Strategic and counterfactual reasoning in AI-assisted decision making

Thesis approved by
the Department of Computer Science
University of Kaiserslautern-Landau
for the award of the Doctoral Degree
Doctor of Natural Sciences (Dr. rer. nat.)

to

Efstratios (Stratis) Tsirtsis

Date of Defense:	19 August 2025
Dean:	Prof. Dr. Christoph Garth
Reviewer:	Prof. Dr. Tobias Gerstenberg
Reviewer:	Dr. Manuel Gomez-Rodriguez
Reviewer:	Prof. Dr. Rupak Majumdar

DE-386

Summary

From finance and healthcare to criminal justice and transportation, various domains that involve critical decisions, traditionally made by humans, are increasingly incorporating artificial intelligence (AI) systems into their decision making processes. Modern AI systems excel at processing vast amounts of data and solving complex problems at a speed and scale unimaginable for humans. However, complete automation of high-stakes decisions is often undesirable due to legal, ethical, and societal concerns. A promising approach, which has attracted significant attention in the machine learning literature, lies in human-AI collaboration: decision making pipelines that leverage the computational strengths of AI systems to enhance the overall quality of decisions while maintaining a degree of human control. In this context, I focus on AI-assisted decision making scenarios characterized by complexity and uncertainty, specifically requiring strategic reasoning about others' actions and counterfactual reasoning about alternatives to past decisions.

First, I focus on strategic reasoning and introduce methods based on game-theoretic modeling to support policy design in strategic environments. These methods enable a decision maker in a resource allocation scenario to design policies, informed by a predictive model, that maximize their utility while accounting for strategic responses from individuals who gain knowledge about the policy and aim to receive a beneficial decision. I provide algorithms for two distinct scenarios with varying levels of information available to individuals: a fully transparent scenario where the policy is disclosed and a partially transparent scenario where the decision maker provides actionable recommendations to individuals rejected by the policy.

Then, I shift focus to counterfactual reasoning and develop methods based on causal modeling to enhance the counterfactual reasoning capabilities of a human decision maker in a sequential decision making task. These methods aim to improve the decision maker's learning process from past experiences by identifying critical time steps where different actions could have led to better outcomes. Specifically, I consider settings where a decision maker observes the state of the environment over time and takes a series of interdependent actions that result in an observed outcome. For both discrete and continuous states, I formalize the problem of finding alternative action sequences, close to the observed one, that would have achieved a better counterfactual outcome, and I provide efficient algorithmic solutions.

Finally, I investigate how people perceive responsibility in human-AI teams. In this context, I propose a computational model based on counterfactual simulations to predict how an external observer attributes responsibility to a human and an AI agent collaborating towards a common goal. To evaluate the model's predictions, I develop a simulation environment that generates stylized instances of sequential human-AI collaboration and conduct a human-subject study in which participants make responsibility judgments about the two agents.

Acknowledgements

Technically speaking, this thesis marks the end of my PhD studies. However, I cannot help but think of it as the end of a much longer journey that started more than a decade ago, when my interaction with computer science began. For that reason, I take the opportunity to use these pages to engage in my own counterfactual thoughts, and to express my gratitude to the people without whom this journey would not have been as exciting and unique as it was.

First and foremost, I would like to thank my advisor, Manuel, for his continuous guidance and support throughout my PhD. It is fair to say that my professional and personal growth during these years is largely due to his mentorship. His selective research taste, his high standards for clarity and rigor, and his motivation to pursue ambitious goals have all contributed to shaping me into an independent researcher. I am truly grateful for all our chats, whether in the office or during our forest runs.

I would also like to thank my committee members, Marius Kloft, Andreas Krause, and Rupak Majumdar, for the time they invested in reading my thesis and for the insightful discussion we had during my defense. Their support during the last stages of my PhD was invaluable. I am also grateful to Adish Singla and Krishna Gummadi for evaluating my progress in the earlier stages of my PhD and for having multiple interesting conversations with me throughout these years.

A special mention goes to Tobi Gerstenberg, not only for being a reviewer of my thesis, but also for hosting me in his lab during my research visit at the Department of Psychology at Stanford University. The time I spent with his group of "causemonauts" gave me a unique opportunity to expand my research horizons beyond the boundaries of computer science. I am deeply grateful for his kindness, his willingness to guide me into a new research area, and his support thereafter.

Reflecting on my time at the Max Planck Institute for Software Systems, I feel exceptionally fortunate, and that is largely due to its people. A PhD is a long and often lonely process, but the word "lonely" seems to be an alien concept at this institute. There is always someone willing to help you practice a talk, brainstorm a research idea, or simply share a few human moments. If something will stay with me forever, it is all the after-lunch sweets, the bike rides, the game nights, and the absurdly crowded dinners that transformed an international group of people into a large family. No matter whether our paths crossed briefly or we became close friends, each person gave me something to remember. Abir, Adrian, Alexandra, Aman, Amir, Ana, Ander, Arabinda, Ari, Ashwani, Bala, Cedric, Chris, Clothilde, Corto, Dimitri, Eirini, Eleni, Ellen, Faezeh, Felix, Filip, Germano, Giovanni, Hasan, Iasona, Irmak, Ivan G., Ivi, James, Kaushik, Kilian, Kimaya, Leo, Lia, Luke, Mahmoud, Maitreyi, Marin, Marko, Mehrdad, Michali, Munko, Nastaran, Numair, Pascal, Pavel, Rajarshi, Ram, Richard, Ritam, Rosa, Sathiya, Satya, Seungeon, Srinidhi, Stanly, Stelio, Suhas, Utkarsh, Xuan, and Yugesh, thank you all for the wonderful moments

we shared together.

I would also like to thank all the non-research employees of the MPI for Software Systems who are the backbone of the institute. They have always been extremely helpful, sometimes even going beyond their duties to make sure that we all had everything we needed to do our research without worries. I would like to particularly thank Andreas, Carina, Geraldine, Gretchen, Lisa, Mary-Lou, Rose, Susanne, Tobias, and Vera who are the ones I mostly bothered with my endless questions.

At the MPI for Software Systems, I was also fortunate to meet a special person, Nina, with whom I have been the closest throughout these years. Being both in office 613, we have always been each other's support system, both professionally and personally. We have shared each other's joys and frustrations, successes and failures, and we have had many wonderful moments, for which I am deeply grateful.

Back in 2019-2020, I was in the unfortunate position of starting my PhD right at the onset of a global pandemic. The health risks and the resulting lockdowns have been a hard experience for everyone and have made my PhD journey even more mentally challenging. However, I was lucky to have a fantastic group of friends with whom I feel that, perhaps surprisingly, I grew even closer during those two years. To my childhood friends, Giorgos & Giorgos, Grigoris, and Stratis, and to my "continental" friends, Davide, Ivan F., Marco & Marco, Pavel, Riccardo, and Simin, thank you for helping me stay sane during those difficult times.

Going further back in time, I feel the need to dedicate a few lines to Dimitris Fotakis, the algorithms professor during my undergraduate studies at the National Technical University of Athens. Looking back, I can clearly remember the day when a serendipitous decision led me to one of his passionate lectures, which restored my at-the-time fading interest in the theoretical side of computer science. It was this pivotal moment and his mentorship thereafter that ultimately set me on the path to pursue a PhD. For being the first person to make me believe that I can become a researcher, I will always be grateful.

It is needless to say that none of this would have been possible without the support of my family. I do not have enough words to thank my mother, Lena. An energetic and dedicated dentist, she has been a tireless professional whose work ethic and drive have always inspired me. I am grateful for her constant belief in me and for always being my sounding board, ensuring that my decisions were well thought out. I am equally grateful to my father, Giorgos. He was the first academic I ever met and the person who introduced me to the world of computer programming, setting me on this trajectory ever since. I am fortunate to have inherited the calmness and persistence he carries as a marathon runner. As I look at the traits that shaped this thesis, I increasingly see a blend of my parents' influences and come to understand the Greek saying that "the apple will fall under the apple tree."

I would like to conclude by thanking two people who have often been in the background but whose influence was absolutely essential for me to take on this journey. I was fortunate to have two fantastic godparents, Apostolis and Maria, who were also the best math teachers I have ever had. They taught me to love problem solving and to appreciate the beauty of abstract thinking. Crucially, they instilled in me the belief that, after visualizing the geometric shapes behind the symbols, intuition is all you need to solve even the most seemingly hard problems. Hence, every inequality, every implication, and every theorem in this thesis carries a little bit of their philosophy. For that, this thesis is dedicated to them.

To Apostolis and Maria,
for teaching me how to think.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Irene sat at the head of the table. She could sense the anticipation in the room. Today was the day the community bank would make its new loan approval criteria public—a bold step based on recommendations from Adam, their AI policy tool. Adam had analyzed years of data and predicted that the new criteria would create incentives for applicants to reduce debts and increase savings, leading to a 5% rise in approvals and a 10% drop in defaults. The team had spent weeks preparing for this moment. Would applicants embrace the transparency and act in the way Adam predicted? Irene stood up to address her colleagues. "This is a risky step forward." she said, her voice steady. "But it's one worth taking—for the bank and for our community." With that, the new webpage went live, and the criteria were made public. Within months, approval rates increased, defaults dropped, and the local economy began to thrive. Irene felt a sense of pride and relief. The decision had paid off.

...

The hospital's conference room was dimly lit, the sound of ventilators humming in the background. Ben's eyes were scanning the list of patients, hopelessly trying to make sense of the situation. Despite his team's best efforts, a troubling trend persisted: certain patients weren't responding to the current treatment protocols. Ben couldn't shake the feeling that there was something they were missing—some small adjustments that could save lives. Desperate for a breakthrough, he decided to consult Marie, the hospital's AI assistant, for additional insights. Marie's response was immediate: "See Cases 17 and 42 from last month, earlier use of steroids may have prevented deterioration. The hand written notes may contain more details." Ben pulled up the patients' records. Marie was right—both patients had mentioned a history of asthma. Ben felt a glimmer of hope as he made a note to administer early steroids to asthma patients. Maybe, just maybe, this could turn things around.

...

The courtroom was still as the judge repeated the question that had loomed over the entire trial: "So, who is responsible for the accident?" The case involved a semi-autonomous car that had went off the road to avoid hitting a pedestrian, crashing into a storefront in the process. The car's AI had detected the pedestrian and, calculating the risks, prompted the human driver to take control, suggesting that swerving was the best option. Julia, the human behind the wheel, had followed the AI's suggestion mere milliseconds before the collision. The pedestrian was grateful to Julia for saving their life, but the store owner was furious at the AI for the destruction of their property. The jury sat in silence, struggling to answer the question: who deserved the credit, or the blame, for the life that was saved and the damage that followed?

1.1 Decisions, humans, and machines

The 21st century has seen rapid improvement in the capabilities of artificial intelligence (AI) systems. Technical advances in statistical machine learning along with a vast availability of computational resources have enabled the development of AI systems that can perform tasks previously thought to be exclusive to humans. For example, AI systems can now diagnose diseases [1, 2], drive cars [3, 4], and make accurate financial predictions [5, 6]. Although their performance and their potential to revolutionize our lives are improving with an unprecedented pace, their use is no panacea. For example, AI systems run the risk of being overconfident in their predictions [7], presenting bias against minorities [8] or performing poorly when deployed in a different environment than the one in which they were trained [9]. These concerns have led to growing discussions on the extent to which AI should be used to automate consequential decisions in critical domains such as healthcare [10], hiring [11] and criminal justice [12]. Responding to these concerns, legislative initiatives, such as the General Data Protection Regulation (GDPR) of the European Union, have established safeguards against decisions that are completely automated [13].

One might wonder why the use of AI systems in decision making has gained traction despite the associated risks. A seemingly natural response to the concerns raised above would be to avoid the use of AI systems altogether in high-stakes situations and, as is common practice in many application domains, rely entirely on human judgment for critical decisions. Although, at first glance, this may seem like a safer alternative, there is significant evidence suggesting that human decision making is also far from perfect. For example, issues such as overconfidence in one's estimates and implicit bias against people of certain demographics have been well documented in multiple domains, including the ones mentioned previously, where decisions made by humans are the norm [14–18].

The nature of human decisions has been a central theme of economic research for decades. At the core of classical economic theory lies the concept of *homo economicus* [19, 20]—the assumption that humans evaluate their choices based on the (expected) utility each option offers and consistently choose the one that maximizes their utility. However, this assumption has been challenged by a large body of research in psychology and behavioral economics [21–24]. Human decisions have often been found to deviate from the principles of utility maximization, influenced by various factors, such as emotions [25] and social interactions [26]. Moreover, substantial empirical evidence indicates that human decisions under uncertainty often result in utility that differs from the normative benchmark—the maximum expected value—due to cognitive biases in processing probabilities [27].

Recent work in cognitive science offers another, more computational, perspective on why human decision making frequently deviates from optimality. The resource-rational analysis approach suggests that imperfect decisions arise due to cognitive processes confined by the biological constraints of the human brain [28–30], which presents a trade-off between maximizing utility and reducing the cognitive costs of computation. This perspective aligns closely with ideas in computational complexity [31, 32], where problems of increased complexity are often addressed with approximation algorithms due to the difficulty of finding exact solutions given limited computational resources (*i.e.*, time and memory) [33].

In this context, it is sensible to argue that high-stakes decisions are precisely the

type of decision where AI systems may be most beneficial, if not essential. This is because these decisions are typically characterized by the combination of two factors that, as discussed earlier, limit human capability: uncertainty and complexity. On the other hand, modern AI systems can efficiently leverage vast amounts of data to make (probabilistic) predictions, often surpassing human experts in accuracy [34, 35]. Furthermore, years of research in mathematical optimization have produced a rich toolbox of computational methods capable of solving even the most challenging problems using sophisticated algorithms and modern hardware [36–40].

This state of affairs presents a clear trade-off between the computational advantages of integrating AI into decision making processes and the associated risks and ethical concerns when human oversight is absent. Consequently, there is a growing interest in the potential of *human-AI teams* to improve decision making in domains that are critical, uncertain, and highly complex. The main research objective in the area of human-AI collaboration is to leverage the computational power of AI systems in ways that enhance the overall quality of decisions while maintaining a degree of human control throughout the process.

Before proceeding further, it is essential to clarify the scope of this thesis. AI systems that collaborate with humans can take various forms and appear in numerous application domains. For example, modern robotic systems increasingly rely on data-driven learning algorithms rather than traditional control theory, leading to a form of physical human-AI collaboration [41, 42]. In addition, AI systems with natural language processing capabilities are widely used to assist humans with tasks such as language learning [43–45] and collaborative writing [46]. Refraining from an exhaustive discussion of all forms and application domains, I use the term "human-AI collaboration" to refer to scenarios where a human is tasked with a decision making problem and the AI serves the role of *decision support*. Moreover, I use the term "AI" to refer to a technological entity that has the capacity to (i) make *predictions* using machine learning, and (ii) find the right *decisions* to make or recommend using optimization algorithms [47–50], while abstracting details related to its physical form and mode of communication with the human.

Within the machine learning literature, one can identify two main paradigms for decision support. First, a human decision maker can maintain complete control over the decision making process, using the AI as a *tool to inform their decisions* [51–56]. For example, a clinician may use an AI system that predicts a patient's risk of developing a disease and recommends a treatment plan. In turn, the clinician may choose to follow the recommendation, modify it, or completely ignore it. Second, a human can share the decision making process with the AI, *deferring some decisions* to it, while making the rest themselves [57–63]. For example, a pilot may set a destination and let an AI system navigate the aircraft based on the weather conditions and air traffic but take control when the plane is about to land.

The AI-assisted decision making settings discussed in this thesis broadly align with these two paradigms. However, these paradigms typically involve reasoning under uncertainty about future outcomes, as in common machine learning tasks such as classification and regression. This thesis focuses on decisions and judgments that are inherently more complex, arising in scenarios that require strategic or counterfactual (*i.e.*, retrospective) reasoning. Such situations demand an understanding not just of the (distribution of) outcomes but also of the underlying dynamics that shape them. Next, I discuss these two types of reasoning in greater detail.

1.2 Strategic and counterfactual reasoning

In many real-world scenarios, human decisions and judgments are based on two key reasoning processes: strategic (*i.e.*, reasoning about others' actions) and counterfactual (*i.e.*, reasoning about alternatives to past events). This thesis focuses on (i) how AI can enhance those two types of reasoning to assist human decision makers, and (ii) how counterfactual reasoning can serve as a computational tool for understanding (human) responsibility judgments in human-AI collaboration.

Strategic reasoning is involved in the process of making (utility-maximizing) decisions in the presence of others who also seek to maximize their own objectives. In such settings, the uncertainty about the outcome of the decision making process arises from a complex interplay between the inherent randomness of the environment and the actions of others, strategically responding to the decision maker's actions. For example, consider a government that decides to increase the value added tax (VAT) on certain goods with the aim of boosting tax revenue. The government must anticipate that consumers may respond by reducing their consumption of those goods, potentially leading to a decrease in tax revenue instead. From a computational perspective, such strategic interactions are studied in (algorithmic) game theory [64], where determining optimal strategies has often been found to be intractable [65, 66]. This complexity casts doubt on the ability of humans, with their limited cognitive resources, to act optimally in strategic environments, especially in the presence of a large and diverse population of other humans. Consequently, it highlights the potential of AI systems to support human decisions in such contexts.

Counterfactual reasoning plays a critical role in retrospectively analyzing past decisions; it involves evaluating what the outcome of a decision making process would have been had the decision maker acted differently in the past. Research in psychology suggests that this type of reasoning plays an important role in the process of generating explanations about events, learning from past experience, and planning future actions [67–69]. However, reasoning counterfactually is particularly challenging, as it requires the decision maker to maintain a (mental) model of the world that captures its causal structure, and perform simulations of alternative scenarios to evaluate the impact of different decisions [70]. This complexity is amplified in sequential decision making tasks, where the decision maker has to mentally undo multiple combinations of past decisions to identify those that could have led to better outcomes. Such challeges highlight the potential of AI systems, grounded in the theory of causal inference [71], to support humans in evaluating counterfactual actions and outcomes, providing a valuable learning signal for their future decisions.

Finally, to build trustworthy AI systems that assist humans in decision making, an essential prerequisite is to understand how people assign responsibility in human-AI collaboration scenarios. Counterfactual reasoning is a key component of this process, as responsibility judgments in such contexts often involve counterfactual questions, such as "What would have happened if the AI had not intervened?" The connection between counterfactual reasoning and responsibility is well established in the literature, however, in the context of collaborative decisions and outcomes, it has focused primarily on how humans hold other humans responsible [72–74]. The increasing development of AI systems that assist and collaborate with humans, rather than replacing them [57, 60–62, 75–78] presents a need to extend this line of research to the human-AI collaboration domain [79].

1.3 Contributions and outline

In the next chapters of the thesis, I make contributions that aim to address (some of) the challenges discussed in the previous section. In the remainder of this chapter, I provide an overview of prior related work and a list of the publications that form the core of the thesis. In Chapter 2, I introduce the technical concepts and frameworks that are used throughout the rest of the thesis. The subsequent chapters are then organized as follows:

1. In Chapter 3, I introduce algorithmic methods to *support policy design in strategic environments*. I consider settings where a decision maker (informed by a predictive model) has to design a policy that allocates resources to a population of individuals under the assumption that each individual may strategically respond to the policy to receive a beneficial decision. For example, if a university discloses the criteria they use for graduate admissions, student applicants may invest effort to improve their applications to be admitted. The methods I propose compute policies that maximize the expected utility of the decision maker (*i.e.*, the university) in such strategic settings, while also incentivizing individuals (*i.e.*, students) to invest in forms of effort that help them self-improve.

2. In Chapter 4, I develop algorithmic methods to *enhance the counterfactual reasoning capabilities* of a human decision maker in sequential decision making tasks. For example, consider a clinician who wants to identify past cases of patients whose condition may have improved had they followed a different treatment plan and closely analyze those cases to inform their future treatment decisions. The methods I propose in this chapter address the problem of searching for action sequences (*i.e.*, sequences of treatments) that, in retrospect, would have led a given episode of the decision making process (*i.e.*, a patient) to a better outcome.

3. In Chapter 5, I explore how humans reason about *responsibility in human-AI teams*. Specifically, I consider settings where a human and an AI agent work together towards a common goal, and I propose a computational model that relies on counterfactual simulations to predict and understand how an external (human) observer assigns responsibility to each agent for the collaborative outcome. To this end, I develop a simulation environment generating instances of human-AI collaboration and use it to conduct a human-subject study to evaluate the model's performance.

I conclude the thesis with Chapter 6, which contains a general discussion highlighting key takeaways and promising directions for future research.

1.4 Related work

This section provides a brief overview of prior work along four directions relevant to the main chapters of this thesis: (i) strategic machine learning, (ii) explainable machine learning, (iii) causal and counterfactual reasoning in sequential decision making, and (iv) responsibility attribution in teams.

Strategic machine learning. Developing predictive models that remain robust against adversarial distribution shifts has received significant attention in the machine learning literature [80]. In this context, the increasing use of predictive models in high-stakes decision making has inspired a line of work on strategic classification [81–91]. In strategic classification, individuals subject to the model's predictions can manipulate their features to receive favorable predictions, while maintaining their original label. By taking into consideration the specific structure of the individuals' incentives, one can anticipate the form that the distribution shift takes, and under certain technical conditions, design predictive models that remain resistant to misclassification errors resulting from such strategic behavior. Research in this area has also explored additional aspects of strategic classification, including fairness concerns [92–94] and different assumptions about the level of information available to individuals and the designer of the predictive model [95, 96].

An adjacent line of work that generalizes strategic classification is that on performative prediction [97–100]. This field studies the stable points that arise when a classifier is repeatedly retrained under distribution shifts caused by its own predictions. However, both strategic classification and performative prediction do not explicitly distinguish between predictions and decisions—an essential distinction in AI-assisted decision making pipelines—and focus solely on maximizing predictive accuracy rather than a decision maker's utility, which is the central focus of Chapter 3. Moreover, their technical assumptions differ significantly from those made in this chapter, making the technical contributions orthogonal.

A more closely related area is the one that focuses on incentive-aware evaluation mechanisms [101–103]. These works aim to design scoring rules that incentivize individuals to invest effort in specific actions (*e.g.*, grading schemes that incentivize studying against copying homework assignments). However, in the modeling frameworks proposed in these works, the decision maker uses neither predictions about the outcome of each individual nor the feature distribution of a population to design their decision policy. Therefore, these approaches are less applicable to AI-assisted decision making contexts involving predictions by a machine learning model.

Explainable machine learning. An important aspect of trustworthy machine learning is the ability to understand the predictions of a model. Although one may favor inherently interpretable models, such as linear models or decision trees, there has been significant interest in developing post-hoc methods for explaining predictions of complex machine learning models, such as neural networks. One such approach is generating feature-based explanations [104–106]. Feature-based explanations help individuals understand the importance of each feature in a particular prediction. Typically, these approaches create an easily explainable local approximation of the model (*e.g.*, linear) to assign weights to individual features.

While there is no consensus in the literature on what constitutes a good post-hoc explanation, a second type that is gaining prominence is that of counterfactual explanations [107–110]. This type of explanation is the focus of Section 3.2 in this thesis. The goal of counterfactual explanations is to identify minimal changes in feature values that would be sufficient for a predictive model to change its prediction for a given sample. These explanations have gained traction because they place no constraints on model complexity, do not require model disclosure, facilitate actionable recourse, and seem to automate compliance with the law [111].

The technical challenges in the field of counterfactual explanations primarily in-

volve satisfying several desiderata. For example, Mothilal et al. [110] emphasize giving options to the explainee by providing sets of explanations that are diverse in terms of features, Ustun et al. [108] focus on finding explanations that are based on actionable feature changes, while Karimi et al. [112] prioritize explanations that are faithful to the underlying causal structure of the world. In contrast, the work presented in Section 3.2 focuses on how a decision maker, using predictions from a predictive model, should provide counterfactual explanations to individuals who received a negative decision from their decision policy. The goal is to maximize the decision maker's utility when individuals respond strategically to the explanations by adapting their features. Since existing work on counterfactual explanations typically focuses solely on explaining predictions without distinguishing between model predictions and decisions, they are not suited to AI-assisted decision making scenarios discussed in Chapter 3 and Section 3.2, in particular. For a comprehensive discussion on counterfactual explanations of predictive models, refer to Karimi et al. [113] and Verma et al. [114].

Causal and counterfactual reasoning in sequential decision making. The field of causal inference has a rich and interdisciplinary history. Economists have developed and extensively studied the potential outcomes framework [115], creating a comprehensive toolkit for designing experimental setups and identifying quantities such as average and conditional treatment effects. Computer scientists have focused on formally expressing causality through graph diagrams and structural equations [71]. This has led to the development of the structural causal model (SCM) framework, which, based on a calculus of interventions [116], allows to reason formally about different types of probability distributions, such as observational, interventional, and counterfactual distributions (see Section 2.3 for further details). Moreover, it is increasingly being integrated into the development and analysis of machine learning models (refer to Peters et al. [117] for an overview). The SCM framework, with a particular emphasis on counterfactual distributions, is central to Chapter 4.

In the context of sequential decision making, causal modeling has been a central component of the broad area of causal reinforcement learning [118]. This term refers to a line of work that aims to design policies for sequential decision making tasks, as in traditional reinforcement learning [119], but it incorporates additional structural assumptions about the environment. This approach serves multiple purposes, from achieving robust performance guarantees across different environments [120–122] and similar yet distinct decision making tasks [123, 124], to maintaining strong performance in the presence of unobserved variables [125–129].

In Chapter 4, our main goal is to identify an action sequence for an observed episode of a sequential decision making process that differs minimally from the original sequence and would have led to a better outcome in retrospect. Therefore, within the aforementioned line of work, the most closely related work is the one that focuses on the development of machine learning methods that employ elements of counterfactual reasoning to improve or retrospectively analyze decisions in sequential settings [130–132]. These works primarily focus on expressing the decision making task's environment as a causal model and using that information to evaluate and efficiently compute decision policies based on counterfactual realizations of logged episodes. Moreover, the work in Chapter 4 has ties to prior work that uses counterfactual reasoning to develop explainable reinforcement learning models [133, 134].

However, none of the aforementioned works aim to find an action sequence, close to the observed sequence of a particular episode, that is counterfactually optimal to support the learning process of a human decision maker.

Responsibility attribution in teams. The study of responsibility attribution is central in both AI and human psychology. Prior work has established strong connections between human perceptions of responsibility and cognitive processes such as causal and counterfactual reasoning [72, 74, 135–143]. Specifically, in team contexts, research has focused on identifying conditions under which one *is* or *should* be held responsible for a collaborative outcome. For example, there has been empirical evidence in psychology that responsibility judgments about a member in a team are influenced by factors such as pivotality [73, 74] (*i.e.*, the extent to which an individual's actions were critical for the outcome) and replaceability [143] (*i.e.*, how easily an individual could have been substituted). Moreover, in AI, prior work has proposed normative frameworks, based on structural causal models, that provide definitions of when an AI system (or agent, more generally) should be held responsible, based on whether it was an actual cause of the collaborative outcome [139, 144].

Although this line of work forms the basis for Chapter 5, it does not explicitly distinguish between human and AI team members, nor does it focus on the differences in how each agent's responsibility is perceived. That said, the work in this chapter is closely related to recent work investigating responsibility and related concepts in the context of AI-assisted decision making. For example, Awad et al. [145] study a scenario of shared control between a human and an AI in a vehicle, finding that the AI agent is consistently blamed less than humans when both make mistakes. Lima et al. [146] explore a setting of AI-assisted bail decision making and find that people hold human decision makers responsible in the sense of having an obligation or authority to make a decision, whereas AI is perceived as responsible in terms of being praised or blamed for specific decisions and outcomes. However, none of these works focus on identifying the cognitive processes that underlie responsibility judgments for humans and AI agents in collaborative settings, which is the objective of Chapter 5.

1.5 Publications

The work presented in Chapters 3, 4 and 5 has been peer-reviewed and published in top-tier venues for machine learning, operations research, and cognitive science. Details of the publication titles, venues, and co-authors are outlined below:

1. Stratis Tsirtsis, Behzad Tabibian, Moein Khajehnejad, Adish Singla, Bernhard Schölkopf, and Manuel Gomez-Rodriguez. Optimal decision making under strategic behavior. Published in *Management Science*, volume 70, issue 12, pages 8506–8519, 2024.

2. Stratis Tsirtsis and Manuel Gomez-Rodriguez. Decisions, counterfactual explanations and strategic behavior. Published in *Advances in Neural Information Processing Systems*, volume 33, pages 16749–16760, 2020.

3. Stratis Tsirtsis, Abir De, and Manuel Gomez-Rodriguez. Counterfactual explanations in sequential decision making under uncertainty. Published in *Ad-*

vances in Neural Information Processing Systems, volume 34, pages 30127–30139, 2021.

4. Stratis Tsirtsis and Manuel Gomez-Rodriguez. Finding counterfactually optimal action sequences in continuous state spaces. Published in *Advances in Neural Information Processing Systems*, volume 36, pages 3220–3247, 2023.

5. Stratis Tsirtsis, Manuel Gomez-Rodriguez, and Tobias Gerstenberg. Towards a computational model of responsibility judgments in sequential human-AI collaboration. Published in *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 46, pages 1039-1046, 2024.

I can be considered the main contributor for all publications except (1), in which I share equal contribution with Behzad Tabibian and Moein Khajehnejad. Behzad and Moein contributed to the development of a preliminary version of the modeling framework and the theoretical results in Section 3.1, specifically concerning Theorem 3.1.1, Algorithm 2, and Proposition 3.1.4. The remaining theoretical analysis, extensive experimental evaluation of the methods, and general refinement of the work through multiple rounds of peer review leading to the journal publication are my own contributions. However, it is important to note that all of the aforementioned publications are the result of collaborative effort and would not have been possible without my co-authors. Therefore, in the presentation of Chapters 3 to 5, I use first-person plural pronouns (*e.g.*, "we" instead of "I") to emphasize their significance.

Chapter 2

Technical concepts and frameworks

This chapter provides a concise introduction to the core modeling frameworks on which the subsequent chapters of the thesis are based. Here, I present the frameworks at an abstract level, complemented by simple examples where appropriate. Note that, each of the following chapters can be read independently, as their individual formulations and results are self-contained. Hence, the goal of this chapter is to serve as a "warm-up" rather than "preliminaries", allowing the reader to gain a basic familiarity with the general technical concepts before directly using them in the specific contexts considered in the subsequent chapters.

I begin by introducing elements of game theory, a framework for modeling strategic interactions between rational agents, which is central to Chapter 3. Then, I present Markov decision processes (MDPs), a standard formulation for describing an agent's sequential decisions in an uncertain environment, alongside their extension to the multi-agent setting. I conclude with the introduction of structural causal models (SCMs), a framework for expressing causal relationships between random variables. MDPs and SCMs form the basis for Chapter 4, while the multi-agent extension of MDPs is used in Chapter 5. Note that, this chapter does not go into details regarding established theoretical results or specific algorithms, deferring more detailed discussions to the chapters where they are most relevant.

In terms of notation, I adopt standard conventions from the machine learning literature. Calligraphic letters (*e.g.*, $\mathcal{A}$) denote sets, while capital letters (*e.g.*, X) denote random variables. When it is clear from the context, certain capital letters (*e.g.*, T in the context of MDPs) denote constants of the respective problem. Lowercase letters represent functions (*e.g.*, $\pi(\cdot)$) or specific variable values and realizations (*e.g.*, x). Bold letters denote multi-dimensional variables, such as vectors or matrices (*e.g.*, $\boldsymbol{x}$), and I use regular letters with subscripts to refer to the individual elements of the respective vector or matrix (*e.g.*, $x_{i,j}$). Generally, the marginal probability of a random variable X taking the value x is written as $P(X = x)$ and conditional probabilities as $P(Y = y \,|\, X = x)$. When the context is clear, marginal probabilities are written as $P(x)$ and conditional probabilities as $P(Y = y \,|\, x)$ or $P(y \,|\, x)$. The notation $P(X)$ refers to the distribution of the random variable X. In addition, for any set $\mathcal{A}$, $\Delta(\mathcal{A})$ denotes the set of all possible distributions over $\mathcal{A}$. Lastly, $\mathbb{1}[\cdot]$ denotes the indicator function, $[n]$ denotes the set of natural numbers ranging from 1 to n and $[n]_0 = \{0\} \cup [n]$.

2.1 Games and equilibria

Game theory is an established field in economics, mathematics and computer science, studying the behavior of rational agents in strategic interactions [147]. It is important to clarify certain terms to avoid ambiguity. In the context of economics, the term "agent" refers to an entity (*e.g.*, an individual or an organization) that makes decisions based on its own preferences and goals, while "rational" typically describes agents whose decisions maximize some measure of the agents' utility [22]. For example, if a customer visits a store to buy cereal with the sole goal of saving money, purchasing the box with the lowest price per kilogram is a rational action—one that is optimal according to their utility. However, if the customer buys the one with the mascot on the packaging that reminds them of their childhood, that is a suboptimal (*i.e.*, irrational) decision.

Game theory focuses on strategic interactions, that is, situations where two rational agents (also called "players") make decisions in each other's presence, with each agent's utility influenced by the other's decision. The simplest form of interaction one can study is when both agents are aware of each other's options and utilities, they are both rational, and they also expect each other to behave rationally [148]. The goal of game theory is to predict the outcome of these interactions in terms of the actions the agents will take and the individual utilities they will obtain. Next, I formally introduce two-player games and use them to express a simple example of a strategic interaction. Although the core ideas can be extended to games involving more than two players, those are not relevant in the context of this thesis.

The possible outcomes of a two-player game correspond to the combinations of the two players' actions, represented as $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2$, where $\mathcal{A}_i$ is a finite set of actions available to player i. Each player's utility, depending on the outcome, is described by a function $u_i : \mathcal{A} \to \mathbb{R}$. The decision player i makes is which strategy σ_i to follow, where σ_i is a probability distribution over $\mathcal{A}_i$, and the player samples an action $A_i \sim \sigma_i$. Such strategies are known as mixed strategies. In the special case where a player selects an action $A_i = a$ deterministically, σ_i is a point mass distribution on a and is referred to as a pure strategy.

