judea Pearl for modeling the causal mechanisms of a system as a set of structural equations which describe the data generation mechanism. The formal definition of SCM is as follows:

Definition 1 (Structural Causal Model)**.** *A structural causal model $\mathcal{M}$ is represented by a quadruple $\langle \mathbf{U}, \mathbf{V}, \mathbf{F}, P(\mathbf{U}) \rangle$ where*

1. *$\mathbf{U}$ is a set of exogenous random variables that are determined by factors outside the model.*

2. *$P(\mathbf{U})$ is a joint probability distribution defined over $\mathbf{U}$.*

3. *$\mathbf{V}$ is a set of endogenous variables that are determined by variables in $\mathbf{U} \cup \mathbf{V}$.*

4. *$\mathbf{F}$ is a set of structural equations from $\mathbf{U} \cup \mathbf{V}$ to $\mathbf{V}$. Specifically, for each $V \in \mathbf{V}$, there is a function $f_V \in \mathbf{F}$ mapping from $\mathbf{U} \cup (\mathbf{V} \backslash V)$ to V, i.e., $v = f_V(pa_V, u_V)$, where pa_V and u_V are realization of a set of endogenous variables $PA_V \in \mathbf{V} \setminus V$ and a set of exogenous variables U_V respectively.*

If all exogenous variables in $\mathbf{U}$ are assumed to be mutually independent, then the causal model is called a *Markovian model*; otherwise, it is called a *semi-Markovian model.* Each causal model $\mathcal{M}$ is associated with a causal graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ where $\mathcal{V}$ is a set of nodes and $\mathcal{E}$ is a set of edges. Each node of $\mathcal{V}$ corresponds to a variable of $\mathbf{V}$ in $\mathcal{M}$. Each edge in $\mathcal{E}$, denoted by a directed arrow $\rightarrow$, points from a node $V_i \in PA_j$ to a different node $V_j \in \mathbf{V}$, which represents the direct causal relationship from V_i to V_j. In a causal graph, a *path* is a sequence of directed edges and a *directed path* is a path whose edges point to the same direction.

Throughout this dissertation, we only consider Markovian models and their associated causal graphs are *Directed Acyclic Graph* (DAG) which means there are no cyclic dependencies among the variables. In our causal graphs, all exogenous variables are omitted.

Following factorization formula [52], the observable distribution derived from SCM $\mathcal{M}$ can be decomposed into:

$$P(\mathbf{v}) = \sum_{\mathbf{u}} \prod_{\{i|V_i \in \mathbf{V}\}} P(v_i \mid pa_i, u_i)P(\mathbf{u}), \tag{3.1}$$

where every conditional probability $P(v_i \mid pa_i, u_i)$ is governed by the corresponding structural equation f_i.

3.2.1 Intervention

The *do-operation*[29] is proposed to define the intervention in SCMs. Specifically, the do-calculus introduces a new operation $do(\mathbf{X} = \mathbf{x})$ or $do(\mathbf{x})$, which means the intervention forces some variables $\mathbf{X}$ to take $\mathbf{x}$. According to the form of $\mathbf{x}$, the intervention can be classified into *hard intervention* and *soft intervention*. This new operation also leads to a new probability distribution, called the interventional distribution [53]:

Definition 2 (Interventional Distribution or Post-interventional Distribution)**.** *The interventional distribution $P(\boldsymbol{y} \mid do(\boldsymbol{x}))$ or $P(\boldsymbol{Y}(\boldsymbol{x}))$ denotes the distribution of variables $\boldsymbol{y}$ when the variables $\boldsymbol{X}$ are forced to be set to $\boldsymbol{x}$.*

3.2.1.1 Hard Intervention

In a SCM, hard intervention $do(\mathbf{x})$ is define as the substitution of equation $x = f_{\mathbf{X}}(\mathbf{pa}_{\mathbf{X}}, \mathbf{u}_{\mathbf{X}})$ with constant $\mathbf{x}$. The hard intervention results in a new structural causal model, which is referred as sub-model $\mathcal{M}_{\mathbf{x}}$. The causal graph $\mathcal{G}_{\bar{\mathbf{X}}}$ associated with $\mathcal{M}_{\mathbf{x}}$ is the same as $\mathcal{G}$ except that all edges incoming to nodes $\mathbf{X}$ are removed. After the hard intervention, the sub-model $\mathcal{M}_{\bar{\mathbf{x}}}$ induces a interventional distribution $P(\mathbf{v} \mid do(\mathbf{x}))$ using Eq. 3.1:

$$
\begin{aligned}
P(\mathbf{v} \mid do(\mathbf{x})) &= \sum_{\mathbf{u}} \prod_{\{i \mid V_i \in \mathbf{V}\}} P(v_i \mid pa_i, u_i, do(\mathbf{x})) P(\mathbf{u} \mid do(\mathbf{x})) \\
&= \sum_{\mathbf{u}} \prod_{\{i \mid V_i \in \mathbf{V} \setminus \mathbf{X}\}} P(v_i \mid pa_i, u_i) P(\mathbf{u}).
\end{aligned}
\tag{3.2}
$$

which is also called the *truncated factorization*.

3.2.1.2 Soft Intervention

The soft intervention extends the hard intervention such that it forces variables $\mathbf{X}$ to take functional relationship $g(\mathbf{z})$ in responding to some other variables $\mathbf{Z}$, which is denoted by σ [54]. The soft intervention substitutes equation $\mathbf{x} = f_{\mathbf{X}}(\mathbf{pa}_{\mathbf{X}}, \mathbf{u}_{\mathbf{X}})$ with a new function $\mathbf{x} = g(\mathbf{z})$. The soft intervention results in a new model $\mathcal{M}_{\sigma_{\mathbf{X}}}$ with a causal graph $\mathcal{G}_{\sigma_{\mathbf{X}}}$ where a node σ_{X_i} for every $X_i \in \mathbf{X}$ and an edge $\sigma_{X_i} \to X_i$ are added. Similar to hard intervention, the causal model $\mathcal{M}_{\sigma_{\mathbf{X}}}$ also induces a interventional distribution $P(\mathbf{v}; \sigma_{\mathbf{X}})$ or $P(\mathbf{v}(\sigma_{\mathbf{X}}))$ using Eq. 3.1:

$$
\begin{aligned}
P(\mathbf{v}; \sigma_{\mathbf{X}}) &= \sum_{\mathbf{u}^*} \prod_{\{i \mid V_i \in \mathbf{V}\}} P(v_i \mid pa_i, u_i; \sigma_{\mathbf{x}}) P(\mathbf{u}^* \mid \sigma_{\mathbf{x}}) \\
&= \sum_{\mathbf{u}^*} \prod_{\{i \mid V_i \in \mathbf{V} \setminus \mathbf{X}\}} P(v_i \mid pa_i, u_i) P(\mathbf{u}) \prod_{\{i \mid V_i \in \mathbf{X}\}} P(v_i \mid pa_i, u_i; \sigma_{\mathbf{X}}) P(\mathbf{u}^* \setminus \mathbf{u}; \sigma_{\mathbf{X}}).
\end{aligned}
\tag{3.3}
$$

where $\mathbf{U}^*$ is the set of all exogenous variables.

3.2.2 Causal Effects

After defining interventions, we can now compute the causal effects, which is the main task of causal inference. Causal effects permit us to predict how systems would respond to a hypothetical intervention. For the total causal effect of X on Y, it represents the effect of a intervention transmitted along all causal paths from cause X to effect Y, which is as follows:

Definition 3 (Total Causal Effect)**.** *The total causal effect $TE(x_1, x_0)$ of X on Y measures the causal effect of changing cause X from x_0 to x_1 on effect Y transmitted along all causal paths:*

$$TE(x_1, x_0) = P(y \mid do(x_1)) - P(y \mid do(x_0)).$$

Moreover, when we specify the causal effect of X on Y as the effect of a intervention transmitted along a certain subset of all causal paths from X to Y, the causal effect is referred to as path-specific effect [55]:

Definition 4 (Path-specific Effect)**.** *Given a causal path set π, the path-specific effect $PE(x_1, x_0)$ of X on Y measures the causal effect of changing cause X from x_0 to x_1 on effect Y transmitted along the path set π:*

$$PE(x_1, x_0) = P(y \mid do(x_{1|\pi}), do(x_{0|\bar{\pi}})) - P(y \mid do(x_0)),$$

where $P(y \mid do(x_{1|\pi}), do(x_{0|\bar{\pi}}))$ represents the interventional distribution of Y where the effect of intervention $do(x_1)$ is transmitted along π while the effect of intervention $do(x_0)$ is transmitted along other paths.

Depending on the choice of path set π, the path-specific effect can be divided into *direct causal effect* and *indirect causal effect*. If path set π only contains the direct path from X to Y, i.e., $X \to Y$, then the path-specific effect is called the direct causal effect. If path set π contains some paths other than $X \to Y$, the path-specific effect is called the indirect causal effect. In later chapters, these concepts will be defined and used more specifically.

3.3 Fairness Notions

Broadly, the goal of fair machine learning is to make fair decisions without any prejudice or favoritism towards an individual or a group based on their intrinsic or acquired traits [20]. In order to fight against discrimination and achieve fairness, a number of association-based and causality-based fairness notions have been proposed to measure how fair an algorithm is. Specifically, the binary sensitive attribute S is used to indicate the unprotected and protected groups, i.e., $S = \{s^+, s^-\}$. The decision attribute is also a binary variable denoted by $Y = \{y^+, y^-\}$.

3.3.1 Association-Based Fairness

In this dissertation, association-based fairness algorithms are often used as baselines to compare with our algorithms. In this part, we introduce three commonly used association-based fairness notions: demographic parity [56, 57], equal opportunity and equal odds [15]. More association-based fairness notions could be found in [20, 1].

Definition 5 (Demographic Parity)**.** *Given a sensitive feature S and a prediction $\hat{Y}$, the prediction $\hat{Y}$ is considered fair with respect to S if $P(\hat{Y} = y^+|S = s^+) = P(\hat{Y} = y^+|S = s^-)$.*

Definition 6 (Equal Opportunity)**.** *Given an observational data S, X, Y and a prediction $\hat{Y}$, the prediction $\hat{Y}$ is considered to satisfy the equal opportunity with respect to S and Y if $P(\hat{Y}^+|S^+, Y^+) = P(\hat{Y}^+|S^-, Y^+)$.*

Definition 7 (Equal Odds)**.** *Given an observational data S, X, Y and a prediction $\hat{Y}$, the prediction $\hat{Y}$ is considered to satisfy the equal odds with respect to S and Y if $P(\hat{Y}^+|S^+, Y^+) = P(\hat{Y}^+|S^-, Y^+)$ and $P(\hat{Y}^+|S^+, Y^-) = P(\hat{Y}^+|S^-, Y^-)$.*

3.3.2 Causality-Based Fairness

Compared with association-based fairness notions, causality-based fairness notions consider more knowledge of the data generation mechanism, depicted by a causal graph. In the perspective of causality, discrimination can be viewed as the causal effects of sensitive attribute S on decision attribute Y. Based on the paths through which S affects Y, various causality-based fairness notions have been proposed. Here we only introduce several notions used in the dissertation, others can be found in [19, 58].

Definition 8 (Total Causal Fairness)**.** *Given the sensitive attribute S and decision attribute Y, the total causal fairness is achieved if*

$$TE(s^+, s^-) = P(y|do(s^+)) - P(y|do(s^-)) = 0.$$

Definition 9 (Path-Specific Causal Fairness)**.** *Given the sensitive attribute S and decision attribute Y, and a causal path set π that contains some paths from S to Y, the path-specific causal fairness is achieved if*

$$PE(s^+, s^-) = P(y \mid do(s^+_\pi), do(s^-_{\bar{\pi}})) - P(y \mid do(s^-)) = 0.$$

Specifically, if π contains only direct causal path $S \rightarrow Y$, it is called direct causal fairness, and if π contains only indirect causal path from S to Y through redlining/proxy attributes, it is called indirect causal fairness.

4 Fair Multiple Decision Making Through Soft Interventions

4.1 Introduction

Algorithmic decision making models have been widely adopted in various domains to make important decisions, like hiring employees, granting loans, assessing recidivism risks, etc. Traditional learning algorithms are designed to maximize the prediction accuracy. Thus, they may produce biased decision models by inheriting and reinforcing discrimination in the training data [32], and/or introducing additional discrimination during the learning process [15]. How to ensure fairness in algorithmic decision making models is an important task in machine learning [32, 15]. Over the past years, many researchers have been devoted to the design of fair classification algorithms with respect to a pre-defined protected attribute, such as race or sex, and a decision task/model, such as hiring [57, 33, 59]. In particular, one line of the work is to incorporate fairness constraints into classic learning algorithms to build fair classifiers from potentially biased data [28, 60, 38, 61, 62, 63]. Most of previous research generally focuses on a single decision model. However, in reality there usually exist multiple decision models within a system and all of which may contain a certain amount of discrimination, either introduced by themselves or transmitted from upstream models. As a motivating example, consider two decision tasks Y_1, Y_2 where Y_1 is used by the city government to allocate policing resources to different locations and Y_2 is used by a local bank to make personal loan decisions. Due to historically segregated housing, neighborhood racial composition differs based on geographic locations, and there can exist direct racial discrimination in Y_1 as well. Thus, certain locations will be allocated more police resources than others, resulting in larger numbers of criminal arrest records. As a result, when the criminal arrest record is used in Y_2, certain racial group will receive unfair disadvantage in getting loans.

Ideally we would like to build fair models for all decision making tasks. However, if decision models influence one another, it is not a straightforward problem even if we know how to build a fair model for each task. This is because the data distribution can change as a consequence of deploying new models. If we build the model for each task independently using static training datasets, the learning process of each model is based on the fixed distribution given in the training data. However, deploying new fair models would change the distributions of attribute variables that are affected by their decisions as well as the discrimination that is passing down. As a result, the subsequent models built on the original distribution may not perform well in terms of both accuracy and fairness. On the other hand, if we build fair models one by one following a temporal sequential order, each time deploying a model and collecting the output data before building the next one, then the time needed for building all models may not be acceptable for some applications.

In this chapter, we propose an approach that learns multiple fair classifiers simultaneously and only requires a static training dataset. The core idea is to leverage Pearl's structural causal model (SCM) [29], treat each decision model as a soft intervention and infer the post-intervention distributions to formulate the loss function as well as the fairness constraints. The SCM is widely adopted in fair classification research for defining fairness as the causal effect of the protected attribute on the decision [64, 65, 66, 16, 67, 68, 18]. Causal inference in the SCM is often facilitated with the "(hard) intervention" that forces some variable X to take certain constant x, denoted by $do(X = x)$ [29]. "Soft intervention" [69, 54], also known as the "conditional action" [29], extends the hard intervention such that variable X is forced to take a specified functional relationship $g(z)$ in responding to some set Z of other variables, denoted by $do(X = g(z))$. In our approach, the deploying of new decision models is considered as to perform soft interventions on the decisions, whose influence can be inferred as the post-intervention distributions. By quantifying fairness as causal effects of the protected attribute on all decisions, under the hard intervention on the protected attribute and soft interventions on decisions, we formulate fair classification for multiple decisions as a single constrained optimization problem.

Combining multiple decision models together makes the optimization challenging to solve. Similarly to [38], we adopt surrogate functions to smooth the loss function and constraints. However, the difference in our problem is that, each decision model is associated with a surrogate function, and the surrogated protections are used in downstream decision models, resulting non-linear combinations of multiple surrogate functions. As a result, our loss function is different from traditional surrogated loss functions whose excess risks have been analyzed and bounded in [70]. To investigate the excess risk of our loss function, we adopt theoretical tools in [70] and show that nontrivial upper bounds exist on the excess risk in a form that is the same as that for traditional surrogated loss functions given in [70], irrespective of the number of decision models involved.

Contributions. To the best of our knowledge, this is the first work to study fair multiple decision making where the feature distribution may change due to the deployment of decision models. Our approach provides a general way to incorporate fairness constraints into the generic classification formulation such that we can readily employ off-the-shelf classification models and optimization algorithms. The causal inference allows us to train all decision models jointly from a single dataset. Since our approach is based on the SCM, all SCM-based fairness notions, including the total effect [65], direct and indirect discrimination [64, 65, 66], counterfactual fairness [16, 67, 68], and PC-fairness [18], can be naturally applied to our problem formulation. The theoretical results imply that we don't need to worry about additional losses caused by multiple surrogate functions. By conducting experiments on both synthetic and real-world datasets, we show that our approach consistently outperforms the approach which builds fair classifiers for each decision separately.

4.2 Fair Classification

Following the notations used in [38], the problem of fair classification is to learn a mapping $f : \mathbf{X} \mapsto Y$ parameterized with θ, where $\mathbf{X}$ is a set of input attributes and $Y = \{0, 1\}$ is the class label. The learning algorithm aims to minimize the classification error $\mathbb{E}_{\mathbf{X},Y}[\mathbb{1}_{f(\mathbf{x})\neq y}]$, where $\mathbb{1}_A$ is the indicator function, i.e., $\mathbb{1}_A = 1$ if A is true and $\mathbb{1}_A = 0$ if A is false. Usually, f is defined based on another function h that is performed in the real number domain, i.e., $h : \mathbf{X} \mapsto \mathbb{R}$ and $f(\mathbf{x}) = \mathbb{1}_{h(\mathbf{x})\geq 0}$. Thus, the classification error can be reformulated as

$$R(h) = \mathbb{E}_{\mathbf{X}} \left[P(Y = 1|\mathbf{x})\mathbb{1}_{h(\mathbf{x})<0} + P(Y = 0|\mathbf{x})\mathbb{1}_{h(\mathbf{x})\geq 0} \right]. \tag{4.1}$$

By using surrogate functions $\phi(\cdot)$ to smooth and bound the indicator function (i.e., the 0-1 loss), we obtain the ϕ-loss as:

$$R_\phi(h) = \mathbb{E}_{\mathbf{X}} \left[P(Y = 1|\mathbf{x})\phi(h(\mathbf{x})) + (1 - P(Y = 1|\mathbf{x}))\phi(-h(\mathbf{x})) \right],$$

and the optimization problem as $\min_{h\in\mathcal{H}} R_\phi(h)$. Similarly fair classification can be formulated as a constrained optimization problem

$$\min_{h\in\mathcal{H}} \quad R_\phi(h) \quad \text{s.t.} \quad -\tau \leq T_\phi(h) \leq \tau, \tag{4.2}$$

where $T_\phi(h)$ is a measure of ϕ-unfairness depending on the particular fairness notion used which is also smoothed by using surrogate functions. Widely used surrogate functions include the hinges loss, square loss, logistic loss, exponential loss, etc.

4.3 Formulating Fair Classification for Making Multiple Decisions

In this section, we formally formulate the fair classification problem for making multiple decisions. Consider a protected attribute S, a set of non-protected attributes $\mathbf{X} = \{X_1, \cdots, X_m\}$ and a set of decisions $\mathbf{Y} = \{Y_1, \cdots, Y_l\}$. For ease of representation, we assume that the protected attribute and all decisions are binary, i.e., $S = \{s^-, s^+\}$ with s^- denoting the protected group and s^+ denoting the non-protected group, and $Y_k = \{y^-, y^+\}$ for each $Y_k \in \mathbf{Y}$ with y^- denoting the negative decision (i.e., $Y_k = 0$) and y^+ denoting the positive decision (i.e., $Y_k = 1$). Often we abbreviate expressions $Y_k = y^-, y^+$ as y_k^-, y_k^+. Note that decisions can be interdependent such that later decisions may depend on the consequences of earlier decisions either directly and/or indirectly through the change of some features that is mediated between the two decisions. In real situations, such indirect influence may need time to take effect and cannot be observed within a short period of time. Therefore, we only assume that a historical dataset $\mathcal{D} = \{(s^{(i)}, \mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^N$ that reflects the original decision making mechanisms is observed.

Our task is to build a classifier h_k for each decision Y_k from training data $\mathcal{D}$. Classifier

h_k takes some profile attributes $\mathbf{Z}_k \subseteq \{S\} \cup \mathbf{X}$ as the input to make the prediction as $\mathbb{1}_{h_k(\mathbf{z}_k)\geq 0}$. We would like to ensure that any classifier is fair if all classifiers are deployed, with the fairness of all classifiers being measured using the same fairness notion. In this chapter, for simplicity we consider total effect [65] as the fairness notion which is defined based on the total causal effect as follows. Nevertheless, our formulation can be easily extended to other causal-based fairness notions as long as they can be identified and computed with expressions of observational distributions.

Definition 10. *For the classifier built for each decision Y_k, it is considered to be fair if*

$$-\tau \leq P^*(y_k^+|do(s^+)) - P^*(y_k^+|do(s^-)) \leq \tau$$

where τ is a user-defined threshold and P^* is the distribution after all classifiers are deployed.

As shown in [38], the formulation of fair classification consists of a loss function for quantifying the classification error and a number of constraints for enforcing fairness. For the case of a single decision model, the loss function can be directly computed from $\mathcal{D}$, and fairness constraints can be computed from $\mathcal{D}$ as well after performing hard intervention $do(s)$. However, for the case of multiple decision models, due to the change in the data distribution made by model deployment, the loss function and fairness constraints should not be computed from $\mathcal{D}$ but P^* which may be different from the distribution followed by $\mathcal{D}$. Therefore, we propose to adopt the soft intervention to model all model deployments and infer post-intervention distributions.

To this end, we build a causal graph $\mathcal{G}$ to represent the causal structure of the underlying data generation mechanism from dataset $\mathcal{D}$. The research of causal structure discovery is quite active in recent years and many algorithms have been proposed [71]. Given the causal graph, we capture the deployment of classifier $h_k(\mathbf{z}_k)$ as a soft intervention that forces the prediction of decision Y_k to take functional relationship $h_k(\mathbf{z}_k)$, denoted as $do(h_k)$. Consequently, distribution P^* after the deployment of all classifiers can be captured by the post-intervention distribution after performing soft interventions $do(h_1, \cdots, h_l)$. Then, the classification error of $h_k(\mathbf{z}_k)$ after the deployment of classifiers could be measured similarly to Eq. (4.1) as given by

$$R(h_k) = \mathop{\mathbb{E}}_{\mathbf{Z}_k|do(h_1,\cdots,h_l)} \left[P(y_k^+|\mathbf{z}_k)\mathbb{1}_{h_k(\mathbf{z}_k)<0} + P(y_k^-|\mathbf{z}_k)\mathbb{1}_{h_k(\mathbf{z}_k)\geq 0} \right], \tag{4.3}$$

where the expectation is computed on the post-intervention distribution of $\mathbf{Z}_k$. Similarly, the fairness constraints of $h_k(\mathbf{z}_k)$ is given by the total effect

$$T(h_k) = P(y_k^+|do(s^+, h_1, \cdots, h_l)) - P(y_k^+|do(s^-, h_1, \cdots, h_l)), \tag{4.4}$$

which is based on the post-intervention distributions of Y_k after performing both hard intervention $do(s)$ and soft interventions $do(h_1, \cdots, h_l)$. We use Y_k to denote both decision label and predicted decision, and use soft intervention to distinguish between them: if the distribution is pre-interventional such as $P(y_k^+|\mathbf{z}_k)$, Y_k is the label; if the distribution is post-interventional such as $P(y_k^+|do(s^+, h_1, \cdots, h_l))$, Y_k is the prediction.

Take the toy model given in the introduction as an example, where there are two decisions Y_1 and Y_2 representing the policing resource allocation and bank loan decision respectively. We treat race as the protected attribute, denoted by S. We denote the location of the residential area as X_1, and denote the number of criminal arrest records in each area as X_2. The causal graph of this toy model is shown in Figure 4.1. We would like to build two fair

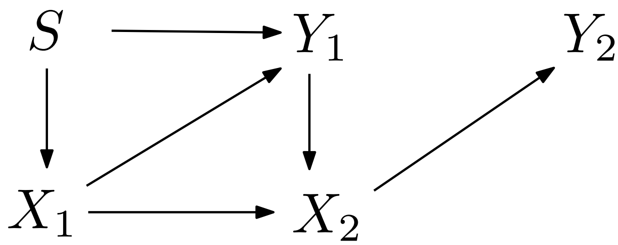


Figure 4.1: Causal graph of toy model.

classifiers $h_1(x_1)$ and $h_2(x_2)$ for predicting Y_1 and Y_2. Note that the inputs of classifiers could be different from the original parents of Y_1 and Y_2, which are $\{S, X_1\}$ and X_2 respectively. Their loss functions are given by $R(h_1)$, $R(h_2)$, and fair constraints are given by $T(h_1)$, $T(h_2)$.

Next, we need to derive $R(h_k)$ and $T(h_k)$, which are given on the post-intervention distribution, as smooth expressions on $\mathcal{D}$, which is the observational data. By using surrogate functions $\phi(\cdot)$ to smooth and bound the indicator function, we finally derive the formulas of $R_\phi(h_k)$ and $T_\phi(h_k)$ that will be used in our formulation of fair multi-decision learning.

4.3.1 Deriving Loss Function and Fair Constraints

In the above example, the learning of classifier h_1 could be done by solving an ordinary fair classification problem. However, when learning classifier h_2, both its loss and fairness are affected by classifier h_1, and in this case the effect is transmitted indirectly through X_2. To accurately measure the loss and fairness of h_2, we need to mathematically express the effect of h_1 as post-intervention distributions. Thus, we apply following three properties of the (soft) intervention to compute post-intervention distributions from observational data.

(1) An intervention on a variable V would not change the distribution of V's non-descendant.

(2) An intervention on V would not change the generation mechanism of another variable W, i.e., distribution $P(w|\mathsf{pa}_W)$ would not be changed.

(3) A soft intervention on V would change its conditional distribution $P(v|\mathsf{pa}_V)$ according to the defined functional relationship.

Next we show how the properties work in the toy example. Note that $R(h_2)$ is given by $\mathbb{E}_{X_2|do(h_1,h_2)}\left[P(y_2^+|x_2)\mathbb{1}_{h_2(x_2)<0} + P(y_2^-|x_2)\mathbb{1}_{h_2(x_2)\geq 0}\right]$, which by definition is equal to $\sum_{X_2} P(x_2|do(h_1,h_2))\left[P(y_2^+|x_2)\mathbb{1}_{h_2(x_2)<0} + P(y_2^-|x_2)\mathbb{1}_{h_2(x_2)\geq 0}\right]$. Due to Property (1), $P(x_2|do(h_1,h_2)) = P(x_2|do(h_1))$, which can be broken down by conditioning on X_1, Y_1 as $\sum_{X_1,Y_1} P(x_2|do(h_1),x_1,y_1)P(x_1,y_1|do(h_1))$. Due to Property (2), we have that $P(x_2|do(h_1),x_1,y_1)$ $= P(x_2|x_1,y_1)$. Meanwhile, we rewrite $P(x_1,y_1|do(h_1))$ as $P(x_1|do(h_1))P(y_1|do(h_1),x_1)$ which is equal to $P(x_1)P(y_1|do(h_1),x_1)$. Then, we further break down $P(y_1|do(h_1),x_1)$ as $\sum_S P(y_1|do(h_1),s,x_1)P(s|do(h_1),x_1)$. Due to Property (1), we have $P(s|do(h_1),x_1) = P(s|x_1)$. Due to Property (3), we have $P(y_1|do(h_1),s,x_1)$ be equal to a new distribution $P_{h_1}(y_1|x_1)$ defined by function h_1, which is given by $\mathbb{1}_{h_1(x_1)\geq 0}$ if $y_1 = y_1^+$ and $\mathbb{1}_{h_1(x_1)<0}$ if $y_1 = y_1^-$ in our case. Finally, combining every components above together and using a surrogate function ϕ to replace each indicator, we obtain that

$$\begin{aligned}
R_\phi(h_2) = \sum_{S,X_1,X_2} P(s,x_1)\big(&\phi(h_2(x_2))\phi(-h_1(x_1))P(y_2^+|x_2)P(x_2|x_1,y_1^+)\\
&+\phi(h_2(x_2))\phi(h_1(x_1))P(y_2^+|x_2)P(x_2|x_1,y_1^-)\\
&+\phi(-h_2(x_2))\phi(-h_1(x_1))P(y_2^-|x_2)P(x_2|x_1,y_1^+)\\
&+\phi(-h_2(x_2))\phi(h_1(x_1))P(y_2^-|x_2)P(x_2|x_1,y_1^-)\big)\,.
\end{aligned}$$

For $T(h_2)$, it is given by $P(y_1^+|do(s^+,h_1,h_2)) - P(y_1^+|do(s^-,h_1,h_2))$, which can be directly rewritten as $P(y_1^+|do(s^+,h_1)) + P(y_1^-|do(s^-,h_1)) - 1$. By similarly applying the three properties, we could obtain that

$$
\begin{aligned}
T_\phi(h_2) =\sum_{X_1,X_2} \big(&\phi(-h_2(x_2))\phi(-h_1(x_1))P(x_1|s^+)P(x_2|x_1,y_1^+)\\
&+\phi(-h_2(x_2))\phi(h_1(x_1))P(x_1|s^+)P(x_2|x_1,y_1^-)\\
&+\phi(h_2(x_2))\phi(-h_1(x_1))P(x_1|s^-)P(x_2|x_1,y_1^+)\\
&+\phi(h_2(x_2))\phi(h_1(x_1))P(x_1|s^-)P(x_2|x_1,y_1^-)\big) - 1.
\end{aligned}
$$

More generally, when there are l classifiers, we could derive $R_\phi(h_k)$ and $T_\phi(h_k)$ by using the factorization formula proposed in [69] which implicitly encode all three properties. For $R_\phi(h_k)$, according to the factorization formula, post-intervention $P(\mathbf{z}_k|do(h_1),\cdots,do(h_l))$ is given by

$$
P(\mathbf{z}_k|do(h_1,\cdots,h_l)) = \sum_{\mathbf{X}\backslash\mathbf{Z}_k,\mathbf{Y}} \prod_{i=1}^{l} P_{h_i}(y_i|\mathbf{z}_i) \prod_{i=1}^{m} P(x_i|\mathsf{pa}_{X_i}), \tag{4.5}
$$

where $P_{h_i}(y_i|\mathbf{z}_i)$ is the distribution of Y_i defined by classifier $h_i(\mathbf{z}_i)$, i.e., $\mathbb{1}_{h_i(\mathbf{z}_i)\geq 0}$ if $y_i = y^+$ and $\mathbb{1}_{h_i(\mathbf{z}_i)<0}$ if $y_i = y^-$. For the sake of simple representation, we assume that S has no parent in the causal graph. Note that all terms in Eq. (4.5) can be computed from data. Then, we can derive the formula for computing $R_\phi(h_k)$.