An important aspect of a game is the order in which the players take actions. The most widely studied class of two-player games involves simultaneous actions, where both players lack information about the exact action that the other will take. In such games, players commit to strategies σ_1, σ_2, from which they sample their actions. Consequently, the expected utility for player i is given by

$$\mathbb{E}_{A_1 \sim \sigma_1, A_2 \sim \sigma_2} \left[u_i(A_1, A_2) \right] = \sum_{a_1 \in \mathcal{A}_1} \sum_{a_2 \in \mathcal{A}_2} u_i(a_1, a_2) P(A_1 = a_1) P(A_2 = a_2).$$

Despite its simplicity, this formulation can characterize a wide range of strategic interactions. A classic example we consider here is the "Bach or Stravinsky?" dilemma. In this scenario, there are two players, Layla and Frank, who both want to attend a musical concert. Layla has a preference for Bach, while Frank has a preference for Stravinsky. However, neither wants to attend a concert alone. They negotiate and each makes a decision about which concert to attend. The utility functions of the two players can be represented by Table 2.1.

The anticipated behavior in this strategic interaction is that both players will adopt strategies that form a Nash equilibrium [149]. That is, a combination of

strategies such that no player can improve their utility by unilaterally changing their strategy while keeping the other's strategy fixed. Formally, the strategies σ_1 and σ_2 form a Nash equilibrium if:

$$\mathbb{E}_{A_1\sim\sigma_1, A_2\sim\sigma_2}\left[u_1(A_1, A_2)\right] \geq \mathbb{E}_{A_1\sim\sigma_1', A_2\sim\sigma_2}\left[u_1(A_1, A_2)\right] \quad \forall\sigma_1' \text{ and}$$
$$\mathbb{E}_{A_1\sim\sigma_1, A_2\sim\sigma_2}\left[u_2(A_1, A_2)\right] \geq \mathbb{E}_{A_1\sim\sigma_1, A_2\sim\sigma_2'}\left[u_2(A_1, A_2)\right] \quad \forall\sigma_2'.$$

To understand the intuition, consider that, in the context of the "Bach or Stravinsky?" example, we restrict the two players' behavior to pure (*i.e.*, non-randomized) strategies. In that case, there are two Nash equilibria, with pure strategies σ_1 and σ_2 assigning point masses to either of two combinations of actions: (Bach, Bach) or (Stravinsky, Stravinsky). In other words, game theory predicts that if Layla and Frank are both rational, they will eventually attend the same concert. Otherwise, any unilateral deviation would result in the player who deviates attending a concert alone and receiving zero utility, an outcome that is non desirable.

The order in which the players take actions is crucial for the formation of equilibria. For example, consider a slightly different scenario in which Layla first selects and goes to a concert, then invites Frank after she arrives. The only natural outcome is that they will both attend the Bach concert, Layla's preferred composer. This interaction belongs to the class of Stackelberg games [150], where a leader (here, Layla) acts first and a follower (here, Frank) best-responds to the action of the leader. Formally, the leader commits to a strategy σ_1 and the follower's best-response is a strategy

$$\sigma_2 = BR(\sigma_1) = \operatorname*{argmax}_{\sigma\in\Delta(\mathcal{A}_2)} \mathbb{E}_{A_1\sim\sigma_1, A_2\sim\sigma} u_2(A_1, A_2),$$

where $\Delta(\mathcal{A}_2)$ is the space of all possible distributions defined over $\mathcal{A}_2$. In that context, the strategies of the two players form a Stackelberg equilibrium if the leader's strategy maximizes their own utility assuming that the follower will best-respond to it. Formally, the equilibrium strategy σ_1 satisfies

$$\sigma_1 = \operatorname*{argmax}_{\sigma'\in\Delta(A_1)} \mathbb{E}_{A_1\sim\sigma', A_2\sim BR(\sigma')} u_1(A_1, A_2).$$

In a Stackelberg game, it is easy to see that the leader can gain an advantage by committing to a strategy that induces a favorable best-response from the follower, enhancing the leader's utility as a result. Games of this type, along with the computation of their equilibria, are the focus of Chapter 3.

Table 2.1: **Utilities in the "Bach or Stravinsky?" game.** Rows correspond to the actions of Layla, and columns correspond to the actions of Frank. Each cell contains a pair of numbers, indicating the utility of Layla on the left and the utility of Frank on the right.

Layla \ Frank	Bach	Stravinsky
Bach	(2,1)	(0,0)
Stravinsky	(0,0)	(1,2)

2.2 Models of sequential decision making

Decision making processes in daily life, from driving and navigation to clinical care, are often sequential in nature. An agent—human or artificial—interacts with its environment over a series of time steps and, at each step, they observe the current state of the environment, take an action, and receive a reward signal indicating the quality of their action. Consequently, the environment's state evolves based on the agent's actions. Markov decision processes (MDPs) are the standard mathematical framework used to model such decision making tasks [119, 151].

2.2.1 Markov decision processes

Here, we focus on the simplest form of sequential decision making, which is captured by finite MDPs. In a finite MDP, the environment is characterized by a finite set of states $\mathcal{S}$, the agent has access to a finite set of actions $\mathcal{A}$, and they have to make decisions in a finite sequence of time steps. One of the key characteristics of MDPs is that they typically describe environments that behave stochastically. Given the state $s_t \in \mathcal{S}$ of the environment at time t and an action $a_t \in \mathcal{A}$ of the agent, there is uncertainty about the future evolution of the state of the environment. Formally, this is captured by a set of conditional transition distributions $P(S_{t+1} \mid S_t, A_t)$, where $P(s' \mid s, a)$ denotes the probability that the environment transitions from state s to s' if the agent takes action a. Note that, these conditional probabilities are sufficient to completely characterize the dynamics due to the Markov property, a fundamental assumption in MDPs. Intuitively, the Markov property states that the probability of transitioning to a specific next state depends only on the current state and action, not on the history of previous states and actions. Formally, the following conditional independence holds:

$$P(S_{t+1} \mid S_t, A_t, S_{t-1}, A_{t-1}, \ldots, S_0, A_0) = P(S_{t+1} \mid S_t, A_t)$$

While interacting with such an environment, the agent receives a reward signal every time they take an action, which is defined as a reward function $r : S \times A \to \mathbb{R}$. A numerical value $r(s, a)$ indicates the quality of the agent's action a while the environment is in state s. Given a finite horizon $T \in \mathbb{N}$, the agent's behavior is described by a policy $\pi : \mathcal{S} \times [T-1]_0 \to \Delta(\mathcal{A})$ that, for each state and time step, yields a distribution over the set of actions $\mathcal{A}$. In that context, starting from an initial state s_0, the agent's goal is to act according to a policy π^* that maximizes their expected total reward over time, that is,

$$\pi^* = \operatorname*{argmax}_{\pi} \mathbb{E}\left[\sum_{t=0}^{T-1} r(S_t, A_t) \mid S_0 = s_0, \pi\right].$$

Optimal policies in finite MDPs are easy to compute using dynamic programming [152]. The first step is to define a value function $V^* : \mathcal{S} \times [T] \to \mathbb{R}$ such that $V^*(s, i)$ represents the maximum expected total reward achievable given that the environment is in state s and there are i time steps left until reaching the time horizon T. Formally,

$$V^*(s, i) = \max_{\pi} \mathbb{E}\left[\sum_{t=T-i}^{T-1} r(S_t, A_t) \mid S_{T-i} = s, \pi\right],$$

with the boundary condition that $V^*(s, 0) = 0$ for all $s \in \mathcal{S}$. A key observation here is that, based on the Markov property, the aforementioned quantity for $i > 0$ can be expressed recursively as

$$V^*(s, i) = \max_{a \in \mathcal{A}} \left[r(s, a) + \sum_{s' \in \mathcal{S}} P(s' \mid s, a) V^*(s', i - 1) \right].$$

This is known as the Bellman equation in the reinforcement learning literature and implies that the problem of computing the function V^* presents an optimal substructure. Hence, one can compute all its values in a bottom-up manner, working backwards from $i = 0$. This approach, referred to as backward induction or value iteration, also allows the computation of the optimal policy π^*. Each value $\pi^*(s, T - i)$ can be determined by selecting the actions $a \in \mathcal{A}$ that maximize the right-hand side of the equation above.

Chapter 4 builds extensively on the framework of MDPs. In this chapter, we define quantities similar to the value function mentioned above. We also develop algorithmic techniques using dynamic programming and recursive computations, as previously discussed.

2.2.2 Decentralized partially observable Markov decision processes

Standard MDPs focus on a single agent that takes actions based on the state of the environment. However, in many real-world scenarios, sequential decision making processes are much more complex. This section introduces decentralized partially observable MDPs (Dec-POMDPs) [153], which involve two key generalizations: (i) the state of the environment is not directly observable, and (ii) there are multiple agents operating independently in the same environment. Similarly to Section 2.1, the formulation is presented with two agents, but the core ideas can be extended to a larger number of agents.

Formally, a finite Dec-POMDP is characterized by a finite set of states $\mathcal{S}$, and each agent i has their own set of actions $\mathcal{A}_i$. The state of the environment at a time step $t + 1$ is determined by both its state $s_t \in \mathcal{S}$ at time t, as well as the combination of actions $a_{t,1}, a_{t,2} \in \mathcal{A}_1, \mathcal{A}_2$ taken by the two agents. The dynamics of a Dec-POMDP are characterized by a set of conditional probabilities $P(s' \mid s, a)$ that express the probability of the environment transitioning to a state s' given that it is in a state s and the two agents took a joint action $a \in \mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2$. Similarly to standard MDPs, the Markov property holds.

At each time step, the two agents receive a reward $r(s, a)$ indicating the quality of their joint action a. It is important to note that the reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is shared between the two agents. In other words, maximizing the total reward each agent receives means that they should act collaboratively. However, in a Dec-POMDP, neither the state of the environment nor each agent's actions are observable. Instead, for each agent i, there is a set of possible observations $\mathcal{O}_i$ they can make. Depending on the state of the environment s_t and the joint action a_t they take, the agents acquire observations $o_{t,1}, o_{t,2}$, sampled from a given conditional distribution $P(O_{t,1}, O_{t,2} \mid S_t, A_t)$. Then, each agent maintains a belief about the state of the environment and acts according to a time-dependent policy

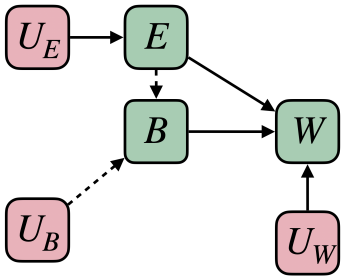


Figure 2.1: **Causal graph representing the relationship between early wake-up times and work productivity.** Green boxes represent endogenous random variables and pink boxes represent exogenous noise variables. The value of each endogenous variable is given by a function of the values of its ancestors in the causal graph. The value of each exogenous noise variable is sampled independently from a given distribution. An intervention $do[B = 1]$ breaks the dependence of the variable B from its ancestors (highlighted by dashed lines) and sets its value to 1. After observing an event $E = 1, B = 0, W = 0$, a counterfactual prediction can be thought of as the result of an intervention $do[B = 1]$ in a modified SCM where the noise variables take values u_E, u_B, u_W from posterior distributions with support such that $1 = g_E(u_E)$, $0 = g_B(1, u_B)$ and $0 = g_W(1, 0, u_w)$.

$\pi_{i,t} : \mathcal{O}_0 \times \mathcal{O}_1 \times \ldots \times \mathcal{O}_{t-1} \to \Delta(\mathcal{A}_i)$ that considers the entire set of observations they have received until time t.

From a computational perspective, the goal in Dec-POMDPs is typically to find a joint policy $\pi = (\pi_1, \pi_2)$ that maximizes the expected total reward. The computation of optimal policies for Dec-POMDPs is a much harder problem than in standard MDPs [154] and goes beyond the scope of this thesis. In Chapter 5, we use Dec-POMDPs solely as a modeling tool to describe the collaboration between a human and an AI agent to introduce a model that helps in understanding people's perception of responsibility of the two agents for a joint outcome in a collaborative task.

2.3 Structural causal models

Our understanding of the world is deeply rooted in causal relationships, such as clouds causing rain or fire generating heat. While traditional probability theory can describe associations between variables through conditional distributions, it does not capture the semantic meaning of causation. For example, a conditional distribution $P(B \mid A)$ does not specify whether A is a noisy generator (*i.e.*, a cause) of B or vice versa—in that case, the distribution could represent a posterior belief about a cause B given the observed evidence A. Structural causal models (SCMs) [71, 117] are a mathematical framework that addresses this limitation by providing a principled way to represent causal relationships.

The key concepts discussed here can best be illustrated using a simple example throughout this section. Consider a scenario in which an office worker might wake up early ($E \in \{0, 1\}$), possibly have breakfast ($B \in \{0, 1\}$), and then go to work. Whether they are productive at work ($W \in \{0, 1\}$) may be caused by several factors but, for simplicity, consider only the following: having breakfast to gain energy and

getting up early to plan the day ahead. An SCM $\mathcal{C}$ describing these relationships can be represented by the directed acyclic graph in Fig. 2.1.

The three variables mentioned above are the endogenous variables of the SCM, which means that they are measurable variables of interest for the analysis. In the causal graph, these variables are represented as nodes (illustrated in green), with edges indicating causal relationships. In this example, the edge $E \to B$ indicates that waking up early allows enough time for breakfast, the edge $B \to W$ indicates that having breakfast provides energy that boosts worker productivity, and the edge $E \to W$ indicates that waking up early allows enough time for planning. Moreover, the graph includes a set of exogenous (noise) random variables $\mathcal{U} = (U_E, U_B, U_W)$ (illustrated in pink) that introduce stochasticity into the system.

To fully specify the relationships between the random variables in the causal graph, an SCM includes additional components beyond the structure of the graph: (i) a set of distributions $\{P^{\mathcal{C}}(U_E), P^{\mathcal{C}}(U_B), P^{\mathcal{C}}(U_W)\}$ for the exogenous variables and (ii) a set of functions $\{g_E, g_B, g_W\}$, known as structural equations or mechanisms. Each of these functions $g_I : \mathcal{PA}_I \times U_I \to \{0, 1\}$ specifies how an endogenous variable I is determined by its endogenous parent variables $\mathcal{PA}_I$ in the causal graph and the corresponding exogenous variable U_I—conceptually, these structural equations resemble assignment statements similar to those found in imperative programming languages. For example, the causal relationships between the three variables E, B and W may be expressed as

$$
\begin{aligned}
E &:= g_E(U_E) = \mathbb{1}\left[U_E \leq 0.9\right] \\
B &:= g_B(E, U_B) = \mathbb{1}\left[E = 1 \wedge U_B \leq 0.9\right] \\
W &:= g_W(E, B, U_W) = \mathbb{1}\left[E = 1 \wedge B = 1 \wedge U_W \leq 0.9\right],
\end{aligned}
$$

where U_E, U_B, U_W are mutually independent and follow uniform distributions in $[0, 1]$. Then, the equations above suggest that (i) the worker wakes up early with probability 0.9, (ii) they have breakfast with probability 0.9, provided that they wake up early, and (iii) they are productive at work with probability 0.9, only if they have breakfast and wake up early to plan the day ahead. In this example, the noise variable U_B could account for unknown factors such as their appetite and food availability, while U_W could account for unpredictable meetings and emergencies in the office.

It is easy to see that, within the framework of SCMs, typical conditional distributions are straightforward to compute. For example, if we are interested in the probability that the worker is productive ($W = 1$) given that they had breakfast ($B = 1$), this is given by

$$
\begin{aligned}
P^{\mathcal{C}}(W = 1 \mid B = 1) &= P^{\mathcal{C}}(W = 1 \mid B = 1, E = 1) P^{\mathcal{C}}(E = 1 \mid B = 1) \\
&\quad + P^{\mathcal{C}}(W = 1 \mid B = 1, E = 0) P^{\mathcal{C}}(E = 0 \mid B = 1) \\
&= P^{\mathcal{C}}(W = 1 \mid B = 1, E = 1) \\
&= \int_0^1 \mathbb{1}\left[u_W \leq 0.9\right] f(u_W) du_W \\
&= 0.9,
\end{aligned}
$$

where $f(u_W)$ represents the probability density function of U_W.

A key characteristic of the probability written above is that it corresponds to an observational distribution, that is, a distribution characterizing a natural probabilistic relationship of the underlying physical system. An additional feature of SCMs is their ability to represent interventional distributions, that is, distributions reflecting what will happen in the system if some external actor performs an intervention. Formally, this is achieved using the $do[\cdot]$ operator [116], which breaks the causal dependency of a variable I from its parents $\mathcal{PA}_I$ in the causal graph, modifies its original structural equation g_I, and replaces it with an assignment $I := i$ to a constant value i.[1] For example, consider that we are interested in computing the probability that the worker is productive ($W = 1$) given that one day their employer decides to offer free breakfast in the office ($do[B = 1]$). Given the SCM $\mathcal{C}$, this probability can be written as

$$\begin{aligned} P^{\mathcal{C};do[B=1]}(W=1) &= P^{\mathcal{C}}(U_W \leq 0.9, E=1) \\ &= \int_0^1 \int_0^1 \mathbb{1}\left[u_W \leq 0.9\right] \mathbb{1}\left[u_E \leq 0.9\right] f(u_E) f(u_W) du_E du_W \\ &= 0.81. \end{aligned}$$

It is important to note that the values of the observational conditional probability $P^{\mathcal{C}}(W = 1 \mid B = 1)$ and the interventional probability $P^{\mathcal{C};do[B=1]}(W = 1)$ differ in this example. The key difference is that, in the observational setting, conditioning on the event that the worker had breakfast ($B = 1$) implies that they woke up early ($E = 1$), which allows them to plan their day ahead and increases their chances of being productive. In contrast, in the interventional setting, the fact that the employer provides breakfast in the office does not offer information about whether the worker woke up early and planned their day, thus making a productive day less certain overall. In general, observational and interventional distributions may (not) differ depending on the structure of the causal graph (refer to Pearl [71] for details).

The expressive power of SCMs is highlighted by their ability to enable reasoning formally about counterfactuals—retrospective "what if?" scenarios that differ from the observed reality. Within an SCM, this type of reasoning is performed in three main steps. First, given observations of the endogenous variables, we infer the posterior distribution of the exogenous variables that led to these observations, a process known as abduction. Next, we perform an intervention using the $do[\cdot]$ operator corresponding to the counterfactual query of interest. Finally, we use the inferred distributions and modified equations to compute the counterfactual distributions of the endogenous variables.

For example, consider a scenario in which the office worker woke up early, did not have breakfast at home (*e.g.*, because they had ran out of food) and consequently was not productive at work. Given that information, one can infer that the noise variables U_E, U_B, U_W must have been such that they allowed the observation $E = 1, B = 0, W = 0$, leading to posterior probability distributions

$$\begin{aligned} &P^{\mathcal{C}}(U_E \mid E=1) = \text{Uniform}[0, 0.9], \\ &P^{\mathcal{C}}(U_B \mid E=1, B=0) = \text{Uniform}[0.9, 1], \text{and} \\ &P^{\mathcal{C}}(U_W \mid E=1, B=0, W=0) = \text{Uniform}[0, 1], \end{aligned}$$

[1] This is known as a hard intervention. In general, the *do* operator also allows for soft interventions that assign a distribution over values instead of a single value.

and let f'_E, f'_B, f'_W be their probability density functions.

A counterfactual quantity of interest could be the probability that the worker would have been productive this particular day, had their employer provided free breakfast in the office ($do[B = 1]$). To answer the counterfactual question, one can simply consider an alternative SCM $\mathcal{C}'$ that is identical to $\mathcal{C}$ except for the fact that the distributions of the noise variables are characterized by the aforementioned probability density functions f'_E, f'_B, f'_W, rather than the ones in $\mathcal{C}$. Then, the counterfactual probability mentioned above can be formally expressed as

$$
\begin{aligned}
&P^{\mathcal{C}|E=1,B=0,W=0\,;\,do[B=1]}(W=1) \\
&\quad = P^{\mathcal{C}'\,;\,do[B=1]}(W=1) \\
&\quad = P^{\mathcal{C}'}(U_W \leq 0.9, E=1) \\
&\quad = \int_0^1 \int_0^1 \mathbb{1}\left[u_W \leq 0.9\right] \mathbb{1}\left[u_E \leq 0.9\right] f'_E(u_E) f'_W(u_W) du_E du_W \\
&\quad = 0.9.
\end{aligned}
$$

This result has an intuitive interpretation. Since the worker managed to wake up early, they had enough time to plan their day. Therefore, if the employer had provided breakfast in the office, the productivity of the worker would have depended solely on the exogenous factors U_W. The fact that, in reality, the worker did not have breakfast and was not productive does not offer any posterior information about these exogenous factors for that particular day. Therefore, the counterfactual probability that the worker would have been productive is simply equal to the probability $P^{\mathcal{C}}(U_W \leq 0.9) = P^{\mathcal{C}'}(U_W \leq 0.9) = 0.9$.

Counterfactual distributions in SCMs are at the center of Chapter 4. In this chapter, we will build upon the technical tools presented here and in Section 2.2 to work with a causal formulation of sequential decision making and reason about the counterfactual effects of action sequences different from those observed in reality.

Chapter 3

Supporting policy design in strategic environments

Decisions across a wide variety of domains, from banking and hiring to insurances, are increasingly informed by data-driven predictive models. In all these domains, the decision maker aims to employ a decision policy that maximizes their utility while the predictive model aims to provide an accurate prediction of the outcome of the process from a set of observable features. For example, in loan decisions, a bank may decide whether or not to offer a loan to an applicant on the basis of a predictive model's estimate of the probability that the individual would repay the loan. The bank's policy in such a setting could be a simple threshold rule, such as granting the loan if the estimated probability exceeds 80%.

Due to the consequential nature of these decisions, there is an increasing pressure on decision makers to be transparent about the decision policies, the predictive models, and the features they use. However, even with access to a highly accurate predictive model, transparency can introduce significant complexities in the decision making process. This is because revealing the decision policy to the individuals who are subject to it induces *strategic behavior*, as it shows them how they could alter their features to receive a favorable decision. This creates a non-trivial feedback loop between the policy and the feature distribution on which the decision maker's utility depends. Therefore, to maximize their utility in that setting, the decision maker may have to consider policies beyond simple threshold rules and analyze the individuals' anticipated responses. This task can be time-consuming or even intractable for a human decision maker, as it involves reasoning about a large number of candidate policies and individuals with diverse characteristics and potential responses.

In this chapter, we address this problem by introducing algorithmic methods that combine predictions from a machine learning model with natural assumptions about how individuals respond to a given policy to compute policies that maximize the decision maker's utility under transparency and strategic behavior. The resulting policies are based on a discrete and relatively small set of feature values, making them easier for a human decision maker to evaluate before implementation. It is important to note that transparency can manifest in various forms. In the following sections, we focus on two approaches towards transparency that have attracted significant interest in the machine learning literature.

In Section 3.1, we look into a setting in which the decision maker publicly shares their entire policy. For example, in loan decisions, this can involve a bank publishing

a set of criteria (*e.g.*, annual income greater than $70,000 and credit card debt less than $10,000) that they deem necessary to grant a loan. This setting, which we refer to as the *complete transparency* scenario, is closely related to strategic classification [84, 85, 93, 155]. This line of work develops classifiers that can maintain their accuracy when data points keep their original label fixed but strategically modify their features to achieve a favorable classification.

In Section 3.2, we focus on an alternative approach to achieve transparency that we call the *partial transparency* scenario; the decision maker opts not to reveal their entire policy but instead provides counterfactual explanations to individuals. In the context of explainable machine learning, a counterfactual explanation for a negatively classified data point is another positively classified data point that differs minimally in terms of feature values [107, 110, 114]. Similarly, throughout the chapter, we use the term "counterfactual explanation" to describe a personalized recommendation given by the decision maker to an individual. This recommendation specifies which features need to be changed, and by how much, for an individual to receive a positive decision. In the lending example, the bank could advise an applicant to increase their income by $10,000 and/or repay half of their credit card debt, while committing to grant the loan once the applicant performs these changes.[1]

In the following sections, we introduce game-theoretic modeling frameworks for the aforementioned scenarios, formalize the relevant optimization problems, analyze their complexity, introduce algorithms to solve them, and evaluate these algorithms using synthetic and real data. The code used for all experiments in Chapter 3 is available at https://github.com/Networks-Learning/strategic-decisions.

3.1 Decision making under complete transparency

As discussed previously, by being transparent about the decision policy they use, the decision maker creates incentives for individuals to invest effort strategically to receive a beneficial decision. Depending on the policy and the features used by the decision maker, individuals may direct their effort towards genuine self-improvement—a win-win situation for both parties—or may attempt to superficially change their feature values to "game" the decision maker's policy [156]. The latter, more skeptical, view has been the key motivation in previous work on strategic classification, which has focused on protecting predictive models against misclassification errors resulting from malicious strategic behavior. Here, instead of focusing on predictive accuracy, we assume that the decision maker knows the probabilistic relationship between features and individual outcomes, and we introduce algorithms to compute decision policies that maximize the utility of the decision maker in a strategic setting.

[1] In the machine learning literature, counterfactual explanations are used to address two distinct questions that are often mistakenly perceived as synonymous: (i) Why was a data point negatively classified? and (ii) What feasible feature changes can lead to a positive classification? The former question is retrospective, focusing on model interpretability [107], while the latter is prospective, focusing on providing algorithmic recourse [108, 113]. Here, we use the term with its latter interpretation. Therefore, counterfactual explanations, as presented in this chapter, although related, present differences with the concept of counterfactual reasoning discussed in Chapters 4, 5, which has a retrospective nature.

Once we focus on the utility achieved by a decision policy, it is overly pessimistic to always view an individual's strategic effort as some form of gaming, and thus undesirable—several studies in economics note that an individual's effort in changing their features may sometimes lead to self-improvement [157–159]. For example, in hiring decisions, if a law firm uses the number of internships to decide whether to offer a job to an applicant, the applicant may feel compelled to do more internships during their studies to increase their chances of getting hired, and this will improve their job performance. In such cases, the decision maker (*i.e.*, the law firm) may like to use a machine learning model to estimate an individual's probability of success (*i.e.*, high job performance) based on their features and find a decision policy that incentivizes individuals to invest in efforts that increase the decision maker's utility (*i.e.*, overall workforce performance).

Incentivizing individuals to invest additional effort to increase the utility of the decision maker may initially appear as an undesirable immediate cost for individuals. However, the resulting self-improvement could potentially prevent events with a larger cost in the long term (*e.g.*, preventing an employee from being fired due to poor job performance). As a consequence, one can also argue that self-improvement can increase social welfare in the long term (*e.g.*, leading to a more skilled workforce). In this context, it is also worth noting that strategy-aware policies that trade-off immediate costs to individuals and beneficial long-term effects are commonly met in public policy whenever governments impose higher taxes to incentivize desirable social behavior. Prominent examples are the taxation of high-emission vehicles [160], unhealthy food [161, 162] and tobacco products [163]. For instance, in the case of high-emission vehicles, the legislator may want to design a taxation system that maximizes their utility—a function of total emissions, state revenue, and the citizens' well-being—while consulting a simulation model [164] that accounts for the stakeholders' strategic responses—for changes in the drivers' buying patterns or the manufacturers' supply. Importantly, economic problems of this form are relevant even when individuals cannot resort to any form of gaming, for example, when tax authorities have introduced proper mechanisms to control tax evasion.

In this context, we cast the problem of utility maximization as a Stackelberg game [150] in which the decision maker moves first by sharing their decision policy before individuals best-respond and invest effort to maximize their chances of receiving a beneficial decision. Importantly, we assume that decisions are based on low-dimensional feature vectors, so that the decision policies are relatively easy for a human decision maker to review and evaluate before implementing them; as argued elsewhere, in many real-world scenarios, the data is summarized by just a small number of summary statistics (*e.g.*, FICO scores) [165, 166]. Then, we characterize how this strategic investment of effort leads to a change in the feature distribution at a population level. More specifically, we derive an analytical expression for the feature distribution induced by any policy in terms of the original feature distribution by solving an optimal transport problem [167]. Based on this analytical expression, we make the following contributions:

1. We show that the problem of finding the optimal decision policy is NP-hard by using a novel reduction of the Boolean satisfiability (SAT) problem [168].

2. We show that there are cases in which deterministic policies are suboptimal in terms of utility, in contrast with the non-strategic setting, where deterministic

threshold rules are optimal [48, 169].

3. Under a natural monotonicity assumption on the cost individuals pay to change features [85, 92], we show that one can narrow down the search for the optimal policy to a particular family of decision policies with a set of desirable properties. Leveraging that observation, we introduce a polynomial time heuristic search algorithm using dynamic programming to find close to optimal decision policies.

4. Under no assumptions on the cost individuals pay to change features, we introduce an iterative search algorithm that is guaranteed to converge to locally optimal decision policies.

Finally, we experiment with synthetic and real credit card data to illustrate our theoretical findings and show that the decision policies found by our algorithms achieve higher utility than several competitive baselines. Moreover, we also show that our decision policies maintain their competitive advantage even under imperfect conditions, such as errors in utility estimates arising from inaccuracies in the predictive model and potential investments of effort that do not lead to self-improvement.

3.1.1 Policies, utilities, and benefits

Given an individual with a feature vector $\boldsymbol{x} \in \{1, \dots, n\}^d$ there is a (stochastic) label $Y \in \{0, 1\}$ and a decision $D \in \{0, 1\}$, which may also be stochastic, that controls whether the label Y is *realized*.[2] This setting fits a variety of real-world scenarios, where continuous features are often discretized into (percentile) ranges. As an example, in a loan decision, the decision specifies whether the individual receives a loan ($D = 1$) or their application is rejected ($D = 0$); the label indicates whether an individual repays the loan ($Y = 1$) or defaults ($Y = 0$) upon receiving it; and the feature vector ($\boldsymbol{x}$) may include an individual's salary percentile, education, or credit history. Moreover, we denote the number of feature values using $m = n^d$, assuming that the number of features d is small, as discussed earlier.

Individuals' features follow a distribution $P(\boldsymbol{X})$ and, for each individual with features $\boldsymbol{x}$, their decision D is sampled from a decision policy $\pi(D \,|\, \boldsymbol{x})$ and their label Y is sampled from $P(Y \,|\, \boldsymbol{x})$. Throughout the section, for brevity, we will write $\pi(\boldsymbol{x}) = \pi(D = 1 \,|\, \boldsymbol{x})$, and we will say that the decision policy satisfies *outcome monotonicity* if the higher an individual's outcome (*i.e.*, their likelihood of $Y = 1$), the higher their chances of receiving a positive decision, that is,

$$P(Y = 1 \,|\, \boldsymbol{x}_i) < P(Y = 1 \,|\, \boldsymbol{x}_j) \Leftrightarrow \pi(\boldsymbol{x}_i) < \pi(\boldsymbol{x}_j). \tag{3.1}$$

Moreover, we adopt a Stackelberg game-theoretic formulation [150] in which the decision maker moves first by publishing their decision policy π before individuals best-respond. As it will become clearer in the next section, individual best-responses lead to a change in the feature distribution at a population level—we will say that the new feature distribution $P(\boldsymbol{X}; \pi)$ is induced by the policy π. Then, we measure

[2] We assume features are discrete and, without loss of generality, each feature takes n discrete values. In the real dataset we used in the evaluation of our algorithms in Section 3.1.6, discretizing continuous features causes a negligible difference in terms of predictive accuracy. Refer to footnote 12 for more details.

the (immediate) utility a decision maker obtains using a policy π as the average overall profit they obtain [48, 169, 170], that is,

$$\begin{aligned} u(\pi, \gamma) &= \mathbb{E}_{\boldsymbol{X} \sim P(\boldsymbol{X}; \pi), Y \sim P(Y \mid \boldsymbol{X}), D \sim \pi(D \mid \boldsymbol{X})}[Y \cdot D - \gamma \cdot D] \\ &= \mathbb{E}_{\boldsymbol{X} \sim P(\boldsymbol{X}; \pi), D \sim \pi(D \mid \boldsymbol{X})}[P(Y = 1 \mid \boldsymbol{X}) \cdot D - \gamma \cdot D], \end{aligned} \tag{3.2}$$

where $\gamma \in (0, 1)$ is a given constant reflecting economic considerations of the decision maker. For example, in a loan scenario, the term $P(Y = 1 \mid \boldsymbol{X}) \cdot D$ is proportional to the expected number of individuals who receive and repay a loan, the term $\gamma \cdot D$ is proportional to the number of individuals who receive a loan, and γ measures the cost of offering a loan in units of repaid loans. Alternatively, one can think of γ as a lower bound on the probability $P(Y = 1 \mid \boldsymbol{X})$ above which the loan provider would consider it rational to offer a loan. Here, note that γ is bounded by the collateral against the loan, which caps the maximum potential cost to the loan provider. Finally, we define the (immediate) individual benefit an individual with features $\boldsymbol{x}$ obtains from a policy π as

$$b(\boldsymbol{x}) = \mathbb{E}_{D \sim \pi(D \mid \boldsymbol{x})}[f(D)], \tag{3.3}$$

where the function $f(\cdot)$ is problem dependent. Here, for ease of exposition, we will assume that $f(D) = D$ and thus $b(\boldsymbol{x}) = \mathbb{E}_{D \sim \pi(D \mid \boldsymbol{x})}[D] = \pi(\boldsymbol{x})$, however, our results can be extended to any function $f(\cdot)$ that is monotonically increasing in D.

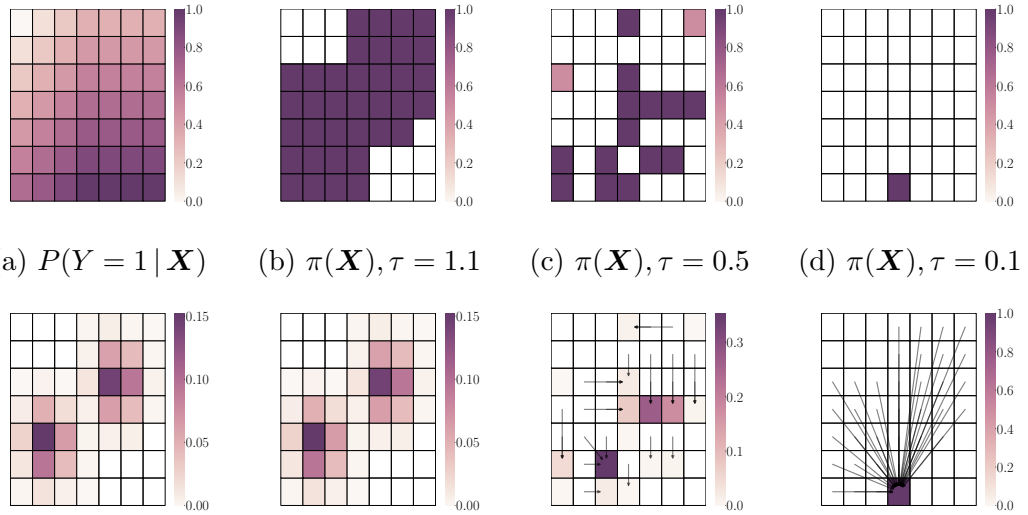
Remarks on strategic classification. Due to Goodhart's law, if the true causal effect between the observed features $\boldsymbol{X}$ and the outcome variable Y is partially described by unobserved features, then $\boldsymbol{X}$ can lose predictive power for Y after individuals best-respond, that is, $P(Y \mid \boldsymbol{X})$ may change [97]. This has been a key insight by previous work on strategic classification [85, 92, 93], which aims to develop accurate predictive models $P_\theta(Y \mid \boldsymbol{X})$ in a strategic setting. Even if there is no unmeasured confounding, a predictive model $P_\theta(Y \mid \boldsymbol{X})$ trained using empirical risk minimization, that is, $\theta^* = \operatorname{argmin}_\theta \mathbb{E}_{\boldsymbol{X} \sim P(\boldsymbol{X}), Y \sim P(Y \mid \boldsymbol{X})}[\ell(\boldsymbol{X}, Y, \theta)]$, where $\ell(\cdot)$ is a given loss function, may decrease its accuracy after best-response. This is because, once individuals best-respond to a decision policy π, θ^* may be suboptimal with respect to the feature distribution induced by the policy, that is,

$$\mathbb{E}_{\boldsymbol{X}, Y \sim P(\boldsymbol{X}; \pi), P(Y \mid \boldsymbol{X})}[\ell(\boldsymbol{X}, Y, \theta^*)] \geq \min_\theta \mathbb{E}_{\boldsymbol{X}, Y \sim P(\boldsymbol{X}; \pi), P(Y \mid \boldsymbol{X})}[\ell(\boldsymbol{X}, Y, \theta)].$$

In this context, Miller et al. [156] have argued that, to distinguish between gaming and improvement, it is necessary to have access to the full underlying causal graph between the features and the outcome variable. In our theoretical and methodological contributions, we assume that there are no unobserved confounders affecting the outcome, that is, $P(Y \mid \boldsymbol{X})$ does not change, and $P_\theta(Y \mid \boldsymbol{X}) = P(Y \mid \boldsymbol{X})$. However, we relax this assumption in our experimental evaluation in Section 3.1.6. In that context, we consider the development of optimal policies that account for changes on $P(\boldsymbol{X})$, $P(Y \mid \boldsymbol{X})$ and $P_\theta(Y \mid \boldsymbol{X})$ after individuals best-respond a very interesting direction for future work [171, 172].

3.1.2 Problem formulation

Similarly as in most previous work in strategic classification [82–85, 92], we consider a Stackelberg game in which the decision maker moves first before individuals best-respond. Moreover, we assume every individual is rational and aims to maximize



(a) $P(Y = 1 \mid \boldsymbol{X})$ (b) $\pi(\boldsymbol{X}), \tau = 1.1$ (c) $\pi(\boldsymbol{X}), \tau = 0.5$ (d) $\pi(\boldsymbol{X}), \tau = 0.1$

(e) $P(\boldsymbol{X})$ (f) $P(\boldsymbol{X}; \pi), \tau = 1.1$ (g) $P(\boldsymbol{X}; \pi), \tau = 0.5$ (h) $P(\boldsymbol{X}; \pi), \tau = 0.1$

Figure 3.1: **Optimal decision policies and induced feature distributions.** Panels (a) and (e) visualize $P(Y = 1 \mid \boldsymbol{X})$ and $P(\boldsymbol{X})$, respectively. Panels (b, c, d) show different cases of an optimal decision policy π, while panels (f, g, h) show the respective induced feature distribution $P(\boldsymbol{X}; \pi)$. Here, the cost of adapting from a feature value $\boldsymbol{x}_i$ to $\boldsymbol{x}_j$ is set to their Manhattan distance, that is, $c(\boldsymbol{x}_i, \boldsymbol{x}_j) = \tau[|x_{i0} - x_{j0}| + |x_{i1} - x_{j1}|]$, where τ is a scaling parameter. In all panels, each cell corresponds to a different feature value $\boldsymbol{x}_i$ and darker colors correspond to higher values. As the cost of changing features for individuals decreases (*i.e.*, τ decreases), the optimal decision policy only provides positive decisions for a few $\boldsymbol{x}$ values with high $P(Y = 1 \mid \boldsymbol{x})$, encouraging individuals to move to those values.

their individual benefit. However, in contrast with previous work, we assume the decision maker shares their decision policy rather than the predictive model. Then, our goal is to find the (optimal) policy that maximizes utility, as defined in Eq. 3.2, that is,

$$\pi^* = \operatorname*{argmax}_{\pi} u(\pi, \gamma), \tag{3.4}$$

under the assumption that each individual best-responds. For each individual, their best-response is to change from their initial set of features $\boldsymbol{x}_i$ to a set of features

$$\boldsymbol{x}_j = \operatorname*{argmax}_{k \in [m]} \left\{ b(\boldsymbol{x}_k) - c(\boldsymbol{x}_i, \boldsymbol{x}_k) \right\}, \tag{3.5}$$

where $c(\boldsymbol{x}_i, \boldsymbol{x}_k)$ is the cost[3] they pay for changing from $\boldsymbol{x}_i$ to $\boldsymbol{x}_k$. Throughout the section, we will assume that (i) it holds that $c(\boldsymbol{x}_i, \boldsymbol{x}_j) > 0$ for all $i \neq j$ such that $P(Y = 1 \mid \boldsymbol{x}_j) \geq P(Y = 1 \mid \boldsymbol{x}_i)$ and (ii) if there are ties in Eq. 3.5, the individual chooses to move to the set of features $\boldsymbol{x}_j$ with the highest $P(Y = 1 \mid \boldsymbol{x}_j)$. Moreover, we will say that the cost satisfies *outcome monotonicity* [92, 93], if improving an

[3]The cost $c(\boldsymbol{x}_i, \boldsymbol{x}_k)$ for each pair of feature values can be non symmetric and, in practice, may be given by a parameterized function. We assume these costs are known to the decision maker, similarly to previous work in the strategic machine learning literature [85, 92, 93, 95, 96, 103, 156].

individual's outcome requires increasing amount of effort, that is,

$$P(Y=1\,|\,\boldsymbol{x}_i) < P(Y=1\,|\,\boldsymbol{x}_j) < P(Y=1\,|\,\boldsymbol{x}_k) \Leftrightarrow \\ [c(\boldsymbol{x}_i,\boldsymbol{x}_j) < c(\boldsymbol{x}_i,\boldsymbol{x}_k)] \wedge [c(\boldsymbol{x}_j,\boldsymbol{x}_k) < c(\boldsymbol{x}_i,\boldsymbol{x}_k)]\,, \quad (3.6)$$

and worsening an individual's outcome requires no effort, that is, $P(Y=1\,|\,\boldsymbol{x}_i) > P(Y=1\,|\,\boldsymbol{x}_j) \Leftrightarrow c(\boldsymbol{x}_i,\boldsymbol{x}_j) = 0$.

At a population level, this best-response results into a transportation of mass between the original distribution and the induced distribution, that is, from $P(\boldsymbol{X})$ to $P(\boldsymbol{X};\pi)$, as exemplified by Fig. 3.1. In particular, we can readily derive an analytical expression for the induced feature distribution in terms of the original feature distribution:

$$P(\boldsymbol{x}_j;\pi) = \sum_{i\in[m]} P(\boldsymbol{x}_i)\mathbb{1}\left[\boldsymbol{x}_j = \underset{k\in[m]}{\operatorname{argmax}}\left\{b(\boldsymbol{x}_k) - c(\boldsymbol{x}_i,\boldsymbol{x}_k)\right\}\right]. \quad (3.7)$$

Note that the transportation of mass between the original and the induced feature distribution has a natural interpretation in terms of optimal transport theory [167]. More specifically, the probability values of the induced feature distribution are given by $P(\boldsymbol{x}_j;\pi) = \sum_{i\in[m]} f_{i,j}$, where $f_{i,j}$ denotes the flow between $P(\boldsymbol{x}_i)$ and $P(\boldsymbol{x}_j|\pi)$ and it is the solution to the following optimal transport problem:

$$\begin{aligned} \underset{\{f_{i,j}\}}{\text{maximize}} \quad & \sum_{i,j\in[m]\times[m]} f_{i,j}[b(\boldsymbol{x}_j) - c(\boldsymbol{x}_i,\boldsymbol{x}_j)] \\ \text{subject to} \quad & f_{i,j} \geq 0 \quad \forall i,j \quad \text{and} \quad \sum_{j\in[m]} f_{i,j} = P(\boldsymbol{x}_i). \end{aligned}$$

Finally, we can combine Eqs. 3.2, 3.4, 3.5 and 3.7, and rewrite our goal as follows:

$$\pi^* = \underset{\pi}{\operatorname{argmax}}\Bigg\{\sum_{i,j\in[m]\times[m]} \left(P\left(Y=1\,|\,\boldsymbol{x}_j\right) - \gamma\right)\pi(\boldsymbol{x}_j) \\ \cdot \left(P(\boldsymbol{x}_i)\mathbb{1}\left[\boldsymbol{x}_j = \underset{k\in[m]}{\operatorname{argmax}}\left\{b(\boldsymbol{x}_k) - c(\boldsymbol{x}_i,\boldsymbol{x}_k)\right\}\right]\right)\Bigg\}, \quad (3.8)$$

where note that, by definition, $0 \leq \pi(\boldsymbol{x}_j) \leq 1$ for all j, the optimal policy π^* may not be unique and, in practice, the distribution $P(\boldsymbol{X})$ and the conditional distribution $P(Y\,|\,\boldsymbol{X})$ may be approximated using models trained on historical data (see remarks on gaming in Section 3.1.1).

Unfortunately, the following Theorem tells us that, in general, we cannot expect to find the optimal policy that maximizes utility in polynomial time using a novel reduction of the Boolean satisfiability (SAT) problem [168]):[4]

Theorem 3.1.1. *The problem of finding the optimal decision policy π^* that maximizes utility in a strategic setting is NP-hard.*

[4]All proofs for Section 3.1 can be found in Appendix A.1.

The above result readily implies that, in contrast with the non strategic setting, where there is no distribution shift, optimal decision policies are not always deterministic threshold rules [48, 169], that is,

$$\pi^*(D = 1 \,|\, \boldsymbol{x}) = \begin{cases} 1 & \text{if } P(Y = 1 \,|\, \boldsymbol{x}) \geq \gamma \\ 0 & \text{otherwise.} \end{cases} \tag{3.9}$$

In addition, in a strategic setting, there are many instances in which optimal decision policies are not deterministic [88], even under outcome monotonic costs. For example, assume $\boldsymbol{x} \in \{1, 2, 3\}$ with $\gamma = 0.1$,

$$\begin{aligned} P(\boldsymbol{x}) &= 0.1\,\mathbb{1}[\boldsymbol{x} = 1] + 0.4\,\mathbb{1}[\boldsymbol{x} = 2] + 0.5\,\mathbb{1}[\boldsymbol{x} = 3], \\ P(Y = 1 \,|\, \boldsymbol{x}) &= 1.0\,\mathbb{1}[\boldsymbol{x} = 1] + 0.7\,\mathbb{1}[\boldsymbol{x} = 2] + 0.4\,\mathbb{1}[\boldsymbol{x} = 3], \text{and} \\ c(\boldsymbol{x}_i, \boldsymbol{x}_j) &= \begin{bmatrix} 0.0 & 0.0 & 0.0 \\ 0.3 & 0.0 & 0.0 \\ 1.2 & 0.3 & 0.0 \end{bmatrix}. \end{aligned}$$

In the non-strategic setting, the optimal policy is clearly $\pi^*(D = 1 \,|\, \boldsymbol{X} = 1) = 1$, $\pi^*(D = 1 \,|\, \boldsymbol{X} = 2) = 1$ and $\pi^*(D = 1 \,|\, \boldsymbol{X} = 3) = 1$. However, in the strategic setting, a brute force search reveals that the optimal policy is stochastic, and it is given by $\pi^*(D = 1 \,|\, \boldsymbol{X} = 1) = 1$, $\pi^*(D = 1 \,|\, \boldsymbol{X} = 2) = 0.7$ and $\pi^*(D = 1 \,|\, \boldsymbol{X} = 3) = 0$, inducing a transportation of mass from $P(\boldsymbol{X} = 3)$ to $P(\boldsymbol{X} = 2; \pi)$ and from $P(\boldsymbol{X} = 2)$ to $P(\boldsymbol{X} = 1; \pi)$. Moreover, note that the optimal policy in the strategic setting achieves higher utility than its counterpart in the non-strategic setting.

3.1.3 Outcome monotonic costs

In this section, we show that, if the cost individuals pay to change features satisfies outcome monotonicity, as defined in Eq. 3.6, we can narrow down the search for the optimal policy to a particular family of decision policies with a set of desirable properties. Here, without loss of generality, we will index the feature values in decreasing order with respect to their corresponding outcome, that is, $i < j \Rightarrow P(Y = 1 \,|\, \boldsymbol{x}_i) \geq P(Y = 1 \,|\, \boldsymbol{x}_j)$.

Given any instance of the utility maximization problem, as defined in Eq. 3.4, it is easy to show that the optimal policy will always decide positively about the feature value with the highest outcome[5], that is, $\pi^*(\boldsymbol{x}_1) = 1$, and negatively about the feature values with outcome lower than γ, that is, $P(Y = 1 \,|\, \boldsymbol{x}_i) < \gamma \Rightarrow \pi^*(\boldsymbol{x}_i) = 0$. However, if the cost individuals pay to change features satisfies outcome monotonicity, we can further characterize a particular family of decision policies that is guaranteed to contain a policy that achieves the optimal utility. In particular, we start by showing that there exists an optimal policy that is outcome monotonic.

Proposition 3.1.1. *Let π^* be an optimal policy that maximizes utility. If the cost $c(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is outcome monotonic then there exists an outcome monotonic policy π such that $u(\pi, \gamma) = u(\pi^*, \gamma)$.*

In the above, note that, given an individual with an initial set of features $\boldsymbol{x}_i$, an outcome monotonic policy always induces a best-response $\boldsymbol{x}_j$ such that

[5] As long as $P(Y = 1 \,|\, \boldsymbol{x}_1) > \gamma$.

$P(Y = 1 \mid \boldsymbol{x}_j) \geq P(Y = 1 \mid \boldsymbol{x}_i)$. Otherwise, a contradiction would occur since, by assumption, it would hold that $\pi(\boldsymbol{x}_i) \geq \pi(\boldsymbol{x}_j)$ and $\pi(\boldsymbol{x}_j) \geq \pi(\boldsymbol{x}_j) - c(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Next, we consider additive costs (*i.e.*, $c(\boldsymbol{x}_i, \boldsymbol{x}_j) + c(\boldsymbol{x}_j, \boldsymbol{x}_k) = c(\boldsymbol{x}_i, \boldsymbol{x}_k)$), and afterwards move on to subadditive costs (*i.e.*, $c(\boldsymbol{x}_i, \boldsymbol{x}_j) + c(\boldsymbol{x}_j, \boldsymbol{x}_k) \geq c(\boldsymbol{x}_i, \boldsymbol{x}_k)$).

Additive costs. If the cost is additive, we first show that we can narrow down the search for the optimal policy to the policies π that satisfy that

$$\pi(\boldsymbol{x}_i) = \pi(\boldsymbol{x}_{i-1}) \vee \pi(\boldsymbol{x}_i) = \max(0, \pi(\boldsymbol{x}_{i-1}) - c(\boldsymbol{x}_i, \boldsymbol{x}_{i-1})) \tag{3.10}$$

for all $i > 1$ such that $P(Y = 1 \mid \boldsymbol{x}_i) > \gamma$. In the remainder, we refer to any policy with this property as an outcome monotonic binary policy. More formally, we have the following theorem.

Theorem 3.1.2. *Let π^* be an optimal policy that maximizes utility. If the cost $c(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is additive and outcome monotonic then there exists an outcome monotonic binary policy π such that $u(\pi, \gamma) = u(\pi^*, \gamma)$.*

Moreover, we can further characterize the best-responses of individuals under outcome monotonic binary policies and additive costs.

Proposition 3.1.2. *Let π be an outcome monotonic binary policy, $c(\boldsymbol{x}_i, \boldsymbol{x}_j)$ be an additive and outcome monotonic cost, $\boldsymbol{x}_i$ be an individual's initial set of features, and define $j = \max\{k : k \leq i \wedge (\pi(\boldsymbol{x}_k) = 1 \vee \pi(\boldsymbol{x}_k) = \pi(\boldsymbol{x}_{k-1}))\}$. If $P(Y = 1 \mid \boldsymbol{x}_i) > \gamma$, the individual's best-response is $\boldsymbol{x}_j$ and, if $P(Y = 1 \mid \boldsymbol{x}_i) \leq \gamma$, the individual's best-response is $\boldsymbol{x}_j$ if $\pi(\boldsymbol{x}_j) \geq c(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $\boldsymbol{x}_i$ otherwise.*

This proposition readily implies that $P(\boldsymbol{x}_i; \pi) = 0$ for all $\boldsymbol{x}_i$ such that $\pi(\boldsymbol{x}_i) \neq \pi(\boldsymbol{x}_{i-1})$ with $\pi(\boldsymbol{x}_i) > 0$. Therefore, it lets us think of the feature values $\boldsymbol{x}_i$ with $\pi(\boldsymbol{x}_i) = \pi(\boldsymbol{x}_{i-1})$ as *blocking* states and those with $\pi(\boldsymbol{x}_i) \neq \pi(\boldsymbol{x}_{i-1})$ as *non-blocking* states. Moreover, the above results facilitate the development of a highly effective heuristic search algorithm based on dynamic programming to find close to optimal (outcome monotonic binary) policies in polynomial time.

Algorithm 1 summarizes the dynamic programming algorithm and Fig. 3.2 helps visualize the entire procedure. The main idea is to recursively create a set of decision subpolicies $\{\pi_{i,j}(\boldsymbol{x}_k)\}$ where $i, j = 1, \ldots, m$ with $j < i$, $k = j, \ldots, m$, which we later use to build the entire decision policy π. At a high level, a subpolicy $\pi_{i,j}$ is defined for all feature values $\boldsymbol{x}_k$ "on the right" of $\boldsymbol{x}_j$, and it has the form of a "staircase" (*i.e.*, no blocking feature values) between $\boldsymbol{x}_j$ and $\boldsymbol{x}_i$ (refer to Figs. 3.2(b, c, d) for visualized examples of subpolicies). Depending on the structure of the costs and feature and label distributions, the algorithm may need to perform several rounds and, in each round, create a new set of decision subpolicies, which are used to set only some values of the decision policy.

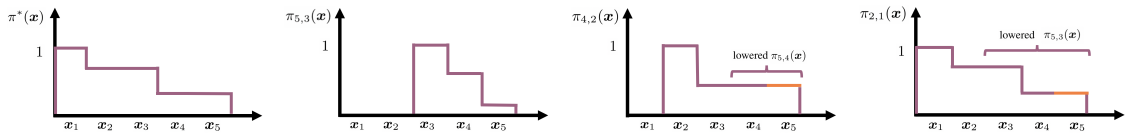
In each round, we proceed in decreasing order of i and j (lines 5–6) until the feature value index s, which is computed in the previous round (line 26) and marks that the computation of policy values for indexes $1, \ldots, s-1$ is finalized. For each subpolicy $\pi_{i,j}$: (i) we fix $\pi_{i,j}(\boldsymbol{x}_j) = \pi_{2,1}(\boldsymbol{x}_s)$, $\pi_{i,j}(\boldsymbol{x}_k) = \pi(\boldsymbol{x}_{k-1}) - c(\boldsymbol{x}_k, \boldsymbol{x}_{k-1})$ for all $j < k < i$ and $\pi_{i,j}(\boldsymbol{x}_k) = 0$ for all k such that $P(Y = 1 \mid \boldsymbol{x}_k) \leq \gamma$ (line 4); (ii) we decide whether to *block* or not to block the feature value $\boldsymbol{x}_i$, that is, set $\pi_{i,j}(\boldsymbol{x}_i)$ to either $\pi_{i,j}(\boldsymbol{x}_{i-1})$ or $\pi_{i,j}(\boldsymbol{x}_{i-1}) - c(\boldsymbol{x}_i, \boldsymbol{x}_{i-1})$, based on previously computed subpolicies within the round (line 12); and, (iii) after we decide whether to block the

Algorithm 1: DynamicProgramming: It searches for the decision policy that maximizes utility under additive and outcome monotonic costs.

input : number of feature values m, constant γ, distributions $P(\boldsymbol{X})$ and $P(Y \mid \boldsymbol{X})$, and cost function $c(\cdot, \cdot)$
output : policy π and associated utility $u(\pi, \gamma)$

1 $\{\pi_{i,j}\} \leftarrow$ `initialize_subpolicies`()
2 $s \leftarrow 1$ // Initialize the round's starting index
3 **repeat**
4 $\quad r, \{\pi_{i,j}\}, F \leftarrow$ `compute_base_subpolicies`$(c, P(\boldsymbol{X}), P(Y \mid \boldsymbol{X}), \pi_{2,1}(\boldsymbol{x}_s))$
5 $\quad$ **for** $i \leftarrow r - 1$ **to** $s + 1$ **do**
6 $\quad\quad$ **for** $j \leftarrow i - 1$ **to** s **do**
7 $\quad\quad\quad$ **if** $c(\boldsymbol{x}_{i-1}, \boldsymbol{x}_j) > \pi_{2,1}(\boldsymbol{x}_s)$ **then**
8 $\quad\quad\quad\quad$ **continue** // Skip this subpolicy as invalid
9 $\quad\quad\quad$ $\sigma \leftarrow c(\boldsymbol{x}_{i-1}, \boldsymbol{x}_j)$
10 $\quad\quad\quad$ $G \leftarrow (P(Y = 1 \mid \boldsymbol{x}_j) - \gamma) \sum_{k\,:\,j \le k < i} P(\boldsymbol{x}_k)$ // Utility gained by the population with indices $j, \ldots, i-1$
11 $\quad\quad\quad$ $\pi', F', v' \leftarrow$ `lower`$(\pi_{i+1,i}, F(i+1, i), \sigma)$
12 $\quad\quad\quad$ **if** $F(i+1, j) \ge F' + G$ ***and*** $c(\boldsymbol{x}_i, \boldsymbol{x}_j) \le \pi_{2,1}(\boldsymbol{x}_s)$ **then**
13 $\quad\quad\quad\quad$ $\pi_{i,j}(\boldsymbol{x}_i) \leftarrow \pi_{i,j}(\boldsymbol{x}_{i-1}) - c(\boldsymbol{x}_i, \boldsymbol{x}_{i-1})$ // Set $\boldsymbol{x}_i$ as a non-blocking feature value
14 $\quad\quad\quad\quad$ $F(i, j) \leftarrow F(i+1, j)$
15 $\quad\quad\quad\quad$ **for** $l \leftarrow i + 1$ **to** m **do**
16 $\quad\quad\quad\quad\quad$ $\pi_{i,j}(\boldsymbol{x}_l) \leftarrow \pi_{i+1,j}(\boldsymbol{x}_l)$ // Set subpolicy values based on the previously computed subpolicy
17 $\quad\quad\quad\quad$ $V(i, j) \leftarrow V(i+1, j)$
18 $\quad\quad\quad$ **else**
19 $\quad\quad\quad\quad$ $\pi_{i,j}(\boldsymbol{x}_i) \leftarrow \pi_{i,j}(\boldsymbol{x}_{i-1})$ // Set $\boldsymbol{x}_i$ as a blocking feature value
20 $\quad\quad\quad\quad$ $F(i, j) \leftarrow F' + G$
21 $\quad\quad\quad\quad$ **for** $l \leftarrow i + 1$ **to** m **do**
22 $\quad\quad\quad\quad\quad$ $\pi_{i,j}(\boldsymbol{x}_l) \leftarrow \pi'(\boldsymbol{x}_l)$ // Set subpolicy values based on the lowered subpolicy
23 $\quad\quad\quad\quad$ $V(i, j) \leftarrow v'$
24 $\quad$ **for** $l \leftarrow s$ **to** $V(s+1, s)$ **do**
25 $\quad\quad$ $\pi(\boldsymbol{x}_l) \leftarrow \pi_{s+1,s}(\boldsymbol{x}_l)$ // Set policy values that will not be revisited
26 $\quad$ $s \leftarrow V(s+1, s)$ // Set next round's starting index
27 **until** $V(s+1, s) = m$
28 **return** $\pi, u(\pi, \gamma)$

feature value $\boldsymbol{x}_i$ or not, we set the remaining policy values (with indexes $i+1, \ldots, m$) by *appending* the *best* of these previously computed subpolicies in terms of overall utility (lines 16 and 22). Here, note that there is a set of base subpolicies, those with $i = r$ where $r = \max\{k : P(Y = 1 \mid \boldsymbol{x}_k) > \gamma\}$ and $1 - c(\boldsymbol{x}_{i-1}, \boldsymbol{x}_j) \ge 0$, which can be computed directly, without recursion (line 4). Intuitively, these correspond to a "staircase" form (see Fig. 3.2(b)), where we only need to decide whether to block or not the "rightmost" feature value, depending on which option gives the greatest utility. Moreover, if we decide to block $\boldsymbol{x}_i$ in a subpolicy $\pi_{i,j}$, we need to lower the



(a) Optimal policy $\pi^*(\boldsymbol{x})$ (b) Subpolicy $\pi_{5,3}(\boldsymbol{x})$ (c) Subpolicy $\pi_{4,2}(\boldsymbol{x})$ (d) Subpolicy $\pi_{2,1}(\boldsymbol{x})$

Figure 3.2: **Optimal policy and subpolicies after Algorithm 1 performs its first round.** Panel (a) shows the optimal policy $\pi^*(\boldsymbol{x})$, which contains blocking states in $\boldsymbol{x}_3$ and $\boldsymbol{x}_5$. Panel (b) shows the subpolicy $\pi_{5,3}(\boldsymbol{x})$, which is a base subpolicy that can be computed without recursion. Panel (c) shows the subpolicy $\pi_{4,2}(\boldsymbol{x})$, which contains a blocking state in $\boldsymbol{x}_4$ and uses a lowered version of the subpolicy $\pi_{5,4}(\boldsymbol{x})$ to set the feature value $\boldsymbol{x}_5$. Since $\pi_{4,2}(\boldsymbol{x}_4) - c(\boldsymbol{x}_5, \boldsymbol{x}_4) < 0$, this value is set equal to $\pi_{4,2}(\boldsymbol{x}_4)$. Panel (d) shows the subpolicy $\pi_{2,1}(\boldsymbol{x})$, which contains a blocking state in $\boldsymbol{x}_3$ and uses a lowered version of the subpolicy $\pi_{5,3}(\boldsymbol{x})$ to set the feature values $\boldsymbol{x}_4$ and $\boldsymbol{x}_5$. Since in $\pi_{2,1}(\boldsymbol{x})$, the feature value $\boldsymbol{x}_5$ became negative and was set as blocking, the algorithm will perform a second round, starting from $\boldsymbol{x}_3$, which is the last blocking state before $\xi = 4$.

values of the previously computed subpolicies down (line 11) by $\sigma = c(\boldsymbol{x}_{i-1}, \boldsymbol{x}_j)$ before appending them so that $\pi_{i,j}(\boldsymbol{x}_i) = \pi_{2,1}(\boldsymbol{x}_s) - c(\boldsymbol{x}_{i-1}, \boldsymbol{x}_j)$ eventually. However, some of these values may become negative and are thus decided to be blocking states, *i.e.*, $\pi'(\boldsymbol{x}_k) = \pi_{i+1,i}(\boldsymbol{x}_d) - \sigma \; \forall k : r \geq k > \xi$ where $\xi = \max\{l : \pi_{i+1,i}(\boldsymbol{x}_l) - \sigma \geq 0\}$. If during this procedure the lowered policy makes some individual change their best-response, the policy values starting from the last blocking state v' before ξ will be revisited in another round (line 23). Figs. 3.2(c, d) show examples of subpolicies $\pi_{i,j}$ where the lowering procedure is performed.

Within the algorithm, the function `initialize_subpolicies`() initializes the subpolicies $\{\pi_{i,j}\}$, `compute_base_subpolicies`(...) computes $r = \max\{k : P(Y = 1 \,|\, \boldsymbol{x}_k) > \gamma\}$, the base subpolicies and their utilities, and `lower`(...) computes a policy π' with $\pi'(\boldsymbol{x}_k) = \pi_{i+1,i}(\boldsymbol{x}_k) - \sigma$ if that quantity is non-negative and $\pi'(\boldsymbol{x}_k) = \pi_{i+1,i}(\boldsymbol{x}_\xi) - \sigma$ otherwise, its corresponding utility F', and calculates the index v' of the last blocking state before ξ as described in the previous paragraph.

As mentioned above, the algorithm might need more than one round to terminate. Since each round consists of one dynamic programming execution, an array of utility values of all subpolicies needs to be computed, having a size of $\mathcal{O}(m^2)$, considering that each state variable i, j takes values from the set $\{1, 2, .., m\}$. Given an outcome monotonic binary policy π, according to Proposition 3.1.2, we can easily characterize the best-response of each individual and it can be easily seen that the overall utility $u(\pi, \gamma)$ can be computed with a single pass over the feature values. Therefore, computing each entry's value in the aforementioned array takes $\mathcal{O}(m)$ time, leading to a total round complexity of $\mathcal{O}(m^3)$.

Now, consider the total number of rounds. It can be observed that a second round is executed iff $s \neq m$ at the end of the first one, implying that at least one feature value was blocked since the value of $V(i, j)$ might get altered only when choosing to block a feature value because of the `lower`(...) operation. Therefore, we can deduce that during each round ending with $s \neq m$, at least one feature value gets blocked, leading to a $\mathcal{O}(m)$ bound on the total number of rounds. As a

consequence, the overall complexity of the algorithm is $\mathcal{O}(m^4)$.

Subadditive costs. If the cost is subadditive, we can show that we need to instead narrow down the search for the optimal policy to the policies π that satisfy

$$\pi(\boldsymbol{x}_i) = \pi(\boldsymbol{x}_{i-1}) \vee \bigvee_j \pi(\boldsymbol{x}_i) = \max(0, \pi(\boldsymbol{x}_{i-1}) - c(\boldsymbol{x}_j, \boldsymbol{x}_{i-1})) \tag{3.11}$$

for all $i > 1$ such that $P(Y = 1 \,|\, \boldsymbol{x}_i) > \gamma$ and $j = i, \dots k$ with $k = \max\{j : \pi(\boldsymbol{x}_{i-1}) - c(\boldsymbol{x}_j, \boldsymbol{x}_{i-1}) > 0\}$. More formally, we have the following proposition, which can be easily shown using a similar reasoning as the one used in the proof of Theorem 3.1.2:

Proposition 3.1.3. *Let π^* be an optimal policy that maximizes utility. If the cost $c(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is subadditive and outcome monotonic then there exists an outcome monotonic policy π satisfying Eq. 3.11 such that $u(\pi, \gamma) = u(\pi^*, \gamma)$.*

Similarly as in the case of additive costs, it is possible to characterize the best-response of the individuals[6] and adapt the above mentioned heuristic search algorithm to find close to optimal (outcome monotonic binary) policies with subadditive costs, however, the resulting algorithm is rather impractical due to its complexity and therefore we omit the details.

Remarks on computational hardness. We conjecture that, even in the simplest case of additive and outcome monotonic costs, the problem of finding an optimal policy π^* remains NP-hard. The reason is that one can view the problem as a version of the 0-1 knapsack or the traveling salesman with profits [173] problems where the profit of an item (node) is a function of the other items (nodes) present in the knapsack (path). More specifically, one has to make a binary decision about every feature value $\boldsymbol{x}_i$, that is, to set it either as blocking or non-blocking. However, from Eq. 3.8 and Proposition 3.1.2, it follows that the portion of utility gained by deciding to block a single feature value $\boldsymbol{x}_i$ is a product that depends, not only on $\pi(\boldsymbol{x}_i)$ and $P(Y = 1 \,|\, \boldsymbol{x}_i)$, but also on the policy values $\pi(\boldsymbol{x}_j)$ of all feature values $\boldsymbol{x}_j$ with $j \neq i$. Unfortunately, this additional structure makes any reduction from the above classic NP-complete problems highly non-trivial. We leave this as an open problem for future work.

3.1.4 General costs

In this section, we first show that, under no assumptions on the cost people pay to change features, the optimal policy might not be outcome monotonic. Then, we introduce an efficient iterative algorithm that it is guaranteed to terminate and find locally optimal decision policies.[7] Finally, we propose a variation of the algorithm that can significantly reduce its running time when working with real data.

There may not exist an optimal policy that satisfies outcome monotonicity under general costs. Our starting point is the toy example introduced at the end of Section 3.1.2. Here, we just modify the cost individuals pay to change features so that it violates outcome monotonicity of the costs. More specifically,

[6] In this case, each possible decision policy value blocks zero, one or more feature values.

[7] We refer to a policy π as locally optimal if there exists no $\pi' \neq \pi$, differing in exactly one feature value $\boldsymbol{x}_i$, such that $u(\pi', \gamma) > u(\pi, \gamma)$.

Algorithm 2: ITERATIVE: It approximates the optimal decision policy that maximizes utility under general costs.

input : number of feature values m, constant γ, distributions $P(\boldsymbol{X})$ and $P(Y \mid \boldsymbol{X})$, and cost function $c(\cdot, \cdot)$
output : policy π and associated utility $u(\pi, \gamma)$

1 $\pi \leftarrow$ `initialize_policy()`
2 **repeat**
3 $\pi_{\text{old}} \leftarrow \pi$
4 **for** $i \leftarrow 1$ **to** m **do**
5 $\pi(\boldsymbol{x}_i) \leftarrow$ `solve`$(i, \pi, c, P(\boldsymbol{X}), P(Y \mid \boldsymbol{X}))$ `// Optimize the value of` $\pi(\boldsymbol{x}_i)$`, keeping` $\pi(\boldsymbol{x}_j)$ `for` $j \neq i$ `fixed`
6 **until** $u(\pi, \gamma) = u(\pi_{old}, \gamma)$
7 **return** $\pi, u(\pi, \gamma)$

assume $\boldsymbol{x} \in \{1, 2, 3\}$ with $\gamma = 0.1$,

$$P(\boldsymbol{x}) = 0.1\,\mathbb{1}[\boldsymbol{x} = 1] + 0.4\,\mathbb{1}[\boldsymbol{x} = 2] + 0.5\,\mathbb{1}[\boldsymbol{x} = 3],$$
$$P(Y = 1 \mid \boldsymbol{x}) = 1.0\,\mathbb{1}[\boldsymbol{x} = 1] + 0.7\,\mathbb{1}[\boldsymbol{x} = 2] + 0.4\,\mathbb{1}[\boldsymbol{x} = 3], \text{and}$$
$$c(\boldsymbol{x}_i, \boldsymbol{x}_j) = \begin{bmatrix} 0.0 & 0.2 & 0.3 \\ 0.3 & 0.0 & 0.7 \\ 1.2 & 1.1 & 0.0 \end{bmatrix}$$

Now, in the strategic setting, it is easy to see that every policy given by $\pi^*(D = 1 \mid \boldsymbol{X} = 1) = 1$, $\pi^*(D = 1 \mid \boldsymbol{X} = 2) \leq 0.7$ and $\pi^*(D = 1 \mid \boldsymbol{X} = 3) = 1$ is optimal and induces a transportation of mass from $P(\boldsymbol{X} = 2)$ to $P(\boldsymbol{X} = 1; \pi)$. Therefore, optimal policies are not necessarily outcome monotonic under general costs.