However, it may not be ideal to directly apply Eq. (4.5) to our problem formulation. First, some computations in Eq. (4.5) are not necessary since $\mathbf{Z}_k$ should not be affected by interventions on the non-ancestors of Y_k. More importantly, if any X_i is a continuous variable, its corresponding summation in Eq. (4.5) would become an integral, making the gradient difficult to compute. Thus, to further simplify Eq. (4.5), we index all attributes in $\mathbf{X}$ and $\mathbf{Y}$ according to the topological ordering, and denote the subsets of $\mathbf{X}$ and $\mathbf{Y}$ that are prior to Y_i (or X_i) in the topological order as $\mathbf{X}'_{Y_i}$ and $\mathbf{Y}'_{Y_i}$ ($\mathbf{X}'_{X_i}$ and $\mathbf{Y}'_{X_i}$). Then, by canceling out all terms that are after Y_k in the order, it follows that

$$\begin{aligned}
&P(\mathbf{z}_k|do(h_1,\cdots,h_l))\\
=&\sum_{\{S,\mathbf{X}'_{Y_k}\}\backslash\mathbf{Z}_k,\mathbf{Y}'_{Y_k}} P(s)\prod_{Y_i\in\mathbf{Y}'_{Y_k}} P_{h_i}(y_i|\mathbf{z}_i)\prod_{X_i\in\mathbf{X}'_{Y_k}} P(x_i|\mathsf{pa}_{X_i})\\
=&\sum_{\{S,\mathbf{X}'_{Y_k}\}\backslash\mathbf{Z}_k,\mathbf{Y}'_{Y_k}} P(s)\prod_{Y_i\in\mathbf{Y}'_{Y_k},y_i^+} \mathbb{1}_{h_i(\mathbf{z}_i)\geq 0}\prod_{Y_i\in\mathbf{Y}'_{Y_k},y_i^-} \mathbb{1}_{h_i(\mathbf{z}_i)<0}\prod_{X_i\in\mathbf{X}'_{Y_k}} P(x_i|\mathsf{pa}_{X_i}).
\end{aligned}\tag{4.6}$$

We can rewrite $P(s)\prod_{X_i\in\mathbf{X}'_{Y_k}} P(x_i|\mathsf{pa}_{X_i})$ as

$$\begin{aligned}
&P(s)\prod_{X_i\in\mathbf{X}'_{Y_k}}P(x_i|s,\mathbf{x}'_{X_i},\mathbf{y}'_{X_i})\\
=&P(s)\prod_{X_i\in\mathbf{X}'_{Y_k}}P(x_i|s,\mathbf{x}'_{X_i})\frac{P(\mathbf{y}'_{X_i}|s,x_i,\mathbf{x}'_{X_i})}{P(\mathbf{y}'_{X_i}|s,\mathbf{x}'_{X_i})}\\
=&P(s,\mathbf{x}'_{X_i})\prod_{X_i\in\mathbf{X}'_{Y_k}}\frac{P(\mathbf{y}'_{X_i}|s,x_i,\mathbf{x}'_{X_i})}{P(\mathbf{y}'_{X_i}|s,\mathbf{x}'_{X_i})}.
\end{aligned}$$

Thus, we can rewrite Eq. (4.6) as an expectation over $S,\mathbf{X}'_{Y_k}$. With the surrogate function we obtain

$$\begin{aligned}
R_\phi(h_k)=\mathop{\mathbb{E}}_{S,\mathbf{X}'_{Y_k}}\Bigg[&P(y_k^+|\mathbf{z}_k)\phi(h_k(\mathbf{z}_k))\sum_{\mathbf{Y}'_{Y_k}}\prod_{Y_i\in\mathbf{Y}'_{Y_k},y_i^+}\phi(-h_i(\mathbf{z}_i))\prod_{Y_i\in\mathbf{Y}'_{Y_k},y_i^-}\phi(h_i(\mathbf{z}_i))\\
&\prod_{X_i\in\mathbf{X}'_{Y_k}}\frac{P(\mathbf{y}'_{X_i}|s,x_i,\mathbf{x}'_{X_i})}{P(\mathbf{y}'_{X_i}|s,\mathbf{x}'_{X_i})}+P(y_k^-|\mathbf{z}_k)\phi(-h_k(\mathbf{z}_k))\sum_{\mathbf{Y}'_{Y_k}}\prod_{Y_i\in\mathbf{Y}'_{Y_k},y_i^+}\phi(-h_i(\mathbf{z}_i))\\
&\prod_{Y_i\in\mathbf{Y}'_{Y_k},y_i^-}\phi(h_i(\mathbf{z}_i))\prod_{X_i\in\mathbf{X}'_{Y_k}}\frac{P(\mathbf{y}'_{X_i}|s,x_i,\mathbf{x}'_{X_i})}{P(\mathbf{y}'_{X_i}|s,\mathbf{x}'_{X_i})}\Bigg].
\end{aligned}\tag{4.7}$$

We can see that, in Eq. (4.7), only probabilities of categorical decisions are involved, and the expectation can be estimated as an empirical risk.

Similarly, for $T(h_k)$, Eq. (4.4) can be directly rewritten as $T(h_k) = P(y_k^+|do(s^+, h_1, \cdots, h_l)) + P(y_k^-|do(s^-, h_1, \cdots, h_l)) - 1$, and $P(y_k^+|do(s, h_1, \cdots, h_l))$ can be given by

$$\mathop{\mathbb{E}}_{\mathbf{X}'_{Y_k}|S=s}\left[\mathbb{1}_{h_k(\mathbf{z}_k)>0}\sum_{\mathbf{Y}'_{Y_k}}\prod_{Y_i\in\mathbf{Y}'_{Y_k},y_i^+}\mathbb{1}_{h_i(\mathbf{z}_i)>0}\prod_{Y_i\in\mathbf{Y}'_{Y_k},y_i^-}\mathbb{1}_{h_i(\mathbf{z}_i)<0}\prod_{X_i\in\mathbf{X}'_{Y_k}}\frac{P(\mathbf{y}'_{X_i}|s,x_i,\mathbf{x}'_{X_i})}{P(\mathbf{y}'_{X_i}|s,\mathbf{x}'_{X_i})}\right].$$

By applying surrogate function ϕ, we obtain that

$$\begin{aligned}
&T_\phi(h_k) = \\
&\mathop{\mathbb{E}}_{\mathbf{X}'_{Y_k}|S=s^+}\left[\phi(-h_k(\mathbf{z}_k))\sum_{\mathbf{Y}'_{Y_k}}\prod_{Y_i\in\mathbf{Y}'_{Y_k},y_i^+}\phi(-h_i(\mathbf{z}_i))\prod_{Y_i\in\mathbf{Y}'_{Y_k},y_i^-}\phi(h_i(\mathbf{z}_i))\prod_{X_i\in\mathbf{X}}\frac{P(\mathbf{y}'_{X_i}|s^+,x_i,\mathbf{x}'_{X_i})}{P(\mathbf{y}'_{X_i}|s^+,\mathbf{x}'_{X_i})}\right]+ \\
&\mathop{\mathbb{E}}_{\mathbf{X}'_{Y_k}|S=s^-}\left[\phi(h_k(\mathbf{z}_k))\sum_{\mathbf{Y}'_{Y_k}}\prod_{Y_i\in\mathbf{Y}'_{Y_k},y_i^+}\phi(-h_i(\mathbf{z}_i))\prod_{Y_i\in\mathbf{Y}'_{Y_k},y_i^-}\phi(h_i(\mathbf{z}_i))\prod_{X_i\in\mathbf{X}}\frac{P(\mathbf{y}'_{X_i}|s^-,x_i,\mathbf{x}'_{X_i})}{P(\mathbf{y}'_{X_i}|s^-,\mathbf{x}'_{X_i})}\right]-1.
\end{aligned} \tag{4.8}$$

4.3.2 Problem Formulation

Now we are ready to formulate the classification problem. For each classifier $h_k(\mathbf{z}_k)$, we derive its ϕ-loss $R_\phi(h_k)$ and ϕ-unfairness $T_\phi(h_k)$. Then, we minimize the summation of the ϕ-loss over all classifiers. Meanwhile, given a fairness threshold τ_k, we want the ϕ-unfairness to be bounded within the range $[-\tau_k, \tau_k]$, so we require that $-\tau_k \leq T_\phi(h_k) \leq \tau_k$. Generally, large thresholds indicate loose fairness requirements and small ones indicate strict fairness requirements. However, due to the application of surrogate functions, τ_k may not be equal to threshold τ that is placed on the original fairness metric (e.g., 0.05 on the total effect used in the literature). In practice, we need to test different values of τ_k in order to find a good balance between fairness and accuracy.

Problem Formulation 1. *The problem of fair multiple decision making for* $\mathbf{Y} = \{Y_1, \cdots, Y_l\}$ *is formulated as the following constrained optimization problem:*

$$\min_{h_1,\cdots,h_l \in \mathcal{H}} \sum_{k=1}^{l} R_\phi(h_k) \quad \text{s.t.} \quad \forall k, \quad -\tau_k \leq T_\phi(h_k) \leq \tau_k.$$

where the formulas of $R_\phi(h_k)$ *and* $T_\phi(h_k)$ *are shown in Eqs.* (4.7) *and* (4.8)*, respectively.*

From Section 4.3.1 we see that both the loss function and constraints in this formulation involve non-linear combinations of surrogate functions. Specifically, for each Y_k, surrogate functions of Y_i that are ancestors of Y_k are involved as multiplications. In essence, this is because the surrogated predictions of one classifier are used in computing the loss of downstream classifiers. As each surrogate function is involved as a single term in the multiplication in Eqs. (4.7) and (4.8), the gradients of $R_\phi(h_k)$ and $T_\phi(h_k)$ can be easily computed. However, it is important to know whether such "passing down" process would accumulate surrogate errors and affect the accuracy of classification. We analyze the risk bound of the optimization in the next subsection. s

4.3.3 Excess Risk Bound

Our main result of the excess risk bound is on the unconstrained optimization of the loss function. We show that for each classifier h_k, ϕ-loss $R_\phi(h_k)$ approaching its unconstrained optimum R_ϕ^* indicates that classification error $R(h_k)$ also approaching its unconstrained optimum R^*, no matter how many classifiers are involved in the formulation. Although this result does not directly give the risk bound to our constrained optimization problem, it can be easily extended to the constrained situation if we treat the constraints as penalty terms that are to be added to the loss function.

We first formally define R^* and R^*_ϕ, which are optimums of $R(h_k)$ and $R_\phi(h_k)$ over all possible classifiers. By replacing each $h_i(\mathbf{z}_i)$ in $R(h_k)$ and $R_\phi(h_k)$ with a real-valued variable α_i, we define that $R^* = \inf_{\forall i, \alpha_i \in \mathbb{R}} R(h_k)$ and $R^*_\phi = \inf_{\forall i, \alpha_i \in \mathbb{R}} R_\phi(h_k)$. Then, we define the generic ϕ-conditional risk $C^\eta_\phi(\alpha)$, optimal ϕ-conditional risk $H_\phi(\eta)$, constrained optimal ϕ-conditional risk $H^-_\phi(\eta)$, and ψ-transform. All these definitions are consistent to those in [70]. Let η be a value in $[0,1]$, $\bar{\eta} = 1 - \eta$, α be an arbitrary real value, and ϕ be a surrogate function, we define that $C^\eta_\phi(\alpha) = \eta\phi(\alpha) + \bar{\eta}\phi(-\alpha)$, $H_\phi(\eta) = \inf_{\alpha \in \mathbb{R}} C^\eta_\phi(\alpha)$, $H^-_\phi(\eta) = \inf_{\alpha : \alpha(2\eta - 1) \leq 0} C^\eta_\phi(\alpha)$. As in [70], we require ϕ to be classification-calibrated, i.e., for any $\eta \neq 1/2$, $H^-_\phi(\eta) > H_\phi(\eta)$. We then define ψ by $\psi = \tilde{\psi}^{**}$ where $\tilde{\psi}(\gamma) = H^-_\phi(\frac{1+\gamma}{2}) - H_\phi(\frac{1+\gamma}{2})$ and g^{**} is the Fenchel–Legendre biconjugate of g. We show that, by without loss of generality assuming that $\phi(0) = 1$ and $\inf_{\alpha \in \mathbb{R}} \phi(\alpha) = 0$, we can obtain a risk bound for h_k that is the same as that for the single decision model optimization [70], as given in the following theorem.

Theorem 1. *For any classification-calibrated surrogate function ϕ satisfying $\phi(0) = 1$ and $\inf_{\alpha \in \mathbb{R}} \phi(\alpha) = 0$, any measurable function h_k for predicting Y_k, we have*

$$\psi(R(h_k) - R^*) \leq R_\phi(h_k) - R^*_\phi,$$

where $\psi(\delta)$ is a non-decreasing function mapping from $[0,1]$ to $[0,\infty)$.

Lemma 1. *For ψ, H_ϕ and H^-_ϕ, they have following properties.*

1. *For $\lambda \in [0,1]$ and $\gamma \in \mathbb{R}$, $\psi(\lambda\gamma) \leq \lambda\psi(\gamma)$.*

2. *$H^-_\phi(\eta) \geq H_\phi(\eta)$ for $\eta \in [0,1]$.*

3. $\eta \leq H_\phi(\eta)$ *for* $\eta \in [0, 1/2]$.

4. $\eta \leq 1 \leq H_\phi^-(\eta)$ *for* $\eta \in [0, 1]$.

Proof. Parts 1,2,3 are proved in [70]. For Part 4, note that H_ϕ is concave and symmetric about 1/2, meaning that it gets its minimum at $\eta = 0, 1$ and maximum at $\eta = 1/2$ [70]. We have $H_\phi(0) = H_\phi(1) = \inf_{\alpha\in\mathbb{R}} \phi(\alpha) = 0$. Meanwhile, we have $H_\phi(1/2) = 1/2 \cdot \inf_{\alpha\in\mathbb{R}}(\phi(\alpha)+\phi(-\alpha))$. Due to the convexity and symmetry between $\phi(\alpha)$ and $\phi(-\alpha)$, we can see that $H_\phi(1/2) = \phi(0) = 1$. Then, since H_ϕ is concave, we have $\eta H_\phi(1/2) + \bar{\eta} H_\phi(0) \leq H_\phi(\eta/2 + \bar{\eta} \cdot 0)$, which leads to $\eta \leq H_\phi(\eta/2) \leq H_\phi(\eta)$ for $\eta \in [0, 1/2]$.

For Part 5, note that H_ϕ^- is concave on $[0, 1/2]$ and on $[1/2, 1]$ and also symmetric about 1/2 [70]. Since $H_\phi^-(1/2) = H_\phi(1/2) = 1$ and $H_\phi^-(0) = H_\phi^-(1) = \inf_{\alpha\leq 0} \phi(\alpha) = \phi(0) = 1$, we have $H_\phi^-(\eta) \geq 1 \geq \eta$. □

Next, we first prove Theorem 1 based on the toy example in the main paper, and then explain how this proof can be extended to general situations.

4.3.3.1 Proof of Theorem 1 Based on Toy Example

Proof of Theorem 1. The causal graph of the toy example is shown in Fig. 4.1. In the example, we have two classifiers h_1, h_2. Note that $R_\phi(h_1)$ is the same as that of a single decision model, so we focus on $R_\phi(h_2)$. Denoting $\mathbf{Z} = \{S, X_1, X_2\}$, we define

$$c_1(\mathbf{z}) = \frac{P(y_1^+|s, x_1, x_2)}{P(y_1^+|s, x_1)} + \frac{P(y_1^-|s, x_1, x_2)}{P(y_1^-|s, x_1)},$$

and define

$$\eta_1(\mathbf{z}) = \frac{P(y_1^-|s, x_1, x_2)}{c_1(\mathbf{z})}, \quad \eta_2(\mathbf{z}) = P(y_2^+|x_2),$$

and

$$\bar{\eta}_1(\mathbf{z}) = 1 - \eta_1(\mathbf{z}), \quad \bar{\eta}_2(\mathbf{z}) = 1 - \eta_2(\mathbf{z}),$$

For simplifying representation, in the remaining of this file we omit $(\mathbf{z})$ in all expressions.

Note that

$$\begin{aligned}
R_\phi(h_2) &= \mathop{\mathbb{E}}_{\mathbf{z}}\left[c_1\left(\eta_1\eta_2\phi(h_1(x_1))\phi(h_2(x_2)) + \bar{\eta}_1\eta_2\phi(-h_1(x_1))\phi(h_2(x_2))\right.\right. \\
&\quad \left.\left. +\eta_1\bar{\eta}_2\phi(h_1(x_1))\phi(-h_2(x_2)) + \bar{\eta}_1\bar{\eta}_2\phi(-h_1(x_1))\phi(-h_2(x_2))\right)\right] \\
&= \mathop{\mathbb{E}}_{\mathbf{z}}\left[c_1(\eta_1\phi(h_1(x_1)) + \bar{\eta}_1\phi(-h_1(x_1) > 0))(\eta_2\phi(h_2(x_2)) + \bar{\eta}_2\phi(-h_2(x_2)))\right] \\
&= \mathop{\mathbb{E}}_{\mathbf{z}}\left[c_1 C_\phi^{\eta_1}(h_1(x_1))C_\phi^{\eta_2}(h_2(x_2))\right],
\end{aligned}$$

we can express $R_\phi(h_2)$ using the generic ϕ-conditional risk $C_\phi^\eta(\alpha)$. According to the definition of R_ϕ^*, we correspondingly have

$$R_\phi^* = \mathop{\mathbb{E}}_{\mathbf{z}}\left[c_1 H_\phi(\eta_1)H_\phi(\eta_2)\right].$$

Similarly we can also express $R(h_2)$ and R^* as

$$R(h_2) = \mathop{\mathbb{E}}_{\mathbf{z}}\left[c_1 C^{\eta_1}(h_1(x_1))C^{\eta_2}(h_2(x_2))\right],$$

$$R^* = \mathop{\mathbb{E}}_{\mathbf{z}}[c_1 H(\eta_1)H(\eta_2)],$$

where $C^\eta(\alpha)$ and $H(\eta)$ are defined by replacing ϕ with $\mathbb{1}$ in $C_\phi^\eta(\alpha)$ and $H_\phi(\eta)$. Note that $H(\eta)$ is always obtained when the sign of α is the same as the sign of $\eta - 1/2$.

Denote by α^* the signs of solutions $\{\text{sign}(\eta_1 - 1/2), \text{sign}(\eta_2 - 1/2)\}$. Then, we have

$$\begin{aligned} R(h_2) - R^* &= \mathop{\mathbb{E}}_{\mathbf{z}} \left[c_1 \left(C^{\eta_1}(h_1(x_1)) C^{\eta_2}(h_2(x_2)) - H(\eta_1) H(\eta_2) \right) \right] \\ &= \mathop{\mathbb{E}}_{\mathbf{z}} \left[c_1 \mathbb{1}(\text{sign}(h) \neq \alpha^*) \left(C^{\eta_1}(h_1(x_1)) C^{\eta_2}(h_2(x_2)) - H(\eta_1) H(\eta_2) \right) \right]. \end{aligned}$$

Since ψ is convex [70], it follows that

$$\psi(R(h_2) - R^*) \leq \mathop{\mathbb{E}}_{\mathbf{z}} \left[c_1 \mathbb{1}(\text{sign}(h) \neq \alpha^*) \psi \left(C^{\eta_1}(h_1(x_1)) C^{\eta_2}(h_2(x_2)) - H(\eta_1) H(\eta_2) \right) \right].$$

Without loss of generality, assume $\eta_1 \leq \bar{\eta}_1$ and $\eta_2 \leq \bar{\eta}_2$. Thus, according to the definition, $H(\eta_1) = \eta_1$ and $H(\eta_2) = \eta_2$. Then, we want to show that for any h_1, h_2 whose signs are not equivalent to α^*, we have

$$\psi \left(C^{\eta_1}(h_1(x_1)) C^{\eta_2}(h_2(x_2)) - H(\eta_1) H(\eta_2) \right) \leq H_\phi^-(\eta_1) H_\phi^-(\eta_2) - H_\phi(\eta_1) H_\phi(\eta_2). \tag{4.9}$$

To this end, we consider two cases: (1) only one classifier from h_1, h_2 makes the prediction that is opposite to α^*; and (2) both h_1, h_2 make predictions that are opposite to α^*.

For Case (1), assume that h_1 makes the opposite prediction. Thus, $C^{\eta_1}(h_1(x_1)) = \bar{\eta}_1$, and $C^{\eta_2}(h_2(x_2)) = \eta_2$. Then, we have

$$\psi \left(C^{\eta_1}(h_1(x_1)) C^{\eta_2}(h_2(x_2)) - H(\eta_1) H(\eta_2) \right) = \psi \left((\bar{\eta}_1 - \eta_1) \eta_2 \right).$$

Based on Lemma 1, Part 1, it follows that

$$\psi \left((\bar{\eta}_1 - \eta_1) \eta_2 \right) \leq \eta_2 \psi \left(\bar{\eta}_1 - \eta_1 \right) = \eta_2 \left(H_\phi^-(\eta_1) - H_\phi(\eta_1) \right).$$

Based on Lemma 1, Part 3, we have $\eta_2 \leq H_\phi(\eta_2)$. So it follows that

$$\psi\left((\bar{\eta}_1 - \eta_1)\eta_2\right) \leq \left(H_\phi^-(\eta_1) - H_\phi(\eta_1)\right) H_\phi(\eta_2).$$

Based on Lemma 1, Part 2, we prove Eq. (4.9).

For Case (2), we have $C^{\eta_1}(h_1(x_1)) = \bar{\eta}_1$, and $C^{\eta_2}(h_2(x_2)) = \bar{\eta}_2$. Thus,

$$\psi\left(C^{\eta_1}(h_1(x_1))C^{\eta_2}(h_2(x_2)) - H(\eta_1)H(\eta_2)\right) = \psi\left(\bar{\eta}_1\bar{\eta}_2 - \eta_1\eta_2\right).$$

Without loss of generality, assume $\eta_1 \leq \eta_2$, i.e., $\bar{\eta}_2 \leq \bar{\eta}_1$. We have that

$$\begin{aligned}
&\bar{\eta}_1\bar{\eta}_2 - \eta_1\eta_2 = \bar{\eta}_1\bar{\eta}_2 - \eta_1\eta_2 - \eta_1\bar{\eta}_2 + \eta_1\bar{\eta}_2 \\
&= \bar{\eta}_2(\bar{\eta}_1 - \eta_1) + \eta_1(\bar{\eta}_2 - \eta_2) \\
&\leq \bar{\eta}_1(\bar{\eta}_1 - \eta_1) + \eta_1(\bar{\eta}_2 - \eta_2).
\end{aligned} \tag{4.10}$$

Since ψ is convex, we have

$$\begin{aligned}
&\psi\left(\bar{\eta}_1\bar{\eta}_2 - \eta_1\eta_2\right) \leq \psi\left(\bar{\eta}_1(\bar{\eta}_1 - \eta_1) + \eta_1(\bar{\eta}_2 - \eta_2)\right) \\
&\leq \bar{\eta}_1\psi(\bar{\eta}_1 - \eta_1) + \eta_1\psi(\bar{\eta}_2 - \eta_2).
\end{aligned}$$

According to the definition of ψ, we have $\psi(\bar{\eta} - \eta) = H_\phi^-(\eta) - H_\phi(\eta)$. According to Lemma 1, Parts 4&3, we have $\bar{\eta}_1 \leq 1 \leq H_\phi^-(\eta_2)$, $\eta_1 \leq H_\phi(\eta_1)$. As a result, we have

$$\psi\left(\bar{\eta}_1\bar{\eta}_2 - \eta_1\eta_2\right) \leq H_\phi^-(\eta_2)\left(H_\phi^-(\eta_1) - H_\phi(\eta_1)\right) + H_\phi(\eta_1)\left(H_\phi^-(\eta_2) - H_\phi(\eta_2)\right)$$

which proves Eq. (4.9).

Finally, we have

$$
\begin{aligned}
\psi(R(h_2) - R^*) &\leq \mathop{\mathbb{E}}_{\mathbf{z}} \left[c_1 \mathbb{1}(\mathrm{sign}(h) \neq \alpha^*) \left(H_\phi^-(\eta_1) H_\phi^-(\eta_2) - H_\phi(\eta_1) H_\phi(\eta_2) \right) \right] \\
&\leq \mathop{\mathbb{E}}_{\mathbf{z}} \left[c_1 \mathbb{1}(\mathrm{sign}(h) \neq \alpha^*) \left(C_\phi^{\eta_1}(h_1(x_1)) C_\phi^{\eta_2}(h_2(x_2)) - H_\phi(\eta_1) H_\phi(\eta_2) \right) \right] \\
&\leq \mathop{\mathbb{E}}_{\mathbf{z}} \left[c_1 \left(C_\phi^{\eta_1}(h_1(x_1)) C_\phi^{\eta_2}(h_2(x_2)) - H_\phi(\eta_1) H_\phi(\eta_2) \right) \right] \\
&= R_\phi(h_2) - R_\phi^*.
\end{aligned}
$$

□

4.3.3.2 Extending to General Situations

We prove that Theorem 1 can be extended to h_3, then, it can be similarly extended to any k. Note that the key is to prove

$$
\begin{aligned}
&\psi\left(C^{\eta_1}(h_1(x_1)) C^{\eta_2}(h_2(x_2)) C^{\eta_3}(h_3(x_3)) - H(\eta_1) H(\eta_2) H(\eta_3)\right) \\
&\leq H_\phi^-(\eta_1) H_\phi^-(\eta_2) H_\phi^-(\eta_3) - H_\phi(\eta_1) H_\phi(\eta_2) H_\phi(\eta_3).
\end{aligned}
\tag{4.11}
$$

Similarly, we consider three cases: (1) only one classifier from h_1, h_2, h_3 makes the prediction that is opposite to α^*; (2) two classifiers from h_1, h_2, h_3 make predictions that are opposite to α^*; and (3) all three classifiers make predictions that are opposite to α^*.

For Case (1), the proof is similar to that in Section 4.3.3.1.

For Case (2), assume that h_1, h_2 make the opposite predictions. Then, we have

$$
\begin{aligned}
&\psi\left(C^{\eta_1}(h_1(x_1)) C^{\eta_2}(h_2(x_2)) C^{\eta_3}(h_3(x_3)) - H(\eta_1) H(\eta_2) H(\eta_3)\right) \\
&= \psi\left((\bar{\eta}_1 \bar{\eta}_2 - \eta_1 \eta_2) \eta_3\right) \leq \eta_3 \psi\left(\bar{\eta}_1 \bar{\eta}_2 - \eta_1 \eta_2\right).
\end{aligned}
$$

Thus, based on Eq. (4.9), we can prove Eq. (4.11).