An iterative algorithm for general costs. Next, we introduce an efficient iterative algorithm that is guaranteed to terminate and find locally optimal decision policies. The iterative algorithm is based on the following key insight: fix the decision policy $\pi(\boldsymbol{x})$ for all feature values $\boldsymbol{x} = \boldsymbol{x}_k$ except $\boldsymbol{x} = \boldsymbol{x}_i$. Then, Eq. 3.8 reduces to searching over $\mathcal{O}(m)$ values for $\pi(\boldsymbol{x}_i)$.

Exploiting this insight, the iterative algorithm proceeds iteratively and, at each each iteration, it optimizes the decision policy for each of the feature values while fixing the decision policy for all other values. Algorithm 2 summarizes the overall procedure. Within the algorithm, `initialize_policy()` initializes the decision policy to $\pi(\boldsymbol{x}) = 0$ for all $\boldsymbol{x}$, `solve`$(\ldots)$ finds the best policy $\pi(\boldsymbol{x}_i)$ for $\boldsymbol{x}_i$ given $\pi(\boldsymbol{x}_k)$ for all $\boldsymbol{x}_k \neq \boldsymbol{x}_i$, the cost function c, and the distributions $P(\boldsymbol{X})$ and $P(Y \mid \boldsymbol{X})$ by searching over $\mathcal{O}(m)$ critical values, where the best-response of some $\boldsymbol{x}_k$ might change. In practice, we proceed over feature values in decreasing order with respect to $P(Y = 1 \mid \boldsymbol{x}_i)$ because we have observed it improves performance. However, our theoretical results do not depend on such ordering. In the following, we refer to lines 2-7 of Algorithm 2 as one *iteration* and line 5 as one *step*.

Theoretical guarantees of the iterative algorithm. We start our theoretical analysis with the following Proposition, which shows that our algorithm is guaranteed to terminate after a finite number of steps:

Proposition 3.1.4. *Algorithm 2 terminates after at most* $m^{1+\frac{1}{\bar{u}}} - 1$ *steps, where* $\bar{u}$ *is the greatest common denominator of all elements in the set* $A = \{c(\boldsymbol{x}_i, \boldsymbol{x}_j) -$

$c(\boldsymbol{x}_i, \boldsymbol{x}_k) : \boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_k \in [m] \times [m] \times [m]\} \cup 1$.[8]

Note that, at each step, our iterative algorithm leads to a policy π which is better or equivalent to the policy computed in the previous step π', that is, $u(\pi, \gamma) \geq u(\pi', \gamma)$. This holds because `solve`$(\ldots)$ either returns a strictly better policy or returns the given policy unchanged. Also, following from line 6 of Algorithm 2, the algorithm terminates only if it performs a full iteration where all of its m steps fail to increase the policy's utility, that is, there exists no $\pi \neq \pi_{\text{old}}$ differing in exactly one feature value $\boldsymbol{x}_i$ such that $u(\pi, \gamma) > u(\pi_{\text{old}}, \gamma)$. As a direct consequence, we can conclude that Algorithm 2converges to locally optimal decision policies.

Moreover, we can characterize the computational complexity of the algorithm as follows. At each iteration, the algorithm calls `solve` m times and, within `solve`, there are $\mathcal{O}(m)$ candidate values for $\pi(\boldsymbol{x}_i)$ when $\pi(\boldsymbol{x}_k)$ is fixed for all $\boldsymbol{x}_k \neq \boldsymbol{x}_i$, and they can all be evaluated in $\mathcal{O}(m^2)$. Therefore, the iteration complexity of Algorithm 2 is $\mathcal{O}(m^3)$.

Speeding up the iterative algorithm in the presence of non-actionable features. Here, we discuss a highly effective strategy to speed up the iterative algorithm whenever some of the features are non-actionable, which is amenable to parallelization. As an example, assume there is an *Age Group* feature which takes values $\{< 30, 30-60, > 60\}$. Now, consider two individuals with initial feature values $\boldsymbol{x}_i, \boldsymbol{x}_j$ such that $\boldsymbol{x}_{i,AgeGroup} = < 30$ and $\boldsymbol{x}_{j,AgeGroup} = > 60$. Since individuals cannot change their age, it holds that $c(\boldsymbol{x}_i, \boldsymbol{x}_j) = c(\boldsymbol{x}_j, \boldsymbol{x}_i) = \infty$. Let $\mathcal{G}$ be an undirected graph where each node v_i represents a feature value $\boldsymbol{x}_i$ and there is an edge $e_{i,j}$ between two nodes v_i and v_j iff $c(\boldsymbol{x}_i, \boldsymbol{x}_j) \leq 1 \vee c(\boldsymbol{x}_j, \boldsymbol{x}_i) \leq 1$. Then, if there are non-actionable features, it is easy to see that the graph $\mathcal{G}$ may be composed of several independent connected components. Assume v_i and v_j belong to two different connected components. Then, whatever value is picked for $\pi(\boldsymbol{x}_i)$, the best-response of individuals with initial features $\boldsymbol{x}_j$ will never be $\boldsymbol{x}_i$ since $\pi(\boldsymbol{x}_i) \leq 1 \Rightarrow \pi(\boldsymbol{x}_i) - c(\boldsymbol{x}_j, \boldsymbol{x}_i) \leq 1 - c(\boldsymbol{x}_j, \boldsymbol{x}_i) < 0 \leq \pi(\boldsymbol{x}_j)$ and therefore $\boldsymbol{x}_j$ will always be a better response. Similarly, the best-response of individuals with initial features $\boldsymbol{x}_i$ will never be $\boldsymbol{x}_j$ independently of the value of $\pi(\boldsymbol{x}_j)$. As a consequence, we can find the values of the optimal policy by running the iterative algorithm independently on each independent component.

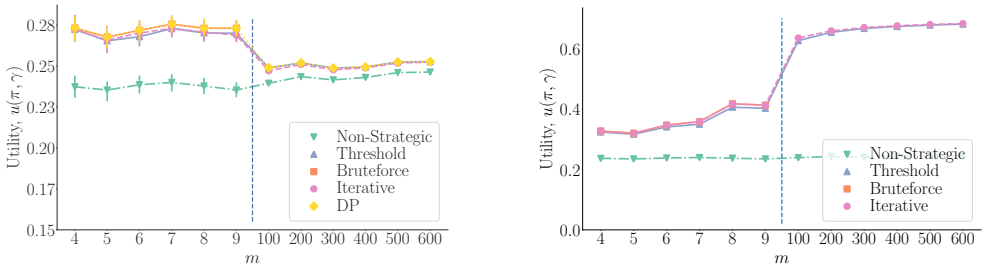
3.1.5 Experiments on synthetic data

In this section, we evaluate both our dynamic programming algorithm (Algorithm 1) and our iterative algorithm (Algorithm 2) on outcome monotonic and general costs. We first compare the utility achieved by the decision policies found by our algorithms and those found by several competitive baselines. Then, we compare their computational complexity both in terms of running time and number of rounds (or iterations) to termination.[9]

Performance evaluation. We compare the utility achieved by the decision policies found by our algorithms and those found by several baselines. More specifically, we consider

[8] The common denominator $\bar{u}$ satisfies that $\frac{a}{\bar{u}} \in \mathbb{Z}\ \forall a \in A \cup \{1\}$. Such $\bar{u}$ exists if and only if $\frac{a}{b}$ is rational $\forall a, b \in A$.

[9] All experiments for Section 3.1 ran on a machine equipped with 48 Intel(R) Xeon(R) 3.00GHz CPU cores and 1.2TB memory.



(a) Outcome monotonic additive costs

(b) General costs

Figure 3.3: **Performance evaluation on synthetic data.** Panels show the utility obtained by several decision policies against the number of feature values m. Here, note that the dynamic programming (DP) algorithm (Algorithm 1) only works with outcome monotonic additive costs and thus only appears in Panel (a). In Panel (a), we set $\kappa = 0.1$ and, in Panel (b), we set $\kappa = 0.25$. In both panels, we repeat each experiment 100 times, and error bars indicate 95% confidence intervals.

(i) *Non-Strategic*: the optimal deterministic threshold rule in a non-strategic setting (see Eq. 3.9),

(ii) *Threshold*: the optimal deterministic threshold rule in a strategic setting, found via bruteforce search over all deterministic threshold rules,

(iii) *Bruteforce*: the optimal (stochastic) decision policy in a strategic setting, found via brute force search,

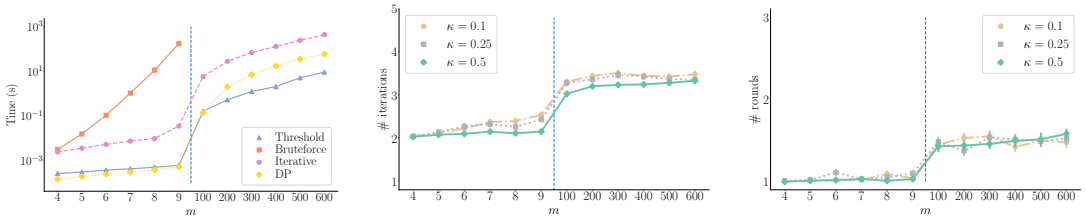
(iv) *DP*: the (stochastic) decision policy found by our dynamic programming algorithm (Algorithm 1), which we can only run for instances with outcome monotonic additive, and

(v) *Iterative*: the (stochastic) decision policy found by our iterative algorithm (Algorithm 2).

Here, we consider unidimensional features with m discrete values $\boldsymbol{x} \in [m]$ and compute $P(\boldsymbol{x} = i) = p_i / \sum_j p_j$, where p_i is sampled from a Gaussian distribution $N(\mu = 0.5, \sigma = 0.1)$ truncated from below at zero. Then, we sample the values $P(Y = 1 \,|\, \boldsymbol{x})$ from a Uniform$[0, 1]$ and set $\gamma = 0.3$.

For instances with outcome monotonic additive costs, we initially set $c(\boldsymbol{x}_i, \boldsymbol{x}_j) = 0 \;\forall \boldsymbol{x}_i, \boldsymbol{x}_j : P(Y = 1 \,|\, \boldsymbol{x}_j) \leq P(Y = 1 \,|\, \boldsymbol{x}_i)$. Then, we take $m - 1$ samples from $U[0, 1/\kappa]$ and assign them to $c(\boldsymbol{x}_m, \boldsymbol{x}_i) \;\forall i < m$ such that $c(\boldsymbol{x}_m, \boldsymbol{x}_i) > c(\boldsymbol{x}_m, \boldsymbol{x}_j) \;\forall i < j$ and $\kappa \in (0, 1]$. Finally, we set the remaining values $c(\boldsymbol{x}_i, \boldsymbol{x}_j)$, in decreasing order of i and j such that $c(\boldsymbol{x}_i, \boldsymbol{x}_j) = c(\boldsymbol{x}_{i-1}, \boldsymbol{x}_j) - c(\boldsymbol{x}_{i-1}, \boldsymbol{x}_i)$. It is easy to observe that, proceeding this way, individuals with feature values $\boldsymbol{x}_i$ can move (on expectation) to at most κm *better* states, that is, $c(\boldsymbol{x}_i, \boldsymbol{x}_j) \leq 1$ for all $\boldsymbol{x}_i, \boldsymbol{x}_j$ such that $\max(1, i - \kappa m) \leq j < i$. For instances with general costs, we sample the cost between feature values $c(\boldsymbol{x}_i, \boldsymbol{x}_j)$ from a Uniform$[0, 1]$ for a fraction κ of all pairs and set $c(\boldsymbol{x}_i, \boldsymbol{x}_j) = \infty$ for the remaining pairs.

Fig. 3.3 summarizes the results for both outcome monotonic and general costs. In both cases, we observe that the optimal decision policy in a non-strategic setting has an underwhelming performance. For outcome monotonic additive costs, we observe that the policies found using our dynamic programming algorithm and brute force



(a) Running time vs. # feature values

(b) # iterations vs. # feature values

(c) # rounds vs. # feature values

Figure 3.4: **Running time analysis on synthetic data with outcome monotonic and additive costs.** Panel (a) shows the running time of the brute force search, the threshold policy baseline, our iterative algorithm and our dynamic programming algorithm. Panels (b) and (c) show the number of iterations and rounds required by the iterative and dynamic programming algorithms until termination, respectively, for different κ values. In Panel (a), we set $\kappa = 0.1$. In all panels, we repeat each experiment 100 times, and error bars indicate 95% confidence intervals.

search closely match each other in terms of utility and they consistently outperform the policies found by the iterative algorithm. For general costs, we find that our iterative algorithm and the threshold policy baseline are the top performers. We obtain qualitatively similar results under additional values of the parameter κ and alternative cost functions (refer to Appendix C.1).

Running time and number of iterations/rounds. To compare the running time of all the aforementioned algorithms, we consider the same configuration as in the performance evaluation with outcome monotonic and additive costs. Fig. 3.4a summarizes the results, which show several interesting insights. We find that brute force search quickly becomes computationally intractable. Moreover, we observe that the dynamic programming algorithm, is significantly faster than the iterative algorithm, making it the most efficient of the proposed algorithms. Recall that the complexity of one round in the dynamic programming algorithm and one iteration in the iterative algorithm is $\mathcal{O}(m^3)$. The results show that, although in theory, the dynamic programming algorithm needs $\mathcal{O}(m)$ rounds to terminate, in practice, it rarely needs more than two rounds. This is in contrast with the iterative algorithm which might need a larger number of iterations to converge, especially for large values of m. Overall, the above results let us conclude that, under outcome monotonic additive costs, the dynamic programming algorithm is a highly effective and efficient heuristic.

3.1.6 Experiments on real data

In this section, we evaluate our iterative algorithm using real credit card data. Since in our experiments, the cost individuals pay to change features is not always monotonic, we only experiment with our iterative algorithm.

Experimental setup. We use the publicly available *credit* dataset [174], which contains information about a bank's credit card payoffs.[10] For each accepted credit card holder, the respective dataset contains various demographic characteristics and

[10]We used a preprocessed version of the credit dataset by Ustun et al. [108].

financial status indicators which serve as features $\boldsymbol{X}$ and the current credit payoff status which serves as label Y. Among the features, we distinguish both numerical and discrete-valued features as well as actionable (*e.g.*, most recent bill amount) and non-actionable (*e.g.*, age group) features [108]. Refer to Appendix B.1.1 for more details on the specific features we used.

To approximate the conditional distribution $P(Y \mid \boldsymbol{X})$, we first cluster the credit card holders into k groups based on the original numerical features using k-means clustering[11] and then, for each credit card holder, we replace their initial numerical features with the respective identifier of the cluster they belong to, represented using a one-hot encoding. After this preprocessing step, the discrete feature values $\boldsymbol{x}_i$ consist of all possible value combinations of discrete non-actionable features and cluster identifiers. Then, we train four types of classifiers (multi-layer perceptron, support vector machine, logistic regression, decision tree) using `scikit-learn` [176] with default parameters. Finally, we choose the pair of classifier type and number of clusters k that maximizes accuracy, estimated using 5-fold cross validation, to approximate the values of $P(Y \mid \boldsymbol{X})$.[12]

To set the cost function $c(\boldsymbol{x}_i, \boldsymbol{x}_j)$ values, we use the *maximum percentile shift* [108]. More specifically, let $\mathcal{L}$ be the set of actionable (numerical) features and $\bar{\mathcal{L}}$ be the set of non-actionable (discrete-valued) features. Then, for each pair of feature values, we set the cost function $c(\boldsymbol{x}_i, \boldsymbol{x}_j)$ to:

$$c(\boldsymbol{x}_i, \boldsymbol{x}_j) = \begin{cases} \tau \cdot \max_{l \in \mathcal{L}} |q_l(x_{j,l}) - q_l(x_{i,l})| & \text{if } x_{i,l} = x_{j,l} \ \forall l \in \bar{\mathcal{L}} \\ \infty & \text{otherwise,} \end{cases} \tag{3.12}$$

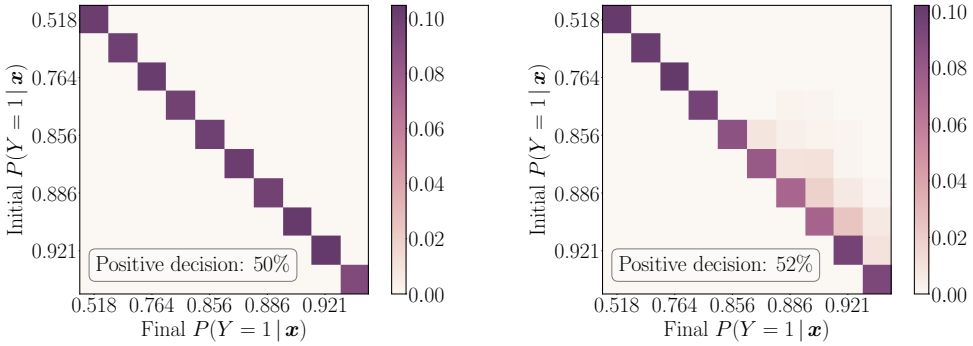
where $x_{j,l}$ is the value of the l-th feature for the feature value $\boldsymbol{x}_j$, $q_l(\cdot)$ is the CDF of the numerical feature $l \in \mathcal{L}$ and $\tau \geq 1$ is a scaling factor which controls the difficulty of changing features. As an exception, we always set the cost $c(\boldsymbol{x}_i, \boldsymbol{x}_j)$ between two feature values to ∞ if $q_l(x_{j,l}) < q_l(x_{i,l})$ for $l \in \{$Total overdue counts, Total months overdue$\}$, not allowing the history of overdue payments to be erased.

Finally, we set the parameter γ to the 50-th percentile of all the individuals' $P(Y = 1 \mid \boldsymbol{x})$, such that 50% of the population is accepted by the optimal threshold policy in the non strategic setting, and we compare the performance of the decision policies found by our iterative algorithm (*Iterative*) with two baselines: (i) *Non-Strategic*, the optimal deterministic threshold rule in a non-strategic setting (Eq. 3.9), and (ii) *Threshold*, the optimal deterministic threshold rule in a strategic setting found via bruteforce search over all deterministic threshold rules. Refer to Appendix B.1.2 for further details on the experimental setup.

Results. We first look into the transportation of mass induced by the decision policy found by our iterative algorithm for different τ values in Fig. 3.5. We observe that, as the cost of changing features increases, there is a higher transportation of mass towards feature values with the highest outcomes $P(Y = 1 \mid \boldsymbol{x})$. Moreover,

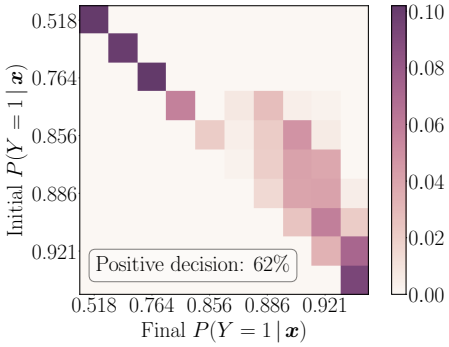
[11] One could use alternatives approaches to partition a feature space into a discrete set of regions (*e.g.*, such that a classifier is calibrated on each region [175]). However, our goal is not to advance the state of the art in calibration or clustering algorithms and, therefore, we resort to k-means clustering for simplicity.

[12] The best pair of classifier type and number of clusters k achieved an accuracy equal to 80.49%. We trained the same four types of classifiers on the raw (non-discretized) features, achieving a maximum accuracy equal to 80.59%, indicating that, in the given dataset, the discretization procedure causes negligible losses in terms of predictive accuracy.

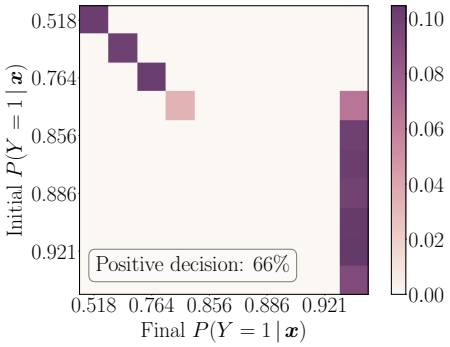


(a) Non-strategic setting

(b) Strategic setting, $\tau = 10$



(c) Strategic setting, $\tau = 3.3$

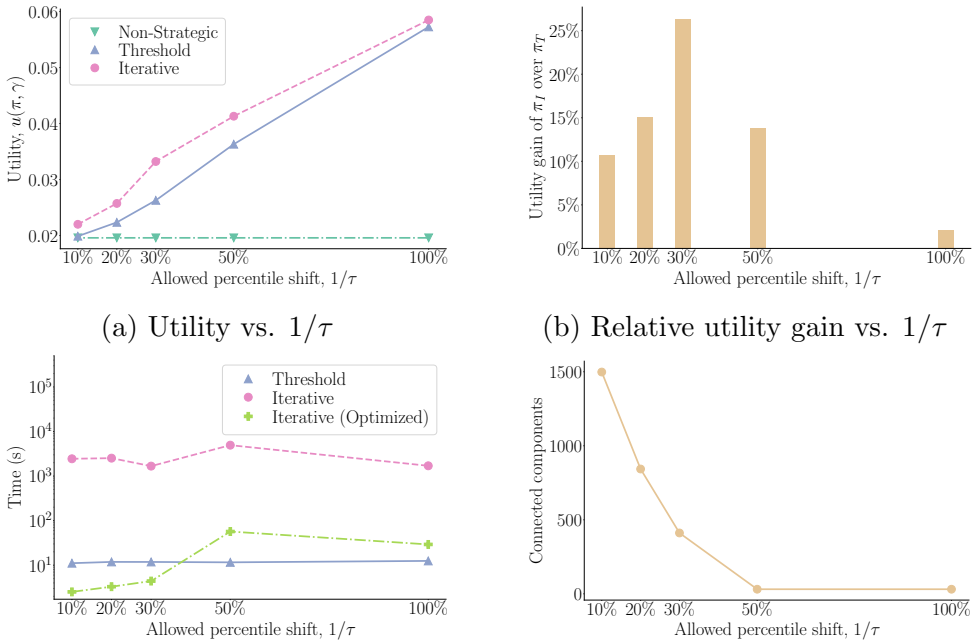


(d) Strategic setting, $\tau = 1$

Figure 3.5: **Transportation of mass in the credit dataset as induced by the policies found via the iterative algorithm (Algorithm 2).** For each individual in the population, we record their outcome $P(Y = 1 \,|\, \boldsymbol{x})$ before the best-response (Initial $P(Y = 1 \,|\, \boldsymbol{x})$) and after the best-response (Final $P(Y = 1 \,|\, \boldsymbol{x})$). Panel (a) shows the transportation of mass in the non-strategic setting, while panels (b-d) show the transportation of mass for several values of τ, which controls the difficulty of changing features. In each panel, the color illustrates the percentage of individuals with the corresponding initial and final $P(Y = 1 \,|\, \boldsymbol{x})$ values. The overlayed boxes indicate the percentage of the population that receives a positive decision, that is, the sum $\sum_{\boldsymbol{x}} \pi(\boldsymbol{x}')P(\boldsymbol{x})$, where $\boldsymbol{x}'$ is the best-response of the individuals with an initial feature value $\boldsymbol{x}$.

whenever individuals can arbitrarily change actionable features (*i.e.*, $\tau = 1$), the best-response of individuals is either feature values with the highest outcomes or their initial features if their recourse may be limited due to non-actionable features (*e.g.*, history of overdue payments). Finally, we observe that the decision policies found by our algorithm consistently lead to a higher number of individuals receiving a positive decision in comparison to the non-strategic setting, accross all τ values.

Next, we compare the utility of the decision policy found by our iterative algorithm and the policies found by the baselines. Here, we do not compare with the optimal (stochastic) decision policy because brute force search does not scale to the size of the dataset. Figs. 3.6a and 3.6b summarize the results for several values of the cost scaling factor τ, which show that the decision policy found by the iterative algorithm outperforms the baselines and, as the cost of changing features becomes smaller (*i.e.*, τ decreases), the utility value increases. Moreover, we observe that the decision policy given by the iterative algorithm achieves a significant relative gain



(a) Utility vs. $1/\tau$

(b) Relative utility gain vs. $1/\tau$

(c) Running time vs. $1/\tau$

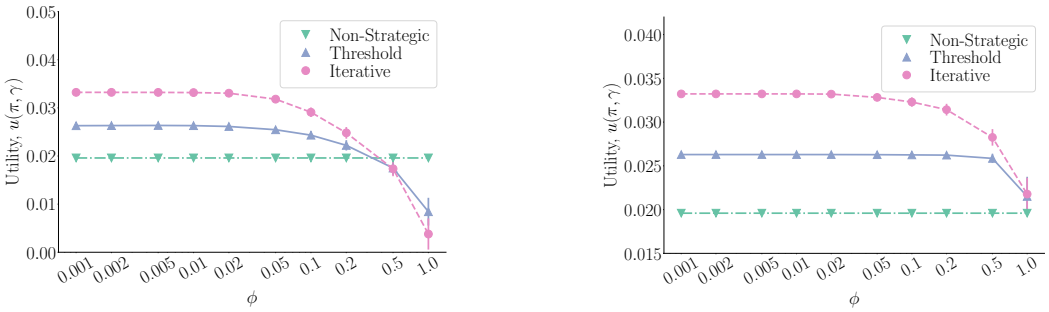
(d) # components vs. $1/\tau$

Figure 3.6: **Effectiveness and efficiency of the proposed algorithms.** Panel (a) shows the utility achieved by three types of decision policies in the credit dataset, against the value of the parameter τ, which controls how difficult it is for the individuals to change their features. Panel (b) shows the relative gain in utility achieved by the policy found via our iterative algorithm (π_I) in comparison with the policy found via the threshold baseline (π_T), against the value of the parameter τ. Panel (c) shows the running time of the threshold baseline algorithm and our iterative algorithm, with and without the speed-up discussed in Section 3.1.4. Panel (d) shows the number of connected components in the graph $\mathcal{G}$. In all panels, we repeat each experiment 100 times, and error bars indicate 95% confidence intervals. Note that, whenever we implement the iterative algorithm with the speed-up, we solve the subproblems corresponding to independent components sequentially, however, the procedure is amenable to parallelization.

in utility compared to the one given by the threshold baseline, especially when the individuals' ability to change their features is limited (*i.e.*, the cost scaling factor τ is large). We obtain qualitatively similar results under additional values of the parameter γ and one alternative cost function (refer to Appendix C.2).

Further, we compare the running time of the threshold baseline and the iterative algorithm with and without the speed-up that exploits the presence of non-actionable features, described in Section 3.1.4. Figs. 3.6c, 3.6d summarize the results. We observe that, whenever the cost to change features is high, there exist many independent connected components and the speed up provides a significant advantage. In those cases, the iterative algorithm with the speed-up performs faster than the threshold baseline while the running time of the two algorithms remains comparable, even when the cost to change features is low.

To conclude, we investigate to what extent the utilities of the decision policies found by our iterative algorithm and the threshold baseline are affected by (i) a misspecification of the conditional probabilities $P(Y = 1 \,|\, \boldsymbol{x})$ and the cost values



(a) Utility vs. ϕ under misspecified $P(Y = 1 \mid \boldsymbol{x})$

(b) Utility vs. ϕ under misspecified $c(\boldsymbol{x}_i, \boldsymbol{x}_j)$

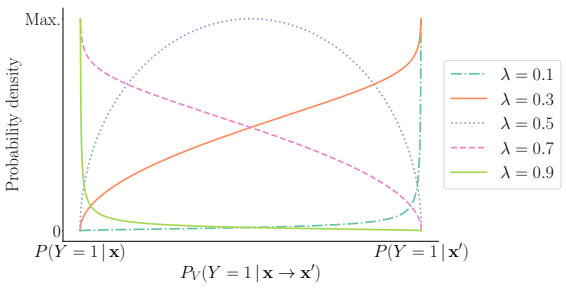
Figure 3.7: **Sensitivity of the proposed algorithms to misspecifications.** Panel (a) shows the utility achieved by three types of decision policies in the credit dataset against the amount of noise in the estimation of the conditional probabilities $P(Y = 1 \mid \boldsymbol{x})$. Panel (b) shows the utility achieved by the same three types of decision policies in the credit dataset against the amount of noise in the estimation of the cost values $c(\boldsymbol{x}_i, \boldsymbol{x}_j)$. In both panels, we set $\tau = 3.3$ and the horizontal axis shows the value of the parameter ϕ, which controls the level of noise (or misspecification). We repeat each experiment 100 times, and error bars indicate 95% confidence intervals.

$c(\boldsymbol{x}_i, \boldsymbol{x}_j)$, for example, due to imperfect estimations, and (ii) violations of our assumption of no unobserved confounding.

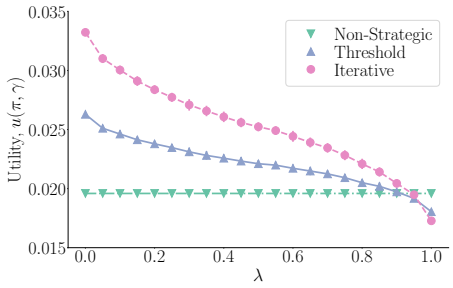
Regarding the misspecification of $P(Y = 1 \mid \boldsymbol{x})$, let σ_P and σ_c be the standard deviations of the values $\{P(Y = 1 \mid \boldsymbol{x}_l)\}$ for $l \in \{1, \ldots, m\}\}$ and $\{c(\boldsymbol{x}_i, \boldsymbol{x}_j)\}$ for $i, j \in [m] \times [m]$ such that $c(\boldsymbol{x}_i, \boldsymbol{x}_j) \neq \infty$, respectively. To model the case of misspecified $P(Y = 1 \mid \boldsymbol{x})$ values, for each feature value $\boldsymbol{x}$, we provide as input to the two algorithms a distorted value $\tilde{P}(Y = 1 \mid \boldsymbol{x}) = P(Y = 1 \mid \boldsymbol{x}) + \epsilon$, where ϵ is a random noise sampled from a Gaussian distribution with mean 0 and standard deviation $\phi \cdot \sigma_P$ and $\phi \in (0, 1)$ controls the level of misspecification. Then, we truncate the values $\tilde{P}(Y = 1 \mid \boldsymbol{x})$ at 0 and 1, to make sure they are valid probabilities. For the case of misspecified cost values, we follow a similar approach where, we introduce an additive noise term ϵ to each cost value $c(\boldsymbol{x}_i, \boldsymbol{x}_j)$, sampled from a Gaussian distribution with mean 0 and standard deviation $\phi \cdot \sigma_c$, truncating the distorted values $\tilde{c}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ at 0. Fig. 3.7 summarizes the results for several values of the scaling factor ϕ. We observe that, in both cases, the utility of the decision policy found by our iterative algorithm drops as the values of $\tilde{P}(Y = 1 \mid \boldsymbol{x})$ and $\tilde{c}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ become more distorted. However, unless the level of misspecification is exceptionally high, our algorithm's decision policy outperforms both the policy given by the threshold baseline and the optimal policy in the non-strategic setting.

Regarding violations of the assumption of no unobserved confounding, we refrain from making domain-specific causal modeling assumptions, and we focus on a general model of confounding that allows us to control the balance between gaming and improvement whenever individuals change their features from a value $\boldsymbol{x}$ to a value $\boldsymbol{x}'$. More specifically, whenever individuals change their features from $\boldsymbol{x}$ to $\boldsymbol{x}'$, we assume that their labels Y are sampled from a *confounded* conditional distribution $P_V(Y \mid \boldsymbol{x} \to \boldsymbol{x}')$ with

$$P_V(Y = 1 \mid \boldsymbol{x} \to \boldsymbol{x}') = V_{\boldsymbol{x},\boldsymbol{x}'} \cdot P(Y = 1 \mid \boldsymbol{x}') + (1 - V_{\boldsymbol{x},\boldsymbol{x}'}) \cdot P(Y = 1 \mid \boldsymbol{x}),$$



(a) Probability of $Y = 1$ after a change from $\boldsymbol{x}$ to $\boldsymbol{x}'$



(b) Utility vs. λ

Figure 3.8: **Sensitivity of the proposed algorithms to unobserved confounding.** Panel (a) shows the probability density function of the *confounded* conditional probability $P_V(Y = 1 \,|\, \boldsymbol{x} \to \boldsymbol{x}')$ for different levels of unobserved confounding where, to facilitate visibility, we scale all probability density functions to have the same maximum value. Panel (b) shows the utility achieved by three types of decision policies in the credit dataset against the level of unobserved confounding, where we set $\tau = 3.3$ and repeat each experiment 100 times, with error bars indicating 95% confidence intervals.

where $V_{\boldsymbol{x},\boldsymbol{x}'} \sim \text{Beta}(\alpha(\lambda), \beta(\lambda))$ and $\lambda \in [0, 1]$ is a parameter controlling the level of confounding. Fig. 3.8a shows the probability density of $P_V(Y = 1 \,|\, \boldsymbol{x} \to \boldsymbol{x}')$ for different λ values. As $\lambda \to 0$, the distribution of $P_V(Y = 1 \,|\, \boldsymbol{x} \to \boldsymbol{x}')$ is more concentrated towards $P(Y = 1 \,|\, \boldsymbol{x}')$ and thus our assumption of no observed confounding becomes (approximately) valid. As $\lambda \to 1$, the distribution of $P_V(Y = 1 \,|\, \boldsymbol{x} \to \boldsymbol{x}')$ is more concentrated towards $P(Y = 1 \,|\, \boldsymbol{x})$ and feature changes do not modify an individual's outcome, matching the setting studied by early work on strategic classification [85, 92, 93]. Refer to Appendix B.1.3 for more details about the specific functional form we used to define $\alpha(\lambda)$ and $\beta(\lambda)$. Fig. 3.8b shows the utility of the decision policy found by our iterative algorithm and the policies found by the baselines for different λ values. As one may have expected, we observed that, as the level of unobserved confounding increases, the utility of the decision policy found by our iterative algorithm drops. However, unless the observed feature changes correspond almost always to gaming (*i.e.*, $\lambda \approx 1$), the decision policy given by the iterative algorithm maintains its competitive advantage in comparison to the policy given by the threshold baseline and the optimal policy in the non-strategic setting.

3.2 Decision making under partial transparency

Although the model described in the previous section is general enough to apply to various real-life scenarios, decision makers may be reluctant to reveal their entire policy to the public due to reasons such as trade secrets [111]. In practice, however, decision makers who use predictive models to make decisions are not required to publish their full policies. Instead, they have to provide explanations to individuals regarding the decisions they receive—for instance, there already exists a legal requirement in the European Union to grant individuals, subject to (semi)-automated decision making, the right-to-explanation [177, 178]. Consider a bank that denies a

loan to an applicant based on a predictive model's estimate of the likelihood that the applicant repays the loan. The bank can meet the requirement for providing explanations by indicating which features the applicant needs to change and by how much (*e.g.*, increase their income by $5,000) to improve their financial situation and become eligible for a loan.

In response to increasing calls for transparency in the use of machine learning models in high-stakes decision making, there has been a surge in work on explainable machine learning [104–109, 179, 180], with particular emphasis on *counterfactual explanations* [107–109, 180]. Given a negatively classified data point, these explanations explain that individual prediction by identifying an alternative data point that, although classified positively, differs minimally in the feature space from the negatively classified one. However, these works do not distinguish between decisions and predictions. Consequently, they cannot be readily used to provide explanations for decisions taken by a decision maker (informed by a predictive model), which is ultimately what individuals who are subject to (semi)-automated decision making typically care about.

Similarly to the previous section, we build upon a recent line of work that explicitly distinguishes between predictions and decisions [48–50, 169, 181] and introduce methods to find counterfactual explanations for decisions made by a decision maker informed by a data-driven predictive model. These explanations serve as actionable recommendations that help individuals understand what they would have to change to receive a beneficial decision, rather than a positive prediction. In this context, we highlight that individuals may use the information gained through the explanations they receive to invest effort strategically and maximize their chances of receiving a beneficial decision, an aspect overlooked in previous work on explainable machine learning. Then, similarly to the previous section, we use that as an opportunity to find decision policies and counterfactual explanations that maximize the utility of the decision maker while incentivizing individuals to self-improve.

We extend the model introduced in Section 3.1 and cast the above problem as a Stackelberg game in which the decision maker moves first and shares their counterfactual explanations before individuals best-respond to these explanations and invest effort to receive a beneficial decision. Similarly as before, we assume that the decision maker takes decisions based on low-dimensional feature vectors, so that the decision policies are relatively easy to evaluate before implementation. Under this problem formulation, we make the following contributions:

1. We show that, given a predefined policy, the problem of finding the optimal set of counterfactual explanations is NP-hard by using a novel reduction of the Set Cover problem [168].

2. We show that the corresponding objective function is monotone and submodular, and, as a direct consequence, it readily follows that a standard greedy algorithm offers approximation guarantees [182].

3. Given a predefined set of counterfactual explanations, we show that the optimal policy is deterministic and can be computed in polynomial time. Building on this result, we can reduce the problem of jointly finding both the optimal policy and set of counterfactual explanations to maximizing a non-monotone submodular function, a problem that can also be solved with approximation guarantees [183].

4. We demonstrate that, by incorporating a matroid constraint into the problem formulation, we can increase the diversity of the optimal set of counterfactual explanations and incentivize individuals across the whole spectrum of the population to self-improve.

Experiments using real lending and credit card data illustrate our theoretical findings and show that the counterfactual explanations and decision policies found by the above algorithms achieve higher utility than several competitive baselines.

3.2.1 A game-theoretic model of counterfactual explanations

Here, we introduce a model that extends the one introduced in Section 3.1 to capture the ability of a decision maker to provide counterfactual explanations instead of publishing their entire policy. An individual with a feature vector $\boldsymbol{x} \in \{1, \dots, n\}^d$ has a (stochastic) label $Y \in \{0, 1\}$ and a decision $D \in \{0, 1\}$ controls whether the corresponding label is realized. For example, in university admissions, the decision specifies whether a student is admitted ($D = 1$) or rejected ($D = 0$), the label indicates whether the student completes the program ($Y = 1$) or drops out ($Y = 0$) upon acceptance, and the feature vector $\boldsymbol{x}$ may include their GRE scores, undergraduate GPA percentile, or research experience. Going forward, we will denote the set of feature values as $\mathcal{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_m\}$, where $m = n^d$ and assume that the number of features d is small, as discussed previously.

Each decision is sampled from a decision policy $D \sim \pi(D \,|\, \boldsymbol{x})$, where, for brevity, we will write $\pi(\boldsymbol{x}) = \pi(D = 1 \,|\, \boldsymbol{x})$. For each individual, the label Y is sampled from a conditional probability distribution $P(Y \,|\, \boldsymbol{X})$ and, without loss of generality, we index the feature values in decreasing order with respect to their corresponding outcome, that is, $i < j \Rightarrow P(Y = 1 \,|\, \boldsymbol{x}_i) \geq P(Y = 1 \,|\, \boldsymbol{x}_j)$.

Similarly as in Section 3.1, we adopt a Stackelberg game-theoretic formulation in which each individual with initial feature value $\boldsymbol{x}_i$ receives a counterfactual explanation from the decision maker by means of a feature value $\mathsf{e}(\boldsymbol{x}_i) \in \mathcal{A}$ before they (best-)respond, where $\mathcal{A} \subseteq \mathcal{P}_\pi$ is a set of counterfactual explanations and $\mathcal{P}_\pi = \{\boldsymbol{x} \in \mathcal{X} : \pi(\boldsymbol{x}) = 1\}$.[13] This formulation fits a variety of real-world applications. For example, insurance companies often provide online car insurance simulators that, on the basis of a customer's initial feature value $\boldsymbol{x}_i$, let the customer know whether they are eligible for a particular deal. In case the customer does not qualify, the simulator could provide a counterfactual explanation $\mathsf{e}(\boldsymbol{x}_i)$ under which the individual is guaranteed to be eligible. In the remainder, for each individual with initial feature value $\boldsymbol{x}_i$, we assume they do not know anything about the other counterfactual explanations $\mathcal{A} \backslash \mathsf{e}(\boldsymbol{x}_i)$ other individuals may receive nor the decision policy $\pi(\boldsymbol{x})$.

Now, let $c(\boldsymbol{x}, \mathsf{e}(\boldsymbol{x}_i))$ be the cost an individual pays for changing from $\boldsymbol{x}_i$ to $\mathsf{e}(\boldsymbol{x}_i)$ and $b(\boldsymbol{x}) = \mathbb{E}_{D \sim \pi(D \,|\, \boldsymbol{x})}[D]$ be the (immediate) benefit they obtain from a policy π, which is equal to the probability that the individual receives a positive decision. Then, each individual's best-response is to change from their initial feature value $\boldsymbol{x}_i$ to $\mathsf{e}(\boldsymbol{x}_i)$ iff the gained benefit they would obtain outweighs the cost they would pay

[13] In practice, individuals with initial feature values $\boldsymbol{x}_i$ such that $\pi(\boldsymbol{x}) = 1$ may not receive any explanation since they are guaranteed to receive a positive decision.

for changing features, that is,

$$\mathsf{e}(\boldsymbol{x}_i) \in \mathcal{R}(\boldsymbol{x}_i) = \{\boldsymbol{x}_j \in \mathcal{X} \,:\, b(\boldsymbol{x}_j) - c(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq b(\boldsymbol{x}_i)\},$$

and it is to keep their initial feature value $\boldsymbol{x}_i$ otherwise. Here, we will refer to $\mathcal{R}(\boldsymbol{x}_i)$ as the *region of adaptation*. Then, at a population level, the above best-response results into a transportation of mass between the original feature distribution $P(\boldsymbol{X})$ and a new feature distribution $P(\boldsymbol{X}; \pi, \mathcal{A})$ induced by the policy π and the set of counterfactual explanations $\mathcal{A}$. More specifically, we can readily derive an analytical expression for the induced feature distribution in terms of the original feature distribution, that is, for all $\boldsymbol{x}_j \in \mathcal{X}$,

$$P(\boldsymbol{x}_j; \pi, \mathcal{A}) = P(\boldsymbol{x}_j)\mathbb{1}\left[\mathcal{R}(\boldsymbol{x}_j) \cap \mathcal{A} = \emptyset\right] + \sum_{i \in [m]} P(\boldsymbol{x}_i)\mathbb{1}\left[(\mathsf{e}(\boldsymbol{x}_i) = \boldsymbol{x}_j \wedge \boldsymbol{x}_j \in \mathcal{R}(\boldsymbol{x}_i)\right].$$

Similarly as in Section 3.1, we assume that there are no unobserved confounders (*i.e.*, $P(Y \mid \boldsymbol{X})$ does not change). Moreover, we assume that the decision maker has access to (an estimation of) the original feature distribution $P(\boldsymbol{X})$, and aims to maximize the (immediate) utility $u(\pi, \gamma, \mathcal{A})$, which is the expected overall profit they obtain, that is,

$$\begin{aligned} u(\pi, \gamma, \mathcal{A}) &= \mathbb{E}_{\boldsymbol{x} \sim P(\boldsymbol{X}; \pi, \mathcal{A}), Y \sim P(Y \mid \boldsymbol{X}), D \sim \pi(\boldsymbol{X})}\left[Y \cdot D - \gamma \cdot D\right] \\ &= \mathbb{E}_{\boldsymbol{X} \sim P(\boldsymbol{X}; \pi, \mathcal{A})}\left[\pi(\boldsymbol{X})(P(Y = 1 \mid \boldsymbol{X}) - \gamma)\right], \end{aligned} \tag{3.13}$$

where $\gamma \in (0, 1)$ is a given constant reflecting economic considerations of the decision maker. For example, in university admissions, the term $\pi(\boldsymbol{X})P(Y = 1 \mid \boldsymbol{X})$ is proportional to the expected number of students who are admitted and complete the program, the term $\pi(\boldsymbol{X})\gamma$ is proportional to the number of students who are admitted, and γ measures the cost of education in units of graduated students. As a direct consequence, given a feature value $\boldsymbol{x}_i$ and a set of counterfactual explanations $\mathcal{A}$, we can conclude that, if $\mathcal{R}(\boldsymbol{x}_i) \cap \mathcal{A} \neq \emptyset$, the decision maker will decide to provide the counterfactual explanation $\mathsf{e}(\boldsymbol{x}_i)$ that provides the largest utility gain under the assumption that individuals best-respond, that is,

$$\mathsf{e}(\boldsymbol{x}_i) = \underset{\boldsymbol{x} \in \mathcal{A} \cap \mathcal{R}(\boldsymbol{x}_i)}{\operatorname{argmax}} \; P(Y = 1 \mid \boldsymbol{x}) \text{ for all } \boldsymbol{x}_i \in \mathcal{X} \setminus \mathcal{P}_\pi \text{ such that } \mathcal{R}(\boldsymbol{x}_i) \cap \mathcal{A} \neq \emptyset, \tag{3.14}$$

and, if $\mathcal{R}(\boldsymbol{x}_i) \cap \mathcal{A} = \emptyset$, we arbitrarily assume that $\mathsf{e}(\boldsymbol{x}_i) = \operatorname{argmin}_{\boldsymbol{x} \in \mathcal{A}} c(\boldsymbol{x}_i, \boldsymbol{x})$.[14]

Given the above preliminaries, our goal is to help the decision maker to first find the optimal set of counterfactual explanations $\mathcal{A}$ for a pre-defined policy in Section 3.2.2 and then both the optimal policy π and set of counterfactual explanations $\mathcal{A}$ in Section 3.2.3.

Remarks. Given an individual with initial feature value $\boldsymbol{x}$, one may think that, by providing the counterfactual explanation $\mathsf{e}(\boldsymbol{x}) \in \mathcal{A} \cap \mathcal{R}(\boldsymbol{x})$ that gives the largest utility gain, the decision maker is not acting in the individual's best interest but rather selfishly. This is because there may exist another counterfactual explanation $\mathsf{e}_m(\boldsymbol{x}) \in \mathcal{A} \cap \mathcal{R}(\boldsymbol{x})$ with lower cost for the individual, that is, $c(\boldsymbol{x}, \mathsf{e}_m(\boldsymbol{x})) \leq c(\boldsymbol{x}, \mathsf{e}(\boldsymbol{x}))$. In our work, we argue that the provided counterfactual explanations help

[14] Note that, if $\mathcal{A} \cap \mathcal{R}(\boldsymbol{x}_i) = \emptyset$, the individual's best-response is to keep their initial feature value $\boldsymbol{x}_i$ and thus any choice of counterfactual explanation $\mathsf{e}(\boldsymbol{x}_i)$ leads to the same utility.

the individual to achieve a greater self-improvement and this is likely to result in a superior long-term well-being. For example, consider a bank issuing credit cards who wants to maintain credit for trustworthy customers and incentivize the more risky ones to improve their financial status. In this case, $\mathsf{e}(\boldsymbol{x})$ is the explanation that maximally improves the financial status of the individual, making the repayment more likely, but requires them to pay a larger (immediate) cost. In contrast, $\mathsf{e}_m(\boldsymbol{x})$ is an alternate explanation that requires the individual to pay a smaller (immediate) cost but, in comparison with $\mathsf{e}(\boldsymbol{x})$, would result in a higher risk of default. In this context, note that the individual would be "willing" to pay the cost of following either $\mathsf{e}(\boldsymbol{x})$ or $\mathsf{e}_m(\boldsymbol{x})$ since both explanations lie within the region of adaptation $\mathcal{R}(\boldsymbol{x})$. Refer to Section 3.2.6 for an anecdotal real-world example of $\mathsf{e}(\boldsymbol{x})$ and $\mathsf{e}_m(\boldsymbol{x})$.

3.2.2 Finding the optimal counterfactual explanations for a policy

In this section, our goal is to find the optimal set of counterfactual explanations $\mathcal{A}^*$ for a pre-defined policy π, that is,

$$\mathcal{A}^* = \underset{\mathcal{A} \subseteq \mathcal{P}_\pi \,:\, |\mathcal{A}| \leq k}{\operatorname{argmax}} \; u(\pi, \gamma, \mathcal{A}), \tag{3.15}$$

where the cardinality constraint on the set of counterfactual explanations balances the decision maker's obligation to be transparent with trade secrets [111]. More specifically, note that, without this constraint, an adversary could reverse-engineer the entire decision policy $\pi(\boldsymbol{x})$ by impersonating individuals with different feature values $\boldsymbol{x}$ [184]. As it will become clearer in the experimental evaluation in Sections 3.2.5 and 3.2.6, our results may persuade decision makers to be transparent about their decision policies, something they are typically reluctant to be, despite the increasing legal requirements, since we show that transparency increases the utility of the policies.

Throughout this section, we assume that the aim of the decision maker who picks the pre-defined policy is to maximize their utility and, therefore, $\pi(\boldsymbol{x}) = 0$ for all $\boldsymbol{x} \in \mathcal{X}$ such that $P(Y = 1 \,|\, \boldsymbol{x}) < \gamma$. Moreover, we assume that the policy is outcome monotonic (see Eq. 3.1). Outcome monotonicity just implies that the higher an individual's outcome $P(Y = 1 \,|\, \boldsymbol{x})$, the higher their chances of receiving a positive decision $\pi(\boldsymbol{x})$.[15]

Unfortunately, using a novel reduction of the Set Cover problem [168], the following theorem reveals that we cannot expect to find the optimal set of counterfactual explanations in polynomial time:[16]

Theorem 3.2.1. *The problem of finding the optimal set of counterfactual explanations that maximizes utility under a cardinality constraint is NP-Hard.*

Even though Theorem 3.2.1 is a negative result, we will now show that the objective function in Eq. 3.15 satisfies a set of desirable properties, specifically, non-negativity, monotonicity and submodularity, which allow a standard greedy algorithm to enjoy approximation guarantees at solving the problem. To this end,

[15] If the policy π is deterministic, our results also hold for non outcome monotonic policies.

[16] All proofs for Section 3.2 can be found in Appendix A.2.

Algorithm 3: Standard greedy algorithm [182]

input : ground set of counterfactual explanations $\mathcal{P}_\pi$, parameter k, and utility function f

output : set of counterfactual explanations $\mathcal{A}$

$\mathcal{A} \leftarrow \varnothing$

while $|\mathcal{A}| \leq k$ **do**

  $\boldsymbol{x}^* \leftarrow \text{argmax}_{\boldsymbol{x} \in \mathcal{P}_\pi \setminus \mathcal{A}} \{f(\mathcal{A} \cup \{\boldsymbol{x}\}) - f(\mathcal{A})\}$

  $\mathcal{A} \leftarrow \mathcal{A} \cup \{\boldsymbol{x}^*\}$ `// Add the feature value` x `that maximizes the marginal difference of` f

return $\mathcal{A}$

with a slight abuse of notation, we first express the objective function as a set function $f(\mathcal{A}) = u(\pi, \gamma, \mathcal{A})$, which takes values over the ground set of counterfactual explanations $\mathcal{P}_\pi$. Then, we have the following proposition:

Proposition 3.2.1. *The function f is non-negative, submodular and monotone. Formally, all three of the following conditions are satisfied:*

1. *$f(\mathcal{A}) \geq 0$ for all $\mathcal{A} \subseteq \mathcal{P}_\pi$.*
2. *For all $\mathcal{A}, \mathcal{B} \subseteq \mathcal{P}_\pi : \mathcal{A} \subseteq \mathcal{B}$ and $\boldsymbol{x} \in \mathcal{P}_\pi \setminus \mathcal{B}$, it holds that $f(\mathcal{A} \cup \{\boldsymbol{x}\}) - f(\mathcal{A}) \geq f(\mathcal{B} \cup \{\boldsymbol{x}\}) - f(\mathcal{B})$.*
3. *For all $\mathcal{A} \subseteq \mathcal{P}_\pi$ and $\boldsymbol{x} \in \mathcal{P}_\pi$, it holds that $f(\mathcal{A} \cup \{\boldsymbol{x}\}) \geq f(\mathcal{A})$.*

The above result directly implies that the standard greedy algorithm [182] for maximizing a non-negative, submodular and monotone function will find a solution $\mathcal{A}$ to the problem such that $f(\mathcal{A}) \geq (1 - 1/e) f(\mathcal{A}^*)$, where $\mathcal{A}^*$ is the optimal set of counterfactual explanations. The algorithm starts from a solution set $\mathcal{A} = \emptyset$ and it iteratively adds to $\mathcal{A}$ the counterfactual explanation $\boldsymbol{x} \in \mathcal{P}_\pi \setminus \mathcal{A}$ that provides the maximum marginal difference $f(\mathcal{A} \cup \{\boldsymbol{x}\}) - f(\mathcal{A})$. Algorithm 3 provides a pseudocode implementation of the algorithm.

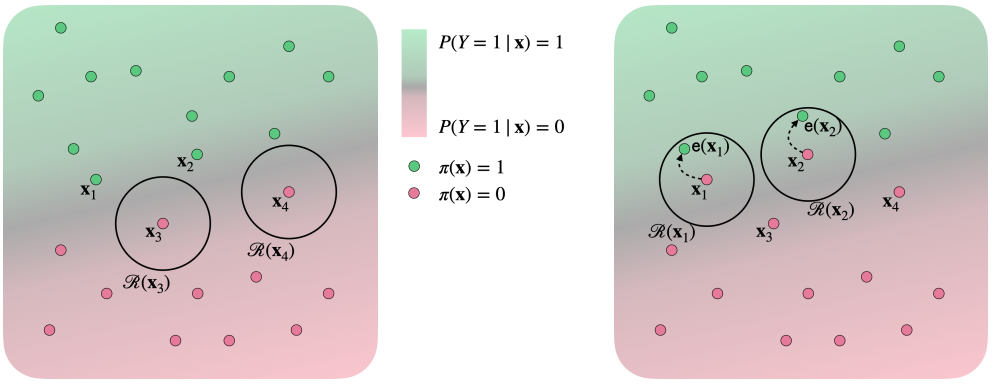
Finally, note that the greedy algorithm computes the marginal difference of f for at most m elements per iteration and, following from the proof of Proposition 3.2.1, the marginal difference $f(\mathcal{A} \cup \{\boldsymbol{x}\}) - f(\mathcal{A})$ can be computed in $\mathcal{O}(m)$. Therefore, it immediately follows that, in our problem, the greedy algorithm has an overall complexity of $\mathcal{O}(km^2)$.

3.2.3 Finding the optimal policy and counterfactual explanations

In this section, our goal is to jointly find the optimal decision policy and set of counterfactual explanations $\mathcal{A}^*$, that is,

$$\pi^*, \mathcal{A}^* = \underset{(\pi, \mathcal{A}) : \mathcal{A} \subseteq \mathcal{P}_\pi \wedge |\mathcal{A}| \leq k}{\text{argmax}} u(\pi, \gamma, \mathcal{A}), \tag{3.16}$$

where, similarly as in the previous section, k is the maximum number of counterfactual explanations the decision maker is willing to provide to the population to balance the right to explanation with trade secrets. By jointly optimizing both the



Non-strategic policy Strategic policy

Figure 3.9: **Jointly optimizing the decision policy and the counterfactual explanations can offer additional gains.** The left panel shows the optimal (deterministic) decision policy π in the non-strategic setting, as given by Eq. 3.18. Here, there does not exist a set of counterfactual explanations $\mathcal{A} \in \mathcal{P}_\pi$ that increases the utility of the policy. This happens because the area of adaption of $\boldsymbol{x}_3$ and $\boldsymbol{x}_4$ does not include any feature value that receives a positive decision. The right panel shows the decision policy and counterfactual explanations that are (jointly) optimal in terms of utility, as given by Eq. 3.16. Here, the individuals with feature values $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ receive $\mathsf{e}(\boldsymbol{x}_1)$ and $\mathsf{e}(\boldsymbol{x}_2)$, respectively, as counterfactual explanations. Since these explanations are within their areas of adaptation $\mathcal{R}(\boldsymbol{x}_1)$ and $\mathcal{R}(\boldsymbol{x}_2)$, they change their initial feature values in order to receive a positive decision.

decision policy and the counterfactual explanations, we may obtain an additional gain in terms of utility in comparison with just optimizing the set of counterfactual explanations given the optimal decision policy in a non-strategic setting. For a visual illustration to understand the intuition behind this, refer to Fig. 3.9. Moreover, as we will show in the experimental evaluation in Section 3.2.6, this additional gain will be significant.

Similarly as in Section 3.2.2, we cannot expect to find the optimal policy and set of counterfactual explanations in polynomial time. More specifically, we have the following negative result, which easily follows from Proposition 3.2.2 and slightly extending the proof of Theorem 3.2.1:

Theorem 3.2.2. *The problem of jointly finding both the optimal policy and the set of counterfactual explanations that maximize utility under a cardinality constraint is NP-hard.*

However, while the problem of finding both the policy and the set of counterfactual explanations appears significantly more challenging than the problem of finding just the set of counterfactual explanations given a pre-defined policy (*i.e.*, the problem given in Eq. 3.15), the following proposition shows that the problem is not inherently *harder*. More specifically, for each possible set of counterfactual explanations, it shows that the policy that maximizes the utility can be easily computed.

Proposition 3.2.2. *Let* $\mathcal{Y} = \{\boldsymbol{x} \in \mathcal{X} : P(Y = 1 \,|\, \boldsymbol{x}) \geq \gamma\}$. *Given a set of counterfactual explanations* $\mathcal{A} \subseteq \mathcal{Y}$[17], *the policy* $\pi^*_{\mathcal{A}} = \operatorname{argmax}_{\pi:\mathcal{A}\subseteq\mathcal{P}_\pi} u(\pi, \gamma, \mathcal{A})$ *is*

[17]Note that, since the decision maker is rational, they will never provide an explanation that contributes negatively to their utility.

deterministic, can be found in polynomial time, and is given by

$$\pi^*_{\mathcal{A}}(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \Big(\{\boldsymbol{x}' \in \mathcal{A} : P(Y=1 \,|\, \boldsymbol{x}') > P(Y=1 \,|\, \boldsymbol{x}) \wedge c(\boldsymbol{x}, \boldsymbol{x}') \leq 1\} \\ & \qquad = \emptyset \wedge \boldsymbol{x} \in \mathcal{Y}\Big) \vee \boldsymbol{x} \in \mathcal{A} \\ 0 & \text{otherwise.} \end{cases} \tag{3.17}$$

Intuitively, the optimal policy assigns $\pi^*_{\mathcal{A}}(\boldsymbol{x}) = 1$ if $\boldsymbol{x}$ is required to serve as a counterfactual explanation (*i.e.*, $\boldsymbol{x} \in \mathcal{A}$), or if all counterfactual explanations in $\mathcal{A}$ are not within the region of adaptation of $\boldsymbol{x}$ or adapting to them would not improve the individual's outcome. Proposition 3.2.2 implies that, to set all the values of the optimal decision policy, we only need to perform $\mathcal{O}(km)$ comparisons. Moreover, it reveals that, in contrast with the non strategic setting, the optimal policy given a set of counterfactual explanations is not a deterministic threshold rule with a single threshold [48, 169], that is,

$$\pi(\boldsymbol{x}) = \begin{cases} 1 & \text{if } P(Y=1 \,|\, \boldsymbol{x}) \geq \gamma \\ 0 & \text{otherwise,} \end{cases} \tag{3.18}$$

but rather a more conservative deterministic decision policy that does not depend only on the outcome $P(Y=1 \,|\, \boldsymbol{x})$ and γ but also on the cost individuals pay to change features. Moreover, we can build upon the above result to prove that the problem of finding the optimal decision policy and set of counterfactual explanations can be reduced to maximizing a non-monotone submodular function. To this aim, let $\pi^*_{\mathcal{A}}$ be the optimal policy induced by a given set of counterfactual explanations $\mathcal{A}$, as in Proposition 3.2.2, and define the set function $h(\mathcal{A}) = u(\pi^*_{\mathcal{A}}, \gamma, \mathcal{A})$ over the ground set $\mathcal{Y}$. Then, we have the following proposition:

Proposition 3.2.3. *The function h is non-negative, submodular and non-monotone.*

Fortunately, there exist efficient algorithms with global approximation guarantees for maximizing a non-monotone submodular function under cardinality constraints. In our work, we use the randomized polynomial time algorithm by Buchbinder et al. [183], which can find a solution $\mathcal{A}$ such that $h(\mathcal{A}) \geq (1/e)h(\mathcal{A}^*)$, where $\mathcal{A}^*$ and $\pi^*_{\mathcal{A}^*}$ are the optimal set of counterfactual explanations and decision policy, respectively (*i.e.*, the solutions to the problem given by Eq. 3.16). The algorithm is just a randomized variation of the standard greedy algorithm. It starts from a solution set $\mathcal{A} = \emptyset$ and it iteratively adds one counterfactual explanation $\boldsymbol{x} \in \mathcal{Y} \backslash \mathcal{A}$. However, instead of greedily choosing the element $\boldsymbol{x}$ that provides the maximum marginal difference $h(\mathcal{A} \cup \{\boldsymbol{x}\}) - h(\mathcal{A})$, it sorts all the candidate elements with respect to their marginal difference and picks one at random among the top k. Algorithm 4 provides a pseudocode implementation of the algorithm.

To enjoy a $1/e$ approximation guarantee, Algorithm 4 requires that there are $2k < m$ candidate feature values whose marginal contribution to any set is zero. In our problem, this can be trivially satisfied by adding $2k$ feature values $\boldsymbol{x}$ to $\mathcal{X}$ such that $P(Y=1 \,|\, \boldsymbol{x}) = \gamma$, $P(\boldsymbol{x}) = 0$ and $c(\boldsymbol{x}, \boldsymbol{x}_j) = c(\boldsymbol{x}_j, \boldsymbol{x}) = 2 \; \forall \boldsymbol{x}_j \in \mathcal{X}$. If the algorithm adds some of those counterfactual explanations to the set $\mathcal{A}$, it is easy to see that we can ignore them without causing any difference in utility or best-responses.

Algorithm 4: Randomized algorithm by Buchbinder et al. [183]

input : ground set of counterfactual explanations $\mathcal{Y}$, parameter k, and utility function h

output : set of counterfactual explanations $\mathcal{A}$

$\mathcal{A} \leftarrow \varnothing$

while $|\mathcal{A}| \leq k$ **do**

  $\mathcal{B} \leftarrow$ `get_top_k`$(\mathcal{Y}, \mathcal{A}, h)$

  $\boldsymbol{x}^* \sim \mathcal{B}$

  $\mathcal{A} \leftarrow \mathcal{A} \cup \{\boldsymbol{x}^*\}$ `// Add a feature value` x^* `sampled from the top-k in terms of marginal increase of` h

return $\mathcal{A}$

Finally, note that, following from the proof of Proposition 3.2.3, the marginal difference of h can be computed in $\mathcal{O}(m)$. Therefore, since the above randomized algorithm has a complexity of $\mathcal{O}(km)$, it readily follows that, in our problem, the algorithm has an overall complexity of $\mathcal{O}(km^2)$.

3.2.4 Increasing the diversity of counterfactual explanations

In many cases, decision makers may like to ensure that individuals across the whole spectrum of the population are incentivized to self-improve. For example, in a loan scenario, the bank may use the age group as a feature to estimate the probability that a customer repays the loan, however, it may like to deploy a decision policy that incentivizes individuals across all age groups in order to improve the financial situation of all. To this aim, the decision maker can increase the diversity of the optimal set of counterfactual explanations by incorporating a matroid constraint into the problem formulation, rather than a cardinality constraint.

Formally, consider disjoint sets $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_l$ such that $\bigcup_i \mathcal{X}_i = \mathcal{X}$ and integers $d_1, d_2, \ldots, d_l$ such that $k = \sum_i d_i$. Then, a partition matroid is the collection of sets $\{S \subseteq 2^{\mathcal{X}} : |S \cap \mathcal{X}_i| \leq d_i \ \forall i \in [l]\}$. In the loan example, the decision maker could search for a set of counterfactual explanations $\mathcal{A}$ within a partition matroid where each one of the $\mathcal{X}_i$'s corresponds to the feature values covered by each age group and $d_i = k/l \ \forall i \in [l]$. This way, the set of counterfactual explanations $\mathcal{A}$ would include explanations for every age group.

In this case, the decision maker could rely on a variety of polynomial time algorithms with global guarantees for submodular function maximization under matroid constraints, for example, the algorithm by Calinescu et al. [185].

3.2.5 Experiments on synthetic data

In this section, we evaluate Algorithms 3 and 4 using synthetic data and show that the counterfactual explanations and decision policies found by our algorithms achieve higher utility than several competitive baselines.[18]

Experimental setup. For simplicity, we consider feature values $\boldsymbol{x} \in \{1, \ldots, m\}$ and $P(\boldsymbol{x} = i) = p_i / \sum_j p_j$ where p_i is sampled from a Gaussian distribution

[18] All experiments for Section 3.2 ran on a machine equipped with 48 Intel(R) Xeon(R) 3.00GHz CPU cores and 1.2TB memory.

$N(\mu = 0.5, \sigma = 0.1)$ truncated from below at zero. We also sample $P(Y = 1 \mid \boldsymbol{x}) \sim$ Uniform$[0, 1]$, $c(\boldsymbol{x}_i, \boldsymbol{x}_j) \sim$ Uniform$[0, 1]$ for 50% of all pairs and $c(\boldsymbol{x}_i, \boldsymbol{x}_j) = 2$ for the rest. Finally, we set $\gamma = 0.3$.

In our experiments, we compare the utility of the following decision policies and counterfactual explanations:

— *Black box:* decisions are taken by the optimal decision policy in the non-strategic setting, given by Eq. 3.18, and individuals do not receive any counterfactual explanations.

— *Minimum cost:* decisions are taken by the optimal decision policy in the non-strategic setting, given by Eq. 3.18, and individuals receive counterfactual explanations of minimum cost with respect to their initial feature values, similarly as in previous work [108, 109, 186]. More specifically, we cast the problem of finding the set of counterfactual explanations as the minimization of the weighted average cost individuals pay to change their feature values to the closest counterfactual explanation, that is,

$$\mathcal{A}_{mc} = \underset{\mathcal{A} \subseteq \mathcal{P}_\pi \,:\, |\mathcal{A}| \leq k}{\operatorname{argmin}} \sum_{\boldsymbol{x}_i \in \mathcal{X} \setminus \mathcal{P}_\pi} P(\boldsymbol{x}_i) \min_{\boldsymbol{x}_j \in \mathcal{A}} c(\boldsymbol{x}_i, \boldsymbol{x}_j),$$

and realize that this problem is a version of the k-median problem, which we can solve using a greedy heuristic [187].

— *Diverse:* decisions are taken by the optimal decision policy in the non-strategic setting, given by Eq. 3.18, and individuals receive a set of diverse counterfactual explanations of minimum cost with respect to their initial feature values, similarly as in previous work [180, 188], that is,

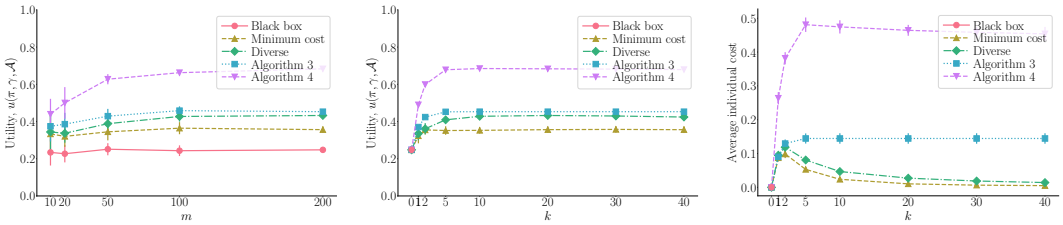
$$\mathcal{A}_{d} = \underset{\mathcal{A} \subseteq \mathcal{P}_\pi \,:\, |\mathcal{A}| \leq k}{\operatorname{argmax}} \sum_{\boldsymbol{x}_i \in \mathcal{X} \setminus \mathcal{P}_\pi} P(\boldsymbol{x}_i) \mathbb{1} \left[\mathcal{R}(\boldsymbol{x}_i) \cap \mathcal{A} \neq \emptyset \right].$$

To solve the above problem, we realize it can be reduced to the weighted version of the maximum coverage problem, which can be solved using a well-known greedy approximation algorithm [189].