For Case (3), without loss of generality, assume that $\eta_1 \leq \eta_2 \leq \eta_3$, i.e., $\bar{\eta}_3 \leq \bar{\eta}_2 \leq \bar{\eta}_1$. Then, we have

$$\begin{aligned}
&\psi\left(C^{\eta_1}(h_1(x_1))C^{\eta_2}(h_2(x_2))C^{\eta_3}(h_3(x_3)) - H(\eta_1)H(\eta_2)H(\eta_3)\right)\\
&= \psi\left(\bar{\eta}_1\bar{\eta}_2\bar{\eta}_3 - \eta_1\eta_2\eta_3\right)\\
&= \psi\left(\bar{\eta}_3(\bar{\eta}_1\bar{\eta}_2 - \eta_1\eta_2) + \eta_1\eta_2(\bar{\eta}_3 - \eta_3)\right).
\end{aligned}$$

Based on Eq. (4.10), it follows that

$$\begin{aligned}
&\psi\left(\bar{\eta}_3(\bar{\eta}_1\bar{\eta}_2 - \eta_1\eta_2) + \eta_1\eta_2(\bar{\eta}_3 - \eta_3)\right)\\
&= \psi\left(\bar{\eta}_3\bar{\eta}_2(\bar{\eta}_1 - \eta_1) + \bar{\eta}_3\eta_1(\bar{\eta}_2 - \eta_2) + \eta_1\eta_2(\bar{\eta}_3 - \eta_3)\right)\\
&\leq \psi\left(\bar{\eta}_1\bar{\eta}_2(\bar{\eta}_1 - \eta_1) + \bar{\eta}_2\eta_1(\bar{\eta}_2 - \eta_2) + \eta_1\eta_2(\bar{\eta}_3 - \eta_3)\right)\\
&\leq \bar{\eta}_1\bar{\eta}_2\psi\left(\bar{\eta}_1 - \eta_1\right) + \bar{\eta}_2\eta_1\psi\left(\bar{\eta}_2 - \eta_2\right) + \eta_1\eta_2\psi\left(\bar{\eta}_3 - \eta_3\right).
\end{aligned}$$

Then, since $\bar{\eta}_1 \leq 1 \leq H_\phi^-(\eta_2)$, $\bar{\eta}_2 \leq 1 \leq H_\phi^-(\eta_3)$, $\eta_1 \leq H_\phi(\eta_1)$, $\eta_2 \leq H_\phi(\eta_2)$, it follows that

$$\begin{aligned}
&\bar{\eta}_1\bar{\eta}_2\psi\left(\bar{\eta}_1 - \eta_1\right) + \bar{\eta}_2\eta_1\psi\left(\bar{\eta}_2 - \eta_2\right) + \eta_1\eta_2\psi\left(\bar{\eta}_3 - \eta_3\right)\\
&\leq H_\phi^-(\eta_2)H_\phi^-(\eta_3)\left(H_\phi^-(\eta_1) - H_\phi(\eta_1)\right) + H_\phi^-(\eta_3)H_\phi(\eta_1)\left(H_\phi^-(\eta_2) - H_\phi(\eta_2)\right)\\
&\quad + H_\phi(\eta_1)H_\phi(\eta_2)\left(H_\phi^-(\eta_3) - H_\phi(\eta_3)\right)\\
&= H_\phi^-(\eta_1)H_\phi^-(\eta_2)H_\phi^-(\eta_3) - H_\phi(\eta_1)H_\phi(\eta_2)H_\phi(\eta_3),
\end{aligned}$$

which proves the Eq. (4.11).

The meaning of Theorem 1 clearly gives the following corollary.

Corollary 1. $R_\phi(h_k) \to R_\phi^*$ *indicates* $R(h_k) \to R^*$.

4.4 Experiments

4.4.1 Experiment Setup

We evaluate our method using both synthetic and real-world data. Table 4.1 provides a summary of two datasets' statistics. For the synthetic data, we manually define a causal graph with five variables S, X_1, X_2, Y_1, Y_2 shown in Fig. 4.2. Then, a conditional probability table is defined for each attribute over its parents, and the data is generated by sampling each attribute in topological order according to the conditional probability. For the real-world data, we use the Adult dataset [72] and build the causal graph by using the PC algorithm implemented in the Tetrad [73]. We follow the settings in [67] to select 7 out of 11 attributes and binarize their domain values. The significant threshold for conditional independence testing is set as 0.01, and three tiers in the partial order are used. We handle this imbalanced data using the over-sampling technique [74]. The resultant dataset consists of 10,1472 records. The causal graph is shown in Fig. 4.3. We treat `Age` as the protected attribute S, and `Workclass` and `Income` as two decisions Y_1, Y_2. By default, we use 0.05 as the threshold for judging fairness.

We design an evaluation process which simulates the real model deployment procedure. The dataset is randomly split to training and testing datasets. We deploy and evaluate the learned classifiers sequentially according to their topological order. The first classifier h_1 is deployed first, and evaluated on the original testing dataset. After that, it produces

Table 4.1: Dataset statistics

Dataset	#Instances	#Attributes	Sensitive Variable	Decision Variable
Sythetic	10,000	5	S	Y_1, Y_2
Adult	101,472	7	age	workclass, income

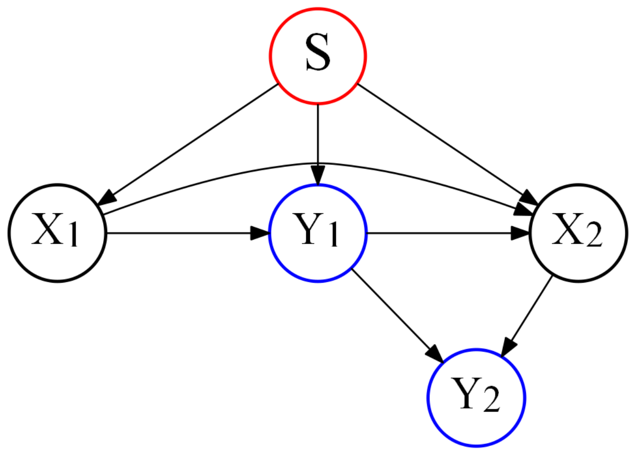


Figure 4.2: The causal graph for the synthetic dataset.

predicted decisions for Y_1, which are then used to re-generate the values of the subsequent variables in the order, as well as the true values of the next classifier h_2, by using the causal graph. In the end, we evaluate h_2 based on the re-generated data.

For training, our method (referred to as the joint method) formulates the optimization problem on the training data to learn all classifiers simultaneously. We also consider a simplified version of our method (referred to as the serial method) that learns classifiers sequentially following the topological order similarly to the deployment procedure. Each classifier only uses the direct parents of the label. After each classifier is learned, it is treated as a soft intervention such that the post-intervention distribution is inferred and used to train subsequent classifiers. We compare our methods with a baseline method (referred to the separate method) where each classifier is learned using the direct parents separately on the training data.

4.4.2 Implementation

All classifiers are implemented as empirical risk minimization classifiers where the logistic surrogate function is used. For unconstrained, separate, and serial methods, each

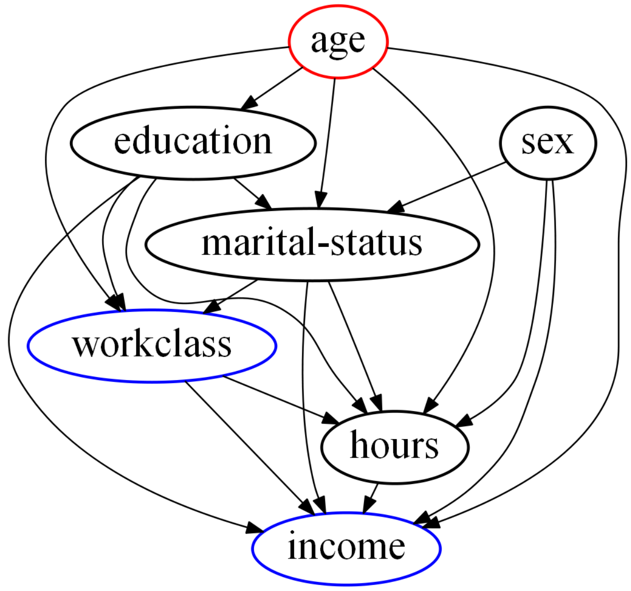


Figure 4.3: The causal graph for the Adult dataset.

classifier is learned individually as a convex optimization problem. Thus, we use the CVXPY package [75] to directly solve the unconstrained/constrained convex optimization problem. For the joint method, since the objective function and constraints are non-convex, we add constraints as penalty terms to the objective function and adopt PyTorch [76] to optimize it using the Adam optimizer. The convergence of Adam algorithms for non-convex optimization has been studied, e.g., in [77]. All experiments are conducted in a PC with 8GB RAM and Intel Core i5-1035G1 CPU.

4.4.3 Experimental Results

As discussed, since separate training does not consider the change in the data distribution caused by the deployment of new classifiers, it fails to achieve fairness in testing even if the classifier is fair in training. To demonstrate this, Table 4.2 and Table 4.3 show the results of one typical setting for each method on both synthetic and Adult datasets, obtained from 5-fold cross-validation. For all methods, we manage to build classifiers that

Table 4.2: Accuracy and unfairness from Unconstrained, Separate, Serial and Joint methods on synthetic data (bold values indicate violation of fairness).

Phase			Synthetic			
			Uncons.	Separate	Serial	Joint
Train	h_1	Acc. (%)	80.32	75.35	75.35	75.35
		Unfairness	**0.15**	0.01	0.01	0.01
	h_2	Acc. (%)	90.13	75.79	84.02	82.77
		Unfairness	**0.23**	0.04	0.03	0.04
Test	h_1	Acc. (%)	80.70	75.54	75.54	75.54
		Unfairness	**0.15**	0.01	0.01	0.01
	h_2	Acc. (%)	89.95	77.06	84.16	82.09
		Unfairness	**0.13**	**0.09**	0.03	0.03

are fair in training. For the Adult dataset, we use 0.1 as the fairness threshold for h_2. We can see that, in testing, the serial and joint methods achieve consistent performance, but the separate method cannot guarantee to achieve fairness for h_2. We also did a grid search on thresholds τ_1, τ_2 on the synthetic data to find classifier pairs h_1, h_2 whose fairness is between -0.05 and 0.05 in training. Then, we evaluated these classifiers in testing. We observe that, even if we use the training data for testing to avoid any generalization error, in 71.43% of these pairs produced by the separate method, h_2 exceeded the interval [-0.05, 0.05] and hence violated the fairness criterion. On the contrary, all classifiers produced by the serial and joint methods are fair in testing.

Comparing the serial and joint methods, they obtain similar results. This is expected since both of them apply the soft intervention to capture the model deployment. The advantage of the joint method is that it can adjust all classifiers simultaneously to obtain a better overall performance. This is not shown in current experiments probably due to the small scale of the problem. We will study whether the joint method would outperform the serial method in larger problems in our future work.

Table 4.3: Accuracy and unfairness from Unconstrained, Separate, Serial and Joint methods on Adult data (bold values indicate violation of fairness).

Phase			Synthetic			
			Uncons.	Separate	Serial	Joint
Train	h_1	Acc. (%)	55.71	55.64	55.63	55.63
		Unfairness	**0.15**	0.05	0.05	0.05
	h_2	Acc. (%)	76.75	71.17	68.90	69.31
		Unfairness	**0.24**	0.10	0.10	0.10
Test	h_1	Acc. (%)	55.63	55.56	55.57	55.57
		Unfairness	**0.15**	0.05	0.05	0.05
	h_2	Acc. (%)	77.07	73.33	68.91	69.40
		Unfairness	**0.23**	**0.17**	0.10	0.10

4.5 Summary

In this chapter, we proposed an approach that learns multiple fair classifiers from a static training dataset, which is a general way to incorporate fairness constraints into the generic classification formulation such that we can readily employ off-the-shelf classification models and optimization algorithms. We treated the deployment of each classifier as a soft intervention and inferred the distributions after the deployment as post-intervention distributions. We adopted surrogate functions to smooth the loss function and fair constraints to formulate the fair classification problem as a constrained optimization problem. In addition, we theoretically showed that combining multiple decision models in the optimization would not introduce additional surrogate errors. By conducting experiments on both synthetic and real-world datasets, we showed that our approach consistently outperforms the approach of building fair classifiers for each decision independently, and performs closely to the sequential learning approach where new data needs to be generated and collected after each model deployment.

5 Achieving Long-term Fairness in Sequential Decision Making

5.1 Introduction

Fair machine learning has received increasing attention in the past years, especially in decision making tasks such as hiring [1], college admissions [3] and bank loans [5]. Many algorithms for achieving fair decision making have been proposed based on various fairness notions (e.g. demographic parity [78], equalized odds [79] and counterfactual fairness [80]). At present, the majority of studies on fair machine learning focus on the static or one-shot classification setting. However, in practice, decision making systems are usually operating in a dynamic manner such that the classifier makes sequential decisions over a period of time. In many situations, each decision made by the classifier may change the underlying data population and further affect subsequent decisions. For example, suppose a person applies to a bank for a loan and the bank estimates the risk of default according to his/her credit score. Then, the bank's decision on the loan application (e.g., whether to grant the loan and the interest rate assigned) may in turn affect the default risk and change the person's credit score (e.g., the credit score will decrease if the loan is granted but he/she defaults on the loan) which will affect his/her next loan application. If the bank's decision leads to a long-term decrease in the credit score, then it imposes a negative long-term effect on future decisions for this person. Therefore, fair decision making should concern not only the fairness of a single decision but more importantly, whether a decision model can impose fair long-term effects on different groups. This notion of fairness is referred to as long-term fairness in recent studies [24, 22, 51].

The challenge of achieving long-term fairness comes in two folds. Firstly, different from static settings, decisions made by models may change users' behaviors, and/or affect their status such as reputation, qualification, etc., and impact subsequent decisions via feedback loops. Without knowing how the population would be reshaped by decisions, enforcing any fairness constraint may create negative feedback loops and eventually harm fairness in the long run. Recent research has demonstrated that existing fairness criteria cannot guarantee fairness and sometimes undermine fairness even if only one time step is taken into consideration [24, 81, 48, 49]. Secondly, due to the feedback loops, the deployment of the decision model will cause changes in the data distribution that is originally used for training. This can be viewed as a distribution shift problem as the distribution of the training data (i.e., distribution before the model deployment) is different from the distribution of the test data (i.e., distribution after the model deployment). Ignoring the distribution shift will critically affect the achievement of long-term fairness, as long-term fairness is affected by all decisions made by the model along the time.

In this chapter, we propose a framework for achieving long-term fair sequential decision making by addressing both above challenges. We model the dynamics of the decision-making process by employing Pearl's Structural Causal Model (SCM) [29], in which the relations among user features and decisions and how those decisions affect the data distribution can be encoded in a probabilistic graphical model. Specifically, we leverage the time-lagged causal graph [82] to describe the causal relations over time, and adopt the soft intervention [54] for modeling the model deployment and inferring its impacts on the underlying population. Then, we measure long-term fairness as the path-specific effect on the time-lagged causal graph under both the hard intervention on the sensitive attribute and the soft intervention on the predicted decisions. A constrained optimization problem is formulated to strike a trade-off between long-term fairness and model utility, as well as certain short-term fairness requirement that may be stipulated by law or regulations. On the other hand, we show that the constrained optimization problem can be converted to a performative risk optimization problem [83]. Then, we employ the repeated risk minimization (RRM) training technique [83] for dealing with the distribution shift problem. The performative optimality and stability properties of the proposed method are theoretically and empirically evaluated which shows its effectiveness.

To the best of our knowledge, this work is the first to propose a causality-based long-fairness notion. The proposed learning framework is general such that it could incorporate different combinations of surrogate functions, utility loss functions, as well as causal paths regarding long-term fairness used to fit different applications. The experiment results show that the proposed method can achieve long-term fairness for multiple time steps, while the fairness performance deteriorates with time if no fairness constraint or static fairness constraints are used.

5.2 Fairness-aware Classification

The classification problem is to learn a functional mapping $f : \mathcal{X} \mapsto \mathcal{Y}$ from the labeled training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ where $\mathbf{x}_i \in \mathcal{X}$, $y_i \in \mathcal{Y}$ and $\mathcal{Y} = \{-1, 1\}$, by minimizing the 0-1 loss function $\mathbb{E}_{\mathbf{X},Y}[\mathbb{1}[f(\mathbf{X}) \neq Y]]$ where $\mathbb{1}[\cdot]$ is an indicator function. In general, f is made up of another function h set up in the real number domain, i.e., $h : \mathcal{X} \mapsto \mathbb{R}$ and $\mathbb{1}[f(\mathbf{X}) \neq Y] = \mathbb{1}[Yh(\mathbf{X}) \geq 0]$. Since directly minimizing the indicator is intractable, one can replace it with a smooth and differentiable surrogate function ϕ. Then, the loss function can can be reformulated as $\mathbb{E}_{\mathbf{X},Y}[\phi(Yh(\mathbf{X}))]$. Similarly, one can also formulate fairness constraints as smoothed expressions using surrogate functions. As a result, fair classification problems can be formulated as constrained optimization problems [38, 84]. We follow the notations used in [38, 84] in our formulations.

5.3 Formulating Long-term Fairness

We start by formally formulating the long-term fairness in sequential decision making. Assume we have access to a temporal dataset $\mathcal{D} = \{(S, \mathbf{X}^t, Y^t)\}_{t=1}^{l}$ where S is a time-invariant protected attribute, $\mathbf{X}^t$ is a set of time-dependent unprotected attributes and Y^t is a time-dependent class label. Note that this setting can be viewed as observing the data of a set of individuals at all time steps, or a more general situation where a population is subject to the decision cycles and the data is sampled at each time step. For ease of discussion, we assume both class label and protected attribute are binary variables, i.e., $S = \{s^+, s^-\}$ with s^+ denoting the unprotected group and s^- denoting the protected group, and $Y = \{1, -1\}$ with 1 denoting the positive decision and -1 denoting the negative decision, but proposed concepts could be extended to multiple protected attributes and multiple/continuous labels situations. A predictive decision model $h_\theta(\cdot)$ parameterized by θ is trained on $\mathcal{D}$. Then, it is deployed to make predicted decisions $\hat{Y}^t$ from $(S, \mathbf{X}^t)$ repeatedly at each time, i.e., $\hat{Y}^t = 1$ if $h_\theta(s, \mathbf{x}^t) \geq 0$ and $\hat{Y}^t = -1$ otherwise, forming a sequential decision making process. Such sequential decision making process is common in practice. For example, a bank repeatedly makes lending decisions based on applicants' profile such as credit score, income, etc., and a predictive policing algorithm repeatedly makes decision about where to send police for patrolling based on the crime discovered in the neighborhood. The ultimate goal of long-term fair machine learning is to ensure that the model $h_\theta(\cdot)$ is fair in a long-term stage denoted by t*. In this chapter, we assume there is sufficient historical training data such that $l \geq t^*$.

5.3.1 Causality-based Long-term Fairness

We develop the long-term fairness notion by leveraging Pearl's SCM. First, we assume a time-lagged causal graph $\mathcal{G}$ for describing the causal relationship among variables over time. In recent years, structure learning algorithms have been proposed for constructing time-lagged causal graphs from data, including both constrained-based approaches [85, 86, 87] and continuous optimization-based approaches [88, 89] which can be leveraged to learn the time-lagged causal graph from data. Figure 5.1 shows a typical example of the time-lagged causal graph in our settings: the edge from S to $\mathbf{X}^0$ represents the bias in the distribution of $\mathbf{X}$ due to historical reasons; the edges from S and $\mathbf{X}^t$ to Y^t represent that S and $\mathbf{X}^t$ are used as the input to compute $\hat{Y}^t$; and the edges from $\mathbf{X}^t$ and Y^t to $\mathbf{X}^{t+1}$ represent how the distribution of $\mathbf{X}$ would be reshaped via feedback after each decision.

Next, we formulate long-term fairness as path-specific effects that are transmitted in the time-lagged causal graph along certain paths. The path-specific effects reflect how the intervention affects each variable on the path in a topological order and hence are appropriate for capturing dynamics in sequential decision making. Similar to the indirect discrimination in static fair machine learning [64, 66], we can also justify the use of the path-specific effect by the need to distinguish discriminatory effects from explainable effects. We consider discriminatory effects as those which are due to biased decisions made by the decision making system in the past and will continue to influence future decisions. Correspondingly, we consider explainable effects as those which are attributed to external factors and cannot be eliminated within the decision making system. To this end, we categorize unprotected attributes $\mathbf{X}$ into two disjoint subsets: irrelevant attributes $\mathbf{X}_i$ and relevant attributes $\mathbf{X}_r$. We define irrelevant attributes as those which are justifiable in decision making, and meanwhile evolved autonomously or/and altered by external factors only. We define the rest of attributes as relevant attributes, which could be unjustifiable in decision making or reshaped by the decision over time. Then, we define long-term fairness as the causal effect where the influence of the hard intervention on S is transmitted in the causal graph by passing through relevant attributes only. Note that the influence of the soft intervention on Y is still transmitted through all causal paths.

Finally, we propose to adopt soft interventions as a key technique for modeling decision model deployment and inferring its impacts on the underlying population. We treat the deployment of the decision model at each time step as to perform a soft intervention on the decision variable. More specifically, we force the structural equation associated with Y^t in the causal model to be replaced by the decision model $h_\theta(\cdot)$ that outputs $\hat{Y}^t$. Thus, the change to underlying population could be inferred as the post-intervention distribution after performing the soft intervention. Meanwhile, to quantify fairness as causal effects of the protected attribute on the decision, we perform hard intervention on the protected attribute in order to answer the "what if" question, i.e., "what would the decision be if we intervene the gender of applications to female?" As a result, we perform both hard intervention and soft intervention simultaneously for measuring long-term fairness as causal effects.

Symbolically, denote by π the set of causal paths from S to $\hat{Y}^{t^*}$ through relevant attributes $\mathbf{X}_r^1, \cdots, \mathbf{X}_r^{t^*}$ and $\hat{Y}^1, \cdots, \hat{Y}^{t^*-1}$ but not through irrelevant attributes $\mathbf{X}_i^0, \cdots, \mathbf{X}_i^{t^*}$. Meanwhile, as we conduct path-specific hard intervention on S and soft interventions on Y to deploy decision model $h_\theta(\cdot)$, we denote the post-intervention distribution of $\hat{Y}^t$ by $\hat{Y}^t(s_\pi, \theta)$ which explicitly shows that the soft intervention depends on parameters θ. Then, we can readily propose the quantitative notion for long-term fairness.

Definition 11 (Long-term Fairness)**.** *The long-term fairness of a decision model* $h_\theta(\cdot)$ *is*

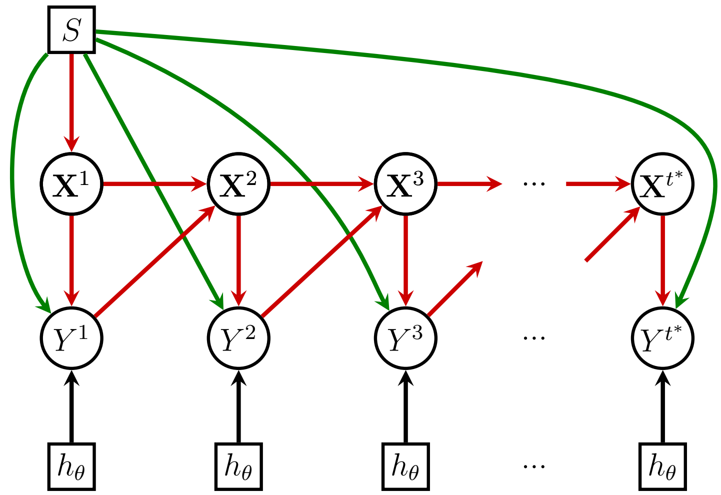


Figure 5.1: A time-lagged causal graph for sequential decision making. Long-term fairness is captured by paths in red, and short-term fairness is captured by paths in green.

measured by $P(\hat{Y}^{t*}(s_\pi^+,\theta)) - P(\hat{Y}^{t*}(s_\pi^-,\theta))$ *where* π *is a set of paths from* S *to* $\hat{Y}^{t*}$ *passing through* $\boldsymbol{X}_r^1$, $\hat{Y}^1$, $\cdots$, $\boldsymbol{X}_r^{t*-1}$, $\hat{Y}^{t*-1}$, $\boldsymbol{X}_r^{t*}$, s_π *represents the path-specific hard intervention and* θ *represents the soft intervention through all paths.*

5.3.2 Loss Function and Short-term Fairness

In addition to long-term fairness, a desired fair decision model should also satisfy two other requirements. Firstly, it is a natural desire for a predictive decision model to maximize the institution utility, e.g., the loan granting model of a bank certainly wants to maximize the expected return from loans. Secondly, the decision model should also satisfy certain short-term fairness requirement at each time step to enforce local equality, which may be stipulated by law or regulations. For example, the Equal Credit Opportunity Act, 1974, prohibits lending decisions from being influenced by race, age, religion, etc. Similar to the direct discrimination in static fair machine learning, we consider a subset of relevant attributes $\tilde{\mathbf{X}}_r \subset \mathbf{X}_r$ which are unprotected but cannot be justifiably used in the decision making either directly or indirectly, referred to as the redlining attributes [64]. Then, we measure the short-term fairness by the causal effect of S on $\hat{Y}^t$ along paths that pass through $\tilde{\mathbf{X}}_r$, i.e., $S \to \tilde{\mathbf{X}}_r \to \hat{Y}^t$, as well as the direct edge $S \to \hat{Y}^t$ at each time step t.

We note that trade-off may exist between fairness and utility, as well as between long-term and short-term fairness. The long-term fairness focuses on remedying past discrimination existed in the system, but has no constraint on the biases in the decision at each time step. The short-term fairness, on the other hand, cares about fairness in the decision making process at each time step, but pays no attention in correcting past discrimination in the population. One should combine long-term and short-term fairness to force the decision model to take into consideration both factors and to remove discrimination in the system gradually with time. Therefore, we similarly propose quantitative notions for short-term fairness and institution utility as follows.

Definition 12 (Short-term Fairness)**.** *The short-term fairness of a decision model $h_\theta(\cdot)$ at time t is measured by the causal effect transmitted through paths involved in time t, i.e., $P(\hat{Y}^t(s^+_{\pi^t}, \theta)) - P(\hat{Y}^t(s^-_{\pi^t}, \theta))$, where $\pi^t = \{S \to \tilde{\mathbf{X}}_r \to \hat{Y}^t, S \to \hat{Y}^t\}$ with redlining attributes $\tilde{\mathbf{X}}_r$, s_π is the path-specific hard intervention and θ represents the soft intervention.*

Definition 13 (Institution Utility)**.** *The institution utility of decision model $h_\theta(\cdot)$ is measured by the aggregate loss given by $\sum_{t=1}^{t*} \mathbb{E}[\mathcal{L}(Y^t, \hat{Y}^t)]$ where $\mathcal{L}(\cdot)$ is the loss function.*

5.4 Learning Fair Decision Models

After formulating related notions, we are ready to formulate the fair sequential decision making problem given a time-lagged causal graph. To ease the representation, in following discussions we consider the simplified causal graph shown in Figure 5.1 where only relevant attributes with no redlining attributes exist. In this case, the long-term fairness is captured by paths from S to $\hat{Y}^{t^*}$ through $\mathbf{X}^1, \hat{Y}^1, \cdots, \mathbf{X}^{t^*}$ as shown in red, and the short-term fairness is captured by the direct edge $S \rightarrow \hat{Y}^t$ at each time t as shown in green. However, all our discussions can be applied to our general formulation that includes both relevant and irrelevant features.

The goal is to learn a functional mapping $h_\theta : (\mathbf{X}^t, S) \mapsto Y^t$ parameterized with θ, i.e., $\hat{Y}^t = h_\theta(\mathbf{X}^t, S)$. Based on the discussions above, we formulate a constrained optimization problem which minimizes the loss while subject to long-term fairness and short-term fairness constrains simultaneously. The thresholds τ_l and τ_t control the strictness of constraints.