— *Algorithm 3:* decisions are taken by the optimal decision policy in the non-strategic setting, given by Eq. 3.18, and individuals receive counterfactual explanations given by Eq. 3.14, where $\mathcal{A}$ is found using Algorithm 3.

— *Algorithm 4:* decisions are taken by the decision policy given by Eq. 3.17 and individuals receive counterfactual explanations given by Eq. 3.14, where $\mathcal{A}$ is found using Algorithm 4.

Results. Figs. 3.10a, 3.10b show the utility achieved by each of the decision policies and counterfactual explanations for several numbers of feature values m and counterfactual explanations k. We find several interesting insights: (i) the counterfactual explanations found by Algorithm 4 and the decision policies given by Eq. 3.17 beat all other alternatives by large margins across the whole spectrum, showing that jointly optimizing the decision policy and the counterfactual explanations offers clear additional gains; (ii) the counterfactual explanations found by Algorithms 3 and 4 provide higher utility gains as the number of feature values increases and thus the search space of counterfactual explanations becomes larger; (iii) a small number of counterfactual explanations is enough to provide significant gains in terms of utility with respect to the optimal decision policy without counterfactual explanations.



(a) Utility vs. # of feature values

(b) Utility vs. # of explanations

(c) Individual cost vs. # of explanations

Figure 3.10: **Results on synthetic data.** Panels (a) and (b) show the utility achieved by six types of decision policies and counterfactual explanations against the total number of feature values m and the number of counterfactual explanations k, respectively. Panel (c) shows the average cost individuals had to pay to change from their initial features to the feature value of the counterfactual explanation they receive under the same five types of decision policies and counterfactual explanations. In Panel (a), we set $k = 0.1m$ and, in Panels (b) and (c), we set $m = 200$. In all panels, we repeat each experiment 20 times, and error bars indicate 95% confidence intervals.

Fig. 3.10c shows the average cost individuals had to pay to change from their initial features to the feature value of the counterfactual explanation they receive. As one may have expected, the results show that, under the counterfactual explanations of minimum cost (Minimum cost and Diverse), the individuals invest less effort to change their initial features and the effort drops as the number of counterfactual explanations increases. In contrast, our methods incentivize the individuals to achieve the highest self-improvement, particularly when we jointly optimize the decision policy and the counterfactual explanations.

3.2.6 Experiments on real data

In this section, we evaluate the utility achieved by Algorithms 3 and 4 using real loan and credit card data, comparing them with the same baselines as before. Moreover, we experiment with a modified version of Algorithm 3, designed to increase the diversity of counterfactual explanations, as described in Section 3.2.4.

Experimental setup. We experiment with two publicly available datasets: (i) the *lending* dataset [190], which contains information about all accepted loan applications in LendingClub during the 2007-2018 period and (ii) the *credit* dataset [174], which contains information about a bank's credit card payoffs (*i.e.*, the same dataset as in Section 3.1.6). For each accepted loan applicant (or credit card holder), we use various demographic information and financial status indicators as features $\boldsymbol{X}$ and the current loan status (or credit payoff status) as label Y. Appendix B.2 contains more details on the specific features we used to experiment with the lending dataset.

The procedures we followed to (i) approximate $P(Y \mid \boldsymbol{X})$, and (ii) determine the values of the cost function $c(\boldsymbol{x}_i, \boldsymbol{x}_j)$ based on the maximum percentile shift are identical to the ones used in Section 3.1.6. Recall that, we distinguish between actionable and non-actionable features. In the credit dataset, non-actionable features contain Marital Status, Age Group and Education Level, Total Overdue Counts, Total Months Overdue, while all other features are actionable. In the lending dataset,

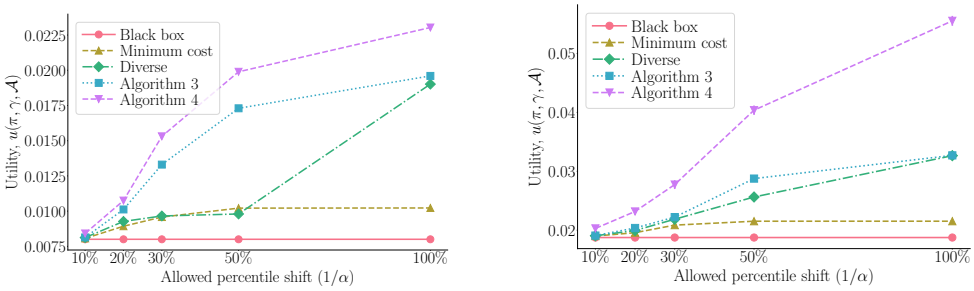
all features are considered actionable. Additionally, a scaling factor α controls the difficulty of changing features, with greater difficulty for higher α values.

Results. We start by focusing on the credit dataset, and we examine an anecdotal example that illustrates the intuitive differences between the counterfactual explanations $\mathsf{e}_m(\boldsymbol{x})$ and $\mathsf{e}(\boldsymbol{x})$ provided by the minimum cost baseline and Algorithm 3, respectively. To this end, for fixed α and k, we first track down the individuals whose best-response under both methods is to change their initial features to the provided counterfactual explanation. Then, for each of these individuals, we compare the counterfactual explanations provided by each of both methods. Table 3.1 shows the initial features $\boldsymbol{x}$ together with the counterfactual explanations $\mathsf{e}_m(\boldsymbol{x})$ and $\mathsf{e}(\boldsymbol{x})$ for one of the above individuals picked at random. In this example, the individual is a university student, unmarried and under the age of 25 who is advised to follow the counterfactual explanations to maintain their credit. Since the marital status, age group and level of education are all non-actionable features, both counterfactual explanations maintain the initial values for those features. Under the minimum cost baseline, the bank would advise the individual to reduce their monthly credit card bill by ∼\$150 and limit high spending to 2 months per semester so that their risk of default would decrease from 16% to 13%. However, under Algorithm 3, the bank would advise to reduce their monthly credit card bill by ∼\$400, limit high spending to 1 month per semester and increase their monthly credit card payoff slightly, so that their risk of default would decrease to 11%. Since by construction, both $\mathsf{e}_m(\boldsymbol{x})$ and $\mathsf{e}(\boldsymbol{x})$ are within the region of adaptation of $\boldsymbol{x}$, the individual is guaranteed to follow the advice in both cases, however, under Algorithm 3, the individual would be less likely to default and more likely to achieve a superior long-term well being.

Next, we compare the utility achieved by the decision policies and counterfactual explanations found by all algorithms considered in the previous section across both datasets, for various values of the parameter α, which controls the individuals'

Table 3.1: **Examples of counterfactual explanations in the credit dataset.** The columns $\mathsf{e}_m(\boldsymbol{x})$ and $\mathsf{e}(\boldsymbol{x})$ correspond to counterfactual explanations provided by the minimum cost baseline and Algorithm 3, respectively, to an individual with initial feature value $\boldsymbol{x}$. Initially, the individual's outcome is $P(Y = 1 \,|\, \boldsymbol{x}) = 0.84$ and, after they best-respond, their outcome is $P(Y = 1 \,|\, \mathsf{e}_m(\boldsymbol{x})) = 0.87$ and $P(Y = 1 \,|\, \mathsf{e}(\boldsymbol{x})) = 0.89$, respectively. In both methods, we set $\alpha = 2$ and $k = 160$.

Feature	$\boldsymbol{x}$	$\mathsf{e}_m(\boldsymbol{x})$	$\mathsf{e}(\boldsymbol{x})$
Married	No	No	No
Age group	Under 25	Under 25	Under 25
Education	Student	Student	Student
Maximum Bill Amount Over Last 6 Months	\$2246	\$2084	\$1929
Maximum Payment Amount Over Last 6 Months	\$191	\$188	\$221
Months With Zero Balance Over Last 6 Months	0	0	0
Months With Low Spending Over Last 6 Months	0	0	0
Months With High Spending Over Last 6 Months	4	2	1
Most Recent Bill Amount	\$2145	\$2003	\$1750
Most Recent Payment Amount	\$123	\$124	\$100
Total Overdue Counts	0	0	0
Total Months Overdue	0	0	0



(a) Lending dataset

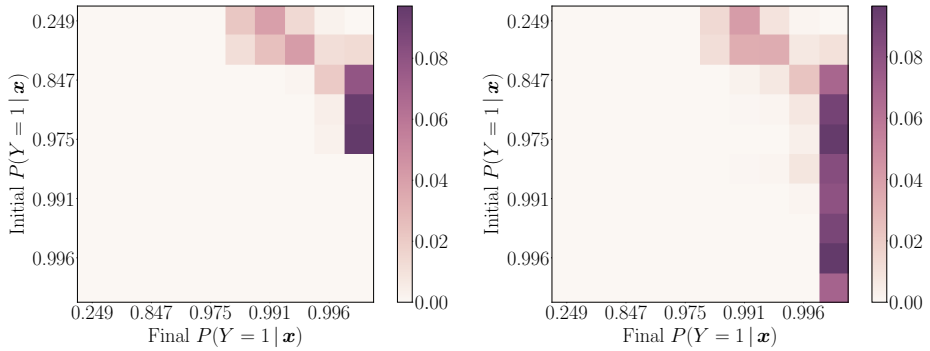
(b) Credit dataset

Figure 3.11: **Utility of different policies against α in the lending and credit datasets.** In panel (a), the number of feature values is $m = 400$ and, in panel (b), it is $m = 3200$. In both panels, we set $k = 0.05m$, we repeat each experiment 20 times, and error bars indicate 95% confidence intervals.

difficulty in changing features. Fig. 3.11 summarizes the results, which show that Algorithm 3 and Algorithm 4 consistently outperform all baselines and, as the cost of adapting to feature values with higher outcome values decreases (smaller α), the competitive advantage by jointly optimizing the decision policy and the counterfactual explanations (Algorithm 4) grows significantly. This competitive advantage is more apparent in the credit dataset because it contains non-actionable features (*e.g.*, credit overdue counts) and, under the optimal decision policy in the non-strategic setting, it is difficult to incentivize individuals who receive a negative decision to improve only by optimizing the set of counterfactual explanations they receive.

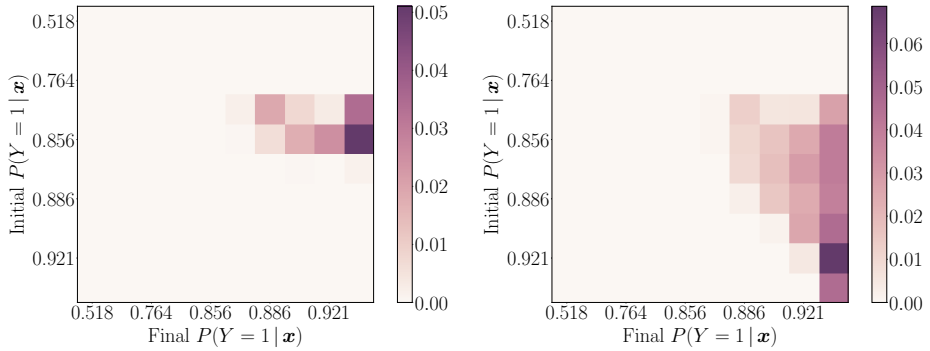
To understand the differences in utility caused by the two proposed algorithms, we measure the transportation of mass induced by the policies and counterfactual explanations used in Algorithm 3 and 4 in both datasets, as follows. For each individual in the population whose best-response is to change their feature value, we record their outcome $P(Y = 1 \,|\, \boldsymbol{x})$ before and after the best-response. Then, we discretize the outcome values using percentiles. Fig. 3.12 summarizes the results, which show several interesting insights. In the lending dataset, we observe that a large portion of individuals do improve their outcome even if we only optimize the counterfactual explanations (Panel (a)). In contrast, in the credit dataset, we observe that, if we only optimize the counterfactual explanations (Panel (c)), most individuals do not improve their outcome. That being said, if we jointly optimize the decision policy and counterfactual explanations (Panels (b) and (d)), we are able to incentivize a large portion of individuals to self improve in both datasets.

Further, we focus on the lending dataset and evaluate the sensitivity of our algorithms. First, we measure the influence that the number of counterfactual explanations has on the utility achieved by each of the decision policies and counterfactual explanations. As shown in Fig. 3.13a, our algorithms require only a small number of counterfactual explanations to provide significant gains in terms of utility with respect to all baselines. Second, we challenge the assumption that individuals do not share the counterfactual explanations they receive with other individuals with different feature values. To this end, we assume that, given the set of counterfactual explanations $\mathcal{A}$ found by Algorithm 4, individuals with initial feature value $\boldsymbol{x}$ receive the counterfactual explanation $\mathsf{e}(\boldsymbol{x}) \in \mathcal{A}$ given by Eq. 3.14 and, with probability p_l, they also receive an additional explanation $\mathsf{e}'(\boldsymbol{x})$ chosen at random from $\mathcal{A}$



(a) Alg. 3, lending

(b) Alg. 4, lending



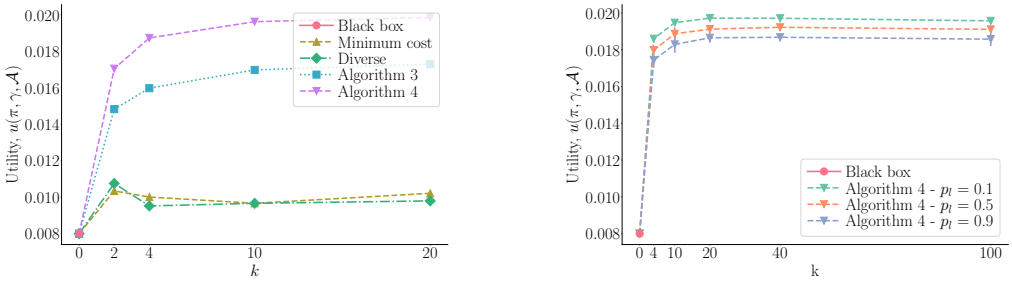
(c) Alg. 3, credit

(d) Alg. 4, credit

Figure 3.12: **Transportation of mass in the lending and credit datasets.** We compare the transportation of mass induced by the policies and counterfactual explanations found by Algorithm 3 and 4. For each individual in the population, whose best-response is to change their feature value, we record their outcome $P(Y = 1\,|\,\boldsymbol{x})$ before the best-response (Initial $P(Y = 1\,|\,\boldsymbol{x})$) and after the best-response (Final $P(Y = 1\,|\,\boldsymbol{x})$). In each panel, $\alpha = 2$, and the color is proportional to the percentage of individuals who move from initial $P(Y = 1\,|\,\boldsymbol{x})$ to final $P(Y = 1\,|\,\boldsymbol{x})$.

and follow the counterfactual explanation that benefits them the most. Fig. 3.13b summarizes the results for several values of p_l and number of counterfactual explanations, which show that the policies and explanations provided by Algorithm 4 present a significant utility advantage even when the leakage probability p_l is large.

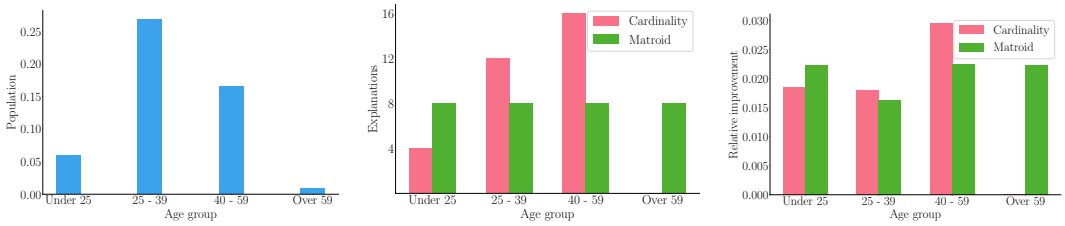
Finally, we focus on the credit dataset and consider a scenario in which a bank aims not only to continue providing credit to the customers that are more likely to repay but also provide explanations that incentivize individuals across all age groups to maintain their credit. To this end, we incorporate a partition matroid constraint that ensures the counterfactual explanations are diverse across age groups, as described in Section 3.2.4, and use a slightly modified version of Algorithm 3 to solve the constrained problem [182], which enjoys a 1/2 approximation guarantee. Fig. 3.14 summarizes the results, which show that: (i) optimizing under a cardinality constraint leads to an unbalanced set of explanations, favoring the more populated age groups (25 to 59) while completely ignoring the recourse potential of individuals older than 60; (ii) the relative group improvement, defined as $\sum_{\boldsymbol{x}_i \in \mathcal{X}_z \setminus \mathcal{P}_\pi} P(\boldsymbol{x}_i)[P(Y = 1\,|\,\boldsymbol{x}_j^i) - P(Y = 1\,|\,\boldsymbol{x}_i)] / \sum_{\boldsymbol{x}_i \in \mathcal{X}_z \setminus \mathcal{P}_\pi} P(\boldsymbol{x}_i)$, where $\mathcal{X}_z$ is the set of feature values corresponding to age group z and $\boldsymbol{x}_j^i$ is the best-response of individuals with initial feature value $\boldsymbol{x}_i \in \mathcal{X}_z$, is more balanced across age groups,



(a) Utility vs. k

(b) Utility vs. k under leakage

Figure 3.13: **Sensitivity to the number of counterfactual explanations and information leakage.** Panel (a) shows the utility achieved by five types of decision policies and counterfactual explanations against the number of counterfactual explanations k. Panel (b) shows the utility achieved by Algorithm 4 against the number of counterfactual explanations k for several values of the leakage probability p_l. In both panels, we use the lending dataset, the number of feature values is $m = 400$, we set $\alpha = 2$, and we repeat each experiment involving randomization 20 times, where error bars indicate 95% confidence intervals.



(a) Rejected population by age

(b) Explanations by age

(c) Relative improvement by age

Figure 3.14: **Increasing the diversity of the provided counterfactual explanations.** Panel (a) shows the population per age group, rejected by the optimal threshold policy in the non strategic setting. Panel (b) shows a comparison of the age distribution of counterfactual explanations in $\mathcal{A}$ produced by the greedy algorithm under a cardinality and a matroid constraint. Panel (c) shows the relative improvement of each age group. In all panels, we use the credit dataset and we set $k = 32$ and $\alpha = 2$.

showing that the matroid constraint can be used to generate counterfactual explanations that help the entire spectrum of the population to self-improve.

3.3 Chapter conclusions

In this chapter, we have studied problems of utility maximization in decision making under strategic behavior. We have introduced game-theoretic models in which a decision maker, informed by a predictive model, designs a decision policy and shares information about it with the individuals who are subject to it. Through theoretical analysis and experiments on real data, we have demonstrated that transparency has the potential to increase a decision maker's utility if the policy is designed to incentivize forms of effort that lead individuals to self-improvement. We hope that

our positive results serve as motivation for decision makers to be more transparent about their decision policies in practice.

We have focused on two specific forms of achieving transparency: (i) complete transparency, where individuals have full access to the decision maker's policy, and (ii) partial transparency, where individuals only have access to counterfactual explanations. As future work, it would be interesting to explore alternative forms of transparency. For example, we have assumed that individuals do not share information about the policy with each other. Relaxing this assumption and developing game-theoretic models in which individuals form a social network and share information with their peers could lead to additional insights into decision making under strategic behavior, reflecting more realistic settings.

Additionally, the models we have introduced assume that there is no unobserved confounding that might enable individuals to "game" the decision maker's policy. Selecting the right set of features to make this assumption a reality is a challenging problem in itself [156]. It would be interesting to investigate the use of causally aware feature selection methods [191] in strategic settings and develop algorithms that find optimal decision policies under different types of unobserved confounding.

Finally, although we have evaluated our methods through a series of experiments using real data, an interesting direction for future work involves implementing these methods within a real-world decision making pipeline. This would include conducting an empirical analysis of all the steps involved, such as feature selection and the iterative process of automated policy design and refinement of the problem parameters (*e.g.*, the cost function) with the aid of a human decision maker. Additionally, the collection of a dataset on how individuals best-respond to a transparent policy would be of great value to the strategic machine learning community, which has remained largely theoretical.

Chapter 4

Enhancing counterfactual reasoning in sequential decision making

Counterfactual reasoning refers to the ability that humans have to mentally simulate alternative worlds where events from the past play out differently than they did in reality [192]. Counterfactuals—thoughts about "what might have been"—arise in a variety of decision making scenarios ranging from simple everyday decisions (*e.g.*, what would have been the outcome of the game had the chess player made a different move?) to more critical ones (*e.g.*, would the patient's condition have improved had the physician administered a different drug?). Research in psychology suggests that these thoughts are closely related to our understanding of causality [74, 193], while also playing an important role in the process in which we generate explanations about events, learn from past experience, and plan future actions [67–69].

In many real-world decision making settings, decisions are sequential—the decision maker takes a sequence of actions over multiple time steps. However, for each sequence of actions taken, there is an exponential number of sequences of actions *not* taken that would have led the decision making process to a different counterfactual outcome. Contemplating all possible alternatives can be overwhelming for a decision maker and, in practice, people tend to focus on a limited number of time steps and actions depending on their recency [194–196]. As a result, a decision maker who uses counterfactual reasoning as a learning signal may overlook useful past situations, potentially hindering their learning process.

For example, consider a clinician reflecting on the efficacy of a series of treatment decisions made for a patient. To think about potential improvements to their treatment policy, they may attempt to identify critical time steps in the patient's care where different decisions could have improved the patient's health. This task is intractable for a human, as there are multiple treatments they could have administered at various stages of a patient's hospital stay. Counterfactual reasoning becomes even more challenging if the clinician aims to analyze multiple patients they have treated over a longer time window.

In this chapter, we address this issue by introducing methods to aid in the retrospective analysis of past decisions in sequential decision making tasks. For each observed episode—a sequence of observed states (*e.g.*, patient vitals) and taken actions (*e.g.*, treatments)—our methods find action sequences that, under the cir-

cumstances of that particular episode, could have led to a better counterfactual outcome. Importantly, we focus on action sequences that are "close" to the one taken in reality, thereby making them more informative for human decision makers by highlighting only the most critical time steps and actions in each episode. Throughout the chapter, we refer to these (constrained) action sequences as *counterfactually optimal*.

Recall the clinical example above and consider a clinician analyzing data of a patient whose vitals have not improved after a certain period of time. A counterfactually optimal action sequence could highlight to the clinician a small set of time steps in the treatment process where, had they administered different drug dosages, the patient's severity would have been lower. Moreover, identifying such action sequences for a large cohort of patients could pinpoint "interesting cases" for the clinician to revisit, where manual inspection of these cases and the corresponding time steps could provide insights to the clinician on possible ways to improve their treatment policy.

In Section 4.1, we formalize sequential decision making using Markov decision processes [119] combined with structural causal models [71]. Based on this formulation, we introduce and present a solution for the optimization problem involved in computing counterfactually optimal action sequences within environments having discrete states and actions. In Section 4.2, we extend this formulation to environments with continuous states. We introduce a variation of the original problem, analyze its computational complexity, and propose an algorithmic solution. Our theoretical findings are supported by experiments using synthetic and real (medical) data.

4.1 Sequential decisions in discrete state spaces

In this section, our goal is to find counterfactually optimal action sequences for decision making processes in which multiple, dependent actions are taken sequentially over time. To operationalize the notion of "closeness" mentioned earlier, we focus on sequences of actions that differ from the observed sequence in at most k actions and could have led the process realization to a better outcome. Because the final outcome depends on the entire sequence of actions and there is (typically) uncertainty in the counterfactual dynamics of the environment, different action sequences may be (counterfactually) optimal under different possible realizations of these dynamics. Consequently, our goal here is not to find a single counterfactually optimal action sequence, but rather a counterfactual policy that leads to action sequences that differ in at most k actions from the observed one for every realization of the counterfactual transition dynamics.

The standard framework for modeling sequential decision making are Markov decision processes (MDPs) [119]. As mentioned in Chapter 2, an MDP is defined by a set of environment states (*e.g.*, vital signs of a patient), a set of actions (*e.g.*, dosages of a drug), and a set of probability distributions that characterize the transitions between states at each time step, conditioned on the action taken. Moreover, each pair of state and action is associated with a numerical reward (*e.g.*, satisfaction inversely proportional to the patient's severity). Typically, the decision maker's goal is to find an action sequence that maximizes their total reward over time.

Although the transition probabilities of an MDP carry sufficient information to answer questions about the future, such as identifying the reward that an action sequence would give in expectation, they do not allow us to answer counterfactual questions. To infer how a particular episode would have evolved under a different action sequence than the one taken in reality, one needs to represent the stochastic state transitions of the environment using a structural causal model (SCM) [71, 197]. This has also been a key aspect of a nascent line of work at the intersection of counterfactual reasoning and reinforcement learning (RL), which has focused on evaluating and improving RL policies using offline data [130–132].

We start by formally characterizing a sequence of discrete actions and discrete states using finite horizon MDPs, and we model the transition probabilities between a pair of states, given an action, using the Gumbel-Max structural causal model [131]. This model has been shown to have a desirable counterfactual stability property and, given a sequence of actions and states, it allows us to reliably estimate the counterfactual outcome under an alternative sequence of actions. Building upon this causal representation of MDPs, we make the following contributions:

1. We formally state the problem of finding counterfactually optimal action sequences for an observed episode in the presence of uncertainty in the counterfactual transition dynamics. Specifically, we cast the problem as a constrained optimization problem over the set of policies that would have resulted in action sequences that differ in at most k actions from the observed sequence.

2. We present a polynomial-time algorithm based on dynamic programming that finds the optimal solution to the aforementioned problem.

Finally, we validate our algorithm using both synthetic and real data from cognitive behavioral therapy and show that counterfactually optimal action sequences can provide valuable insights to enhance sequential decision making under uncertainty. The code used for all experiments in Section 4.1 is available at https://github.com/Networks-Learning/counterfactual-explanations-mdp.

4.1.1 A causal model of sequential decision making

Our starting point is the following stylized setting that resembles a variety of real-world sequential decision making processes. At each time step $t \in [T-1]_0$, where T is a time horizon, the decision making process is characterized by a state $s_t \in \mathcal{S}$, where $\mathcal{S}$ is a space of n states, an action $a_t \in \mathcal{A}$, where $\mathcal{A}$ is a space of m actions, and a reward $r(s_t, a_t) \in \mathbb{R}$. Moreover, given a realization of a decision making process $\tau = \{(s_t, a_t)\}_{t=0}^{T-1}$, we define the *outcome* of the decision making process $o(\tau) = \sum_t r(s_t, a_t)$ as the sum of the rewards.

Given the above setting, we characterize the relationship between actions, states and outcomes using finite horizon Markov decision processes (MDPs). More specifically, we consider an MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, T)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the set of actions, P denotes the transition probability $P(S_{t+1} = s_{t+1} \,|\, S_t = s_t, A_t = a_t)$, r denotes the immediate reward $r(s_t, a_t)$, and T is the time horizon. While this characterization is helpful to make predictions about future states and design action policies [119], it is not sufficient to make counterfactual predictions. For example, given a realization of a decision making process $\tau = \{(s_t, a_t)\}_{t=0}^{T-1}$, we cannot know

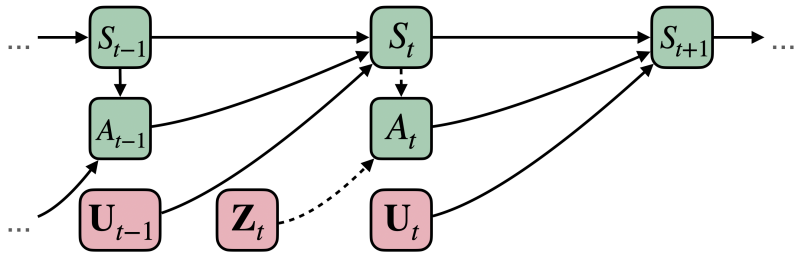


Figure 4.1: **Causal graph of an SCM $\mathcal{C}$ representing a Markov decision process.** Green boxes represent endogenous random variables and pink boxes represent exogenous noise variables. The value of each endogenous variable is given by a function of the values of its ancestors in the causal graph. The value of each exogenous noise variable is sampled independently from a given distribution. An intervention $do[A_t = a']$ breaks the dependence of the variable A_t from its ancestors (highlighted by dotted lines) and sets its value to a'. After observing an event $S_{t+1} = s_{t+1}, S_t = s_t, A_t = a_t$, a counterfactual prediction can be thought of as the result of an intervention $do[A_t = a']$ in a modified SCM where $\boldsymbol{U}_t$ takes values $\boldsymbol{u}_t$ from a posterior distribution with support such that $s_{t+1} = g_S(s_t, a_t, \boldsymbol{u}_t)$.

what would have happened if, instead of taking action a_t at time t, we had taken action $a' \neq a_t$. To be able to overcome this limitation, we will now augment the above characterization using a particular class of structural causal model (SCM) [71, 117] with desirable properties.

Let $\mathcal{C}$ be a structural causal model defined by the assignments

$$S_{t+1} := g_S(S_t, A_t, \boldsymbol{U}_t) \quad \text{and} \quad A_t := g_A(S_t, \boldsymbol{Z}_t), \tag{4.1}$$

where S_t, A_t, S_{t+1} are endogenous categorical variables, and $\boldsymbol{U}_t$ and $\boldsymbol{Z}_t$ are n- and m-dimensional independent exogenous (noise) variables, respectively. Here, g_S and g_A are two given functions, and we refer to the function g_S as the *transition mechanism*. Let $P^{\mathcal{C}}$ denote the distributions and probabilities entailed by $\mathcal{C}$. Note that, we assume there is no unobserved confounding, meaning all noise variables $\boldsymbol{U}_t$ and $\boldsymbol{Z}_t$ are mutually independent. Then, as argued by Buesing et al. [130], we can always find a distribution for the noise variables and a transition mechanism g_S, such that the transition probability of the MDP of interest is given by the following interventional distribution over the SCM $\mathcal{C}$:

$$P(S_{t+1} = s_{t+1} \,|\, S_t = s_t, A_t = a_t) = P^{\mathcal{C}\ ;\ do[A_t = a_t]}(S_{t+1} = s_{t+1} \,|\, S_t = s_t), \tag{4.2}$$

where, recall that, $do[A_t = a_t]$ denotes a (hard) intervention in which the second assignment in Eq. 4.1 is replaced by the value a_t.

Under this view, given an observed realization of a decision making process $\tau = \{(s_t, a_t)\}_{t=0}^{T-1}$, we can compute the posterior distribution $P^{\mathcal{C} \,|\, S_t = s_t, S_{t+1} = s_{t+1}, A_t = a_t}(\mathbf{U}_t)$ of each noise variable $\boldsymbol{U}_t$ and, building on the conditional density function of this posterior distribution, which we denote as $f_{\mathbf{U}_t}^{\mathcal{C} \,|\, S_t = s_t, S_{t+1} = s_{t+1}, A_t = a_t}(\boldsymbol{u})$, we can define a (non-stationary) counterfactual transition probability

$$\begin{aligned} &P_{\tau,t}(S_{t+1} = s' \,|\, S_t = s, A_t = a) \\ &\quad = P^{\mathcal{C} \,|\, S_t = s_t, S_{t+1} = s_{t+1}, A_t = a_t\ ;\ do[A_t = a]}(S_{t+1} = s' \,|\, S_t = s) \end{aligned}$$

$$
\begin{aligned}
&= \int_{\mathbb{R}^n} P^{\mathcal{C} \,|\, S_t=s_t, S_{t+1}=s_{t+1}, A_t=a_t \;;\; do[A_t=a]}(S_{t+1} = s' \,|\, S_t = s, \mathbf{U}_t = \boldsymbol{u}) \\
&\qquad \cdot f^{\mathcal{C} \,|\, S_t=s_t, S_{t+1}=s_{t+1}, A_t=a_t \;;\; do[A_t=a]}_{\mathbf{U}_t}(\boldsymbol{u}) d\boldsymbol{u} \\
&\overset{(a)}{=} \int_{\mathbb{R}^n} \mathbb{1}[s' = g_S(s, a, \boldsymbol{u})] \cdot f^{\mathcal{C} \,|\, S_t=s_t, S_{t+1}=s_{t+1}, A_t=a_t}_{\mathbf{U}_t}(\boldsymbol{u}) d\boldsymbol{u} \\
&= \mathbb{E}_{\mathbf{U}_t \,|\, S_t=s_t, S_{t+1}=s_{t+1}, A_t=a_t}[\mathbb{1}[s' = g_S(s, a, \mathbf{U_t})]],
\end{aligned}
\tag{4.3}
$$

where, in (a), we drop the $do[\cdot]$ because $\mathbf{U}_t$ and A_t are independent in the modified SCM. Importantly, the above counterfactual transition probability allows us to make counterfactual predictions. For example, given that, at time t the state was s_t and, at time $t+1$, the process transitioned to state s_{t+1} after taking action a_t, it allows us to specify the probability of transitioning to state s' after taking action $a \neq a_t$ if the state had been s at time t. For a visual representation of the causal graph associated with the SCM $\mathcal{C}$ and to better understand the concept of counterfactual predictions, refer to Fig. 4.1.