Problem Formulation 2. *The problem of fair sequential decision making is formulated as the constrained optimization:*

$$
\begin{aligned}
& \operatorname*{argmin}_{\theta} \sum_{t=1}^{t^*} \mathbb{E}\left[\mathcal{L}(Y^t, \hat{Y}^t)\right] \\
s.t. \quad & P\Big(\hat{Y}^{t^*}(s_\pi^+, \theta) = 1\Big) - P\Big(\hat{Y}^{t^*}(s_\pi^-, \theta) = 1\Big) \leq \tau_l, \\
& P\Big(\hat{Y}^{t}(s_{\pi^t}^+, \theta) = 1\Big) - P\Big(\hat{Y}^{t}(s_{\pi^t}^-, \theta) = 1\Big) \leq \tau_t, \\
& t = 1, \cdots, t^*
\end{aligned}
$$

where τ_l and τ_t are thresholds for long-term fairness and short-term fairness constraints, respectively.

5.4.1 Formulating as Performative Risk Optimization

Solving the optimization problem in Problem Formulation 1 is not trivial. According to the path-specific effect inference [90] and the definition of soft intervention [54], post-intervention probability $P(\hat{Y}^{t^*}(s_\pi^+, \theta) = 1)$ is given by

$$\sum_{\mathbf{X}^1, Y^1, \cdots, \mathbf{X}^{t^*}} \Big\{ P(\mathbf{x}^1|s^+) P_\theta(y^1|\mathbf{x}^1, s^-) \cdots P(\mathbf{x}^{t^*}|\mathbf{x}^{t^*-1}, y^{t^*-1}) P_\theta(Y^{t^*}=1|\mathbf{x}^{t^*}, s^-) \Big\}, \tag{5.1}$$

where $P_\theta(y|\mathbf{x}, s^-)$ is a probabilistic function determined by $h_\theta(\cdot)$. As a result, $P(\hat{Y}^{t^*}(s^+_\pi, \theta) = 1)$ is a complex nonlinear function of θ, making Problem Formulation 1 difficult to solve. In the following, we show how Problem Formulation 1 is converted to a performative risk optimization problem and then propose an optimization algorithm by leveraging repeated risk minimization.

Following the notation of convex optimization of classification, we denote by ϕ a convex surrogate function. Then, we can formulate the loss function as

$$\mathcal{L}(Y^t, \hat{Y}^t) = \mathbb{1}\left[Y^t h_\theta(\mathbf{X}^t, S) < 0\right] = \phi\left(Y^t h_\theta(\mathbf{X}^t, S)\right).$$

We can also apply the surrogate function to the fairness constraints. For any t, we have

$$P\Big(\hat{Y}^t(s^+_\pi, \theta) = 1\Big) = \sum_{\mathbf{X}^t} P\Big(\mathbf{x}^t(s^+_\pi, \theta)\Big)\, P_\theta(Y^t = 1|\mathbf{x}, s^-).$$

Similar to [84], we estimate $P_\theta(Y^t = 1|\mathbf{x}, s^-)$ by first treating it as $\mathbb{1}\left[h_\theta\left(\mathbf{x}^t, s^-\right) \geq 0\right]$ and then replacing the indicator function by $\phi(\cdot)$:

$$P\Big(\hat{Y}^t(s^+_\pi, \theta) = 1\Big) = \sum_{\mathbf{X}^t} P\Big(\mathbf{x}^t(s^+_\pi, \theta)\Big)\, \phi\left(-h_\theta\left(\mathbf{x}^t, s^-\right)\right) = \mathop{\mathbb{E}}_{\mathbf{X}^t \sim P(\mathbf{X}^t(s^+_\pi, \theta))} \left[\phi\left(-h_\theta\left(\mathbf{X}^t, s^-\right)\right)\right].$$

Similarly, we have

$$-P\Big(\hat{Y}^t(s_\pi^-,\theta)=1\Big) = P\Big(\hat{Y}^t(s_\pi^-,\theta)=0\Big) - 1 = \underset{\mathbf{X}^t\sim P(\mathbf{X}^t(s_\pi^-,\theta))}{\mathbb{E}}\left[\phi\left(h_\theta\left(\mathbf{X}^t,s^-\right)\right)\right] - 1.$$

Then, we define utility loss $l_u(\theta)$, long-term fairness loss $l_l(\theta)$, and short-term fairness loss $l_s(\theta)$ as follows.

$$l_u(\theta) = \sum_{t=1}^{t^*} \underset{S,\mathbf{X}^t,Y^t\sim P(S,\mathbf{X}^t,Y^t)}{\mathbb{E}}\left[\phi\left(Y^t h_\theta(\mathbf{X}^t,S)\right)\right],$$

$$l_l(\theta) = \frac{1}{2}\left\{\underset{\mathbf{X}^{t^*}\sim P(\mathbf{X}^{t^*}(s_\pi^+,\theta))}{\mathbb{E}}\left[\phi\left(-h_\theta\left(\mathbf{X}^{t^*},s^-\right)\right)\right] + \underset{\mathbf{X}^{t^*}\sim P(\mathbf{X}^{t^*}(s_\pi^-,\theta))}{\mathbb{E}}\left[\phi\left(h_\theta\left(\mathbf{X}^{t^*},s^-\right)\right)\right] - 1 - \tau_l\right\},$$

$$l_s(\theta) = \frac{1}{t^*}\sum_{t=1}^{t^*}\left\{\underset{\mathbf{X}^t\sim P(\mathbf{X}^t(s_{\pi^t}^-,\theta))}{\mathbb{E}}\left[\phi\left(-h_\theta\left(\mathbf{X}^t,s^+\right)\right)\right] + \underset{\mathbf{X}^t\sim P(\mathbf{X}^t(s_{\pi^t}^-,\theta))}{\mathbb{E}}\left[\phi\left(h_\theta\left(\mathbf{X}^t,s^-\right)\right)\right] - 1 - \tau_t\right\}.$$

By adding the long-term and short-term fairness losses as regularization terms into the objective function, we obtain an unconstrained optimization problem as given in Problem Formulation 2. The general formulation of the performative risk optimization can be given by $\text{argmin}_\theta \underset{\mathbf{Z}\sim\mathcal{D}(\theta)}{\mathbb{E}} l(\mathbf{Z};\theta)$ where $\mathbf{Z}$ represents the set of all attributes and outcome [83]. Thus, Problem Formulation 2 can be considered as a performative risk optimization problem as all terms in the objective function are represented as expectations of the loss function over the distributions that depend on the loss function parameters. Compare with Problem Formulation 1, Problem Formulation 2 relaxes the fairness constraints and certain amount of violations to the constraints are allowed. However, Problem Formulation 2 can be solve more efficiently by leveraging the repeated risk minimization technique as shown in the next subsection.

Problem Formulation 3. *The problem of fair sequential decision making is reformulated as the performative risk optimization:*

$$\underset{\theta}{\operatorname{argmin}}\, l(\theta) = \lambda_u l_u(\theta) + \lambda_l l_l(\theta) + \lambda_s l_s(\theta) \tag{5.2}$$

where λ_u, λ_l and λ_s are weight parameters and satisfy $\lambda_u + \lambda_l + \lambda_s = 1$.

5.4.2 The Algorithm of Repeated Risk Minimization

Repeated risk minimization (RRM) is an iterative algorithmic heuristic for solving the performative risk optimization problem. The procedure of the RRM is to start from an initial model and repeatedly find a model that minimizes the loss function on the distribution resulting from the previous model, which can symbolically represented as the update rule $\theta_{i+1} = \operatorname{argmin}_\theta \mathop{\mathbb{E}}_{\mathbf{Z}\sim\mathcal{D}(\theta_i)} l(\mathbf{Z};\theta)$ [83]. The RRM converges if the model that minimizes the loss remains unchanged from the previous model, i.e., $\theta_{i+1} = \theta_i$.

To implement the RRM algorithm in our context with three different loss terms, we sample different distributions at each iteration. For computing $l_u(\theta)$, the data distribution does not change with the deployment of new models, and we always use the original dataset $\mathcal{D}$ to compute $l_u(\theta)$. For computing $l_l(\theta)$, the data distribution follows the post-intervention distribution $P(\mathbf{X}^{t^*}(s^+_\pi, \theta))$ (resp. $P(\mathbf{X}^{t^*}(s^-_\pi, \theta))$). Thus, we sample the data according to the inference formula that is similar to Eq. (5.1) where a smooth probabilistic function $P_\theta(y|\mathbf{x}, s)$ is used. Specifically, we first sample $\mathbf{X}^1$ according to the distribution $P(\mathbf{X}^1|s^+)$ (resp. $P(\mathbf{X}^1|s^-)$), and sample the decision for each sample according to $P_\theta(Y^1|\mathbf{x}^1, s^-)$. Then, we sample $\mathbf{X}^2$ according to the distribution $P(\mathbf{X}^2|\mathbf{X}^1, Y^1)$ upon the samples obtained at the first time step. We repeat this process until time t^* to obtain samples for $\mathbf{X}^{t^*}$ for com-

Algorithm 1: Repeated Risk Minimization

Input : Dataset $\mathcal{D} = \{(S, \mathbf{X}^t, Y^t)\}_{t=1}^{l}$, time-lagged causal graph $\mathcal{G}$, convergence threshold δ

Output: The stable model h_θ

Train a classifier on $\mathcal{D}$ according to Eq. (5.2) without the soft intervention to obtain the initial parameter θ_0;
$i \leftarrow 0$;
repeat
    Sampled the post-intervention distributions $P\left(\mathbf{X}^{t^*}(s_\pi^+, \theta_i)\right)$ and $P\left(\mathbf{X}^{t^*}(s_\pi^-, \theta_i)\right)$;
    Sampled the post-intervention distributions $P\left(\mathbf{X}^{t}(s_\pi^+, \theta_i)\right)$ and $P\left(\mathbf{X}^{t}(s_\pi^-, \theta_i)\right)$ for each t;
    Minimize $l(\theta)$ according to Eq. (5.2) to obtain θ_{i+1};
    $\triangle = \|\theta_{i+1} - \theta_i\|_2$;
    $i \rightarrow i + 1$;
until $\triangle < \delta$;
$\theta \leftarrow \theta_i$;
return h_θ;

puting $l_l(\theta)$. For computing $l_s(\theta)$, we similarly sample the distributions $P(\mathbf{X}^t(s_{\pi^t}^+, \theta))$ and $P(\mathbf{X}^t(s_{\pi^t}^-, \theta))$ for each time step t. The procedure of our algorithm starts from an initial model h_{θ_0} directly trained on $\mathcal{D}$, and repeatedly train the model on the re-sample data at each iteration, until the model converges to performative stability. The pseudocode of this procedure is described in Algorithm 1.

5.4.3 Convergence Analysis of RRM

We now conduct performative stability analysis for our algorithm. The convergence of the RRM algorithm depends on the smoothness and convexity of the loss function, as well as the sensitivity of the distribution to the parameters [83]. Specifically, given a general RRM formulation $\theta_{i+1} = \operatorname{argmin}_\theta \underset{\mathbf{Z} \sim \mathcal{D}(\theta_i)}{\mathbb{E}} l(\mathbf{Z}; \theta)$, if loss function $l(\cdot)$ is β-jointly smooth and γ-strongly convex, and distribution $\mathcal{D}(\theta)$ is ε-sensitive, then the RRM converges to a stable point if $\varepsilon < \frac{\beta}{\gamma}$. We similarly analyze these factors for our problem and then give the theoretical convergence result.

Lemma 2. *If the surrogated loss function $(\phi \circ h)(\cdot)$ is γ-strongly convex, then $f(\cdot)$ is γ-strongly convex.*

Lemma 2 can be directly proven according to the sum rule of the gradient.

Next, we study the sensitivity of the distributions. Consider the distribution $P(\mathbf{X}^t(s_\pi, \theta))$ for any t. Its sensitivity to θ depends on to what extend the decisions will impact the attributes via the feedback loop. By assuming that the change of the distribution over the attributes in respond to the change of θ is bounded by a constant, we present following lemma.

Definition 14. *For any t, attributes $\mathbf{X}^{t+1}$ are c-sensitive if*

$$\| \sum_{Y^t} \nabla_\theta P_\theta(y^t | \mathbf{x}^t, s) P(\mathbf{x}^{t+1} | \mathbf{x}^t, y^t) \| \leq c \sum_{Y^t} P(\mathbf{x}^{t+1} | \mathbf{x}^t, y^t).$$

Lemma 3. *For any t, suppose that $\mathbf{X}^{t+1}$ are c-sensitive, then distribution $P(\mathbf{X}^t(s_\pi, \theta))$ is ε-sensitive with $\varepsilon \leq 2mc(t-1)$, where m is the maximum ground distance between two values of $\mathbf{X}^t$.*

Proof. Let $D_{\mathbf{x}^t}(\theta)$ denote probability $P(\mathbf{x}^t | do(s_\pi, \theta))$ and $D(\theta)$ denote the corresponding distribution. We adopt a simple greedy strategy to solve the transportation problem to obtain a upper bound of $W_1(D(\theta), D(\theta'))$. We transverse through each value of $\mathbf{X}^t$. For each $\mathbf{x}^t$, if the amount of dirt in $D_{\mathbf{x}^t}(\theta)$ is larger than that of $D_{\mathbf{x}^t}(\theta')$, then we move the additional dirt to a pool. If the amount of dirt in $D_{\mathbf{x}^t}(\theta)$ is less than that of $D_{\mathbf{x}^t}(\theta')$, then we insert this demand into a queue and move the dirt from the pool to $D_{\mathbf{x}^t}(\theta)$ as soon as there is enough dirt in the pool. As a result, the total amount of dirt moved by this strategy is

$\sum_{\mathbf{X}^t} |D_{\mathbf{x}^t}(\theta) - D_{\mathbf{x}^t}(\theta')|$. Thus, we have

$$W_1(D(\theta), D(\theta')) \leq \sum_{\mathbf{X}^t} |D_{\mathbf{x}^t}(\theta) - D_{\mathbf{x}^t}(\theta')| \cdot m, \tag{5.3}$$

where m is maximum ground distance between two values of $\mathbf{X}^t$. Then, according to the mean value theorem and Cauchy–Schwarz inequality, we have

$$|D_{\mathbf{x}^t}(\theta) - D_{\mathbf{x}^t}(\theta')| = |\nabla D_{\mathbf{x}^t}(\eta) \cdot (\theta - \theta')| \leq \|\nabla D_{\mathbf{x}^t}(\eta)\| \|\theta - \theta'\| \tag{5.4}$$

for some $\eta \in [\theta, \theta']$. By definition of $D_{\mathbf{x}^t}(\theta)$, it follows that

$$D_{\mathbf{x}^t}(\theta) := P(\mathbf{x}^t | do(s_\pi, \theta)) = \sum_{\mathbf{X}^1, Y^1, \cdots, Y^{t-1}} P(\mathbf{x}^1 | s) P_\theta(y^1 | \mathbf{x}^1, s) \cdots P(\mathbf{x}^t | x^{t-1}, y^{t-1}).$$

Thus, we have

$$\nabla D_{\mathbf{x}^t}(\theta) = \sum_{\mathbf{X}^1, Y^1, \cdots, Y^{t-1}} \big\{ P(\mathbf{x}^1 | s) \nabla P_\theta(y^1 | \mathbf{x}^1, s) \cdots P(\mathbf{x}^t | x^{t-1}, y^{t-1})$$
$$+ P(\mathbf{x}^1 | s) P_\theta(y^1 | \mathbf{x}^1, s) P(\mathbf{x}^2 | \mathbf{x}^1, y^1) \nabla P_\theta(y^2 | \mathbf{x}^2, s) \cdots + \cdots \}$$

According to the definition of c-sensitivity, we have

$$\| \sum_{Y^t} \nabla_\theta P_\theta(y^t | \mathbf{x}^t, s) P(\mathbf{x}^{t+1} | \mathbf{x}^t, y^t) \| \leq c \sum_{Y^t} P(\mathbf{x}^{t+1} | \mathbf{x}^t, y^t).$$

By the triangle inequality, it follows that

$$
\begin{aligned}
&\|\nabla D_{\mathbf{x}^t}(\theta)\| \le \sum_{\mathbf{X}^1, Y^1, \cdots, Y^{t-1}} \{P(\mathbf{x}^1|s)cP(\mathbf{x}^2|\mathbf{x}^1, y^1)\cdots P(\mathbf{x}^t|x^{t-1}, y^{t-1}) \\
&+ P(\mathbf{x}^1|s)P_\theta(y^1|\mathbf{x}^1, s)P(\mathbf{x}^2|\mathbf{x}^1, y^1)cP(\mathbf{x}^3|\mathbf{x}^2, y^2)\cdots + \cdots\} \\
&= c \sum_{\mathbf{X}^1, Y^1, \cdots, Y^{t-1}} \{P(\mathbf{x}^1, \mathbf{x}^2, \cdots, \mathbf{x}^t|do(y^1)) \qquad (5.5) \\
&+P(\mathbf{x}^1, y^1, \cdots, \mathbf{x}^t|do(y^2)) + \cdots\} \\
&= c\left\{\sum_{Y^1} P_\theta(\mathbf{x}^t|do(s, y^1)) + \cdots + \sum_{Y^{t-1}} P_\theta(\mathbf{x}^t|do(s, y^{t-1}))\right\},
\end{aligned}
$$

where the second step is based on the truncated factorization formula of computing the *do*-operation. Combining Eqs. (5.3), (5.4), and (5.5), we have

$$
\begin{aligned}
&W_1(D(\theta), D(\theta')) \le mc\sum_{\mathbf{X}}\left\{\sum_{Y^1} P_\eta(\mathbf{x}^t|do(s, y^1)) + \cdots + \sum_{Y^{t-1}} P_\eta(\mathbf{x}^t|do(s, y^{t-1}))\right\}\|\theta - \theta'\| \\
&= 2mc(t-1)\|\theta - \theta'\|.
\end{aligned}
$$

Hence, the lemma is proven. □

After introducing the above two lemmas, we now present our main theoretical result.

Theorem 2. *Suppose that surrogated loss function $(\phi \circ h)(\cdot)$ is β-jointly smooth and γ-strongly convex, and suppose that $\mathbf{X}^{t+1}$ are c-sensitive for any t, then the repeated risk minimization converges to a stable point at a linear rate, if $2mc(t^* - 1) < \frac{\beta}{\gamma}$.*

Proof. This proof basically follows the proof of Theorem 3.5 in [83].

Fix $\theta, \theta' \in \Theta$. Let

$$
f_a(\varphi) = \sum_{t=1}^{t^*} \mathop{\mathbb{E}}_{S, \mathbf{X}^t, Y^t \sim P(S, \mathbf{X}^t, Y^t)} \left[\phi\left(Y^t h_\varphi(\mathbf{X}^t, S)\right)\right],
$$

$$f_l(\varphi) = \frac{1}{2} \left\{ \mathop{\mathbb{E}}_{\mathbf{X}^{t^*} \sim P\left(\mathbf{X}^{t^*} | do((s_\pi^+, \theta)\right)} \left[\phi \left(-h_\varphi \left(\mathbf{X}^{t^*}, s^- \right) \right) \right] + \mathop{\mathbb{E}}_{\mathbf{X}^{t^*} \sim P\left(\mathbf{X}^{t^*} | do((s_\pi^-, \theta)\right)} \left[\phi \left(h_\varphi \left(\mathbf{X}^{t^*}, s^- \right) \right) \right] - 1 \right\},$$

$$f_s(\varphi) = \frac{1}{t^*} \sum_{t=1}^{t^*} \left\{ \mathop{\mathbb{E}}_{\mathbf{X}^{t} \sim P\left(\mathbf{X}^{t} | do((s_{\pi^t}^+, \theta)\right)} \left[\phi \left(-h_\varphi \left(\mathbf{X}^{t^*}, s^- \right) \right) \right] + \mathop{\mathbb{E}}_{\mathbf{X}^{t^*} \sim P\left(\mathbf{X}^{t^*} | do((s_{\pi^t}^-, \theta)\right)} \left[\phi \left(h_\varphi \left(\mathbf{X}^{t^*}, s^- \right) \right) \right] - 1 \right\},$$

and

$$f(\varphi) = \lambda_a f_a(\varphi) + \lambda_l f_l(\varphi) + \lambda_s f_s(\varphi).$$

Define $f'(\varphi)$ similarly to $f(\varphi)$ by replacing θ with θ'. Let $G(\theta) = \mathrm{argmin}_\varphi f(\varphi)$. Since $(\phi \circ h)(\cdot)$ is γ-strongly convex, $f(\cdot)$ is at least γ-strongly convex. Then, we have

$$f(G(\theta)) - f(G(\theta')) \geq (G(\theta) - G(\theta)')^\top \nabla f(G(\theta')) + \frac{\gamma}{2} \|G(\theta) - G(\theta')\|_2^2,$$
$$f(G(\theta')) - f(G(\theta)) \geq \frac{\gamma}{2} \|G(\theta) - G(\theta')\|_2^2.$$

Combining the two inequalities we have

$$-\gamma \|G(\theta) - G(\theta')\|_2^2 \geq (G(\theta) - G(\theta)')^\top \nabla f(G(\theta')). \tag{5.6}$$

On the other hand, since $(\phi \circ h)(\cdot)$ is β-jointly smooth, by applying Cauchy-Schwarz inequality we have that $(G(\theta) - G(\theta)')^\top \nabla \phi(h_{G(\theta')}(\mathbf{x}^{t^*}, s))$ is $\|G(\theta) - G(\theta')\|_2 \beta$-Lipschitz. Using the dual formulation of the optimal transport distance and Lemma 1, we have

$$(G(\theta) - G(\theta)')^\top \nabla f_l(G(\theta')) - (G(\theta) - G(\theta)')^\top \nabla f_l'(G(\theta'))$$
$$\geq -\varepsilon\beta \|G(\theta) - G(\theta')\|_2 \|\theta - \theta'\|_2,$$

$$(G(\theta) - G(\theta)')^\top \nabla f_s(G(\theta')) - (G(\theta) - G(\theta)')^\top \nabla f_s'(G(\theta'))$$
$$\geq -\varepsilon\beta \|G(\theta) - G(\theta')\|_2 \|\theta - \theta'\|_2,$$

where $\varepsilon = 2mc(t^* - 1)$. In addition, we have

$$(G(\theta) - G(\theta)')^\top \nabla f_a(G(\theta')) - (G(\theta) - G(\theta)')^\top \nabla f_a'(G(\theta')) = 0$$

Adding up above three (in)equalities, we have

$$(G(\theta) - G(\theta)')^\top \nabla f(G(\theta')) - (G(\theta) - G(\theta)')^\top \nabla f'(G(\theta'))$$
$$\geq -\varepsilon\beta \|G(\theta) - G(\theta')\|_2 \|\theta - \theta'\|_2.$$

Due to the first-order optimality conditions for convex functions, it follows that

$$(G(\theta) - G(\theta)')^\top \nabla f(G(\theta')) \geq -\varepsilon\beta \|G(\theta) - G(\theta')\|_2 \|\theta - \theta'\|_2. \tag{5.7}$$

Combining Eqs. (5.6) and (5.7), we have

$$-\gamma \|G(\theta) - G(\theta')\|_2^2 \geq -\varepsilon\beta \|G(\theta) - G(\theta')\|_2 \|\theta - \theta'\|_2.$$

By rearranging, we have

$$\|G(\theta) - G(\theta')\|_2 \leq \varepsilon \frac{\beta}{\gamma} \|\theta - \theta'\|_2.$$

Let θ_{PS} be a stable point, i.e., $G(\theta_{\text{PS}}) = \theta_{\text{PS}}$. In addition, by definition we have $\theta_i = G(\theta_{i-1})$. Thus, it follows that

$$\|\theta_i - \theta_{\text{PS}}\| \leq \varepsilon\frac{\beta}{\gamma}\|\theta_{i-1} - \theta_{\text{PS}}\|_2 \leq \left(\varepsilon\frac{\beta}{\gamma}\right)^i \|\theta_0 - \theta_{\text{PS}}\|_2.$$

Therefore, if $\varepsilon = 2mc(t^* - 1) < \frac{\beta}{\gamma}$, the RRM converge to θ_{PS} at a linear rate.

Hence, the theorem is proven.

$\square$

In practice, this theoretical criterion of convergence may be difficult to meet. However, our experimental results show that our algorithm can converge under reasonable conditions.

5.5 Experiments

We conduct experiments on both synthetic and semi-synthetic temporal datasets to evaluate the proposed algorithm. We show that our algorithm is effective in achieving both long-term and short-term fairness, while previous fair algorithms that do not consider the dynamics in sequential decision making actually do not mitigate or even exacerbate the short-term or long-term fairness. We consider three baselines in the experiments which treat the whole temporal dataset as a static dataset and train the decision model on it. Fairness constraints are added following the technique proposed in [38].

- **Logistic Regression (LR)**: An unconstrained logistic regression model which takes user features and labels from all time steps as inputs and outputs.

- **Fair Model with Demographic Parity (FMDP)**: On the basis of the logistic regression model, fairness constraint is added to achieve demographic parity.

- **Fair Model with Equal Opportunity (FMEO)**: On the basis of the logistic regression model, fairness constraint is added to achieve equal opportunity.

5.5.1 Datasets

Synthetic Data. We simulate a process of bank loans following the time-lagged causal graph depicted in Figure 5.1, where S is the protected attribute like race, $\mathbf{X}^t$ represents the financial status of applicants, and Y^t represents the decisions about whether or not to grant loans. At $t = 1$, we generate samples where both values of S are sampled with the equal probability, and the values of $\mathbf{X}^1$ are sampled using two different Gaussian distributions according to the value of S. Then at each time t, we sample predicted decisions $\hat{Y}^t$ and the values of $\mathbf{X}^{t+1}$ as follows. Consider a ground-truth decision model $h_{\theta^*}(\cdot)$ for deciding the probability of whether an individual would default on a loan, given by $\sigma(h_{\theta^*}(\cdot))$ where $\sigma(\cdot)$ is the sigmoid function. Then, we sample the predicted decision $\hat{Y}^t$ (as well as the actual repayment Y^t which is sampled separately) from $\sigma(h_{\theta^*}(\cdot))$ as:

$$P(\hat{Y}^t) = \sigma(h_{\theta^*}(\mathbf{X}^t, S)),\ \hat{Y}^t \sim 2 \cdot \text{Bernoulli}(P(\hat{Y}^t)) - 1.$$

Then, $\mathbf{X}^{t+1}$ is generated according to the update rule below:

$$\mathbf{X}^{t+1} = \begin{cases} \mathbf{X}^t - \epsilon \cdot \theta^t + b & \hat{Y}^t = 1, Y^t = -1 \\ \mathbf{X}^t + \epsilon \cdot \theta^t + b & \hat{Y}^t = 1, Y^t = 1 \\ \mathbf{X}^t + b & \hat{Y}^t = -1 \end{cases} \tag{5.8}$$

where ϵ is a parameter that controls the sensitivity of the update to the predicted decisions, and $b = S \cdot b_1 + (1 - S) \cdot b_0$ is a small base increment at each time step. In the simulation process, we generate a 5-step synthetic dataset with 5000 samples where parameters are set as $\epsilon = 0.5$, $b_0 = 0.2$, $b_1 = 1.0$.

Semi-synthetic Data. We use the Taiwan credit card dataset [91] as the initial data at $t = 1$. To form a balanced dataset, we extract 3000 samples and choose two features PAY_AMT1 and PAY_AMT2 that are appropriate in fitting into our update rule. Then, we generate a 4-step dataset using the same update rule as shown above.

5.5.2 Training and Evaluation

We conduct the training process following the RRM algorithm. At each iteration, we sample the data according to the current decision model and the causal graph. Similar to the data generation process, predicted decisions are sampled according to the probability given by $\sigma(h_\theta(\cdot))$, and the feature values are sampled according to Eq. (5.8). In our experiments, we assume that the true update rule is known in order to remove errors introduced by causal graph construction. In practice, the causal graph learned from data may introduce additional errors.

We then design an evaluation process which simulates the real model deployment procedure and feedback loops. At each time step t, we use the trained decision model $h_\theta(\cdot)$ to make decisions $\hat{Y}^t$, and use the ground-truth model $h_{\theta^*}(\cdot)$ to determine the repayment Y^t. The accuracy is measured by comparing $\hat{Y}^t$ and Y^t, the long-term fairness is measured based on the distribution of $\hat{Y}^{t^*}$ in the evaluation, and the short-term fairness is measured based on the distribution of $\hat{Y}^t$ at different time steps according to proposed definitions.

Alg.	Metric	Time steps				
		$t=1$	$t=2$	$t=3$	$t=4$	$t=5$
RL	Acc	0.912	0.894	0.917	0.921	0.917
	Short	0.152	0.160	0.166	0.164	0.174
	Long	0.058	0.117	0.173	0.246	0.340
FMDP	Acc	0.735	0.706	0.704	0.708	0.725
	Short	0.212	0.216	0.224	0.220	0.232
	Long	0.180	0.306	0.376	0.431	0.481
FMEO	Acc	0.829	0.790	0.795	0.800	0.814
	Short	0.010	0.010	0.010	0.014	0.020
	Long	0.080	0.122	0.190	0.276	0.352
Ours	Acc	0.801	0.754	0.729	0.707	0.692
	Short	0.012	0.008	0.012	0.008	0.002
	Long	0.040	0.024	0.020	0.012	0.002

Table 5.1: Accuracy, short-term and long-term fairness of different algorithms on the synthetic dataset.

Alg.	Metric	Time steps			
		$t=1$	$t=2$	$t=3$	$t=4$
RL	Acc	0.828	0.826	0.841	0.816
	Short	0.015	0.018	0.021	0.012
	Long	0.038	0.088	0.243	0.433
FMDP	Acc	0.830	0.843	0.846	0.841
	Short	0.063	0.066	0.075	0.069
	Long	0.038	0.076	0.223	0.397
FMEO	Acc	0.824	0.830	0.830	0.813
	Short	0.072	0.075	0.087	0.078
	Long	0.006	0.045	0.156	0.295
Ours	Acc	0.648	0.648	0.680	0.687
	Short	0.006	0.006	0.003	0.006
	Long	0.064	0.043	0.016	0.003

Table 5.2: Accuracy, short-term and long-term fairness of different algorithms on the semi-synthetic dataset.

5.5.3 Implementation Details

For baselines FMDP and FMEO, they are formulated as constrained optimization forms which are directly solved by the CVXPY package [75]. For our algorithm, we use the logistic loss function for the surrogate function ϕ and the linear model for the decision model. All algorithms use the l_2-regularization which can equip the logistic loss function with strong convexity. In our algorithm, ReLU activation function is adopted to ensure that the fairness constraints are always non-negative, and we adopt PyTorch [76] to implement optimization with Adam optimizer.

5.5.4 Results

The results of the accuracy and fairness of the baselines and our algorithm on the synthetic dataset are shown in Table 5.1. As can be seen, our algorithm achieves the short-term fairness at all time steps. More importantly, the long-term fairness is improved with time and approaches zero at $t = 5$. For other baselines, there is a clear trend that the long-term fairness continuously accumulates with time. This demonstrates that static fairness notions may harm fairness in the long run. The short-term fairness remains stable with time as it shows the bias in the model that is related to the protected attribute. The experiments on the semi-synthetic dataset produce similar results as shown in Table 5.2. We also observe a trade-off between accuracy and fairness meaning that some accuracy needs to be sacrificed in order to achieve fairness.

We also plot in Figure 5.2 the convergence results of our algorithms for different ϵ values. As mentioned earlier, the value of ϵ controls the sensitivity of $\mathbf{X}^{t+1}$ to the update of θ. Figure 5.2 shows that our algorithm converges when the value of ϵ is reasonably small, which is consistent with the results in [83]. We observe similar results on the semi-synthetic

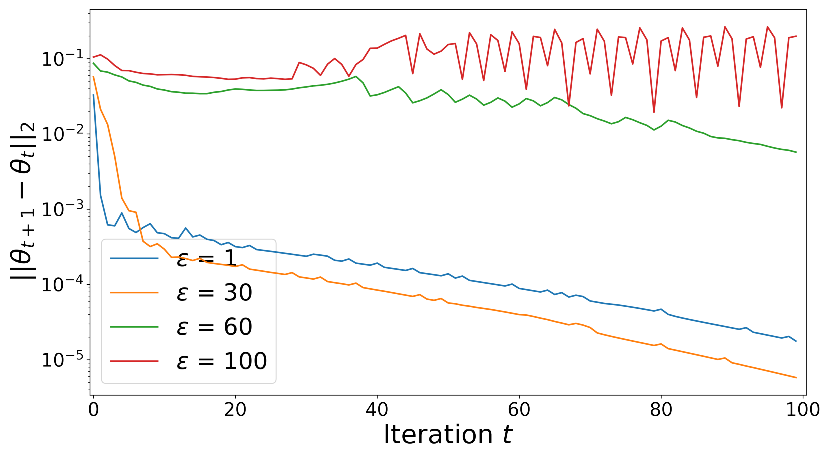


Figure 5.2: The convergence results for different values of ϵ on the synthetic dataset.

dataset.

5.6 Summary

We proposed a framework to achieve long-term fairness in sequential decision making. The decision-making process was modeled by a time-lagged causal graph, in which the hard intervention was performed on the protected attribute and soft interventions were performed on the decisions. We measured both long-term and short-term fairness as path-specific effects. The problem of fair sequential decision making was formulated as a performative risk optimization problem, and repeated risk minimization is adopted to train the model on the datasets sampled from post-intervention distributions. The convergence of the proposed algorithm was analyzed theoretically. Finally, we verified the effectiveness of the proposed framework and algorithm by comparing it with the baselines on two synthetic datasets.

6 Long-term Fair Decision Making Through Deep Generative Models

6.1 Introduction

In the last chapter, we propose to use causal time series graphs to model the system dynamics. The machine learning model deployment is modeled as soft interventions on the graph and the influence of feedback is inferred as the interventional distribution. Long-term fairness is formulated as path-specific effects of the sensitive attribute on the decision at time step T and is achieved by using continuous optimization.

Although the last chapter shows that our approach can reduce the discrimination and bias up to a certain time step, a critical limitation is that to achieve fairness at time step T it requires a time series training dataset whose time length l is greater than T. However, a practical requirement in long-term fair machine learning is to protect disadvantaged groups from pernicious long-term effects in the future that is beyond the data we have, as shown in Figure 6.1. This requires one not only to capture the dynamics in history but also to predict the potential long-term impacts in the future based on the historical data and a fair decision model should minimize such potential long-term impacts. In addition, in order to quantify causal fairness [58], we need to predict not only the observational distribution but also the interventional distribution beyond the training data.

In this chapter, we address the above limitations by developing a deep generative model that can predictively generate data following both observational and interventional distributions, and integrating the prediction and training into a collaborative training framework so that the predicted data could be used as reliable data for training the decision model. To this end, we propose a three-phase approach. In Phase 1, given a training time series within

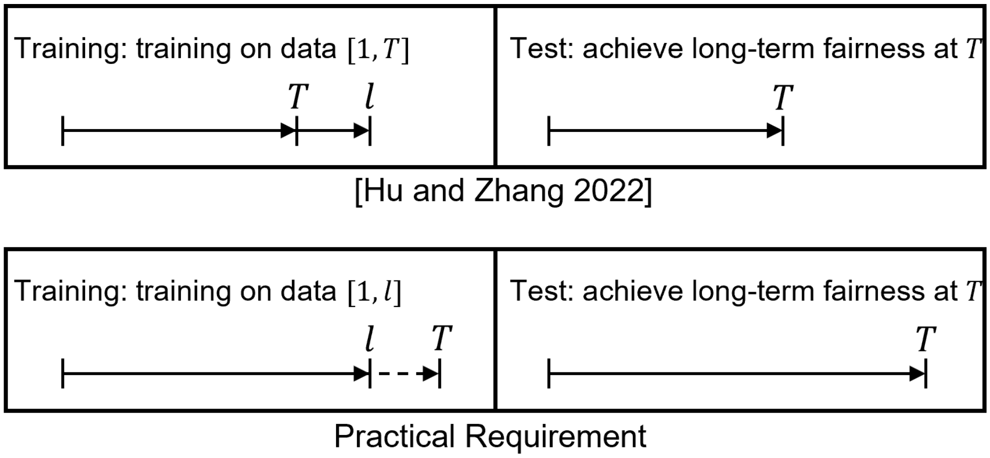


Figure 6.1: Diagram of training and test in long-term fair machine learning where $T \leq l$ (above) and $T > l$ (below).

the time range $[1, l]$, we first train a baseline model for predicting decisions in a static setting on the training data. In Phase 2, we train a recurrent conditional generative adversarial network (RCGAN) which is motivated by [92] for fitting the training time series so that it can generate high-fidelity time series. Finally, in Phase 3, we train a fair decision model on the generated time series within the time range $[1, T]$ ($T > l$) by considering both local and long-term fair constraints. The optimization problem is formulated as a performative risk minimization and solved by using the repeated gradient descent algorithm.

To define long-term fairness, different from the last chapter where the long-term fairness is defined as path-specific risk difference of the decision at time step T, we consider the interventional distribution of features at time step T and measure the 1-Wasserstein distance between the interventional distributions under two different interventions on the sensitive feature. We argue that the long-term fairness metric we proposed is more general than that in the last chapter, because according to the Kantorovich-Rubinstein duality our metric provides an upper bound to the path-specific risk difference of the decision at time step T for a set of reasonable decision models other than a single learned model.

We conduct experiments on both synthetic and semi-synthetic datasets. The results show that given a historical time series, our framework can use the data to train a decision model such that once deployed it can achieve fairness at a certain time step in the future, whereas unfairness may be accumulated if traditional fairness notions are used.

6.2 Background Revisit

We utilize Pearl's structural causal model (SCM) and causal graph [29] for defining the long-term fairness metric and designing the architecture of the deep generative model. For a gentle introduction to SCM please refer to [93]. In this paper, we assume the *Markovian SCM* such that the exogenous variables are mutually independent.

Causal inference in the SCM is facilitated with the interventions [29]. The hard intervention forces some variables to take certain constants. The soft intervention, on the other hand, forces some variables to take certain functional relationships in responding to some other variables [54]. Symbolically, the soft intervention that substitutes equation $X = f_X(\cdot)$ with a new equation $X = g(\cdot)$ is denoted as $\sigma_{X=g(\cdot)}$. The distribution of another variable Y after performing the soft intervention is denoted as $P(Y(\sigma_{X=g(\cdot)}))$.

6.3 Long-term Fairness Metric

6.3.1 Problem Setting

We start by formulating a long-term fairness metric that captures the system dynamics and permits continuous optimization. To ease the representation, we assume a binary sensitive feature for indicating the advantaged and disadvantaged groups denoted by $S \in \mathcal{S} = \{s^+, s^-\}$, as well as a binary decision denoted by $Y \in \mathcal{Y} = \{y^+, y^-\}$. The profile features other than the sensitive feature are denoted by $\mathbf{X} \in \mathcal{X}$. In a sequential decision-making system, if a feature is time-dependent, it means that its value may change from one time step to another. We assume that $\mathbf{X}$ and Y are time-dependent and use the superscript to denote their variants at different time steps, leading to $\mathbf{X}^t$ and Y^t. For S, naturally many sensitive attributes are time-independent, like gender and race. Some sensitive attributes may change over time, but the relative order of individuals in the data does not change, like age. Thus, we treat S as being time-independent in this chapter.

Suppose that we have access to a time series $\mathcal{D} = \{(S, \mathbf{X}^t, Y^t)\}_{t=1}^{l}$. We assume an SCM for describing the data generation mechanism and leverage a causal time series graph for describing the causal relation among $S, \mathbf{X}^t, Y^t$ in the SCM. We make the *stationarity assumption* such that data distribution may shift over time but the data generation mechanism behind it does not change. Figure 6.2 gives an example which shows that at each time step the decision Y^t is made based on the value of $\mathbf{X}^t$ and S. Meanwhile, the value of $\mathbf{X}^t$ is affected by the values of $\mathbf{X}^{t-1}$, Y^{t-1} and S. We will use this graph as a running example throughout the remaining of this chapter. In practice, the causal time series graph can be obtained from the domain knowledge or learned from data using structure learning algorithms (e.g., [94, 87, 88]). Our goal is to learn a decision model $h_\theta : \mathcal{S} \times \mathcal{X} \mapsto \mathcal{Y}$ such that when deployed at every time step, fairness can be achieved at a certain time step T where $T > l$.

To illustrate our problem setting in a real-world scenario, consider an example of a bank loan system. When people apply for bank loans, their personal information (e.g., race, job, assets, credit score, etc.) is used by the bank's decision model to decide whether to grant the loans. Except for race, which is a sensitive feature S, other profile features $\mathbf{X}^t$ represent an applicant's qualification at time step t. The bank's decision Y^t can have impacts on the

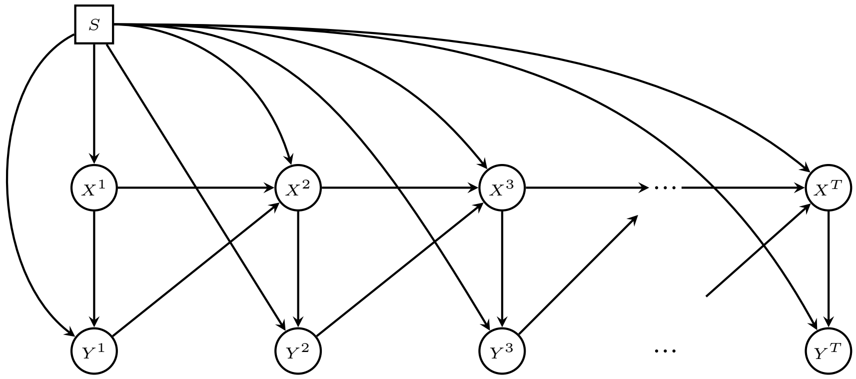


Figure 6.2: A causal time series graph for sequential decision making.

applicants' profile features in $\mathbf{X}^{t+1}$ such as the credit score, which in turn affect the outcomes of their subsequent loans.

6.3.2 Formulate Long-term Fairness

To formulate long-term fairness, the model deployment can be mathematically simulated by soft interventions on Y at all time steps until T [30]. That is, given a decision model h_θ, we use it to replace the original structural equation associated with Y in the SCM. We denote the soft intervention by $\sigma_{Y^t=h_\theta(S,\mathbf{X}^t)}$ and abbreviate it as σ_θ. Then, the influence of the model deployment on feature $\mathbf{X}^t$ can be described by its interventional distribution, denoted by $P(\mathbf{X}^t(\sigma_\theta))$. To define long-term fairness, we focus on the interventional distributions of $\mathbf{X}^T$ conditioning on the advantaged and disadvantaged groups. The rationale behind this is to treat the profile features $\mathbf{X}$ as the representation of the qualification or the reputation of an individual [24, 22, 23]. As a result, long-term fairness is achieved when the disparity in the qualification between the advantaged and disadvantaged groups is eliminated after the model deployment. For example, we say that the loan system achieves long-term fairness if it could eliminate the disparity in the credit score between different race groups at a certain time step T.

We use the 1-Wasserstein distance to measure the difference between the two distributions. The reason is presented in the following proposition.

Proposition 1. *Denote by d the 1-Wasserstein distance between the feature distributions of different groups, i.e., $d = W(P(x|c^+), P(x|c^-))$. For any decision model $h : \mathcal{X} \mapsto \mathcal{A}$ that is Lipschitz continuous, its DP is bounded by $l_h \cdot d$ where l_h is the Lipschitz constant of h. If we assume that the true label is given by another decision model $g : \mathcal{X} \mapsto \mathcal{A}$ that is Lipschitz continuous and satisfies the equal base rate condition, then the EO of h is bounded by $(l_h + l_g)/P(y) \cdot w$ where l_g is the Lipschitz constant of g.*

Proof. According to the definition of DP, we have

$$\mathrm{DP}(h) = |\mathbb{E}[h(x)|c^+] - \mathbb{E}(h(x)|c^-)|.$$

Due to the Kantorovich–Rubinstein duality [95], it is straightforward that

$$\begin{aligned}\mathrm{DP}(h) &\leq \sup_{\|h\| \leq l_h} \left[\mathbb{E}_{x \sim P(x|c^+)}[h(x)] - \mathbb{E}_{x \sim P(x|c^-)}[h(x)]\right] \\ &= l_h \cdot W(P(x|c^+), P(x|c^-)) = l_h \cdot d.\end{aligned}$$

On the other hand, we have

$$\mathrm{EO}(h) = |\mathbb{E}[h(x)|a = 1, c^+] - \mathbb{E}(h(x)|a = 1, c^-)|.$$

Due to the assumption that the true label is given by g and g satisfies the equal base rate condition, it follows that

$$\mathbb{E}[h(x)|a,c] = \int_x h(x)P(x|a,c)dx = \int_x h(x)P(x|c)\frac{P(y|x,c)}{P(y|c)}dx$$
$$= \int_x h(x)P(x|c)\frac{g(x)}{P(y)}dx = \frac{1}{P(y)}\mathbb{E}_{x\sim P(x|c)}[h(x)g(x)].$$

In addition, define $f(x) = h(x)g(x)$ and denote the Lipschitz constant of f as l_f. It is easy to show that $l_f \leq l_h \cdot \sup_x |h(x)| + l_g \cdot \sup_x |g(x)|$. Since $h(x) \leq 1$ and $g(x) \leq 1$, we have $l_f \leq l_h + l_g$. As a result, we have

$$EO(h) \leq \frac{l_h + l_g}{P(y)} W(P(x|c^+), P(x|c^-)) = \frac{l_h + l_g}{P(y)} \cdot d.$$

$\square$

Thus, based on the proposition 1, we obtain the long-term fairness metric defined as follows.

Definition 15. *Given a sequential decision making system, a decision model $h_\theta : \mathcal{S}\times\mathcal{X} \mapsto \mathcal{Y}$, and a time step T, the metric for measuring the long-term fairness produced by deploying h_θ is given by*

$$J_1^T(\theta) \triangleq W(P(\mathbf{X}^T(\sigma_\theta)|S = s^+), P(\mathbf{X}^T(\sigma_\theta)|S = s^-)), \tag{6.1}$$

where W is the 1-Wasserstein distance and σ_θ is the soft intervention.

Definition 16. *Long-term fairness is achieved by a decision model h_θ at time T if $J_1^T(\theta) = 0$.*

Note that although our long-term fairness metric bounds the fairness of the decision model that does not take any sensitive feature as the input, we allow the decision model h_θ that is to be deployed to take the sensitive feature in order to reduce the disparity between the distributions of the two groups. Meanwhile, as will be discussed in the next section, we also consider short-term or local fairness constraints as they may be enforced by law or regulations [96].

6.4 Deep Generative Framework for Achieving Long-term Fairness

In this section, we formally formulate the problem of building the decision model to achieve long-term fairness. We then describe the overview of the proposed three-phase deep generative framework, followed by the details of each phase.

6.4.1 Problem Formulation

In the problem formulation, first, the decision model should make accurate predictions for good utility performance. Typically, loss functions such as the cross-entropy loss are used to penalize inaccurate predictions. In this chapter, we adopt the traditional definitions of the loss function. The loss J_2 defined over the training time series is given as follows.

Definition 17. *Given a time series $\mathcal{D} = \{(S, \mathbf{X}^t, Y^t)\}_{t=1}^{l}$, the loss for the decision model h_θ is given by*

$$J_2(\theta) \triangleq \frac{1}{l} \sum_{t=1}^{l} \mathbb{E}[\mathcal{L}(h_\theta(S, \mathbf{X}^t), Y^t)], \tag{6.2}$$

where $\mathcal{L}$ is any loss function.

Second, as mentioned above, local fairness constraint needs to be required at each time step to restrict the local bias. For the local fairness constraint, we consider the direct discrimination [17] of the decision model h_θ which is enforced on each time step from 1 to T as given below.

Definition 18. *The local fairness constraint for each time step $t \in [1, T]$ is given by*

$$J_3^t(\theta) \triangleq |\mathbb{E}[h_\theta(S = s^+, \mathbf{X}^t(\sigma_\theta))|S = s^-] - \mathbb{E}[h_\theta(S = s^-, \mathbf{X}^t(\sigma_\theta))|S = s^-]|. \tag{6.3}$$

So far, we have considered three factors in the optimization and there are trade-offs between each pair of factors. First, there is a trade-off between the local fairness and the accuracy of the model. This is because if in the training data the decision was made with biases against a certain group, then, a decision model built upon the training data for maximizing the accuracy only will inherit the historical biases by learning from the training data and violate the local fairness constraint. The trade-off between the local fairness and accuracy has been studied in many previous works [97, 98]. Similarly, achieving long-term fairness may be at the cost of accuracy if the training data contain historical biases. In addition, there may exist trade-offs between the long-term and local fairness. For example, making loan decisions exactly according to the credit score is reasonable in terms of local fairness, but might not help in narrowing the gap in credit scores between advantaged and disadvantaged groups. On the other hand, reducing the gap by favoring the disadvantaged group can raise the potential issue of reverse discrimination, which may violate the local fairness constraint.

According to the above analysis, we formulate the objective of the problem by summarizing all three factors, leading to the problem formulation below.

Problem Formulation 4. *The problem of long-term fair decision-making is to solve the optimization problem:*

$$\min_\theta \mathcal{L}(\theta) = \lambda_1 J_1^T(\theta) + \lambda_2 J_2(\theta) + \frac{\lambda_3}{T} \sum_{t=1}^{T} J_3^t(\theta), \tag{6.4}$$

where λ_1, λ_2 and λ_3 are weight parameters.

6.4.2 Overview of the Proposed Framework

There are two main challenges in solving Problem Formulation 1. First, as shown in Eqs. (6.1) and (6.3), the computation of $J_1^T(\theta)$ and $J_3^t(\theta)$ are based on the interventional variants of features $\mathbf{X}^t(\sigma_\theta)$ whose values in turn depend on the model parameter θ. Second, $J_1^T(\theta)$ and $\sum_{t=1}^{T} J_3^t(\theta)$ are computed on time steps that are beyond the range of the training data since $T > l$. Thus, Problem Formulation 1 cannot be solved by the traditional machine learning framework.

We propose a novel three-step framework based on causal inference techniques and deep generative networks. The main idea is to use a deep generative network to simulate an SCM for generating both observational and interventional distributions. It has been proven in [99] that if the structure of a generative network is arranged to reflect the causal structure, then it can be trained with the observational data such that it will agree with the same SCM in terms of any identifiable interventional distributions. In addition, it has been shown that the interventional distribution produced by any soft intervention is identifiable in a Markovian SCM [69]. These previous results have established the theoretical ground for our method.

We illustrate our framework using the example shown in Figure 6.2. In this example, the causal structure at each time step can be mathematically described by two structural equations of the SCM:

$$Y^t = f_Y\left(S, \mathbf{X}^t, U_Y\right), \tag{6.5}$$

$$\mathbf{X}^t = f_{\mathbf{X}}\left(S, \mathbf{X}^{t-1}, Y^{t-1}, U_{\mathbf{X}}\right), \tag{6.6}$$

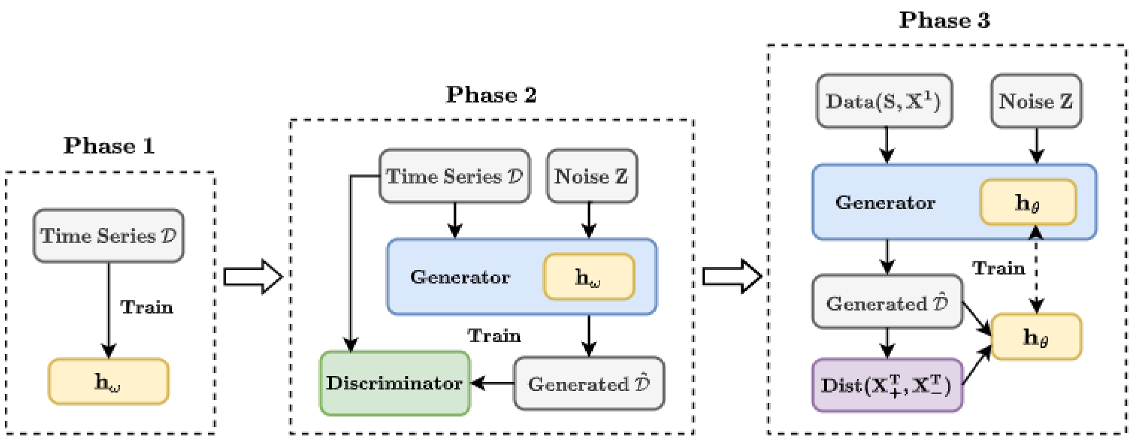


Figure 6.3: The overview of the proposed framework. Solid arrows represent input, and the dashed arrow represents parameter sharing. For Phase 3 only one generator is shown.

where $U_Y, U_{\mathbf{X}}$ are exogenous variables. Note that due to the stationarity assumption, f_Y and $f_{\mathbf{X}}$ are time-independent. According to the principle of independent mechanisms [100], these two structural equations can be learned independently without affecting each other. Motivated by this principle, in Phase 1, we first train a classifier h_ω on the training time series $\mathcal{D}$ to approximate f_Y and make decisions for each time step. Then, in Phase 2, we train a recurrent conditional generative adversarial network (RCGAN) [92] on the same training time series $\mathcal{D}$ using adversarial training. The generator of the RCGAN uses h_ω obtained in Phase 1 for generating decisions. Finally, in Phase 3, we replace h_ω with the decision model h_θ and train it on the data generated from the RCGAN using the objective function Eq. (6.4). The overview of the framework is shown in Figure 6.3 and the pseudo-code is shown in Algorithm 2. Next, we describe the details of each phase.

6.4.3 Phase 1: Train a Decision Classifier

The objective of this phase is to learn a classifier h_ω from the training time series to approximate the mechanism f_Y in Eq. (6.5) for making decisions, i.e., $\hat{Y}^t = h_\omega(S, \mathbf{X}^t)$. In this phase, we train the classifier by maximizing the accuracy only and do not consider any

Algorithm 2: DeepLF

Input : Dataset $\mathcal{D} = \{(S, \mathbf{X}^t, Y^t)\}_{t=1}^{l}$, time-lagged causal graph $\mathcal{G}$, parameters λ_1, λ_2 and λ_3

Output: The fair model h_θ

```
Train a classifier h_ω by minimizing Eq. (6.7) on D;
repeat
    Update the discriminator D_φ according to Eq. (6.12);
    Update the generator G_{ψ,ω} with the classifier h_ω as one of its components
      according to Eq. (6.13);
until convergence;
i ← 0;
Initialize h_{θ_0} according to h_ω;
repeat
    Generate time series using generator G_{ψ,ω};
    Compute ∇_θ L_l(θ) according to Eq. (6.4) using the generated data;
    θ_{i+1} ← θ_i − η_i ∇_θ L_l(θ_i);
    i ← i + 1;
until convergence;
return h_{θ_i};
```

fairness requirement. Since f_Y does not change with time, we aggregate the losses at all time steps in the loss function. Specifically, we use the cross-entropy loss for classification, and the loss function for training h_ω on time series $\mathcal{D}$ is given by

$$\min_{\omega} \mathcal{L}_c(\omega) = -\frac{1}{l} \sum_{t=1}^{l} \mathbb{E}\left[Y^t \log h_\omega(S, X^t)\right]. \tag{6.7}$$

6.4.4 Phase 2: Train an RCGAN

We then train an RCGAN to simulate the distribution shift in features $\mathbf{X}$, which generates the data together with the classifier h_ω obtained in Phase 1. We design the architecture of the RCGAN following the structural equation Eq. (6.6). As a result, according to [99], when the generated data fit the distribution of the real data, the model can also generate interventional data that agree with the same SCM. This is critical for training the

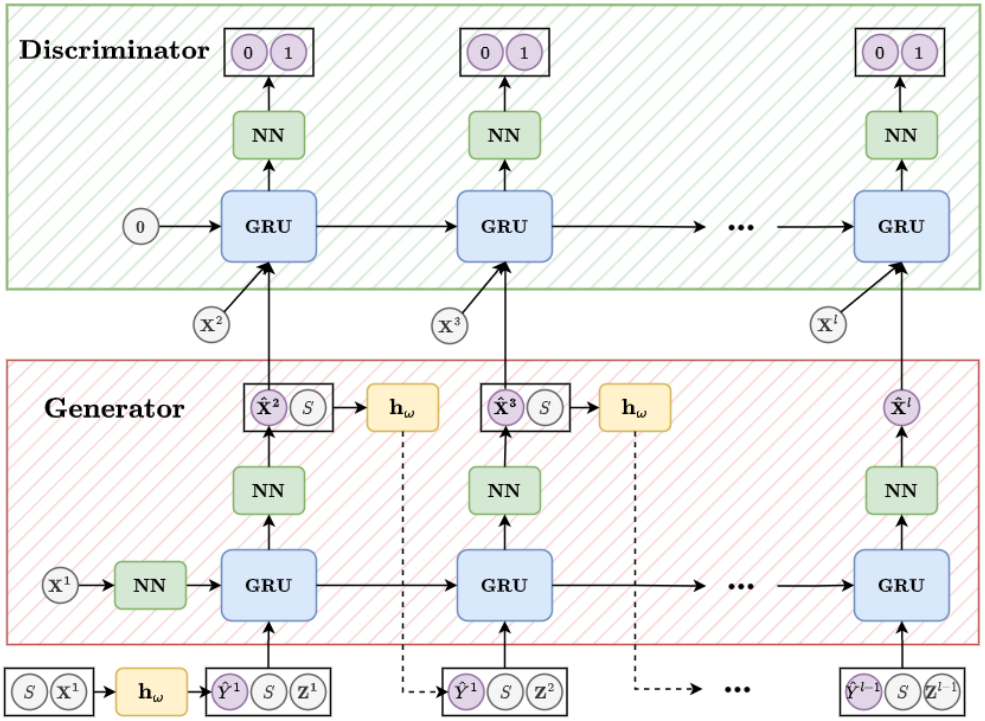


Figure 6.4: The architecture of the RCGAN.

long-term fair decision model in Phase 3 as long-term fairness is defined on the interventional distribution under the soft intervention.