However, for state variables taking discrete values, the posterior distribution of the noise may be non-identifiable without further assumptions, as argued by Oberst and Sontag [131]. This is because there may be several noise distributions and transition mechanisms g_S which give interventional distributions consistent with the MDP's transition probabilities but result in different counterfactual transition distributions. To avoid these non-identifiability issues, we follow Oberst and Sontag and restrict our attention to the class of Gumbel-Max SCMs, that is,

$$
S_{t+1} := g_S(S_t, A_t, \boldsymbol{U}_t) = \operatorname*{argmax}_{s \in \mathcal{S}} \{\log P(S_{t+1} = s \,|\, S_t, A_t) + U_{t,s}\}, \tag{4.4}
$$

where $U_{t,s} \sim \text{Gumbel}(0, 1)$ and $P(S_{t+1} \,|\, S_t, A_t)$ is the transition distribution of the MDP. More specifically, this class of SCMs has been shown to satisfy a desirable counterfactual stability property, which goes intuitively as follows. Assume that, at time t, the process transitioned from state s_t to state s_{t+1} after taking action a_t. Then, in a counterfactual scenario, it is unlikely that, at time t, the process would transition from a state s to a state $s' \neq s_{t+1}$ after taking action a if

$$
\begin{aligned}
P(S_{t+1} = s' \,|\, S_t = s, A_t = a) &\leq P(S_{t+1} = s' \,|\, S_t = s_t, A_t = a_t), \text{ and} \\
P(S_{t+1} = s_{t+1} \,|\, S_t = s, A_t = a) &> P(S_{t+1} = s_{t+1} \,|\, S_t = s_t, A_t = a_t).
\end{aligned}
$$

In words, transitioning to a state other than s_{t+1}—the factual one—is unlikely unless choosing an action that decreases the relative chances of s_{t+1} compared to the other states. More formally, given $\tau = \{(s_t, a_t)\}_{t=0}^{T-1}$, then for all s, s' with $s' \neq s_{t+1}$, the condition

$$
\frac{P(S_{t+1} = s_{t+1} \,|\, S_t = s, A_t = a)}{P(S_{t+1} = s_{t+1} \,|\, S_t = s_t, A_t = a_t)} \geq \frac{P(S_{t+1} = s' \,|\, S_t = s, A_t = a)}{P(S_{t+1} = s' \,|\, S_t = s_t, A_t = a_t)}
$$

implies that $P_{\tau,t}(S_{t+1} = s' \,|\, S_t = s, A_t = a) = 0$. In practice, in addition to solving the non-identifiability issues, the use of Gumbel-Max SCMs allows for an efficient procedure to sample from the corresponding noise posterior distribution $P^{\mathcal{C} \,|\, S_t=s_t, S_{t+1}=s_{t+1}, A_t=a_t}(\mathbf{U}_t)$, described elsewhere [131, 198], and given a set of d samples from the noise posterior distribution, we can compute an unbiased finite sample

Monte-Carlo estimator for the counterfactual transition probability, as defined in Eq. 4.3, as follows:

$$P_{\tau,t}(S_{t+1} = s' \mid S_t = s, A_t = a) \approx \frac{1}{d} \sum_{j \in [d]} \mathbb{1}[s' = g_S(s, a, \boldsymbol{u}_j)] \tag{4.5}$$

On the assumption of no unobserved confounding. The assumption that there are no hidden confounders is a frequent assumption made by work at the intersection of counterfactual reasoning and reinforcement learning [130–132] and, more broadly, in the causal inference literature [199–203]. That said, there is growing interest in developing off-policy methods for partially observable MPDs (POMDPs) that are robust to certain types of confounding [125–127], and in learning dynamic treatment regimes in sequential settings with non-Markovian structure [128, 129]. Moreover, there is a line of work focusing on the identification of counterfactual quantities in non-sequential confounded environments [204–206]. In that context, we consider the computation of (approximately) optimal counterfactual action sequences under confounding as a very interesting direction for future work.

4.1.2 Problem statement

Inspired by previous work on counterfactual explanations in supervised learning [107, 108], we focus on counterfactual outcomes that could have occurred if the alternative action sequence was "close" to the observed one. However, since in our setting, there is uncertainty on the counterfactual dynamics of the environment, we will look for a non-stationary counterfactual policy π that, under every possible realization of the counterfactual transition probability defined in Eq. 4.3, is guaranteed to provide the optimal alternative sequence of actions differing in at most k actions from the observed one.

More specifically, let $\tau = \{(s_t, a_t)\}_{t=0}^{T-1}$ be an observed realization of a decision making process characterized by a Markov decision process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, T)$ with a transition probability defined via a Gumbel-Max structural causal model (SCM), as described in Section 4.1.1. Then, to characterize the effect that any alternative action sequence would have had on the outcome of the above process realization, we start by building a non-stationary counterfactual MDP $\mathcal{M}_\tau = (\mathcal{S}^+, \mathcal{A}, P_\tau^+, r^+, T)$. Here, $\mathcal{S}^+ = \mathcal{S} \times [T-1]_0$ is an enhanced state space such that each $s^+ \in \mathcal{S}^+$ corresponds to a pair (s, l) indicating that the original decision making process would have been at state $s \in \mathcal{S}$ had already taken l actions differently from the observed sequence. Following this definition, r^+ denotes the reward function which we define as $r^+((s, l), a) = r(s, a)$ for any $(s, l) \in \mathcal{S}^+$ and $a \in \mathcal{A}$, that is, the counterfactual rewards remain independent of the number of modifications in the action sequence. Lastly, let P_τ be the counterfactual transition probability, as defined by Eq. 4.3. Then, the transition probability P_τ^+ for the enhanced state space is defined as:

$$P_{\tau,t}^+\left(S_{t+1}^+ = (s', l') \mid S_t^+ = (s, l), A_t = a\right) = \begin{cases} P_{\tau,t}\left(s' \mid s, a\right) \\ \qquad \text{if } (a = a_t \wedge l' = l) \\ \qquad\quad \vee (a \neq a_t \wedge l' = l + 1) \\ 0 \qquad \text{otherwise,} \end{cases} \tag{4.6}$$

Algorithm 5: It samples a counterfactual action sequence from the counterfactual policy π

input : counterfactual policy π, horizon T, counterfactual transition probability P_τ, reward function r, initial state s_0
output : counterfactual trajectory τ', counterfactual outcome $o(\tau')$

$s'_0 \leftarrow s_0$
$l_0 \leftarrow 0$
reward $\leftarrow 0$
for $t \leftarrow 0$ **to** $T-1$ **do**
  $a'_t \leftarrow \pi((s'_t, l_t), t)$ `// Get an action from the counterfactual policy`
  reward $\leftarrow$ reward $+ r(s'_t, a'_t)$
  if $t \neq T-1$ **then**
    $s'_{t+1} \sim P_{\tau,t}(S_{t+1} \,|\, S_t = s'_t, A_t = a'_t)$ `// Sample the next state`
    if $a'_t \neq a_t$ **then**
      $l_{t+1} \leftarrow l_t + 1$ `// Update the counter of action changes`
    else
      $l_{t+1} \leftarrow l_t$
$\tau' \leftarrow \{((s'_t, l_t), a'_t)\}_{t=0}^{T-1}$
$o(\tau') \leftarrow$ reward
return $\tau', o(\tau')$

where note that the dynamics of the original states s are equivalent under $P^+_{\tau,t}$ and $P_{\tau,t}$, however, under $P^+_{\tau,t}$, we also keep track of the number of actions differing from the observed actions. Now, let $\pi : \mathcal{S}^+ \times [T-1]_0 \to \mathcal{A}$ be a policy that deterministically decides about the counterfactual action a'_t that should have been taken if the process's enhanced state had been $s^+_t = (s'_t, l_t)$, that is, the counterfactual state at time t was s'_t after performing l_t action changes. Then, under a counterfactual policy π, the corresponding average counterfactual outcome is given by

$$\bar{o}_\pi(\tau) = \mathbb{E}_{\tau' \sim P^+_\tau \,|\, s^+_0 = (s_0, 0)} \left[\sum_{t=0}^{T-1} r^+((s'_t, l_t), a'_t) \right] \tag{4.7}$$

where $\tau' = \{((s'_t, l_t), a'_t)\}_{t=0}^{T-1}$ is a realization of the non-stationary counterfactual MDP $\mathcal{M}_\tau$ with $a'_t = \pi((s'_t, l_t), t)$ and the expectation is taken over all the realizations induced by the transition probability P^+_τ and the policy π. Here, note that, if $\pi((s_t, 0), t) = a_t$ for all $t \in [T-1]_0$, then $\bar{o}_\pi(\tau) = o(\tau)$ matches the outcome of the observed realization.

Then, our goal is to find the optimal counterfactual policy π^*_τ that maximizes the counterfactual outcome subject to a constraint on the number of counterfactual actions that can differ from the observed ones, that is,

$$\underset{\pi}{\text{maximize}} \quad \bar{o}_\pi(\tau) \qquad \text{subject to} \quad \sum_{t=0}^{T-1} \mathbb{1}[a_t \neq a'_t] \leq k \quad \forall \tau' \sim P^+_\tau \tag{4.8}$$

where $a'_0, \ldots, a'_{T-1}$ is one realization of counterfactual actions and $a_0, \ldots, a_{T-1}$ are the observed actions. The constraint guarantees that any counterfactual action sequence induced by the counterfactual transition probability P^+_τ and the counterfactual policy π can differ in at most k actions from the observed sequence. Finally,

once we have found the optimal policy π_τ^*, we can sample a counterfactual realization of the process and the counterfactually optimal action sequence using Algorithm 5.

4.1.3 A polynomial-time dynamic programming algorithm

To solve the problem defined by Eq. 4.8, we break the problem into several smaller sub-problems. Here, the key idea is to compute the counterfactual policy values that lead to the optimal counterfactual outcome recursively by expanding the expectation and switching the order of the sums in Eq. 4.7.

We start by computing the highest average cumulative reward $h(s, q, c)$ that one could have achieved in the last q steps of the decision making process, starting from state $S_{T-q} = s$, if at most c actions had been different to the observed ones in those last steps. For $c > 0$, we have the recursion

$$h(s,q,c) = \max \Bigg(r(s, a_{T-q}) + \sum_{s' \in \mathcal{S}} P_{\tau, T-q}(s' \mid s, a_{T-q}) h(s', q-1, c), \\ \max_{a \in \mathcal{A} \,:\, a \neq a_{T-q}} \left[r(s,a) + \sum_{s' \in \mathcal{S}} P_{\tau, T-q}(s' \mid s, a) h(s', q-1, c-1) \right] \Bigg), \quad (4.9)$$

and, for $c = 0$, we trivially have that

$$h(s,q,0) = r(s, a_{T-q}) + \sum_{s' \in \mathcal{S}} P_{\tau, T-q}(s' \mid s, a_{T-q}) h(s', q-1, 0), \quad (4.10)$$

with $s \in \mathcal{S}$, $q \in [T]$, $c \in [k]$, and $h(s, 0, c) = 0$ for all s and c. In Eq. 4.9, the first parameter of the outer maximization corresponds to the case where, at time $T - q$, the observed action a_{T-q} is taken and the second parameter corresponds to the case where, instead of the observed action, the best alternative action is taken.

By definition, we can easily conclude that $h(s_0, T, k)$ is the average counterfactual outcome of the optimal counterfactual policy π_τ^*, that is, the objective value of the solution to the optimization problem defined by Eq. 4.8, and we can recover the optimal counterfactual policy π_τ^* by keeping track of the action chosen at each recursive step in Eq. 4.9 and 4.10. The overall procedure, summarized by Algorithm 6, uses dynamic programming. It initially computes the values $h(s, 1, c)$ for all s and c and proceeds with the remaining computations in a bottom-up fashion. The algorithm has complexity $\mathcal{O}(n^2 m T k)$ and is characterized by the following proposition:[1]

Proposition 4.1.1. *The counterfactual policy π_τ^* returned by Algorithm 6 is the solution to the optimization problem defined by Eq. 4.8.*

4.1.4 Experiments on synthetic data

In this section, we evaluate Algorithm 6 on realizations of a synthetic decision making process. To this end, we first look into the average outcome improvement that could have been achieved if at most k actions had been different to the observed ones in every realization, as dictated by the optimal counterfactual policy. Then, we

[1] The proof can be found in Appendix A.3.

Algorithm 6: It returns the optimal counterfactual policy and its average counterfactual outcome

input : states $\mathcal{S}$, actions $\mathcal{A}$, realization τ, horizon T, counterfactual transition probability P_τ, reward function r, constraint k

output : counterfactual policy π^*_τ and average counterfactual outcome $h(s_0, T, k)$

$h(s, q, c) \leftarrow 0$ **for** $s \in \mathcal{S}$, $q \in \{0, \dots, T\}$, $c \in \{0, \dots, k\}$

for $q \leftarrow 1$ **to** T **do**

  for $s \in \mathcal{S}$ **do**

    /* Boundary condition: no action changes left */

    $h(s, q, 0) \leftarrow r(s, a_{T-q})$

    for $s' \in \mathcal{S}$ **do**

      $h(s, q, 0) \leftarrow h(s, q, 0) + P_{\tau, T-q}(s' \mid s, a_{T-q}) h(s', q-1, 0)$

    $\pi^*_\tau((s, k), T-q) \leftarrow a_{T-q}$ // Selected the observed action

for $q \leftarrow 1$ **to** T **do**

  for $c \leftarrow 1$ **to** k **do**

    for $s \in \mathcal{S}$ **do**

      /* Recursion: compute $h(s, q, c)$ using Eq. 4.9 */

      reward $\leftarrow r(s, a_{T-q})$

      for $s' \in \mathcal{S}$ **do**

        reward $\leftarrow$ reward $+ P_{\tau, T-q}(s' \mid s, a_{T-q}) h(s', q-1, c)$

      best_reward $\leftarrow$ reward

      best_action $\leftarrow a_{T-q}$

      for $a \in \mathcal{A} \setminus \{a_{T-q}\}$ **do**

        reward_alt $\leftarrow r(s, a)$

        for $s' \in \mathcal{S}$ **do**

          reward_alt $\leftarrow$ reward_alt $+ P_{\tau, T-q}(s' \mid s, a) h(s', q-1, c-1)$

        if *reward_alt* $>$ *best_reward* **then**

          best_reward $\leftarrow$ reward_alt

          best_action $\leftarrow a$

      $h(s, q, c) \leftarrow$ best_reward

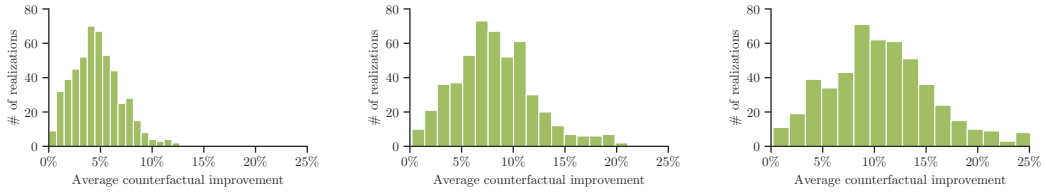
      $\pi^*_\tau((s, k-c), T-q) \leftarrow$ best_action

return $\pi^*_\tau, h(s_0, T, k)$

investigate to what extent the level of uncertainty of the decision making process influences the average counterfactual outcome achieved by the optimal counterfactual policy as well as the number of distinct counterfactual action sequences it provides.[2]

Experimental setup. We characterize the synthetic decision making process using an MDP with states $\mathcal{S} = [n-1]_0$ and actions $\mathcal{A} = [m-1]_0$, where $n = 20$ and $m = 10$, and time horizon $T = 20$. For each state s and action a, we set the immediate reward equal to $r(s, a) = s$, that is, the higher the state, the higher the reward. To set the values of the transition distribution $P(S_{t+1} \mid S_t, A_t)$, we proceed as follows. First we pick one $s^\star \in \mathcal{S}$ uniformly at random and we set a weight $w_{s^\star} = 1$. Then, for the remaining states $s \in \mathcal{S} \setminus s^\star$, we sample weights $w_s \sim \text{Uniform}[0, \alpha]$, where $\alpha \leq 1$. Next, for all $s \in \mathcal{S}$, we set $P(s \mid s_t, a_t) = w_s / \sum_{s' \in \mathcal{S}} w_{s'}$. It is easy to see that, for

[2] All experiments for Section 4.1 ran on a machine equipped with 48 Intel(R) Xeon(R) 3.00GHz CPU cores and 1.5TB memory.



(a) $k = 2$ (b) $k = 5$ (c) $k = 10$

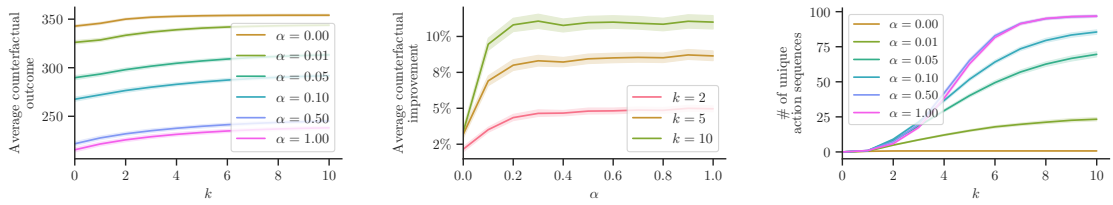
Figure 4.2: **Comparison of factual and counterfactual outcomes.** Each panel shows the empirical distribution of the relative difference between the average counterfactual outcome $\bar{o}_{\pi^*_\tau}(\tau)$ achieved by the optimal counterfactual policy π^*_τ and the observed outcome $o(\tau)$ in a synthetic decision making process for various values of k. In all panels, we set $n = 20$, $m = 10$, $\alpha = 0.4$, $d = 1{,}000$ and estimate the distributions using 500 realizations from 10 different instances of the decision making process (50 realizations per instance), each with different w_s.

each state-action pair (s_t, a_t) at time t, $s_{t+1} = s^\star$ is most likely to be observed in the next timestep $t + 1$. Here, the parameter α controls the level of uncertainty.

Then, we compute the optimal policy that maximizes the average outcome of the decision making process by solving Bellman's equation using dynamic programming (see Section 2.2.1) and use this policy to sample the (observed) realizations as follows. For each realization, we start from a random initial state $s_0 \in \mathcal{S}$ and, at each time step t, we pick the action indicated by the optimal policy with probability 0.95 and a different action uniformly at random with probability 0.05. This leads to action sequences that are slightly suboptimal in terms of average outcome. Finally, to compute the counterfactual transition probabilities $P_{\tau,t}$ for each observed realization τ, we follow the procedure described in Section 4.1.1 with $d = 1{,}000$ samples for each noise posterior distribution.

Results. We first measure to what extent the counterfactual action sequences provided by the optimal counterfactual policy π^*_τ would have improved the outcome of the decision making process. To this end, for each observed realization τ, we compute the relative difference between the average optimal counterfactual outcome and the observed outcome, that is, $(\bar{o}_{\pi^*_\tau}(\tau) - o(\tau))/o(\tau)$. Fig. 4.2 summarizes the results for different values of k. We find that the relative difference between the average counterfactual outcome and the observed outcome is always positive, that is, the sequence of actions specified by the counterfactual policy would have led the process realization to a better outcome in expectation. However, this may not come as a surprise given that the counterfactual policy π^*_τ is optimal, as shown in Proposition 4.1.1; Moreover, as the sequences of actions specified by the counterfactual policy differ more from the observed actions (*i.e.*, k increases), the improvement in terms of expected outcome increases.

Next, we investigate how the level of uncertainty (controlled by the parameter α) of the decision making process influences the average counterfactual outcome achieved by the optimal counterfactual policy π^*_τ as well as the number of distinct counterfactual action sequences π^*_τ provides. Fig. 4.3 summarizes the results, which reveal several interesting insights. As the level of uncertainty α increases, the average counterfactual outcome decreases, as shown in panel (a), however, the relative difference with respect to the observed outcome increases, as shown in panel (b). This



(a) $\mathbb{E}_\tau[\bar{o}_{\pi^*_\tau}(\tau)]$ vs. k

(b) $\mathbb{E}_\tau[(\bar{o}_{\pi^*_\tau}(\tau) - o(\tau))/o(\tau)]$ vs. α

(c) # of unique action sequences vs. k

Figure 4.3: **Effects of the level of uncertainty in the decision making process.** Panel (a) shows the average counterfactual outcome $\bar{o}_{\pi^*_\tau}(\tau)$ achieved by the optimal counterfactual policy π^*_τ. Panel (b) shows the relative difference between the average counterfactual outcome $\bar{o}_{\pi^*_\tau}(\tau)$ and the observed outcome $o(\tau)$. Panel (c) shows the number of distinct counterfactual action sequences π^*_τ provides. In all panels, we set $n = 20$, $m = 10$ and $d = 1{,}000$ and, in each experiment, use 500 realizations from 10 different instances of the decision making process (50 realizations per instance), each with different w_s. In panel (c), for each realization, we sample 100 counterfactual realizations and compute the average number of unique counterfactual action sequences across realizations. Shaded regions correspond to 95% confidence intervals.

suggest that, under high level of uncertainty, the counterfactual action sequences may be more valuable to a decision maker who aims to improve their actions over time. However, in this context, we also find that, under high levels of uncertainty, the number of distinct counterfactual action sequences increases rapidly with k. As a result, it may be preferable to use relatively low values of k to be able to effectively show the counterfactual action sequences to a decision maker in practice.

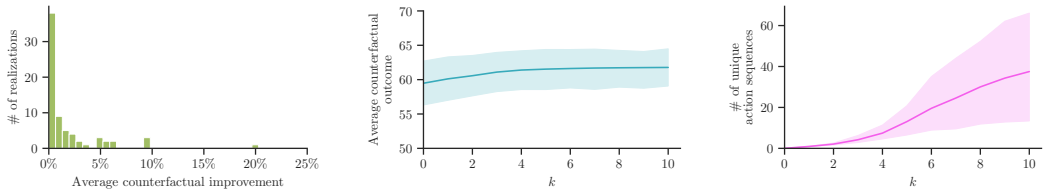
4.1.5 Experiments on real data

In this section, we evaluate Algorithm 6 using real patient data from a series of cognitive behavioral therapy sessions. To this end, similarly as in Section 4.1.4, we first measure the average outcome improvement that could have been achieved if at most k actions had been different to the observed ones in every therapy session, as dictated by the optimal counterfactual policy. Then, we look into individual therapy sessions and showcase how Algorithm 6, together with Algorithm 5, can be used to highlight specific patients and actions of interest for closer inspection.[3] Appendix C.4 contains additional experiments benchmarking the optimal counterfactual policy against several baselines.

Experimental setup. We use anonymized data from a clinical trial comparing the efficacy of hypnotherapy and cognitive behavioral therapy [207] for the treatment of patients with mild to moderate symptoms of major depression.[4] In our experiments, we use data from the 77 patients who received manualized cognitive behavioral therapy, which is one of the gold standards in depression treatment. Among these patients, we discard four of them because they attended less than

[3]Our results should be interpreted in the context of our modeling assumptions and they do not suggest the existence of medical malpractice.

[4]All participants gave written informed consent and the study protocol was peer-reviewed [208].



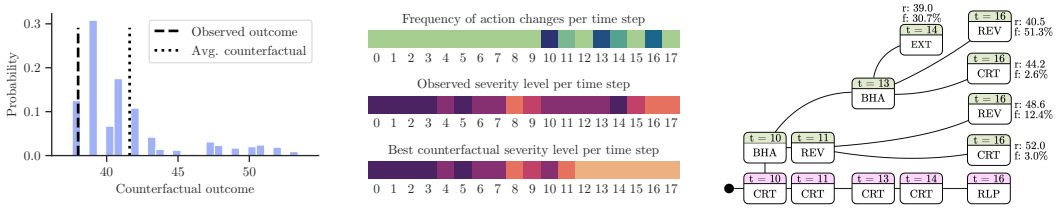
(a) $(\bar{o}_{\pi^*_\tau}(\tau) - o(\tau))/o(\tau)$ (b) $\frac{1}{|\mathcal{T}|}\sum_{\tau\in\mathcal{T}} \bar{o}_{\pi^*_\tau}(\tau)$ vs. k (c) # of unique action sequences vs. k

Figure 4.4: **Performance achieved by the optimal counterfactual policy π^*_τ in a series of manualized cognitive behavioral therapy sessions $\mathcal{T}$.** Panel (a) shows the distribution of the relative difference between the average counterfactual outcome $\bar{o}_{\pi^*_\tau}(\tau)$ achieved by π^*_τ and the observed outcome $o(\tau)$, *i.e.*, $(\bar{o}_{\pi^*_\tau}(\tau) - o(\tau))/o(\tau)$, for $k = 3$. Panels (b) and (c) show the average counterfactual outcome $\bar{o}_{\pi^*_\tau}(\tau)$ achieved by π^*_τ and the average number of unique counterfactual action sequences provided by each π^*_τ, averaged across patients, against the number of actions k differing from the observed ones. In panel (c), for each realization, the average number of unique counterfactual action sequences provided by π^*_τ is estimated using 1,000 counterfactual realizations. In all panels, we set $d = 1{,}000$ and use data from 73 patients. Shaded regions correspond to 95% confidence intervals.

10 sessions. Each patient attended up to 20 weekly therapy sessions and, for each session, the dataset contains the topic of discussion, chosen by the therapist from a pre-defined set of topics (*e.g.*, psychoeducation, behavioural activation, cognitive restructuring techniques). Additionally, a severity score is included, based on a standardized questionnaire [209], filled by the patient at each session, which assesses the severity of depressive symptoms. For more details about the severity score and the pre-defined set of discussion topics, refer to Appendix B.3.

To derive the counterfactual transition probability for each patient, we start by creating an MDP with $n = 5$ states and $m = 9$ actions. Each state $s \in \mathcal{S} = \{0, \dots, 4\}$ corresponds to a severity score, where small numbers represent lower severity, and each action $a \in \mathcal{A}$ corresponds to a topic from the pre-defined list of topics that the therapists discussed during the sessions. Moreover, each realization of the MDP corresponds to the therapy sessions of a single patient ordered in chronological order and time horizon $T \in \{10, \dots, 20\}$ is the number of therapy sessions per patient. Here, we denote the set of realizations for all patients as $\mathcal{T}$.

In addition, to estimate the values of the transition probabilities, we proceed as follows. For every state-action pair (s_i, a), we assume a n-dimensional prior Dirichlet$(\alpha_{i1}, \dots, \alpha_{i1})$ on the probabilities $p_{j\,|\,i,a} = P(s_j \,|\, s_i, a)$, where $\alpha_{i,j} = 1$ if $j \in \{i-1, i, i+1\}$ and $\alpha_{i,j} = 0.01$ otherwise. Then, if we observe c_j transitions from state s_i to each state s_j after action a in the patients' therapy sessions $\mathcal{T}$, we have that the posterior of the probabilities $p_{j\,|\,i,a}$ is a Dirichlet$(\alpha_{i1} + c_1, \dots, \alpha_{in} + c_n)$. Finally, to estimate the value of the transition probability $P(s_j \,|\, s_i, a)$, we take the average of 100,000 samples from the posterior value $p_{j\,|\,i,a}$. This procedure sets the value of the transition probabilities proportionally to the number of times they appeared in the data, however, it ensures that all transition probability values are non zero and transitions between adjacent severity levels are much more likely to happen. Moreover, we set the immediate reward for a pair of state and action (s, a)



(a) Distribution of counterfactual outcomes

(b) Action changes and severity vs. t

(c) Unique counterfactual action sequences

Figure 4.5: **Insights provided by the optimal counterfactual policy π_τ^* for one real patient who received cognitive behavioral therapy.** Panel (a) shows the distribution of the counterfactual outcomes $o(\tau')$ for the counterfactual realizations τ' induced by π_τ^* and P_τ. Panel (b) shows, for each time step, how frequently a counterfactual action sequence changes the observed action as well as the observed severity level and the severity level in the counterfactual realization with the highest counterfactual outcome. Here, darker colors correspond to higher frequencies and higher severities. Panel (c) shows the action changes in the unique counterfactual action sequences (green) provided by π_τ^* along with the mean of counterfactual outcomes (r) that each one achieves and how frequently (f) they appear across the counterfactual realizations. Here, the bottom row shows the observed actions that were changed by at least one of the counterfactual action sequences. Refer to Appendix B.3 for a definition of the actions (*i.e.*, topics). In all panels, we set $d = 1{,}000$ and the results are estimated using 1,000 counterfactual realizations.

equal to $r(s, a) = 5 - s \in \{1, \ldots, 5\}$, that is, the lower the patient's severity level, the higher the reward. Here, if some state-action pair (s, a) is never observed in the data, we set its immediate reward to $r(s, a) = -\infty$. This ensures that those state-action pairs never appear in a realization induced by the optimal counterfactual policy. Finally, to compute the counterfactual transition probability $P_{\tau,t}$ for each realization $\tau \in \mathcal{T}$, we follow the procedure described in Section 4.1.1 with $d = 1{,}000$ samples for each noise posterior distribution.

Results. We first measure to what extent the counterfactual action sequences provided by the optimal counterfactual policy π_τ^* would have improved each patient's severity of depressive symptoms over time. To this end, for each observed realization $\tau \in \mathcal{T}$ corresponding to each patient, we compute the same quality metrics as in experiments on synthetic data in Section 4.1.4. Fig. 4.4 summarizes the results. Panel (a) reveals that, for most patients, the improvement in terms of relative difference between the average optimal counterfactual outcome $\bar{o}_{\pi_\tau^*}(\tau)$ and the observed outcome $o(\tau)$ is rather modest. Moreover, panel (b) also shows that the absolute average optimal counterfactual outcome $\bar{o}_{\pi_\tau^*}(\tau)$, averaged across patients, does not increase significantly even if one allows for more changes k in the sequence of observed actions. These findings suggest that, in retrospect, the choice of topics by most therapists in the sessions was almost optimal. That being said, for 20% of the patients, the average counterfactual outcome improves a $\geq$3 % over the observed outcome and, as we will later discuss, there exist individual counterfactual realizations in which the counterfactual outcome improves much more than 3%. In that context, it is also important to note that, as shown in panel (c), the growth in the number of unique counterfactual action sequences with respect to k is weaker than

the growth found in the experiments with synthetic data and, for $k \leq 4$, the number of unique counterfactual action sequences is smaller than 10. This latter finding suggests that, in practice, it may be possible to effectively show, or summarize, the optimal counterfactual action sequences, a possibility that we investigate next.

We focus on a patient for whom the average counterfactual outcome $\bar{o}_{\pi_\tau^*}(\tau)$ achieved by the optimal policy π_τ^* with $k = 3$ improves 9.5% over the observed outcome $o(\tau)$. Then, using the policy π_τ^*, also with $k = 3$, and the counterfactual transition probability P_τ, we sample multiple counterfactual realizations τ' using Algorithm 5 and look at each counterfactual outcome $o(\tau')$. Fig. 4.5a summarizes the results, which show that, in most of these counterfactual realizations, the counterfactual outcome is greater than the observed outcome—if at most k actions had been different to the observed ones, as dictated by the optimal policy, there is a high probability that the outcome would have improved. Next, we investigate to what extent there are specific time steps within the counterfactual realizations τ' where π_τ^* is more likely to suggest an action change. Fig. 4.5b shows that, for the patient under study, there are indeed time steps that are overrepresented in the optimal counterfactual action sequences, namely $t \in \{10, 13, 16\}$. Moreover, the first of these time steps ($t = 10$) is when the patient had started worsening their depression after an earlier period in which they showed signs of recovery. Remarkably, we find that, in the counterfactual realization τ' with the best counterfactual outcome, the worsening is mostly avoided. Finally, we look closer into the actual action changes suggested by the optimal counterfactual policy π_τ^*. Fig. 4.5c summarizes the results, which reveal that π_τ^* recommends replacing some of the sessions on cognitive restructuring techniques (CRT) by behavioral activation (BHA) consistently across counterfactual realizations τ', particularly at the start of the worsening period. We discussed this recommendation with one of the researchers on clinical psychology who co-authored [207] and told us that, from a clinical perspective, such recommendation is sensible since, whenever the severity of depressive symptoms is high, it is very challenging to apply CRT and instead it is quite common to use BHA. Appendix C.3 contains additional insights about other patients in the dataset.

4.2 Sequential decisions in continuous state spaces

In the previous section, we have introduced a model of sequential decision making in state spaces that are discrete and finite. However, in many real-life applications, the state of the environment is inherently continuous in nature. For example, in critical care, a clinician typically cares about variables affecting the health status of a patient, such as blood pressure, body temperature, and respiratory rate [210]. Thus, in such cases, the state of the environment may be better described by a set of multidimensional vectors than by a finite set of discrete states.

Here, we introduce an extension of the SCM presented in the previous section, adapted to MDPs with continuous states. Specifically, we restrict our attention to the class of *bijective* SCMs [211], which includes multiple models introduced in the causal discovery literature [212–217]. In a bijective SCM, given an observed episode, one can infer the exact values of the exogenous noise variables that led to this particular realization of the transition dynamics (*i.e.*, the posterior distribution is concentrated on a single value). As a result, the counterfactual transition dynamics

are deterministic. Hence, given an observed episode of a decision making process and in contrast to the previous section, our goal here is to find a single action sequence close to the observed one that maximizes the counterfactual outcome that would have been achieved in retrospect.

Building on the characterization of sequential decision making described above, we make the following contributions:

1. We formalize the problem of finding a counterfactually optimal action sequence for a particular episode in environments with continuous states under the constraint that it differs from the observed action sequence in at most k actions.

2. We show that the above problem is NP-hard using a novel reduction from the classic partition problem [168]. This is in contrast to the polynomial time complexity of the variant of the problem in environments with discrete states that we have studied in the previous section.

3. We develop a search method based on the A^* algorithm that, under a natural form of Lipschitz continuity of the environment's dynamics, is guaranteed to return the optimal solution to the problem upon termination.

Finally, we evaluate the performance and qualitative insights of our method by performing a series of experiments using real patient data from critical care. The findings indicate that, despite the increased computational complexity of the problem, the proposed method is very efficient in practice and has the potential to provide valuable insights for sequential decision making tasks. The code used for all experiments in Section 4.2 is available at https://github.com/Networks-Learning/counterfactual-continuous-mdp.

4.2.1 Modeling sequential decisions with bijective SCMs

At each time step $t \in [T-1]_0$, where T is a time horizon, the decision making process is characterized by a d-dimensional vector state $\boldsymbol{s}_t \in \mathcal{S} = \mathbb{R}^d$, an action $a_t \in \mathcal{A}$, where $\mathcal{A}$ is a finite set of m actions, and a reward $r(\boldsymbol{s}_t, a_t) \in \mathbb{R}$ associated with each pair of states and actions. Moreover, given an episode of the decision making process, $\tau = \{(\boldsymbol{s}_t, a_t)\}_{t=0}^{T-1}$, the process's outcome $o(\tau) = \sum_t r(\boldsymbol{s}_t, a_t)$ is given by the sum of the rewards. In the remainder, we will denote the elements of a vector $\boldsymbol{s}_t$ as $s_{t,1}, \ldots, s_{t,d}$.

Further, we characterize the dynamics of the decision making process using the framework of structural causal models (SCMs) [71]. Similarly as in Section 4.1, the endogenous variables of the SCM $\mathcal{C}$ are the random variables representing the states $\boldsymbol{S}_0, \ldots, \boldsymbol{S}_{T-1}$ and the actions $A_0, \ldots, A_{T-1}$. The action A_t at time step t is chosen based on the observed state $\boldsymbol{S}_t$ and is given by a structural (policy) equation

$$A_t := g_A(\boldsymbol{S}_t, \boldsymbol{Z}_t), \tag{4.11}$$

where $\boldsymbol{Z}_t \in \mathcal{Z}$ is a vector-valued noise variable, to allow some level of stochasticity in the choice of the action, and its prior distribution $P^{\mathcal{C}}(\boldsymbol{Z}_t)$ is characterized by a density function $f^{\mathcal{C}}_{\boldsymbol{Z}_t}$. Similarly, the state $\boldsymbol{S}_{t+1}$ in the next time step is given by a structural (transition) equation

$$\boldsymbol{S}_{t+1} := g_S(\boldsymbol{S}_t, A_t, \boldsymbol{U}_t), \tag{4.12}$$

where $\boldsymbol{U}_t \in \mathcal{U}$ is a vector-valued noise variable with its prior distribution $P^{\mathcal{C}}(\boldsymbol{U}_t)$ having a density function $f^{\mathcal{C}}_{\boldsymbol{U}_t}$, and g_S is the transition mechanism. Note that, in Eq. 4.12, the noise variables $\{\boldsymbol{U}_t\}_{t=0}^{T-1}$ are mutually independent and, keeping the sequence of actions fixed, they are the only source of stochasticity in the dynamics of the environment. In other words, a sampled sequence of noise values $\{\boldsymbol{u}_t\}_{t=0}^{T-1}$ and a fixed sequence of actions $\{a_t\}_{t=0}^{T-1}$ result into a single (deterministic) sequence of states $\{\boldsymbol{s}_t\}_{t=0}^{T-1}$. This implicitly assumes that the state transitions are stationary and there are no unobserved confounders. The causal graph G corresponding to the SCM $\mathcal{C}$ is the same as the one presented in Fig. 4.1 in Section 4.1.