The architecture of the RCGAN is illustrated in Figure 6.4, which consists of one generator and one discriminator. For efficiency, we use the gated recurrent unit (GRU) [101] as the core structure of the generator and discriminator. For the generator, it takes the sensitive feature S, a set of noise vectors $\mathbf{Z}$, as well as the features at the first time step $\mathbf{X}^1$ as the input. The hidden state is then initialized by a non-linear transformation (e.g., a multi-layer perceptron) of $\mathbf{X}^1$:

$$\mathbf{h}^1 = \text{MLP}\left(\mathbf{X}^1\right). \tag{6.8}$$

Then, for each time step t, we concatenate the noise vector $\mathbf{Z}^{t-1}$ with the conditional information $\hat{Y}^{t-1}$ and S as the input to each GRU. After the calculation of GRU, we again use a non-linear transformation to convert the hidden states $\mathbf{h}^t$ to predicted $\hat{\mathbf{X}}^{t+1}$. These three steps are shown by the three equations below.

$$\mathbf{I}^{t-1} \leftarrow \left[\hat{Y}^{t-1}, S, \mathbf{Z}^{t-1}\right], \tag{6.9}$$

$$\mathbf{h}^{t} = \mathrm{GRU}\left(\mathbf{I}^{t-1}, \mathbf{h}^{t-1}\right), \tag{6.10}$$

$$\hat{\mathbf{X}}^{t+1} = \mathrm{MLP}\left(\mathbf{h}^{t}\right). \tag{6.11}$$

We also use GRUs for the discriminator to distinguish between generated data and real data. For the generated time series, the discriminator attempts to predict label 0 for each time step, and vice versa, for the real time series, the discriminator attempts to predict label 1 for each time step.

Finally, for training the generator and discriminator, in addition to the objective of the original GAN for minimizing the likelihood of generated data given by the discriminator, the maximum mean discrepancy (MMD) [102] between original data and generated data is also explicitly minimized. The MMD brings two distributions together by comparing their statistics. As a result, the loss functions of the discriminator and the generator are shown below, which are optimized alternatively.

$$\max_{\phi} \mathcal{L}_d(\phi) = \mathbb{E}_{\mathbf{X}}[\log(D_\phi(\mathbf{X}))] + \mathbb{E}_{\mathbf{Z}}[\log(1 - D_\phi(G_{\psi,\omega}(S, \mathbf{Z}, \mathbf{X}^1)))], \tag{6.12}$$

$$\min_{\psi} \mathcal{L}_g(\psi) = \mathbb{E}_{\mathbf{Z}}[\log(1 - D_\phi(G_{\psi,\omega}(S, \mathbf{Z}, \mathbf{X}^1)))] + \gamma \mathrm{MMD}(\mathbf{X}, G_{\psi,\omega}(S, \mathbf{Z}, \mathbf{X}^1)), \tag{6.13}$$

where D_ϕ represents the discriminator, $G_{\psi,\omega}$ represents the generator that uses h_ω as the classifier, and γ controls the strength of the regularization.

6.4.5 Phase 3: Train the Long-term Fair Decision Model

At last, we train a decision model h_θ on the data generated by the RCGAN using the objective function Eq. (6.4). We use the generator obtained in Phase 2 as well as a variant of this generator where the former is for generating the observational distribution and the latter is for performing the soft intervention and generating interventional distribution. Specifically, we first directly apply the generator $G_{\psi,\omega}$ obtained in Phase 2 to generate data for time steps from 1 to l which are used to compute $J_2(\theta)$ in Eq. (6.2). Then, we perform soft intervention σ_θ by replacing h_ω with h_θ to obtain a variant generator $G_{\psi,\theta}$ which is used to generate data for time steps from 1 to T for computing $J_1^T(\theta)$ in Eq. (6.1) and $J_3^t(\theta)$ in Eq. (6.3). In other words, we use $G_{\psi,\theta}$ to generate the interventional data $\mathbf{X}^t(\sigma_\theta)$. Finally, the loss $\mathcal{L}(\theta)$ is computed and h_θ is updated accordingly. Note that the RCGAN trained in Phase 2 will not be updated in this phase.

It is important to note that when we use the RCGAN to generate data samples for computing $\mathcal{L}(\theta)$, those data samples are affected by h_θ as well, due to the fact that h_θ is trained on the interventional distribution after performing soft intervention σ_θ. Such optimization problem is called the performative risk minimization [83] and cannot be solved using traditional empirical risk minimization. In our work, we adopt the repeated gradient descent algorithm (RGD) [83] which is an iterative training approach to address this problem. In the training process of Phase 3, we first initialize h_θ according to h_ω. Then, in each iteration, we use the current version of h_θ for generating data and computing the empirical loss, and h_θ is updated based on the empirical gradient $\nabla_\theta \mathcal{L}(\theta)$. After that, we replace h_θ with its updated version and conduct another iteration of training. This process is repeated until the parameters of h_θ converge.

6.5 Experiments

In this section, we conduct empirical evaluations of our method. We refer to our method as deep long-term fair decision making (**DeepLF**).

6.5.1 Baselines

A multi-layer perceptron (**MLP**) that is trained on the training time series $\mathcal{D}$ without any fairness constraints is used as the first baseline. Two common static fairness constraints, i.e., demographic parity and equal opportunity, are applied to the **MLP** model as fairness constraints respectively, referred to as **MLP-DP** and **MLP-EO**. We also implement the most relevant method proposed in [30] where a logistic regression model is trained with long-term and short-term fairness constraints using the repeated risk minimization [83], referred to as **LRLF**.

6.5.2 Datasets

Many commonly used datasets in fair machine learning [103] are not for dynamic fairness research. In [104], the authors construct a dataset that spans multiple years and allows researchers to study temporal shifts in the distribution level. However, our study requires the longitudinal data that track each instance over time, other than multiple datasets with temporal distribution shifts. Thus, following [30], we generate synthetic and semi-synthetic time series datasets as follows.

6.5.2.1 Synthetic Dataset.

We generate the synthetic time series dataset based on the causal time series graph shown in Figure 6.2. Each sample at each time step in the time series includes a sensitive feature S, profile features $\mathbf{X}^t$ and a decision Y^t. The samples at the initial time step $\mathbf{X}^1, Y^1$ are generated by calling the data generation function (i.e., make_classification) of scikit-learn package. Then, we cluster the generated samples into two groups and assign S to each sample according to the cluster it belongs to. To generate the data samples in the remaining time steps, we design a procedure by simulating the bank loan system in the real world. We first train a neural network classifier h_{θ^*} on $S, \mathbf{X}^1, Y^1$ and treat it as the ground-truth model. For each time step t, classifier h_{θ^*} takes as inputs S and $\mathbf{X}^t$ and outputs a probability distribution over Y^t. We then sample Y^t from the distribution as shown below:

$$P(Y^t) = h_{\theta^*}(S, \mathbf{X}^t) \qquad Y^t \sim \text{Bernoulli}(P(Y^t)) \tag{6.14}$$

After that, we update the value of $\mathbf{X}^t$ to obtain $\mathbf{X}^{t+1}$ based on the value of Y^t. We treat Y^t as the ground-truth of loan repayment ($Y^t = 1$) and default ($Y^t = 0$). An individual with $Y^t = 1$ should have a larger probability to be predicted as 1 in the next time step, and vice versa. Therefore, we update the value of $\mathbf{X}^t$ according to the value of Y^t as well as the gradient of a loss function between the predicted probability and label 1, as given below:

$$\mathbf{X}^{t+1} = \mathbf{X}^t - \epsilon \cdot (2Y^t - 1) \cdot \frac{\partial \mathcal{L}(h_{\theta^*}(S, \mathbf{X}^t), \mathbf{1})}{\partial \mathbf{X}^t} \tag{6.15}$$

where the parameter ϵ controls the magnitude of changes in $\mathbf{X}^t$. As a result, $\mathbf{X}^{t+1}$ will be predicted closer to label 1 if $Y^t = 1$, and will be predicted further from label 1 if $Y^t = 0$. Following above generation rules, we generate a 10-step synthetic time series dataset with 10000 instances and $\mathbf{X}^t$ is 6 dimensional vector. We refer to this dataset SimLoan.

6.5.2.2 Semi-Synthetic Dataset.

We also generate semi-synthetic data by leveraging the real-world Taiwan dataset [91] as the initial data at $t = 1$. A ground-truth classifier and similar generation rules of change are used to generate subsequent decisions $Y^1, ..., Y^l$ and profile features $\mathbf{X}^2, ..., \mathbf{X}^l$. There are 10000 instances in the initial data and they are randomly and equally sampled from groups by S and Y for balance. Like the SimLoan dataset, this dataset is also made up of 10 steps. We refer to this dataset Taiwan.

6.5.3 Implementations Details and Hyperparameters

Experiments are performed on the computer with Intel Core i7-9700K CPU and NVIDIA GeForce GTX 1180 GPU. Except for **LRLF**, other baselines and our framework are used multi-layer fully-connected networks, i.e., **MLP**, as the classifiers. The details of the model architectures and hyperparameters used in our framework on two datasets are given in Tables 6.1 and 6.2. For a fair comparison, we adopt the same network structure and parameter settings for our decision model h_θ. Both datasets are split into train/validation/test sets with the ratio 70/10/20. The models are trained on the train sets and the hyperparameters are chosen on the validation sets. The reported results are calculated on the test sets.

6.5.4 Evaluation

To evaluate the performance of models after deployment, the RCGAN trained in Phase 2 and the decision models that we evaluate are used together to generate interventional data on which the local and long-term fairness are computed. The long-term fairness is measured by Eq. (**??**) computed on the evaluated decision model; the local fairness is measured by the direct discrimination [17] at each time step; and the accuracy of predictions

Table 6.1: The architectures of h_ω and h_θ and hyperparameters for both datasets

Layer	Inputs	Output Dim	
		SimLoan	Taiwan
X		6	6
S		1	1
FC_1	[X, S]	32	16
FC_2	FC_1	64	32
FC_3	FC_2	1	1
Optimizer	Adam		
Learning rate	0.001		
Batch size	512		
λ_1		1.0	1.0
λ_2		128.4	40.0
λ_3		21.0	20.0

Table 6.2: The architecture of RCGAN and hyperparameters for both datasets

Layer	Inputs	Output Dim	
		SimLoan	Taiwan
X/Z		6	6
S/Y		1	1
Generator			
GRU_1	[Z, S, Y]	64	64
GRU_2	GRU_1	64	64
FC_1	GRU_2	6	6
Penalty			
MMD	[X, FC1]	1	1
Discriminator			
GRU_1	FC_1	64	64
GRU_2	GRU1	64	64
FC_1	GUR_2	1	1
Opimizer	Adam		
Learning rate	0.001		
Batch size	512		
γ	100		

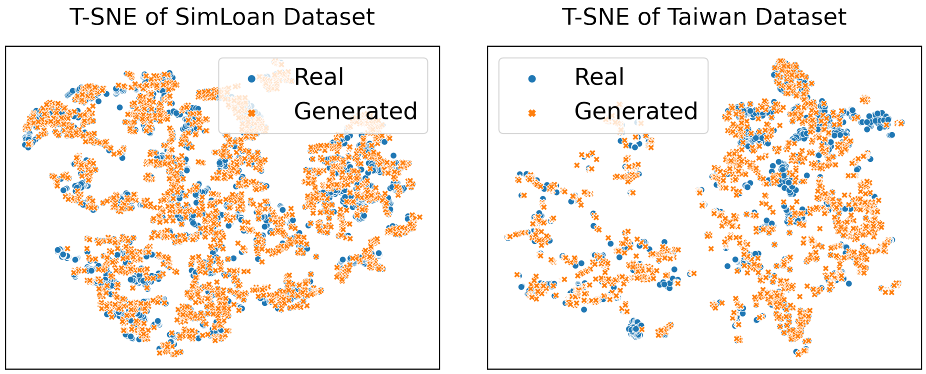


Figure 6.5: T-SNE visualization of real and generated data distributions.

is evaluated based on the ground-truth classifier h_ω at each time step.

6.5.5 Results

6.5.5.1 Performance of RCGAN.

The quality of the time series generated by the RCGAN is crucial for our model training and evaluation. To demonstrate the fidelity of the generated data, we leverage the T-SNE technique [105] to provide the visualization of complex distributions. Figure 6.5 shows the results of T-SNE visualization of real and generated data distributions. As we can see, the real and generated data after dimensionality reduction have very similar structures of data, which verifies the performance of the RCGAN.

6.5.5.2 Fairness and Accuracy of Decision Models.

To evaluate the performance of our algorithm and baselines, we conduct experiments with two settings on both SimLoan and Taiwan datasets. In the first setting, the time step T for achieving long-term fairness is set to 10. We train the models on the 10-step training data (i.e., within the time range $[1, 10]$) and evaluate the models on the 10-step generated datasets with $\mathbf{X}^1$ as input (i.e., also within the time range $[1, 10]$). The results of accuracy

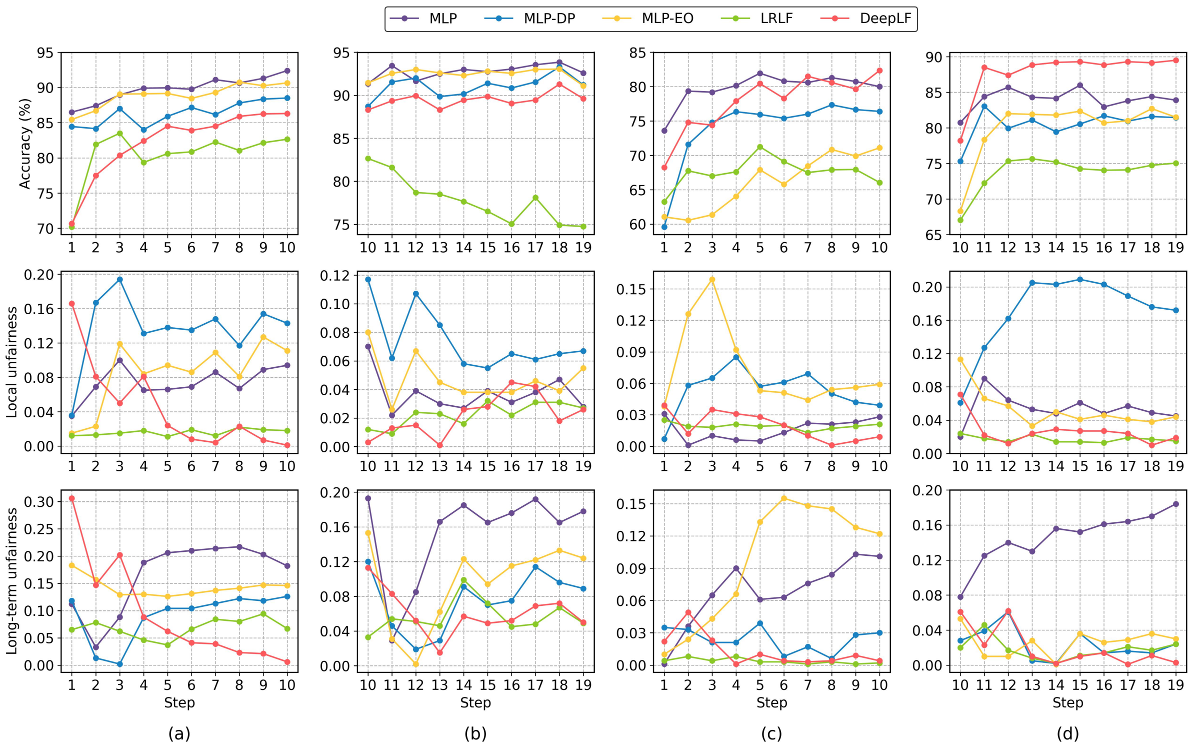
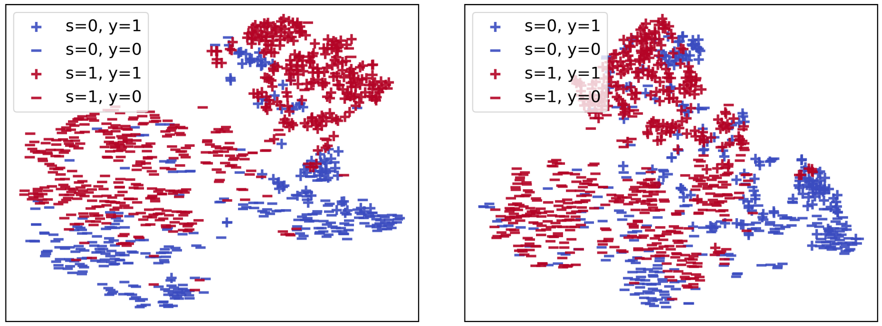
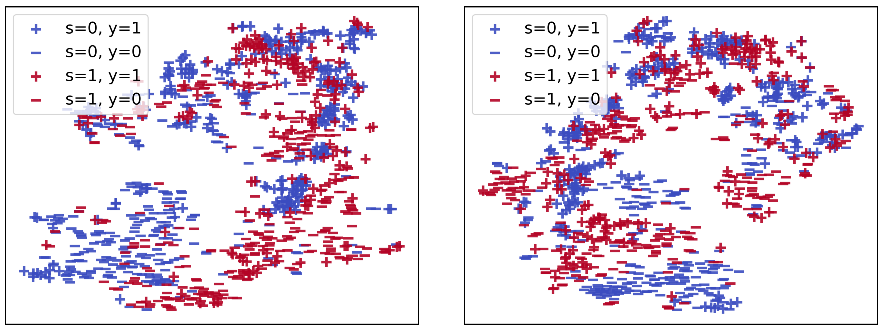


Figure 6.6: Accuracy, local and long-term unfairness of different algorithms on SimLoan ((a) and (b)) and Taiwan ((c) and (d)) datasets. The decision models are trained on generated data within the time range $[1, 10]$. (a) and (c): Results of evaluation on generated data within time range $[1, 10]$. (b) and (d): Results of evaluation on generated data within the time range $[10, 19]$.

and unfairness of all algorithms on the two datasets are shown in Figures 6.6 (a) and (c). We can see that the local and long-term unfairness of our algorithm have the obvious tendency to decline over time and reach low levels at $t = 10$. The trend in the figures shows how our algorithm achieves long-term fairness over time. For **LRLF**, it also produces relatively small local and long-term unfairness at $t = 10$ as expected, but its accuracy is much lower than that of our algorithm, probably due to the capacity of the logistic regression model. For the other three baselines, there is no clear decreasing trend in both local and long-term unfairness, although a relatively higher level of accuracy is achieved. This result verifies the impossibility result in the previous work which shows that static fairness constraints cannot guarantee long-term fairness [24, 23].



(a) SimLoan Dataset



(b) Taiwan Dataset

Figure 6.7: T-SNE of generated data distributions at time step $t = 10$ produced by **MLP** (left) and **DeepLF** (right).

To demonstrate how our algorithm produces a fair qualification distribution, we adopt T-SNE to visualize the distributions of $\mathbf{X}^{10}$ produced by **MLP** and **DeepLF**, as shown in Figure 6.7. It can be seen that compared with the distribution obtained by using the **MLP** as the decision model (left figures), the data samples of two groups ($s = 0, 1$) produced by **DeepLF** (right figures) are more evenly mixed together which implies a fairer qualification distribution.

In the second setting, the time step T for achieving long-term fairness is set to 19. We train the decision models on the same training data as in the first setting but evaluate the models on the 10-step generated data with $\mathbf{X}^{10}$ as the input, i.e., the generated data within the time range $[10, 19]$. The difference between the two settings is that in the first setting we pretend that we could modify the decision model in the past period during which the training data were generated, while in the second setting we only modify the decision model that will be deployed in the future (i.e., starting from $t = 10$). The results are shown in Figures 6.6 (c) and (d). In general, we observe similar results to the first setting where our algorithm achieves the best fairness performance compared with the baseline methods.

6.6 Summary

In this chapter, we proposed a three-phase deep generative framework to achieve long-term fairness by training an RCGAN to predictively generate observational and interventional data. Leveraging the causal time series graph and independent mechanism, we trained a classifier h_ω on the time series $\mathcal{D}$ in the first phase and trained an RCGAN in which the h_ω is used to generate decisions in the second phase. In the last phase, we trained a fair decision model with 1-Wasserstein distance as the long-term fairness constraint and direct discrimination as the local fairness constraint. The optimization was formulated as a performative risk minimization and solved by the repeated gradient descent algorithm. Experiments on both synthetic and semi-synthetic time series datasets showed that our method can achieve a balance among long-term fairness, local fairness, and accuracy.

7 Achieving Long-term Fairness for Dynamic Systems Through Reinforcement Learning

7.1 Introduction

How to achieve long-term fairness has been explored in previous studies through explicit modeling of dynamics and feedback loops. One branch of research is to investigate the impact of current decisions on a target population in application-specific scenarios by leveraging various analytical frameworks, such as the one-step feedback [106] or Pólya urn model [25]. There are analytical results that show that simply enforcing traditional fairness notions at each static decision point may produce adverse effects on disadvantaged groups in the long run [24, 23]. Long-term fairness has also been studied in the context of reinforcement learning (RL) where the system dynamics and feedback loops between decisions and the population are formulated through a Markov Decision Process (MDP) [107]. Following this line of research, [48, 108] establish simulation environments for studying long-term fairness in RL, based on which RL algorithms have been used to learn a decision-making policy that aims to optimize both policy utility and fairness as long-term objectives [31, 50]. However, one limitation of existing methods is that they usually formulate long-term fairness as the difference between the average or instantaneous rewards received by different demographic groups similar to static fairness notions, but do not take into account the inherent difference between traditional fairness notions and long-term fairness requirements.

In this chapter, we consider long-term fairness as a requirement on the states rather than the rewards following the definitions in some well-accepted research [24, 22, 23]. We acknowledge that traditional fairness notions and long-term fairness are distinct requirements that may not necessarily align with one another. Traditional fairness considers the equity of the outcomes or performance of the decision model at a single decision point. It is referred to as short-term fairness later in this chapter for a clear representation. Long-term fairness, on the other hand, refers to a long-term state in which equity is systematically satisfied. Such a state may be achieved by gradually reducing the gap between the qualification distributions of different groups. As a result, imposing short-term fairness constraints may not necessarily lead to long-term fairness even if the constraints are incorporated into a long-term objective. For example, suppose a bank uses different thresholds for making loan decisions for the advantaged and disadvantaged groups in order to ensure fair outcomes. However, this approach may not help narrow the gap between the credit score distributions of the two groups.

To address the above issue, we develop an algorithmic framework that promotes both short-term and long-term fairness simultaneously. Similar to prior works, we utilize the MDP framework to leverage its power in optimizing long-term objectives. By recognizing the distinct requirements of short-term and long-term fairness, we incorporate them into the RL algorithm using different approaches. Since the concept of long-term fairness is aligned with the principle of the MDP framework, we employ an in-processing approach to deal with this constraint. We adopt the 1-Wasserstein distance as the metric of the distribution gap and theoretically show that minimizing the distance can lead to a long-term fair state. On the other hand, we adopt a model-agnostic pre-processing approach to deal with short-term fairness to ensure that it is enforced throughout the sequential decision-making process. We extend a classic pre-processing approach called massaging [109] to the RL setting by integrating it with the policy optimization algorithm. Finally, we show the exact implementation of our algorithmic framework using three case studies, where the experimental results demonstrate that our method is capable of striking a desired balance between short-term fairness, long-term fairness, and the utility of the sequential decision system.

We summarize our contributions as below:

- We propose to achieve systematic equity in sequential decision-making by considering short-term and long-term fairness as distinct fairness requirements.
- We develop an efficient and flexible algorithmic framework that integrates short-term and long-term fairness with the MDP framework as distinct constraints.
- Three case studies within simulation environments are used to prove the effectiveness of our method by evaluating the performance of our method and comparing it with the state-of-the-art baselines.

7.2 Preliminaries

This section introduces the background and preliminaries of fair machine learning, reinforcement learning (RL), and proximal policy optimization (PPO), a prevalent policy optimization algorithm in RL.

7.2.1 Fair Machine Learning

The issue of fairness has become one of the most popular topics in machine learning in recent years. To measure fairness in algorithmic decision-making, a large number of fairness notions have been proposed in the literature. Typical examples include *demographic parity (DP)* and *equal opportunity (EO)*. DP aims to ensure that different demographic groups are represented proportionally in the outcomes of a decision model. EO, on the other hand, refers to the principle of treating individuals or groups fairly by ensuring equal error rates or predictive performance across different demographic subgroups. Then, to address the fairness issues, bias mitigation algorithms are proposed mainly from three perspectives: pre-processing, in-processing, and post-processing. Pre-processing approaches focus on eliminating bias from the training data (e.g., [32, 33, 34]), in-processing approaches aim to avoid introducing bias in model training by proposing new model structures or loss functions (e.g., [110, 111, 60]), and post-processing approaches modify predicted outcomes to resolve fairness issues (e.g., [79, 98, 44]).

7.2.2 Reinforcement Learning

Reinforcement learning consists of two interactive components, an agent and an environment, which interact with each other over time. This interaction process is modeled as a Markov decision process (MDP) [26]. An MDP is denoted by a tuple $M = (\mathcal{S}, \mathcal{A}, P, R, \rho_0, \gamma)$, where $S \in \mathcal{S}$ is a set of states, $A \in \mathcal{A}$ is a set of actions, $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is a transition function that represents the probability of next state given the current state and the action, $R : \mathcal{S} \to \mathbb{R}$ is a reward function, $\rho_0 : \mathcal{S} \to [0, 1]$ is an initial state distribution, and $\gamma \in [0, 1]$ is a discount factor. At each time step t, the agent observes a state $s_t \in S$ from the environment and takes an action $a_t \in A$ following a policy $\pi : \mathcal{S} \to \mathcal{A}$ based on the current state. Then the agent observes a new state $s_{t+1} \in S$ and a reward $r_t \in R$ generated by the environment with the transition probability $P(s_{t+1}|s_t, a_t)$ and the reward function $R(s_t)$ in the next time step $t+1$. The goal of RL is to learn a optimal policy π_θ which maximizes the expected discounted cumulative rewards, defined as below.

$$J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{\infty} \gamma^t R(s_t) \right],$$

where $\tau = (s_0, a_0, s_1, a_1, ...)$ is a trajectory and $\tau \sim \pi_\theta$ means that a trajectory τ is sampled from the policy π_θ following $s_0 \sim \rho_0(s_0), a_t \sim \pi(a_t|s_t), s_{t+1} \sim P(s_{t+1}|s_t, a_t)$.

To address the RL problem, there are several concepts that are often involved in RL algorithms. Let $R(\tau)$ denote the discounted cumulative rewards of a trajectory τ. The state value function V and the state-action value function Q are given by $V(s_t) = \mathbb{E}_{\tau\sim\pi}[R(\tau)|s_t = s]$ and $Q(s_t, a_t) = \mathbb{E}_{\tau\sim\pi}[R(\tau)|s_t = s, a_t = a]$, which evaluate how good a state or a pair of state and action is. The advantage function is the difference between $Q(s_t, a_t)$ and $V(s_t)$, i.e., $A(s_t, a_t) = Q(s_t, a_t) - V(s_t)$, and it can be considered as the advantage of taking a given action over following the policy [112].

7.2.3 Proximal Policy Optimization

Policy optimization methods [113] are a type of reinforcement learning algorithms that improve policies directly by estimating policy gradients and optimizing with stochastic gradient ascent. The most commonly used form of gradient estimator is given by

$$\nabla J(\theta) = \mathbb{E}_{(s_t,a_t)\sim\pi_\theta}[A(s_t, a_t)\nabla_\theta \log \pi_\theta(a_t|s_t)] \tag{7.1}$$

where the expectation is estimated over a batch of samples.

Proximal policy optimization (PPO) [114] is a state-of-the-art policy optimization algorithm stemming from the trust region policy optimization (TRPO) algorithm [115]. It maximizes a clipped surrogate function to prevent the gradient update dramatically, as follows

$$J^{CLIP}(\theta) = \hat{\mathbb{E}}\left[\min(r_t(\theta)A_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)A_t)\right] \tag{7.2}$$

where A_t is short for $A(s_t, a_t)$, $r_t(\theta)$ denotes the probability ratio $\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$, and ϵ is a hyperparameter. To compute variance-reduced advantage function, a neural network is used to estimate the state value function $V(s_t)$ with the squared-error loss

$$L^V(\theta) = \mathbb{E}[(V_\theta(s_t) - R(\tau))^2] \tag{7.3}$$

The clipped surrogate function restricts the magnitude of the gradient update, which not only makes the algorithm more stable, but also allows for multiple updates using a batch of samples, improving the data efficiency.

In this chapter, we adopt PPO as the RL algorithm, but our method can be applied to any policy optimization algorithms.

7.3 Problem Formulation

To develop an RL algorithm for achieving long-term fairness, we start by defining fairness notions in the context of sequential decision-making, presenting the problem formulation, and establishing the setting of the fair RL learning problem.

7.3.1 Fairness Definition for Sequential Decision Making

As stated in the last section, traditional fairness notions are usually concerned about whether the machine learning model produces the same outcome or performance across different groups in a static population. In the context of sequential decision-making, we refer to this type of notions as short-term fairness notions, which are defined based on a cohort of individuals who participate in the decision-making system over a specific period.

Definition 19 (Short-term Fairness)**.** *In a sequential decision-making system, short-term fairness is defined as the equal outcome or performance of the decision model/policy over a participating cohort.*

It is worth emphasizing that, first, short-term fairness notions may be enforced by laws and regulations in some domains, such as the U.S. Equal Employment Opportunity Commission [116] that prohibits employment discrimination. So, it is essential to enforce short-term fairness throughout the sequential decision-making process including both training and evaluation. Second, different short-term fairness notions may conflict and not be achieved simultaneously if historical and/or systemic biases exist [117]. For example, achieving demographic parity may require preferential treatment to account for historical disadvantages, which could potentially impact equal opportunity. Thus, we adopt a single short-term fairness notion only in our algorithm.

On the other hand, long-term fairness has been proposed to account for fairness and equity of the sequential decision-making system in the long run [24]. The general goal of long-term fairness is to reach a state where the historical disadvantages are rectified and systemic biases are removed. Since the trade-off between short-term and long-term fairness is due to historical disadvantages and systemic biases, we presume that long-term fairness also implies a state in which it becomes easier to simultaneously satisfy different short-term fairness notions.

How to quantify long-term fairness? In the literature, features are regarded as indicators or metrics that assess the qualification or competency levels of individuals. Then, long-term fairness is often formulated by measuring the gap in feature distributions between different groups. For example, in [24], the difference in feature distribution of the disadvantaged group between the starting time step and the ending time step, i.e., $\triangle = \mathbb{E}[x_{t=t^*}|c^-] - \mathbb{E}[x_{t=0}|c^-]$, is defined as the measure of long-term fairness. It is called long-term improvement if $\triangle > 0$, stagnation if $\triangle = 0$, and active harm if $\triangle < 0$. In [22], long-term fairness is defined as the parity in the feature distribution between the advantaged and disadvantaged group when the dynamic system is at equilibrium, i.e., $\triangle = |\mathbb{E}[x_{t=t^*}|c^+] - \mathbb{E}[x_{t=t^*}|c^-]|$. In this chapter, we adopt a similar philosophy to [22] and define long-term fairness as follows.

Definition 20 (Long-term Fairness)**.** *Long-term fairness is defined as the equal feature distributions of different groups at a long-term state of the sequential decision-making system.*

After discussing the two types of fairness we consider, we define our problem formulation as follows.

Problem Formulation 5. *Consider a sequential decision-making context. A policy for making the decision is learned through an iterative process of interaction with the environment. Our goal is to learn a fair policy such that: (1) short-term fairness is guaranteed throughout both the training and evaluation processes, and (2) a long-term fair state is reached at the end of the evaluation process.*

7.3.2 Problem Setting for Fair RL

To formulate the problem of long-term fair sequential decision-making, we consider a finite-horizon RL problem and leverage the MDP framework. Specifically, in our context states $\mathcal{S} = \mathcal{C} \times \mathcal{X}$ where $\mathcal{C}$ is the domain of the sensitive feature and $\mathcal{X} \in \mathbb{R}^m$ represents the domain of the profile features. When $\mathcal{C}$ is a binary domain, we use $\{c^+, c^-\}$ to represent the advantaged and disadvantaged groups respectively. Let $\mathcal{A}$ denote the action space. If $\mathcal{A}$ is a binary domain, we use $\{1, 0\}$ to represent the positive and negative actions. Our goal is to learn a stochastic policy $\pi_\theta : \mathcal{S} \rightarrow \mathcal{A}$ which maximizes the agent's cumulative reward while satisfying certain fairness criteria. Note that we generally allow the sensitive attributes to be involved in the policy input and will explicitly adopt constraints to enforce fairness requirements.

We may use the bank loan system as an example to illustrate this problem setting in a real-world scenario. In this example, the bank is treated as the agent, and the population of the applicants is treated as the environment. Assume that the sensitive attribute C is the race of the applicant. As an RL process, at each time step t, an individual from a certain race group is sampled and applies for a loan from the bank, whose state s_t is given by race c_t and personal profile x_t. Then, the bank runs a policy function π_θ to decide whether to approve the loan according to probability distribution $\pi_\theta(a_t|s_t)$. If the loan is approved, depending on whether the individual repays the loan, both the bank's profit and the feature distribution of the population will be affected which determines the new state and the reward. Finally, the goal of the bank is to learn from interactions a lending policy that maximizes its own profit while satisfying fairness.

7.4 Algorithm

In this section, we develop a flexible and effective fair RL algorithm that integrates the pre-processing and in-processing approaches to promote short-term and long-term fairness simultaneously. We assume separate training and evaluation processes where the policy is updated according to the interaction experience with the environment during the training process, and it is further evaluated in a separate environment with the same dynamics after the training completes. However, our method is readily applied to the setting where the evaluation is conducted during the training. In the following, we first provide an overview of the algorithm.

7.4.1 Overview

When developing the fair RL algorithm, it is critical to consider the different requirements for long-term and short-term fairness. Long-term fairness is a state that represents the maximization of the equity of the system in the long run. As it is aligned with the principle of the MDP framework that maximizes the expected total reward over time, we adopt an in-processing approach that regularizes the reward to incorporate the long-term fairness objective so that the RL algorithm can be aware of the long-term fairness status in training. Specifically, we regularize the advantage function of a policy optimization algorithm as:

$$A^{\lambda}(s_t, a_t) = A(s_t, a_t) + \lambda R(s_t), \tag{7.4}$$

where $R(s_t)$ is the regularization that reflects the long-term fairness requirements and λ is a hyperparameter that controls the degree of regularization.

On the other hand, short-term fairness is an instantaneous constraint that may be enforced by laws or regulations at every step of the decision-making. Merely incorporating short-term fairness through regularizing the advantage function may not be sufficient to guarantee short-term fairness throughout the entire training process. Thus, we choose to adopt a pre-processing approach to address this issue. Inspired by the classic pre-processing approach named massaging [109], we propose a method called *action massaging* that selectively alters unfair actions produced by the policy network to fair ones. Specifically, after an action a_t is sampled from the policy network π_θ at time step t, we employ a functional mapping $a'_t = m(s_t, a_t)$ where a'_t may or may not be equal to a_t to generate the trajectory.

By using this altered trajectory to perform policy optimization, the policy gradient becomes

$$\nabla J(\theta) = \mathbb{E}_{(s_t,a_t)\sim\pi_\theta}[A^\lambda(s_t, a'_t)\nabla_\theta \log \pi_\theta(a'_t|s_t)], \tag{7.5}$$

which shows that the policy gradient is computed based on the trajectory formed by a'_t. The rationale of the action massaging is to perform fair actions when the policy network generates biased ones and also encourage the policy to generate fair and high-reward actions. Note that this approach differs from the off-policy RL learning algorithm which optimizes the current policy network based on the trajectories generated by a different policy, and hence our approach does not require importance sampling to correct for the bias.

Next, we describe the above pre-processing and in-processing components in detail.

7.4.2 Action Massaging for Short-term Fairness

The action massaging altered actions according to a pre-defined short-term fairness criterion that is to meet legal and regulatory requirements for decision-making and prevent discrimination against certain groups. As mentioned earlier, we consider group fairness notions such as DP or EO in our work. At each time step t, short-term fairness relies on the current state and action as well as the past states and actions of the system. To facilitate computation, we adopt a sliding window w so that all the states and actions between time step $t-w$ and t form a participating cohort that will be used to measure short-term fairness. We denote this measure as $\triangle_s(s_t, a_t)$. Then, the action massaging $m(s_t, a_t)$ alters the action a_t to a'_t that minimizes $\triangle_s(s_t, a_t)$ to improve short-term fairness.

When designing the mapping $m(s_t, a_t)$, one principle is that the modifications should not significantly damage the utility of the policy. In our method, we treat $\pi_\theta(a_t|s_t)$ as the confidence level for selecting action a_t, and introduce a constraint that limits the difference between the confidence of the original action a_t and the altered action a'_t. The action massaging only alters the current action to a different action when the above difference is smaller than a predefined threshold. When multiple actions satisfy the constraint, the action massaging chooses the one that leads to best short-term fairness. As a result, the action massaging is formulated as follows:

$$\begin{aligned} m(s_t, a_t) = &\operatorname*{argmin}_{a'_t \in \mathcal{A}} \triangle_s(s_t, a'_t) \\ &\text{s.t.} \quad |\pi_\theta(a_t|s_t) - \pi_\theta(a'_t|s_t)| < \tau. \end{aligned} \tag{7.6}$$

The constraint in Eq. (7.6) is an important factor that reflects the trade-off between short-term fairness and utility. It aids in enhancing short-term fairness while minimizing the impact on utility. On one hand, the constraint reduces the number of modifications made by the action massaging, as a large number of modifications will cause instability and deviation of training. On the other hand, the constraint also restricts the modifications to be carried out when current actions have low confidence and hence leads to a smaller reduction in utility. For example, in a special case of binary actions (e.g., the decision of bank loan), the constraint will restrict the modifications to actions with confidence close to 0.5. The exact implementation of the action massaging is task-specific and varies with applications, as will be shown in the case studies in the next section.

7.4.3 Advantage Regularization for Long-term Fairness

The action massaging for short-term fairness is not enough to achieve long-term fairness as the objective of short-term fairness may not exactly align with the objective of long-term fairness. As mentioned above, we leverage the MDP's capacity to maximize long-term returns as a means to attain long-term fairness. Just as done in Chapter 6, to quantify long-term fairness, in our work we employ the 1-Wasserstein distance between the distributions of different groups as the long-term fairness metric.

The proposition 1 in Chapter 6 shows that, by approaching a long-term state where the 1-Wasserstein distance between the feature distributions of different groups is minimized, we can mitigate at that state both the DP and EO of any decision model that is Lipschitz continuous. This implies that a long-term fair state has been reached.

Denote the long-term fairness measure computed at time step t as $\triangle_l(s_t)$. Similar to short-term fairness, we adopt a sliding window to form a participating cohort for estimating the feature distributions. Then, we incorporate $\triangle_l(s_t)$ into the advantage function as the regularization. However, rather than directly adding $\triangle_l(s_t)$ to the advantage function, we further consider the trade-off between short-term fairness and long-term fairness when the two objectives are not aligned. Specifically, we promote the advantage when both short-term and long-term fairness can be improved while demoting the advantage when both short-term and long-term fairness is damaged. When there is a conflict between short-term and long-term fairness, we keep the current advantage unchanged. The regularization term is defined as follows:

$$R(s_t) = \begin{cases} \min(0, \triangle_l(s_t) - \triangle_l(s_{t+1})) & \triangle_s(s_t, a_t) > \delta \\ \max(0, \triangle_l(s_t) - \triangle_l(s_{t+1})) & \triangle_s(s_t, a_t) \leq \delta \end{cases} \tag{7.7}$$

As can be seen, the first term is active when short-term fairness $\triangle_s(s_t, a_t)$ is larger than the threshold δ. Then, this term will penalize the original advantage when the long-term

Algorithm 3: Fair Proximal Policy Optimization (F-PPO)

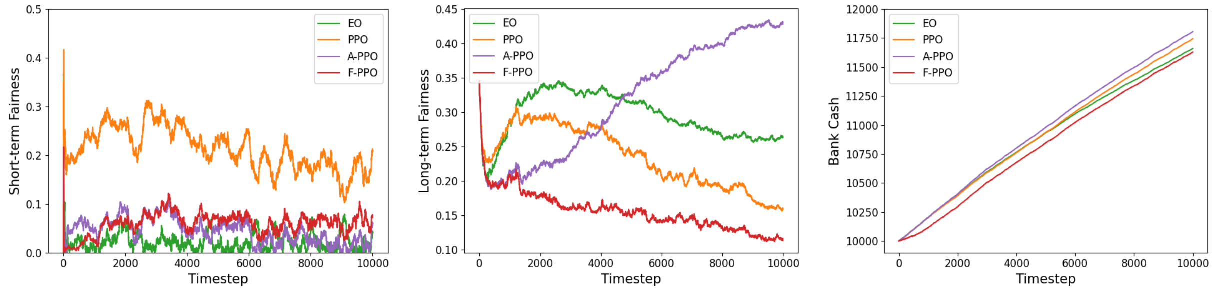
Initialize policy network π_θ and value function network v_ϕ;
for k = *0, 1, 2, ...* **do**
  Collect trajectories $\mathcal{D}_k$ from policy π_θ where actions a_t are sampled from $\pi_\theta(a_t|s_t)$;
  Compute $\triangle_s(s_t, a_t)$ and apply action massaging according to Eq. (7.6) to obtain altered trajectories;
  Compute $\triangle_l(s_t)$ and penalized advantage $A^\lambda(s_t, a_t)$ according to Eqs. (7.4) and (7.7) ;
  Update the policy by maximizing the clipped surrogate function J^{CLIP} according to Eq. (7.2);
  Update the value function network by minimizing the squared-error loss L^V according to Eq. (7.3);
end

fairness measure does not decrease at the next time step $t+1$ compared to the current time step t. The second term is active when short-term fairness $\triangle_s(s_t, a_t)$ is less or equal to δ. In this case, we reward the advantage function when the long-term fairness measure reduces. This approach allows for a continuous improvement of long-term fairness throughout the entire sequence, rather than a sudden change at a specific point, while it remains simple and effective.

Combining the above two methods for fairness, we present the pseudocode of our algorithm in Algorithm 3, which is referred to as F-PPO.

7.5 Experiments

For demonstrating the performance of our proposed method, we make use of the simulation environments [48, 108] that implement toy examples of dynamic systems for supporting studies of long-term consequences of ML-based decision systems. We conduct three case studies in the context of bank loans, allocation of attention, and epidemic control. The proposed method is evaluated with respect to utility, short-term fairness, and long-



(a) The short-term fairness (b) The long-term fairness (c) The amount of bank cash

Figure 7.1: Experimental results for bank loans. The recorded values are averages over 10 evaluation runs.

term fairness. As mentioned earlier, the policies will be first trained by interacting with the environment and then tested separately in the environment.

Baselines. We consider two different categories of baseline agents in our experiments. The first category is human-designed policy agents used in [48, 31], including the EO agent for bank loans, the CPO agent for attention allocation, and the Max agent for epidemic control. The second category consists of learning-based policy agents. We consider the original PPO algorithm that only maximizes the cumulative reward, and the A-PPO algorithm that is the state-of-the-art fair RL algorithm proposed in [31] for achieving fairness through advantage regularization. As an ablation study, we also consider a variant of our method named F-PPO-L that only consists of the long-term fairness component but with the short-term fairness component removed.

7.5.1 Case Study: Bank Loans

In this case study, the bank lending scenario is simulated where an agent plays the role of a bank to make decisions about whether to grant loans to a stream of applicants. The qualification of applicants is described by a discrete credit score, which changes with the loan decisions.

Environment. In this environment, at each time step the bank observes a loan applicant s_t which is sampled with replacement from the pool of applicants. Each applicant consists of a credit score (qualification feature) and a group membership (sensitive feature). The group membership c_t is uniformly sampled from $\{c^+, c^-\}$. The credit score x_t, on the other hand, is drawn from a group-dependent discrete distribution over $X \in \{1, 2, ..., X_{max}\}$ with $\mathbb{E}_{c^-}[X] < \mathbb{E}_{c^+}[X]$. The bank employs a policy to make binary decisions of whether to deny or approve loan applications. If a loan applicant receives a loan and defaults, his/her credit score C_i drops, which is simulated in the distribution by moving a small portion of mass from $P_c(X_i)$ to $P_c(X_{i-1})$ if $i \neq 1$. Similarly, if the loan applicant receives a loan and repays, a small portion of mass will be moved from $P_c(X_i)$ to $P_c(X_{i+1})$ if $i \neq X_{max}$. There is no change in the distribution if the applicant does not receive the loan. The bank's profit increases by the amount of the loan plus the interest on successfully repaid loans, and decreases by the loan amount on defaults. The probability of default is given by a deterministic function of the credit score. The reward of the bank at each time step is defined as the change in its profit at the next time step.

Implementation of F-PPO. In this case study, we implement a policy network $\pi_\theta(a_t|s_t)$ to make binary decisions $a_t \in \{1, 0\}$. We adopt EO as the short-term fairness notion, which is defined as follows:

$$\triangle_s(s_t, a_t) = \left| \frac{\sum_{t-w}^{t} \text{successful\_loan}_{tc^+}}{\sum_{t-w}^{t} \text{will\_repay}_{tc^+}} - \frac{\sum_{t-w}^{t} \text{successful\_loan}_{tc^-}}{\sum_{t-w}^{t} \text{will\_repay}_{tc^-}} \right|, \tag{7.8}$$

where $w = 300$ is the sliding window size. For the action massaging, the action will be flipped if the alternative action is fairer in terms of short-term fairness and the confidence of the action is lower than the threshold. The threshold in Eq. (7.6) is dynamically adjusted according to the number of training iterations. The idea is to perform a cold start in action massaging so that the actions are not altered at the beginning of training. The threshold is initially zero and increases after a certain number of iterations. Specifically, the threshold τ at the ith iteration is defined as $\tau = 1 - 2\tau(i)$ where

$$\tau(i) = \begin{cases} \tau_s \cdot \gamma^{i-i_s} & i \geq i_s \\ 0.5 & \text{otherwise} \end{cases}$$

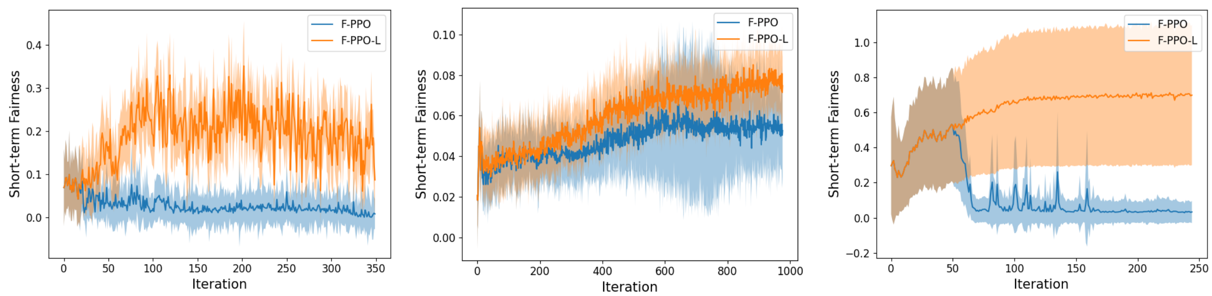
In our experiments, we set $\tau_s = 0.5$, $i_s = 17$, and $\gamma = 0.985$. Finally, long-term fairness is computed as

$$\triangle_l(s_t) = W(P_{t-w:t}(X|c^+), P_{t-w:t}(X|c^-))$$

where $P_{t-w:t}(X|c)$ is the distribution mass of the credit score of group c measured within the sliding window. For other hyperparameters, we set $\lambda = 1$ and $\delta = 0.05$.

Agents. We include PPO, A-PPO, and EO as baselines to compare with our F-PPO, where EO is the agent that maximizes the bank profits subject to constraints of equal opportunity at every time step.

Results. The short-term fairness, long-term fairness, and reward obtained by different agents during the test are shown in Fig. 7.1. As can be seen, despite similar performance in terms of reward (Fig. 7.1c), the fairness performance of different methods is diverse. For short-term fairness (Fig. 7.1a), EO, A-PPO and F-PPO are all able to keep the bias values below 0.1, while the original PPO produces much larger bias values. For long-term fairness (Fig. 7.1b), our F-PPO approach exhibits superior performance among all the methods considered, as it consistently achieves the smallest bias values, and these bias values continue to decrease over time. On the other hand, A-PPO produces the worst performance where



(a) Bank Loans (b) Attention Allocation (c) Epidemic Control

Figure 7.2: Ablation study: mean and standard deviation of short-term fairness in each iteration measured during training.

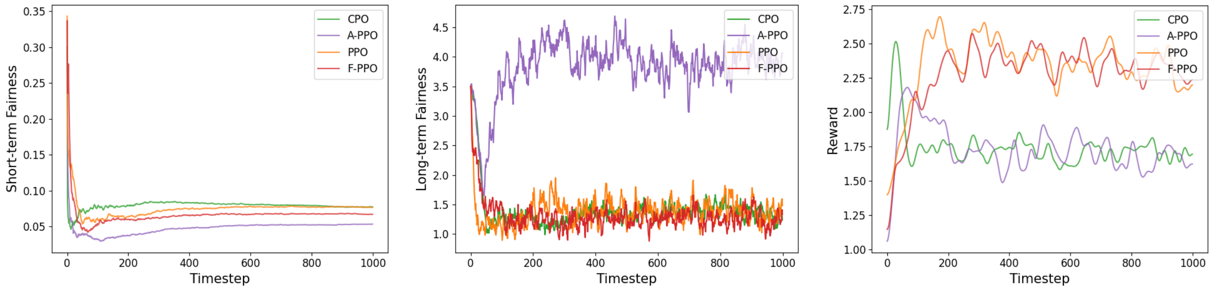
the bias values increase over time. This result shows that simply adding traditional fairness constraints into a long-term objective does not necessarily achieve long-term fairness. By combining the three results, we see that our F-PPO algorithm strikes a desirable balance between short-term fairness, long-term fairness, and utility of the policy.

For the ablation study, Fig. 7.2a shows the mean and standard deviation of short-term fairness achieved by F-PPO and F-PPO-L in each iteration of the training process over 350 iterations. The results show the effectiveness of the action massaging in maintaining short-term fairness throughout both the training and evaluation processes, while the long-term fairness component alone cannot guarantee short-term fairness.

7.5.2 Case Study: Attention Allocation

This scenario aims to simulate incident monitoring and mitigation. In the simulation, the agent's role is to assign attention units to a set of locations. Each attention unit can prevent, or catch, one incident at the location to which it is assigned. The incident rates at each location vary over time in accordance with the number of incident occurrences as well as the agent's decisions on how to assign attention units.

Environment. In the attention allocation environment, let N represent the number



(a) Short-term fairness (b) Long-term fairness (c) Average rewards

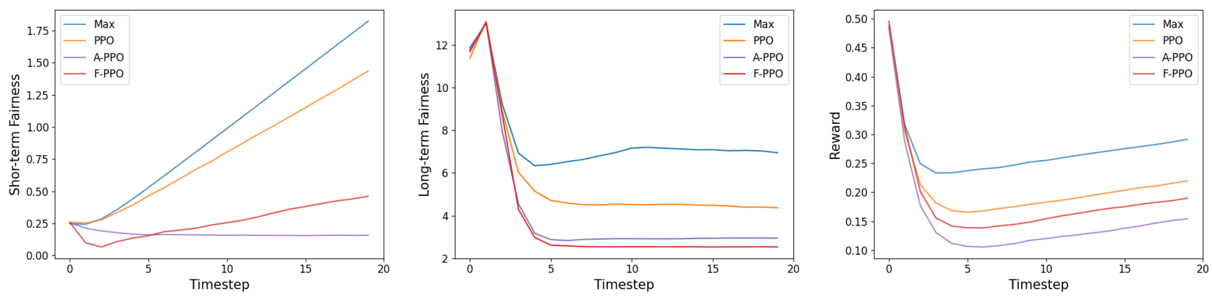
Figure 7.3: Results for the Attention Allocation environment. The recorded values are the averages over 10 evaluation episodes.

of attention units, and K be the number of locations. At each time step t, the agent assigns all N units over the K locations and $a_{k,t}$ denotes the number of units assigned to location k. The number of incidents that occur at each location is sampled from a Poisson distribution as $y_{k,t} \sim \text{Poisson}(\Lambda_{k,t})$, where $\Lambda_{k,t}$ is a dynamic parameter which changes according to

$$\Lambda_{k,t+1} = \begin{cases} \Lambda_{k,t} + \lambda_k^I & \text{if } a_{k,t} = 0 \\ \Lambda_{k,t} - \lambda^D \cdot a_{k,t} & \text{otherwise.} \end{cases}$$

Here, λ_k^I is the increase rate, which may vary between locations k, and λ_D is the decrease rate which is the same across locations. The number of incidents discovered at a location is given by $\hat{y}_{k,t} = \min(a_{k,t}, y_{k,t})$. The reward is defined as $r(s_t) = \zeta_0 \sum_{k=1}^{K} \hat{y}_{k,t} - \zeta_1 \sum_{k=1}^{K} (y_{k,t} - \hat{y}_{k,t})$ which is determined by the fraction of incidents discovered. The parameters ζ_0 and ζ_1 weight the reward function in terms of incidents discovered versus incidents missed. The state s_t is an observation history of length H and each observation is a tuple of vectors $(\hat{y}_t, y_t, a_t, \hat{y}_t \oslash y_t)$, where $\oslash$ denotes the Hadamard division operation.

Implementation of F-PPO. The policy network for attention allocation produces a K dimensional vector of logits which are converted into a probability distribution $P(k)$



(a) Short-term fairness (b) Long-term fairness (c) Average rewards

Figure 7.4: Experimental results for epidemic control. The recorded values are averages over 200 evaluation episodes.

using the softmax function. The action is constructed by iteratively assigning attention units to the locations until all have been assigned. In each iteration, one attention unit is assigned to the location with the highest probability, from which the amount of $\frac{1}{N}$ is removed before the next iteration. For short-term fairness, we adopt DP as the metric defined as follows:

$$\triangle_s(s_t, a_t) = \max_k \left| \frac{\sum_{t'=t-w}^{t} a_{k,t'}}{N \cdot w} - \frac{1}{K} \right|, \tag{7.9}$$

which requires that the number of units assigned should be equal across different locations. The action massaging checks for each pair of locations k_1, k_2 where at least one unit is assigned to k_1, if reallocating one unit from k_1 to k_2 would improve short-term fairness. To minimize the impact of the action massaging on the utility, the algorithm also checks if the difference between $P(k_1)$ and $P(k_2)$ is less than the threshold. When both conditions are met by multiple pairs, the algorithm selects the one that leads to the best short-term fairness performance and performs the reallocation. For simplicity, we use a static threshold of 0.08. Finally, we measure long-term fairness according to the incident distribution over all locations. In training, we estimate the incident distribution based on the number of incident occurrences, but in evaluation, we use $\Lambda_{k,t}$ as the ground truth of the incident distribution.

Agents. For baselines used to evaluate our F-PPO agent, we consider PPO, A-PPO, as well as the CPO agent that aims to discover the most incidents.

Results. The experimental results are shown in Fig. 7.3. Our F-PPO achieves the best long-term fairness performance while maintaining relatively low short-term bias values and high rewards. As a comparison, although A-PPO produces the best short-term fairness performance, its long-term fairness performance and utility are among the worst. For the ablation study, Fig. 7.2b shows the effectiveness of the short-term fairness component.

7.5.3 Case Study: Epidemic Control

The third case study simulates an infectious disease scenario where vaccines are allocated within a social network in a step-by-step manner. At each step, one individual is selected by the policy to receive the vaccine. Meanwhile, healthy individuals have the chance to get infected, and sick individuals have the chance to recover. The task is to optimally allocate vaccines in order to mitigate the spread of disease.

Environment. In this environment, we have a social network $G = (V, E)$ where V is a set of individuals and E is a set of edges representing social connections. The state of the environment is a vector of the health state of all individuals. For each individual, the health state is represented by a four-dimensional one-hot encoding $H = \{S, I, R\}$ that represents the three possible states that an individual can be in, including susceptible (healthy), infected, and recovered. At the initial time step, a random set of individuals V_0 are infected. Then, at each time step t, the probability of a susceptible individual i transitioning to the state of infected is given by $P_I(v_{i,t}) = 1 - (1-\tau)^{|N_I(v_{i,t})|}$, where $N_I(v_{i,t})$ represents the number of infected individuals in the neighbor of individual i and $\tau \in [0, 1]$ is an infectious factor. Meanwhile, the probability of an infected individual recovering is given by $P_R(v) = \rho$ where

$\rho \in [0,1]$ is the recovering factor. If a susceptible individual receives the vaccine, his/her state directly transitions from S to R. The reward is defined as the proportion of the population who are not infected. To study fairness, the Girvan-Newman algorithm [118] is used to partition the network G into two communities corresponding to groups c^+ and c^-.

Implementation of F-PPO. The policy network is a multiclass classifier that outputs the probabilities of $|V|+1$ actions representing either not vaccinating or vaccinating any of the $|V|$ individuals. EO is still adopted as the short-term fairness metric, which measures the vaccination ratio among newly infected individuals in different groups as follows

$$\triangle_s(s_t, a_t) = \left| \frac{\sum_{t-w}^{t} \text{vaccine\_given}_{tc^+}}{\sum_{t-w}^{t} \text{new\_infected}_{tc^+} + 1} - \frac{\sum_{t-w}^{t} \text{vaccine\_given}_{tc^-}}{\sum_{t-w}^{t} \text{new\_infected}_{tc^-} + 1} \right|. \tag{7.10}$$

For the action massaging, at each time step it checks if providing the vaccine to an individual from the other community would result in a fairer allocation. If this condition is met, the algorithm proceeds to check if there exists an individual from the other community whose predicted probability is sufficiently close to that of the current individual. If such an individual is found, the algorithm modifies the action accordingly. A dynamic threshold is again adopted. For the ith iteration, the threshold is defined as:

$$\tau(i) = \begin{cases} \min(\tau_e, \tau_s \cdot \gamma^{i-i_s}) & i \geq i_s \\ 0 & \text{otherwise} \end{cases}$$

Specifically, we set $\tau_s = 0.01$, $\tau_e = 0.35$, $i_s = 50$, $\gamma = 1.2$. Finally, long-term fairness is measured as the distance between the health states of the two communities. For other hyperparameters, we set $\lambda = 0.25$ and $\delta = 0.05$.

Agents. We consider the Max agent in addition to PPO, A-PPO, and F-PPO. The Max agent vaccinates the most susceptible individual each time, which is considered as the individual with the most number of infected neighbors.

Results. The experimental results are shown in Fig. 7.4. As can be seen, F-PPO achieves the best performance in terms of long-term fairness and significantly improves short-term fairness compared with the Max and PPO agents. A-PPO achieves the best performance in terms of short-term fairness, but produces the worst performance in gaining rewards. As expected, the Max agent achieves the highest utility performance, but it also demonstrates the poorest fairness performance. The combination of the results also demonstrates the capability of the F-PPO. The ablation study in Fig. 7.2c shows similar results to the other two case studies.

7.6 Conclusions

In this chapter, we studied the problem of achieving long-term fairness in sequential decision-making systems. We modeled the system as a Markov Decision Process (MDP) and tackled the problem by developing a fair reinforcement learning (RL) algorithm. By acknowledging that short-term fairness and long-term fairness are distinct requirements that may not necessarily align with one another, we developed an algorithmic framework that incorporates both requirements using different bias mitigation approaches, including pre-processing and in-processing approaches. We conducted three simulation case studies. The results show that our method can strike a balance between short-term fairness, long-term fairness, and utility.

8 Conclusions and Future Work

In recent years, with the widespread application of automated decision-making systems, the issue of algorithmic bias has received a lot of attention, and fair machine learning has become a hot research topic. Researchers have proposed various fairness metrics and fairness algorithms. However, current research mainly focuses on studying fairness in static environments, while most real-world problems are dynamic. In this dissertation, we studied fairness in dynamic environments and proposed algorithms to achieve long-term fairness in several different settings. This chapter concludes the whole dissertation and discusses the future work.

8.1 Conclusions

Our dissertation has conducted several works to mainly address the following research problems:

1. How to make multiple interrelated decision models make fair decisions simultaneously in a dynamic system;
2. How to measure and achieve long-term fairness of a decision model in a dynamic system;
3. How to learn a model from collected static data to achieve long-term fairness beyond the data;
4. How to achieve long-term fairness while balancing short-term fairness and utility in a reinforcement learning environment.

To address the above problems, we have summarized the proposed algorithms and definitions as follows.

In Chapter 4, we studied how to train multiple fair classifiers in dynamic settings. It is not feasible to train multiple models using the collected dataset according to the conventional fairness algorithms. Because the previous models will change the data distribution after deployment, the later models are no longer fair. Leveraging structural causal model, we treated each decision model as a soft intervention and inferred the post-interventional distributions. Then, by using do-calculus and surrogate functions, we derived loss functions and fairness constraints based on post-interventional distributions. Multiple fair classifiers were obtained by solving the optimization problem composed of loss functions and fairness constraints. In addition, we theoretically showed that combining multiple decision models in the optimization would not introduce additional surrogate errors. Experimental results showed the effectiveness of our algorithm.

In Chapter 5, sequential decision-making was described by a time-lagged causal graph, and we proposed that the path-specific causal effects of the sensitive attribute on the decision at a certain time step is regarded as its long-term fairness. Similar to Chapter 4, it can be formulated as a constrained optimization problem, but for longer decisions, the derived fairness constraints will be very complex. To simplify the computation, the problem was converted to a performative risk optimization problem and then we proposed an optimization algorithm by leveraging repeated risk minimization. Moreover, the convergence of the proposed algorithm was analyzed theoretically. Finally, we conducted experiments on two synthetic datasets and verified the effectiveness of the proposed framework and algorithm.

In Chapter 6, we have proposed a three-phase deep generative framework that utilizes RCGAN to predictively generate observational and interventional data in order to achieve long-term fairness. The framework employs a causal time series graph and independent mechanism to train a classifier h_ω on the time series $\mathcal{D}$ in the first phase, and an RCGAN in the second phase, where the decisions are generated using h_ω. In the final phase, we train a fair decision model that uses 1-Wasserstein distance as the long-term fairness constraint and direct discrimination as the local fairness constraint. We have formulated the optimization as a performative risk minimization and solved it using the repeated gradient descent algorithm. Experimental results on both synthetic and semi-synthetic time series datasets demonstrate that our method achieves a balance between long-term fairness, local fairness, and accuracy.

In Chapter 7, we presented a novel algorithm F-PPO that can train a fair policy to promote short-term and long-term fairness simultaneously in sequential decision making scenarios. Instead of only considering short-term fairness as previous work, we took both short-term and long-term fairness into account and proposed different methods to promote each separately. To this end, we improved short-term fairness by training the model on fairer data generated by modifying actions, while achieving long-term fairness by regularizing the advantage function. Moreover, we investigated three case studies of bank loans, attention allocation and epidemic control and experimental results showed the effectiveness of the algorithm.

8.2 Future Work

In this dissertation, we proposed causality-based fairness notions and various algorithms to achieve long-term fairness under distribution shift in dynamic environments. In addition, there are many other directions worth exploring and investigating.

In chapters 4 - 6, we explicitly assume the causal models are Markovian models, which means there are no hidden confounders in the models. However, in the real world, an event may involve multiple causes, both direct and indirect, and it is almost impossible to enumerate all of them. Therefore, the existence of confounding factors may be a more common scenario. The causal model used to model this scenario is called a semi-Markov model, which brings about identification issues, i.e., causal effects may not be uniquely identified from the observational data. Some existing works [119] use latent models to model this situation, and in the future, we can try to combine our work with latent models to handle a wider range of problems.

In chapter 7, our algorithm belongs to the field of online reinforcement learning, which requires continuous interaction with the environment. Training models in simulation environments is not a big issue for many tasks, such as go or chess games. However, most of the scenarios where fairness is applied are high-risk or related to human welfare, and such applications are not entirely suitable for using simulated environments and it is also difficult to construct a real simulation environment. Therefore, it is a meaningful research direction to explore how to train a fair policy using only historical data with offline learning algorithms [120, 121].

In recent years, large models [122, 123] have become a trend. Since training large models requires massive resources and time, parameter-efficient fine-tuning [124] has become an important way for ordinary researchers to conduct research. Most existing fairness learning methods require adding fairness constraints during the training process, which is not suitable for pre-trained large models. The mismatch between this demand and current methods motivates us to further investigate how to efficiently enforce fairness on large-scale models.

Bibliography

[1] S. Caton and C. Haas, "Fairness in machine learning: A survey," *ArXiv*, vol. abs/2010.04053, 2020.

[2] C. Schumann, J. Foster, N. Mattei, and J. Dickerson, "We need fairness and explainability in algorithmic hiring," in *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2020.

[3] L. Zhang, Y. Wu, and X. Wu, "Achieving non-discrimination in data release," *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017.

[4] M. S. A. Lee and L. Floridi, "Algorithmic fairness in mortgage lending: from absolute conditions to relational trade-offs," *Minds and Machines*, vol. 31, no. 1, pp. 165–191, 2021.

[5] K. D. Johnson, D. P. Foster, and R. A. Stine, "Impartial predictive modeling: Ensuring fairness in arbitrary models," *Statistical Science*, p. 1, 2016.

[6] R. S. Baker and A. Hawn, "Algorithmic bias in education," *International Journal of Artificial Intelligence in Education*, pp. 1–41, 2021.

[7] A. Beutel, J. Chen, T. Doshi, H. Qian, L. Wei, Y. Wu, L. Heldt, Z. Zhao, L. Hong, E. H. Chi *et al.*, "Fairness in recommendation ranking through pairwise comparisons," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 2212–2220.

[8] Y. Li, H. Chen, S. Xu, Y. Ge, J. Tan, S. Liu, and Y. Zhang, "Fairness in recommendation: A survey," *arXiv preprint arXiv:2205.13619*, 2022.

[9] A. Krishnan, A. Almadan, and A. Rattani, "Understanding fairness of gender classification algorithms across gender-race groups," in *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2020, pp. 1028–1035.

[10] Julia Angwin, "Machine bias: Propublica," 2016, https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing, Last accessed on 2023-1-25.

[11] 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*, vol. 50, no. 1, pp. 3–44, 2021.

[12] S. Barocas and A. D. Selbst, "Big data's disparate impact," *California law review*, pp. 671–732, 2016.

[13] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel, "Fairness through awareness," in *Proceedings of the 3rd innovations in theoretical computer science conference*, 2012, pp. 214–226.

[14] T. Calders and S. Verwer, “Three naive bayes approaches for discrimination-free classification,” *Data mining and knowledge discovery*, vol. 21, pp. 277–292, 2010.

[15] M. Hardt, E. Price, and N. Srebro, “Equality of opportunity in supervised learning,” in *Advances in neural information processing systems*, 2016, pp. 3315–3323.

[16] M. J. Kusner, J. Loftus, C. Russell, and R. Silva, “Counterfactual fairness,” in *Advances in Neural Information Processing Systems*, 2017, pp. 4066–4076.

[17] L. Zhang, Y. Wu, and X. Wu, “A causal framework for discovering and removing direct and indirect discrimination,” *arXiv preprint arXiv:1611.07509*, 2016.

[18] Y. Wu, L. Zhang, X. Wu, and H. Tong, “Pc-fairness: A unified framework for measuring causality-based fairness,” in *Advances in Neural Information Processing Systems*, 2019, pp. 3399–3409.

[19] C. Su, G.-X. Yu, J. Wang, Z. Yan, and L. Cui, “A review of causality-based fairness machine learning,” *Intelligence Robotics*, vol. 2, pp. 244–274, 08 2022.

[20] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, “A survey on bias and fairness in machine learning,” *ACM Computing Surveys (CSUR)*, vol. 54, no. 6, pp. 1–35, 2021.

[21] D. Pessach and E. Shmueli, “A review on fairness in machine learning,” *ACM Computing Surveys (CSUR)*, vol. 55, no. 3, pp. 1–44, 2022.

[22] L. Hu and Y. Chen, “A short-term intervention for long-term fairness in the labor market,” in *Proceedings of the 2018 World Wide Web Conference*, 2018, pp. 1389–1398.

[23] X. Zhang, R. Tu, Y. Liu, M. Liu, H. Kjellström, K. Zhang, and C. Zhang, “How do fair decisions fare in long-term qualification?” *arXiv preprint arXiv:2010.11300*, 2020.

[24] L. T. Liu, S. Dean, E. Rolf, M. Simchowitz, and M. Hardt, “Delayed impact of fair machine learning,” in *International Conference on Machine Learning.* PMLR, 2018, pp. 3150–3158.

[25] D. Ensign, S. A. Friedler, S. Neville, C. Scheidegger, and S. Venkatasubramanian, “Runaway feedback loops in predictive policing,” in *Conference on Fairness, Accountability and Transparency.* PMLR, 2018, pp. 160–171.

[26] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction.* MIT press, 2018.

[27] Z. Tang, J. Zhang, and K. Zhang, “What-is and how-to for fairness in machine learning: A survey, reflection, and perspective,” *arXiv preprint arXiv:2206.04101*, 2022.

[28] 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 26th international conference on world wide web*, 2017, pp. 1171–1180.

[29] J. Pearl, *Causality: Models, Reasoning and Inference*, 2nd ed. New York, NY, USA: Cambridge University Press, 2009.

[30] Y. Hu and L. Zhang, "Achieving long-term fairness in sequential decision making," *arXiv e-prints*, pp. arXiv–2204, 2022.

[31] E. Y. Yu, Z. Qin, M. K. Lee, and S. Gao, "Policy optimization with advantage regularization for long-term fairness in decision systems," *arXiv preprint arXiv:2210.12546*, 2022.

[32] F. Kamiran and T. Calders, "Data preprocessing techniques for classification without discrimination," *Knowledge and Information Systems*, vol. 33, no. 1, pp. 1–33, 2012.

[33] M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian, "Certifying and removing disparate impact," in *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 2015, pp. 259–268.

[34] D. Xu, S. Yuan, L. Zhang, and X. Wu, "Fairgan: Fairness-aware generative adversarial networks," in *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2018, pp. 570–575.

[35] D. Madras, E. Creager, T. Pitassi, and R. Zemel, "Learning adversarially fair and transferable representations," in *International Conference on Machine Learning*. PMLR, 2018, pp. 3384–3393.

[36] F. Calmon, D. Wei, B. Vinzamuri, K. Natesan Ramamurthy, and K. R. Varshney, "Optimized pre-processing for discrimination prevention," *Advances in neural information processing systems*, vol. 30, 2017.

[37] F. Kamiran, T. Calders, and M. Pechenizkiy, "Discrimination aware decision tree learning," in *2010 IEEE International Conference on Data Mining*. IEEE, 2010, pp. 869–874.

[38] Y. Wu, L. Zhang, and X. Wu, "On convexity and bounds of fairness-aware classification," in *The World Wide Web Conference*, 2019, pp. 3356–3362.

[39] D. Xu, Y. Wu, S. Yuan, L. Zhang, and X. Wu, "Achieving causal fairness through generative adversarial networks," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, 2019.

[40] B. H. Zhang, B. Lemoine, and M. Mitchell, "Mitigating unwanted biases with adversarial learning," in *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, 2018, pp. 335–340.

[41] F. Kamiran, A. Karim, and X. Zhang, "Decision theory for discrimination-aware classification," in *2012 IEEE 12th International Conference on Data Mining*. IEEE, 2012, pp. 924–929.

[42] G. Pleiss, M. Raghavan, F. Wu, J. Kleinberg, and K. Q. Weinberger, "On fairness and calibration," *Advances in neural information processing systems*, vol. 30, 2017.

[43] N. Mehrabi, U. Gupta, F. Morstatter, G. V. Steeg, and A. Galstyan, "Attributing fair decisions with attention interventions," *arXiv preprint arXiv:2109.03952*, 2021.

[44] A. K. Menon and R. C. Williamson, "The cost of fairness in binary classification," in *Conference on Fairness, accountability and transparency*. PMLR, 2018, pp. 107–118.

[45] A. Bower, S. N. Kitchen, L. Niss, M. J. Strauss, A. Vargas, and S. Venkatasubramanian, "Fair pipelines," *arXiv preprint arXiv:1707.00391*, 2017.

[46] C. Dwork and C. Ilvento, "Fairness under composition," *arXiv preprint arXiv:1806.06122*, 2018.

[47] X. Zhang and M. Liu, "Fairness in learning-based sequential decision algorithms: A survey," *arXiv preprint arXiv:2001.04861*, 2020.

[48] A. D'Amour, H. Srinivasan, J. Atwood, P. Baljekar, D. Sculley, and Y. Halpern, "Fairness is not static: deeper understanding of long term fairness via simulation studies," in *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 2020, pp. 525–534.

[49] E. Creager, D. Madras, T. Pitassi, and R. Zemel, "Causal modeling for fairness in dynamical systems," in *International Conference on Machine Learning*. PMLR, 2020, pp. 2185–2195.

[50] M. Wen, O. Bastani, and U. Topcu, "Algorithms for fairness in sequential decision making," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021, pp. 1144–1152.

[51] Y. Ge, S. Liu, R. Gao, Y. Xian, Y. Li, X. Zhao, C. Pei, F. Sun, J. Ge, W. Ou *et al.*, "Towards long-term fairness in recommendation," in *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, 2021, pp. 445–453.

[52] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[53] R. Guo, L. Cheng, J. Li, P. R. Hahn, and H. Liu, "A survey of learning causality with data: Problems and methods," *ACM Computing Surveys (CSUR)*, vol. 53, no. 4, pp. 1–37, 2020.

[54] J. Correa and E. Bareinboim, "A calculus for stochastic interventions: Causal effect identification and surrogate experiments," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 06, 2020, pp. 10 093–10 100.

[55] C. Avin, I. Shpitser, and J. Pearl, "Identifiability of path-specific effects," in *IJCAI*, 2005.

[56] I. Žliobaite, F. Kamiran, and T. Calders, "Handling conditional discrimination," in *2011 IEEE 11th International Conference on Data Mining*. IEEE, 2011, pp. 992–1001.

[57] D. Pedreshi, S. Ruggieri, and F. Turini, "Discrimination-aware data mining," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008, pp. 560–568.

[58] A. N. Carey and X. Wu, "The causal fairness field guide: Perspectives from social and formal sciences," *Frontiers in Big Data*, vol. 5, 2022.

[59] A. Agarwal, A. Beygelzimer, M. Dudik, J. Langford, and H. Wallach, "A reductions approach to fair classification," in *International Conference on Machine Learning*, 2018, pp. 60–69.

[60] M. B. Zafar, I. Valera, M. Rodriguez, K. Gummadi, and A. Weller, "From parity to preference-based notions of fairness in classification," in *Advances in Neural Information Processing Systems*, 2017, pp. 229–239.

[61] M. B. Zafar, I. Valera, M. Gomez-Rodriguez, and K. P. Gummadi, "Fairness constraints: A flexible approach for fair classification." *Journal of Machine Learning Research*, vol. 20, no. 75, pp. 1–42, 2019.

[62] L. E. Celis, L. Huang, V. Keswani, and N. K. Vishnoi, "Classification with fairness constraints: A meta-algorithm with provable guarantees," in *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 2019, pp. 319–328.

[63] L. Huang and N. Vishnoi, "Stable and fair classification," in *International Conference on Machine Learning*, 2019, pp. 2879–2890.

[64] L. Zhang, Y. Wu, and X. Wu, "A causal framework for discovering and removing direct and indirect discrimination," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. AAAI Press, 2017, pp. 3929–3935.

[65] J. Zhang and E. Bareinboim, "Fairness in decision-making—the causal explanation formula," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[66] R. Nabi and I. Shpitser, "Fair inference on outcomes," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[67] Y. Wu, L. Zhang, and X. Wu, "Counterfactual fairness: Unidentification, bound and algorithm," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI*, 2019, pp. 10–16.

[68] J. Zhang and E. Bareinboim, "Equality of opportunity in classification: A causal approach," in *Advances in Neural Information Processing Systems*, 2018, pp. 3671–3681.

[69] M. Kocaoglu, A. Jaber, K. Shanmugam, and E. Bareinboim, "Characterization and learning of causal graphs with latent variables from soft interventions," in *Advances in Neural Information Processing Systems*, 2019, pp. 14 346–14 356.

[70] P. L. Bartlett, M. I. Jordan, and J. D. McAuliffe, "Convexity, classification, and risk bounds," *Journal of the American Statistical Association*, vol. 101, no. 473, pp. 138–156, 2006.

[71] P. Spirtes and K. Zhang, "Causal discovery and inference: concepts and recent methodological advances," in *Applied informatics*, vol. 3, no. 1. Springer, 2016, p. 3.

[72] M. Lichman, "UCI Machine Learning Repository," http://archive.ics.uci.edu/ml, 2013.

[73] R. Scheines, P. Spirtes, C. Glymour, C. Meek, and T. Richardson, "The TETRAD Project: Constraint Based Aids to Causal Model Specification," *Multivariate Behavioral Research*, vol. 33, no. 1, pp. 65–117, Jan. 1998.

[74] G. Lemaitre, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning," *Journal of Machine Learning Research*, vol. 18, pp. 17:1–17:5, 2017.

[75] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," *Journal of Machine Learning Research*, vol. 17, no. 83, pp. 1–5, 2016.

[76] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. 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 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[77] X. Chen, S. Liu, R. Sun, and M. Hong, "On the convergence of a class of adam-type algorithms for non-convex optimization," in *7th International Conference on Learning Representations, ICLR 2019*, 2019.

[78] R. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork, "Learning fair representations," in *International conference on machine learning.* PMLR, 2013, pp. 325–333.

[79] M. Hardt, E. Price, and N. Srebro, "Equality of opportunity in supervised learning," in *NIPS*, 2016.

[80] M. J. Kusner, J. R. Loftus, C. Russell, and R. Silva, "Counterfactual fairness," in *NIPS*, 2017.

[81] S. Kannan, A. Roth, and J. Ziani, "Downstream effects of affirmative action," in *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 2019, pp. 240–248.

[82] J. Runge, P. Nowack, M. Kretschmer, S. Flaxman, and D. Sejdinovic, "Detecting and quantifying causal associations in large nonlinear time series datasets," *Science Advances*, vol. 5, no. 11, p. eaau4996, 2019.

[83] J. Perdomo, T. Zrnic, C. Mendler-Dünner, and M. Hardt, "Performative prediction," in *International Conference on Machine Learning.* PMLR, 2020, pp. 7599–7609.

[84] Y. Hu, Y. Wu, L. Zhang, and X. Wu, "Fair multiple decision making through soft interventions," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[85] J. Runge, "Causal network reconstruction from time series: From theoretical assumptions to practical estimation," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 28, no. 7, p. 075310, 2018.

[86] ——, "Conditional independence testing based on a nearest-neighbor estimator of conditional mutual information," in *International Conference on Artificial Intelligence and Statistics.* PMLR, 2018, pp. 938–947.

[87] ——, "Discovering contemporaneous and lagged causal relations in autocorrelated nonlinear time series datasets," in *Conference on Uncertainty in Artificial Intelligence.* PMLR, 2020, pp. 1388–1397.

[88] R. Pamfil, N. Sriwattanaworachai, S. Desai, P. Pilgerstorfer, K. Georgatzis, P. Beaumont, and B. Aragam, "Dynotears: Structure learning from time-series data," in *International Conference on Artificial Intelligence and Statistics.* PMLR, 2020, pp. 1595–1605.

[89] S. Löwe, D. Madras, R. Zemel, and M. Welling, "Amortized causal discovery: Learning to infer causal graphs from time-series data," *arXiv preprint arXiv:2006.10833*, 2020.

[90] C. Avin, I. Shpitser, and J. Pearl, "Identifiability of path-specific effects," in *Proceedings of the 19th international joint conference on Artificial intelligence*, 2005, pp. 357–363.

[91] I.-C. Yeh and C.-h. Lien, "The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients," *Expert Systems with Applications*, vol. 36, no. 2, pp. 2473–2480, 2009.

[92] C. Esteban, S. L. Hyland, and G. Rätsch, "Real-valued (medical) time series generation with recurrent conditional gans," *arXiv preprint arXiv:1706.02633*, 2017.

[93] J. Pearl, "Causal inference." *NIPS Causality: Objectives and Assessment*, pp. 39–58, 2010.

[94] M. J. Vowels, N. C. Camgoz, and R. Bowden, "D'ya like dags? a survey on structure learning and causal discovery," *arXiv preprint arXiv:2103.02582*, 2021.

[95] C. Villani, *Topics in optimal transportation.* American Mathematical Soc., 2021, vol. 58.

[96] E. M. Maltz and F. H. Miller, "The equal credit opportunity act and regulation b," *Okla. L. Rev.*, vol. 31, p. 1, 1978.

[97] M. A. Bakker, A. Noriega-Campero, D. P. Tu, P. Sattigeri, K. R. Varshney, and A. Pentland, "On fairness in budget-constrained decision making," in *Proceedings of the KDD Workshop on Explainable Artificial Intelligence*, 2019.

[98] S. Corbett-Davies, E. Pierson, A. Feller, S. Goel, and A. Huq, "Algorithmic decision making and the cost of fairness," in *Proceedings of the 23rd acm sigkdd international conference on knowledge discovery and data mining*, 2017, pp. 797–806.

[99] M. Kocaoglu, C. Snyder, A. G. Dimakis, and S. Vishwanath, "Causalgan: Learning causal implicit generative models with adversarial training," in *International Conference on Learning Representations*, 2018.

[100] J. Peters, D. Janzing, and B. Schölkopf, *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.

[101] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[102] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola, "A kernel method for the two-sample-problem," *Advances in neural information processing systems*, vol. 19, 2006.

[103] T. Le Quy, A. Roy, V. Iosifidis, W. Zhang, and E. Ntoutsi, "A survey on datasets for fairness-aware machine learning," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, p. e1452, 2022.

[104] F. Ding, M. Hardt, J. Miller, and L. Schmidt, "Retiring adult: New datasets for fair machine learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 6478–6490, 2021.

[105] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

[106] L. T. Liu, S. Dean, E. Rolf, M. Simchowitz, and M. Hardt, "Delayed impact of fair machine learning," in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, ser. Proceedings of Machine Learning Research, J. G. Dy and A. Krause, Eds., vol. 80. PMLR, 2018, pp. 3156–3164.

[107] S. Jabbari, M. Joseph, M. Kearns, J. Morgenstern, and A. Roth, "Fairness in reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1617–1626.

[108] J. Atwood, H. Srinivasan, Y. Halpern, and D. Sculley, "Fair treatment allocations in social networks," *arXiv preprint arXiv:1911.05489*, 2019.

[109] F. Kamiran and T. Calders, "Classifying without discriminating," in *2009 2nd international conference on computer, control and communication.* IEEE, 2009, pp. 1–6.

[110] N. Quadrianto and V. Sharmanska, "Recycling privileged learning and distribution matching for fairness," *Advances in neural information processing systems*, vol. 30, 2017.

[111] M. P. Kim, A. Korolova, G. N. Rothblum, and G. Yona, "Preference-informed fairness," *arXiv preprint arXiv:1904.01793*, 2019.

[112] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," *arXiv preprint arXiv:1506.02438*, 2015.

[113] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," *Advances in neural information processing systems*, vol. 12, 1999.

[114] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[115] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International conference on machine learning.* PMLR, 2015, pp. 1889–1897.

[116] S. Barocas, M. Hardt, and A. Narayanan, "Fairness in machine learning," *Nips tutorial*, vol. 1, p. 2017, 2017.

[117] G. Alves, F. Bernier, M. Couceiro, K. Makhlouf, C. Palamidessi, and S. Zhioua, "Survey on fairness notions and related tensions," *EURO Journal on Decision Processes*, p. 100033, 2023.

[118] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proceedings of the national academy of sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[119] C. Louizos, U. Shalit, J. M. Mooij, D. Sontag, R. Zemel, and M. Welling, "Causal effect inference with deep latent-variable models," *Advances in neural information processing systems*, vol. 30, 2017.

[120] S. Levine, A. Kumar, G. Tucker, and J. Fu, "Offline reinforcement learning: Tutorial, review, and perspectives on open problems," *arXiv preprint arXiv:2005.01643*, 2020.

[121] R. F. Prudencio, M. R. Maximo, and E. L. Colombini, "A survey on offline reinforcement learning: Taxonomy, review, and open problems," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[122] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[123] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[124] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.