The above representation of sequential decision making using an SCM $\mathcal{C}$ is a more general reformulation of a Markov decision process, where a (stochastic) policy $\pi(a \,|\, \boldsymbol{s})$ is entailed by Eq. 4.11, and the transition distribution (*i.e.*, the conditional distribution $P(\boldsymbol{S}_{t+1} \,|\, \boldsymbol{S}_t, A_t)$) is entailed by Eq. 4.12. Specifically, the conditional density function of $\boldsymbol{S}_{t+1} \,|\, \boldsymbol{S}_t, A_t$ is given by

$$p^{\mathcal{C}}(\boldsymbol{S}_{t+1} = \boldsymbol{s} \,|\, \boldsymbol{S}_t = \boldsymbol{s}_t, A_t = a_t) = p^{\mathcal{C}\,;\, do[A_t = a_t]}(\boldsymbol{S}_{t+1} = \boldsymbol{s} \,|\, \boldsymbol{S}_t = \boldsymbol{s}_t)$$
$$= \int_{\boldsymbol{u} \in \mathcal{U}} \mathbb{1}[\boldsymbol{s} = g_S(\boldsymbol{s}_t, a_t, \boldsymbol{u})] \cdot f^{\mathcal{C}}_{\boldsymbol{U}_t}(\boldsymbol{u}) d\boldsymbol{u}, \quad (4.13)$$

where $do[A_t = a_t]$ denotes a (hard) intervention on the variable A_t, whose value is set to a_t. Here, the first equality holds because $\boldsymbol{S}_{t+1}$ and A_t are d-separated by $\boldsymbol{S}_t$ in the sub-graph obtained from G after removing all outgoing edges of A_t[5] and the second equality follows from Eq. 4.12.

Moreover, as argued in Section 4.1, by using an SCM to represent sequential decision making, instead of a standard MDP, we can answer counterfactual questions. More specifically, assume that, at time step t, we observed the state $\boldsymbol{S}_t = \boldsymbol{s}_t$, we took action $A_t = a_t$ and the next state was $\boldsymbol{S}_{t+1} = \boldsymbol{s}_{t+1}$. Retrospectively, we would like to know the probability that the state $\boldsymbol{S}_{t+1}$ would have been $\boldsymbol{s}'$ if, at time step t, we had been in a state $\boldsymbol{s}$, and we had taken an action a, (generally) different from $\boldsymbol{s}_t, a_t$. Using the SCM $\mathcal{C}$, we can characterize this by a counterfactual transition density function

$$p^{\mathcal{C} \,|\, \boldsymbol{S}_{t+1} = \boldsymbol{s}_{t+1}, \boldsymbol{S}_t = \boldsymbol{s}_t, A_t = a_t \,;\, do[A_t = a]}(\boldsymbol{S}_{t+1} = \boldsymbol{s}' \,|\, \boldsymbol{S}_t = s) =$$
$$\int_{\boldsymbol{u} \in \mathcal{U}} \mathbb{1}[\boldsymbol{s}' = g_S(\boldsymbol{s}, a, \boldsymbol{u})] \cdot f^{\mathcal{C} \,|\, \boldsymbol{S}_{t+1} = \boldsymbol{s}_{t+1}, \boldsymbol{S}_t = \boldsymbol{s}_t, A_t = a_t}_{\boldsymbol{U}_t}(\boldsymbol{u}) d\boldsymbol{u}, \quad (4.14)$$

where $f^{\mathcal{C} \,|\, \boldsymbol{S}_{t+1} = \boldsymbol{s}_{t+1}, \boldsymbol{S}_t = \boldsymbol{s}_t, A_t = a_t}_{\boldsymbol{U}_t}$ is the posterior distribution of the noise variable $\boldsymbol{U}_t$ with support such that $\boldsymbol{s}_{t+1} = g_S(\boldsymbol{s}_t, a_t, \boldsymbol{u})$.

In what follows, we will assume that the transition mechanism g_S is continuous with respect to its last argument and the SCM $\mathcal{C}$ satisfies the following form of Lipschitz-continuity:

Definition 4.2.1. *An SCM $\mathcal{C}$ is Lipschitz-continuous iff the transition mechanism g_S and the reward r are Lipschitz-continuous with respect to their first argument,* i.e., *for each $a \in \mathcal{A}$, $\boldsymbol{u} \in \mathcal{U}$, there exists a Lipschitz constant $K_{a,\boldsymbol{u}} \in \mathbb{R}_+$ such that, for any $\boldsymbol{s}, \boldsymbol{s}' \in \mathcal{S}$, $\|g_S(\boldsymbol{s}, a, \boldsymbol{u}) - g_S(\boldsymbol{s}', a, \boldsymbol{u})\| \leq K_{a,\boldsymbol{u}} \|\boldsymbol{s} - \boldsymbol{s}'\|$, and, for each $a \in \mathcal{A}$, there exists a Lipschitz constant $C_a \in \mathbb{R}_+$ such that, for any $\boldsymbol{s}, \boldsymbol{s}' \in \mathcal{S}$, $|r(\boldsymbol{s}, a) - r(\boldsymbol{s}', a)| \leq C_a \|\boldsymbol{s} - \boldsymbol{s}'\|$. In both cases, $\|\cdot\|$ denotes the Euclidean distance.*

[5] This follows directly from the rules of *do*-calculus. For further details, refer to Chapter 3 of Pearl [71].

Note that, although they are not phrased in causal terms, similar Lipschitz continuity assumptions for the environment dynamics are common in prior work analyzing the theoretical guarantees of reinforcement learning algorithms [218–226]. Moreover, for practical applications (*e.g.*, in healthcare), this is a relatively mild assumption to make. Consider two patients whose vitals $\boldsymbol{s}$ and $\boldsymbol{s}'$ are similar at a certain point in time, they receive the same treatment a, and every unobserved factor $\boldsymbol{u}$ that may affect their health is also the same. Intuitively, Definition 4.2.1 implies that their vitals will also evolve similarly in the immediate future, that is, the values $g_S(\boldsymbol{s}, a, \boldsymbol{u})$ and $g_S(\boldsymbol{s}', a, \boldsymbol{u})$ will not differ dramatically. In this context, it is worth mentioning that, when the transition mechanism g_S is modeled by a neural network, it is possible to control its Lipschitz constant during training, and penalizing high values can be seen as a regularization method [227, 228].

Further, we will focus on bijective SCMs [211], a fairly broad class of SCMs, which subsumes multiple models studied in the causal discovery literature, such as additive noise models [212], post-nonlinear causal models [213], location-scale noise models [214] and more complex models with neural network components [215–217].

Definition 4.2.2. *An SCM $\mathcal{C}$ is bijective iff the transition mechanism g_S is bijective with respect to its last argument,* i.e.*, there is a well-defined inverse function $g_S^{-1} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathcal{U}$ such that, for every combination of $\boldsymbol{s}_{t+1}, \boldsymbol{s}_t, a_t, \boldsymbol{u}_t$ with $\boldsymbol{s}_{t+1} = g_S(\boldsymbol{s}_t, a_t, \boldsymbol{u}_t)$, it holds that $\boldsymbol{u}_t = g_S^{-1}(\boldsymbol{s}_t, a_t, \boldsymbol{s}_{t+1})$.*

Importantly, bijective SCMs allow for a more concise characterization of the counterfactual transition density given in Eq. 4.14. More specifically, after observing an event $\boldsymbol{S}_{t+1} = \boldsymbol{s}_{t+1}, \boldsymbol{S}_t = \boldsymbol{s}_t, A_t = a_t$, the value $\boldsymbol{u}_t$ of the noise variable $\boldsymbol{U}_t$ can only be such that $\boldsymbol{u}_t = g_S^{-1}(\boldsymbol{s}_t, a_t, \boldsymbol{s}_{t+1})$, that is, the posterior distribution of $\boldsymbol{U}_t$ is a point mass and its density is given by

$$f_{\boldsymbol{U}_t}^{\mathcal{C} \,|\, \boldsymbol{S}_{t+1}=\boldsymbol{s}_{t+1}, \boldsymbol{S}_t=\boldsymbol{s}_t, A_t=a_t}(\boldsymbol{u}) = \mathbb{1}[\boldsymbol{u} = g_S^{-1}(\boldsymbol{s}_t, a_t, \boldsymbol{s}_{t+1})]. \tag{4.15}$$

Then, for a given episode τ of the decision making process, we have that the (non-stationary) counterfactual transition density is given by

$$\begin{aligned} p_{\tau,t}(\boldsymbol{S}_{t+1} = \boldsymbol{s}' \,|\, \boldsymbol{S}_t = \boldsymbol{s}, A_t = a) &:= p^{\mathcal{C} \,|\, \boldsymbol{S}_{t+1}=\boldsymbol{s}_{t+1}, \boldsymbol{S}_t=\boldsymbol{s}_t, A_t=a_t \,;\, do[A_t=a]}(\boldsymbol{S}_{t+1} = \boldsymbol{s}' \,|\, \boldsymbol{S}_t = \boldsymbol{s}) \\ &= \int_{\boldsymbol{u} \in \mathcal{U}} \mathbb{1}[\boldsymbol{s}' = g_S(\boldsymbol{s}, a, \boldsymbol{u})] \cdot \mathbb{1}[\boldsymbol{u} = g_S^{-1}(\boldsymbol{s}_t, a_t, \boldsymbol{s}_{t+1})] d\boldsymbol{u} \\ &= \mathbb{1}\left[\boldsymbol{s}' = g_S\left(\boldsymbol{s}, a, g_S^{-1}\left(\boldsymbol{s}_t, a_t, \boldsymbol{s}_{t+1}\right)\right)\right]. \end{aligned} \tag{4.16}$$

Since this density is also a point mass, the resulting counterfactual dynamics are purely deterministic. That means, under a bijective SCM , the answer to the question "*What would have been the state at time $t+1$, had we been at state $\boldsymbol{s}$ and taken action a at time t, given that, in reality, we were at $\boldsymbol{s}_t$, we took a_t and the environment transitioned to $\boldsymbol{s}_{t+1}$?*" is just given by $\boldsymbol{s}' = g_S\left(\boldsymbol{s}, a, g_S^{-1}\left(\boldsymbol{s}_t, a_t, \boldsymbol{s}_{t+1}\right)\right)$.

On the counterfactual identifiability of bijective SCMs. Very recently, Nasr-Esfahany and Kiciman [229] have shown that bijective SCMs are in general not counterfactually identifiable when the exogenous variable $\boldsymbol{U}_t$ is multi-dimensional. In other words, even with access to an infinite amount of triplets $(\boldsymbol{s}_t, a_t, \boldsymbol{s}_{t+1})$ sampled from the true SCM $\mathcal{C}$, it is always possible to find an SCM $\mathcal{C}' \neq \mathcal{C}$ with transition mechanism h_S and distributions $P^{\mathcal{C}'}(\boldsymbol{U}_t)$ that entails the same transition

distributions as $\mathcal{C}$ (*i.e.*, it fits the observational data perfectly), but leads to different counterfactual predictions. Although our subsequent algorithmic results do not require the SCM $\mathcal{C}$ to be counterfactually identifiable, the subclass of bijective SCMs we will use in our experiments in Section 4.2.4 is counterfactually identifiable. The defining attribute of this subclass, which we refer to as *element-wise bijective SCMs*, is that the transition mechanism g_S can be decoupled into d independent mechanisms $g_{S,i}$ such that $S_{t+1,i} = g_{S,i}(\boldsymbol{S}_t, A_t, U_{t,i})$ for $i \in \{1, \dots, d\}$. Formally:

Definition 4.2.3. *An SCM $\mathcal{C}$ is element-wise bijective iff it is bijective and there exist functions $g_{S,i} : \mathbb{R} \times \mathcal{A} \times \mathbb{R} \to \mathbb{R}$ with $i \in \{1, \dots, d\}$ such that, for every combination of $\boldsymbol{s}_{t+1}, \boldsymbol{s}_t, a_t, \boldsymbol{u}_t$ with $\boldsymbol{s}_{t+1} = g_S(\boldsymbol{s}_t, a_t, \boldsymbol{u}_t)$, it holds that $s_{t+1,i} = g_{S,i}(\boldsymbol{s}_t, a_t, u_{t,i})$ for $i \in \{1, \dots, d\}$.*

Note that, in an element-wise bijective SCM, it holds that $S_{t+1,i} \perp\!\!\!\perp U_{t,j} \,|\, U_{t,i}, \boldsymbol{S}_t, A_t$ for $j \neq i$, however, $U_{t,i}$, $U_{t,j}$ do not need to be independent. Moreover, under our assumption that the transition mechanism g_S is continuous with respect to its third argument, it is easy to see that, for any element-wise bijective SCM, the functions $g_{S,i}$ are always strictly monotonic functions of the respective $u_{t,i}$. Based on this observation, we have the following theorem of counterfactual identifiability whose proof follows a similar reasoning to proofs found in related work [132, 211]:[6]

Theorem 4.2.1. *Let $\mathcal{C}$ and $\mathcal{C}'$ be two element-wise bijective SCMs with transition mechanisms g_S and h_S, respectively, and, for any observed transition $(\boldsymbol{s}_t, a_t, \boldsymbol{s}_{t+1})$, let $\boldsymbol{u}_t = g_S^{-1}(\boldsymbol{s}_t, a_t, \boldsymbol{s}_{t+1})$ and $\tilde{\boldsymbol{u}}_t = h_S^{-1}(\boldsymbol{s}_t, a_t, \boldsymbol{s}_{t+1})$. Moreover, given any $\boldsymbol{s} \in \mathcal{S}, a \in \mathcal{A}$, let $\boldsymbol{s}' = g_S(\boldsymbol{s}, a, \boldsymbol{u}_t)$ and $\boldsymbol{s}'' = h_S(\boldsymbol{s}, a, \tilde{\boldsymbol{u}}_t)$. If $P^{\mathcal{C}}(\boldsymbol{S}_{t+1} \,|\, \boldsymbol{S}_t = \boldsymbol{s}, A_t = a) = P^{\mathcal{C}'}(\boldsymbol{S}_{t+1} \,|\, \boldsymbol{S}_t = \boldsymbol{s}, A_t = a)$ for all $\boldsymbol{s} \in \mathcal{S}, a \in \mathcal{A}$, it must hold that $\boldsymbol{s}' = \boldsymbol{s}''$.*

4.2.2 Problem statement

Let τ be an observed episode of a decision making process whose dynamics are characterized by a Lipschitz-continuous bijective SCM. To characterize the counterfactual outcome that any alternative action sequence would have achieved under the circumstances of the particular episode, we build upon the formulation of Section 4.2.2, and we define a non-stationary counterfactual MDP $\mathcal{M}^+ = (\mathcal{S}^+, \mathcal{A}, F^+_{\tau,t}, r^+, T)$ with deterministic transitions. Here, $\mathcal{S}^+ = \mathcal{S} \times [T-1]_0$ is an enhanced state space such that each $\boldsymbol{s}^+ \in \mathcal{S}^+$ is a pair $(\boldsymbol{s}, l)$ indicating that the counterfactual episode would have been at state $\boldsymbol{s} \in \mathcal{S}$ with l action changes already performed. Accordingly, r^+ is a reward function which takes the form $r^+((\boldsymbol{s}, l), a) = r(\boldsymbol{s}, a)$ for all $(\boldsymbol{s}, l) \in \mathcal{S}^+$, $a \in \mathcal{A}$, that is, it does not change depending on the number of action changes already performed. Finally, the time-dependent transition function $F^+_{\tau,t} : \mathcal{S}^+ \times \mathcal{A} \to \mathcal{S}^+$ is defined as

$$F^+_{\tau,t}\left((\boldsymbol{s}, l), a\right) = \begin{cases} \left(g_S\left(\boldsymbol{s}, a, g_S^{-1}\left(\boldsymbol{s}_t, a_t, \boldsymbol{s}_{t+1}\right)\right), l+1\right) & \text{if } (a \neq a_t) \\ \left(g_S\left(\boldsymbol{s}, a_t, g_S^{-1}\left(\boldsymbol{s}_t, a_t, \boldsymbol{s}_{t+1}\right)\right), l\right) & \text{otherwise.} \end{cases} \tag{4.17}$$

Intuitively, here we set the transition function according to the point mass of the counterfactual transition density given in Eq. 4.16, and we use the second coordinate to keep track of the changes that have been performed in comparison to the observed action sequence up to the time step t.

[6] All proofs for Section 4.2 can be found in Appendix A.4.

Now, given the initial state $\boldsymbol{s}_0$ of the episode τ and any counterfactual action sequence $\{a'_t\}_{t=0}^{T-1}$, we can compute the corresponding counterfactual episode $\tau' = \{(\boldsymbol{s}'_t, l_t), a'_t\}_{t=0}^{T-1}$. Its sequence of states is given recursively by

$$\begin{aligned} &(\boldsymbol{s}'_1, l_1) = F^+_{\tau,0}\left((\boldsymbol{s}_0, 0), a'_0\right) \text{ and} \\ &\left(\boldsymbol{s}'_{t+1}, l_{t+1}\right) = F^+_{\tau,0}\left((\boldsymbol{s}'_t, l_t), a'_t\right) \text{ for } t \in [T-1], \end{aligned} \tag{4.18}$$

and $o^+(\tau') = \sum_t r^+\left((\boldsymbol{s}'_t, l_t), a'_t\right) = \sum_t r\left(\boldsymbol{s}'_t, a'_t\right)$ is its counterfactual outcome.

Then, our ultimate goal is to find the counterfactual action sequence $\{a'_t\}_{t=0}^{T-1}$ that, starting from the observed initial state $\boldsymbol{s}_0$, maximizes the counterfactual outcome subject to a constraint on the number of counterfactual actions that can differ from the observed ones, that is,

$$\underset{a'_0,\ldots,a'_{T-1}}{\text{maximize}} \quad o^+(\tau') \qquad \text{subject to} \quad \boldsymbol{s}'_0 = \boldsymbol{s}_0 \text{ and } \sum_{t=0}^{T-1} \mathbb{1}[a_t \neq a'_t] \leq k, \tag{4.19}$$

where $a_0, \ldots, a_{T-1}$ are the observed actions. Unfortunately, using a reduction from the classic partition problem [168], the following theorem shows that we cannot hope to find the optimal action sequence in polynomial time:

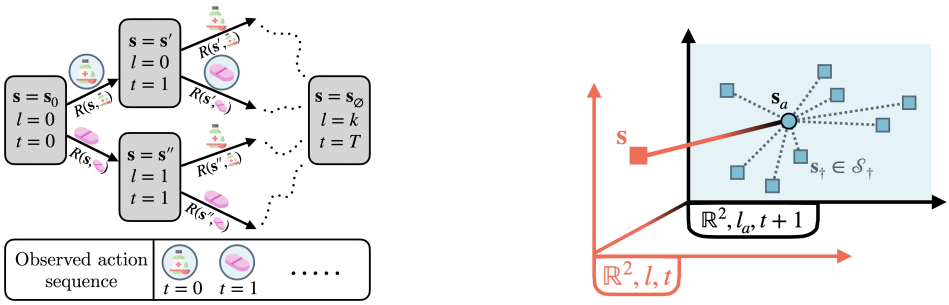
Theorem 4.2.2. *The problem defined by Eq. 4.19. is NP-Hard.*

The proof of the theorem relies on a reduction from the partition problem [168], which is known to be NP-complete, to our problem, defined in Eq. 4.19. At a high-level, we map any instance of the partition problem to an instance of our problem, taking special care to construct a reward function and an observed action sequence, such that the optimal counterfactual outcome $o^+(\tau^*)$ takes a specific value if and only if there exists a valid partition for the original instance. The hardness result of Theorem 4.2.2 motivates our subsequent focus on the design of a method that *always* finds the optimal solution to our problem at the expense of a potentially higher runtime for some problem instances.

4.2.3 An efficient method based on A* search

To deal with the increased computational complexity of the problem, we develop an optimal search method based on the classic A^* algorithm [230], which we have found to be very efficient in practice. Our starting point is the observation that, the problem of Eq. 4.19 presents an optimal substructure, that is, its optimal solution can be constructed by combining optimal solutions to smaller sub-problems. For an observed episode τ, let $V_\tau(\boldsymbol{s}, l, t)$ be the maximum counterfactual reward that could have been achieved in a counterfactual episode where, at time t, the process is at a (counterfactual) state $\boldsymbol{s}$, and there are so far l actions that have been different in comparison with the observed action sequence. Formally,

$$\begin{aligned} V_\tau(\boldsymbol{s}, l, t) = &\max_{a'_t,\ldots,a'_{T-1}} \sum_{t'=t}^{T-1} r(\boldsymbol{s}'_{t'}, a'_{t'}) \\ \text{subject to} \quad &\boldsymbol{s}'_t = \boldsymbol{s} \text{ and } \sum_{t'=t}^{T-1} \mathbb{1}[a_{t'} \neq a'_{t'}] \leq k - l. \end{aligned}$$



(a) Search graph

(b) Heuristic function computation

Figure 4.6: **Main components of our search method based on the A^* algorithm.** Panel (a) shows the search graph for a problem instance with $|\mathcal{A}| = 2$. Here, each box represents a node $v = (\boldsymbol{s}, l, t)$ of the graph, and each edge represents a counterfactual transition. Next to each edge, we include the action $a \in \mathcal{A}$ causing the transition and the associated reward. Panel (b) shows the heuristic function computation, where the two axes represent a (continuous) state space $\mathcal{S} = \mathbb{R}^2$ and the two levels on the z-axis correspond to differences in the (integer) values (l, t) and $(l_a, t+1)$. Here, the blue squares correspond to the finite states in the anchor set $\mathcal{S}_\dagger$ and $(\boldsymbol{s}_a, l_a) = F^+_{\tau,t}((\boldsymbol{s}, l), a)$.

Then, it is easy to see that the quantity $V_\tau(\boldsymbol{s}, l, t)$, for all $\boldsymbol{s} \in \mathcal{S}, l < k$ and $t < T-1$, can be given by the recursive function

$$V_\tau(\boldsymbol{s}, l, t) = \max_{a \in \mathcal{A}} \{r(\boldsymbol{s}, a) + V_\tau(\boldsymbol{s}_a, l_a, t+1)\}, \tag{4.20}$$

where $(\boldsymbol{s}_a, l_a) = F^+_{\tau,t}((\boldsymbol{s}, l), a)$. In the base case of $l = k$ (*i.e.*, all allowed action changes are already performed), we have $V_\tau(\boldsymbol{s}, k, t) = r(\boldsymbol{s}, a_t) + V_\tau(\boldsymbol{s}_{a_t}, l_{a_t}, t+1)$ for all $\boldsymbol{s} \in \mathcal{S}$ and $t < T-1$, and $V_\tau(\boldsymbol{s}, k, T-1) = r(\boldsymbol{s}, a_{T-1})$ for $t = T-1$. Lastly, when $t = T-1$ and $l < k$, we have $V_\tau(\boldsymbol{s}, l, T-1) = \max_{a \in \mathcal{A}} r(\boldsymbol{s}, a)$ for all $\boldsymbol{s} \in \mathcal{S}$.

Given the optimal substructure of the problem, one may be tempted to employ a typical dynamic programming approach to compute the values $V_\tau(\boldsymbol{s}, l, t)$ in a bottom-up fashion. However, the complexity of the problem lies in the fact that, the states $\boldsymbol{s}$ are real-valued vectors whose exact values depend on the entire action sequence that led to them. Hence, to enumerate all the possible values that $\boldsymbol{s}$ might take, one has to enumerate all possible action sequences in the search space, which is equivalent to solving our problem with a brute force search. In what follows, we present our proposed method to find optimal solutions using the A^* algorithm, with the caveat that its runtime varies depending on the problem instance, and it can be equal to that of a brute force search in the worst case.

Casting the problem as graph search. We represent the solution space of our problem as a graph, where each node v corresponds to a tuple $(\boldsymbol{s}, l, t)$ with $\boldsymbol{s} \in \mathcal{S}$, $l \in [k]$ and $t \in [T]_0$. Every node $v = (\boldsymbol{s}, l, t)$ with $l < k$ and $t < T-1$ has $|\mathcal{A}|$ outgoing edges, each one associated with an action $a \in \mathcal{A}$, carrying a reward $r(\boldsymbol{s}, a)$, and leading to a node $v_a = (\boldsymbol{s}_a, l_a, t+1)$ such that $(\boldsymbol{s}_a, l_a) = F^+_{\tau,t}((\boldsymbol{s}, l), a)$. In the case of $l = k$, the node v has exactly one edge corresponding to the observed action a_t at time t. Lastly, when $t = T-1$, the outgoing edge(s) lead(s) to a common node $v_T = (\boldsymbol{s}_\emptyset, k, T)$ which we call the *goal node*, and it has zero outgoing edges itself.

Note that, the exact value of $\boldsymbol{s}_\emptyset$ is irrelevant, and we only include it for notational completeness.

Let $\boldsymbol{s}_0$ be the initial state of the observed episode. Then, it is easy to notice that, starting from the root node $v_0 = (\boldsymbol{s}_0, 0, 0)$, the first elements of each node v_i on a path $v_0, \ldots, v_i, \ldots, v_T$ form a sequence of counterfactual states, and the edges that connect those nodes are such that the corresponding counterfactual action sequence differs from the observed one in at most k actions. That said, the counterfactual outcome $o^+(\tau) = \sum_{t=0}^{T-1} r(\boldsymbol{s}'_t, a'_t)$ is expressed as the sum of the rewards associated with each edge in the path, and the problem defined by Eq. 4.19 is equivalent to finding the path of maximum total reward that starts from v_0 and ends in v_T. Fig. 4.6a illustrates the search graph for a simple instance of our problem.[7] Unfortunately, since the states $\boldsymbol{s}$ are vectors of real values, even enumerating all the graph's nodes requires time exponential in the number of actions $|\mathcal{A}|$, which makes classic algorithms that search over the entire graph non-practical.

To address this challenge, we resort to the A^* algorithm, which performs a more efficient search over the graph by preferentially exploring only parts of it where we have prior information that they are more likely to lead to paths of higher total reward. Concretely, the algorithm proceeds iteratively and maintains a queue of nodes to visit, initialized to contain only the root node v_0. Then, at each step, it selects one node from the queue, and it retrieves all its children nodes in the graph which are subsequently added to the queue. It terminates when the node being visited is the goal node v_T. Algorithm 7 summarizes the procedure. Therein, we represent each node v by an object with 4 attributes: (i) the "tuple" $(\boldsymbol{s}, l, t)$ of the node, (ii) the total reward "rwd" of the path that has led the search from the root node v_0 to the node v, (iii) the parent node "par" from which the search arrived to v, and (iv) the action "act" associated with the edge connecting the node v with its parent. In addition to the queue of nodes to visit, the algorithm maintains a set of explored nodes and adds a new node to the queue only if it has not been previously explored. The algorithm terminates when the goal node is chosen to be visited, that is, the "tuple" attribute of v has the format $(*, *, T)$, where $*$ denotes arbitrary values. Once the goal node v_T has been visited, the algorithm reconstructs and returns the action sequence that led from the root node v_0 to the goal node.

The key element of the A^* algorithm is the criterion based on which it selects which node from the queue to visit next. Let $v_i = (\boldsymbol{s}_i, l_i, t)$ be a candidate node in the queue and r_{v_i} be the total reward of the path that the algorithm has followed so far to reach from v_0 to v_i. Then, the A^* algorithm visits next the node v_i that maximizes the sum $r_{v_i} + \hat{V}_\tau(\boldsymbol{s}_i, l_i, t)$, where $\hat{V}_\tau$ is a *heuristic function* that aims to estimate the maximum reward that can be achieved via any path starting from $v_i = (\boldsymbol{s}_i, l_i, t)$ and ending in the goal node v_T, that is, it gives an estimate for the quantity $V_\tau(\boldsymbol{s}_i, l_i, t)$. Intuitively, the heuristic function can be thought of as an "eye into the future" of the graph search, that guides the algorithm towards nodes that are more likely to lead to the optimal solution and the algorithm's performance depends on the quality of the approximation of $V_\tau(\boldsymbol{s}_i, l_i, t)$ by $\hat{V}_\tau(\boldsymbol{s}_i, l_i, t)$. Next, we will look for a heuristic function that satisfies *consistency*. Formally, a heuristic function $\hat{V}_\tau$ is consistent iff, for nodes $v = (\boldsymbol{s}, l, t)$, $v_a = (\boldsymbol{s}_a, l_a, t+1)$ connected with an edge associated with action a, it satisfies $\hat{V}_\tau(\boldsymbol{s}, l, t) \geq r(\boldsymbol{s}, a) + \hat{V}_\tau(\boldsymbol{s}_a, l_a, t+1)$ [231]. Given a consistent

[7] We would like to give credit to creators Freepik and vectorsmarket15 from flaticon.com whose icons we have used to design Fig. 4.6a.

Algorithm 7: Graph search via A^*

input : states $\mathcal{S}$, actions $\mathcal{A}$, observed action sequence $\{a_t\}_{t=0}^{t=T-1}$, horizon T, transition function $F_{\tau,t}^{+}$, reward function r, constraint k, initial state $\boldsymbol{s}_0$, heuristic function $\hat{V}_\tau$.

output : optimal counterfactual action sequence $\{a_t^*\}_{t=0}^{T-1}$

NODE $v_0 \leftarrow \{\texttt{tuple} : (\boldsymbol{s}_0, 0, 0), \texttt{rwd} : 0, \texttt{par} : Null, \texttt{act} : Null\}$
STACK action_sequence $\leftarrow [\,]$
QUEUE $Q \leftarrow \{root\}$
SET explored $\leftarrow \emptyset$
while *True* **do**
  $v \leftarrow \text{argmax}_{v' \in Q}\{v'.\texttt{rwd} + \hat{V}_\tau(v'.\texttt{tuple})\}$ `// Next node to visit`
  $Q \leftarrow Q \setminus v$
  if $v.\texttt{tuple} = (*, *, T)$ **then**
    while $v.\texttt{par} \neq Null$ **do**
      action_sequence.push(v.act) `// Retrieve final action sequence`
      $v \leftarrow v.\texttt{par}$
    return action_sequence
  explored $\leftarrow$ explored $\cup \{v\}$ `// Set node` v `as explored`
  if $l = k$ **then**
    availabe_actions $\leftarrow \{a_t\}$
  else
    availabe_actions $\leftarrow \mathcal{A}$
  for $a \in$ *available_actions* **do**
    $(\boldsymbol{s}, l, t) \leftarrow v.\texttt{tuple}$
    $(\boldsymbol{s}_a, l_a) \leftarrow F_{\tau,t}^{+}((\boldsymbol{s}, l), a)$ `// Identify` v`'s children nodes`
    $v_a \leftarrow \{\texttt{tuple} : (\boldsymbol{s}_a, l_a, t+1), \texttt{rwd} : v.\texttt{rwd} + r(\boldsymbol{s}, a), \texttt{par} : v, \texttt{act} : a\}$
    if $v_a \notin Q$ ***and*** $v_a \notin$ *explored* **then**
      $Q \leftarrow Q \cup \{v_a\}$ `// Add them to the queue if unexplored`

heuristic function, the A^* algorithm as described above is guaranteed to return the optimal solution upon termination [230].

Computing a consistent heuristic function. We first propose an algorithm that computes the function's values $\hat{V}_\tau(\boldsymbol{s}, l, t)$ for a finite set of points such that $l \in [k]$, $t \in [T-1]_0$, $\boldsymbol{s} \in \mathcal{S}_\dagger \subset \mathcal{S}$, where $\mathcal{S}_\dagger$ is a pre-defined *finite* set of states—an *anchor set*—whose construction we discuss later. Then, based on the Lipschitz-continuity of the SCM $\mathcal{C}$, we show that these computed values of $\hat{V}_\tau$ are valid upper bounds of the corresponding values $V_\tau(\boldsymbol{s}, l, t)$ and we expand the definition of the heuristic function $\hat{V}_\tau$ over all $\boldsymbol{s} \in \mathcal{S}$ by expressing it in terms of those upper bounds. Finally, we prove that the function resulting from the aforementioned procedure is consistent.

To compute the upper bounds $\hat{V}_\tau$, we exploit the observation that the values $V_\tau(\boldsymbol{s}, l, t)$ satisfy a form of Lipschitz-continuity, as stated in the following lemma:

Lemma 4.2.1. *Let* $\boldsymbol{u}_t = g_S^{-1}(\boldsymbol{s}_t, a_t, \boldsymbol{s}_{t+1})$, $K_{\boldsymbol{u}_t} = \max_{a \in \mathcal{A}} K_{a,\boldsymbol{u}_t}$, $C = \max_{a \in \mathcal{A}} C_a$ *and the sequence* $L_0, \ldots, L_{T-1} \in \mathbb{R}_+$ *be such that* $L_{T-1} = C$ *and* $L_t = C + L_{t+1} K_{\boldsymbol{u}_t}$ *for* $t \in [T-2]$. *Then, it holds that* $|V_\tau(\boldsymbol{s}, l, t) - V_\tau(\boldsymbol{s}', l, t)| \leq L_t \|\boldsymbol{s} - \boldsymbol{s}'\|$, *for all* $t \in [T-1]_0$, $l \in [k]$ *and* $\boldsymbol{s}, \boldsymbol{s}' \in \mathcal{S}$.

Algorithm 8: It computes upper bounds $\hat{V}_\tau(\boldsymbol{s}, l, t)$ for the values $V_\tau(\boldsymbol{s}, l, t)$

input : states $\mathcal{S}$, actions $\mathcal{A}$, observed action sequence $\{a_t\}_{t=0}^{t=T-1}$, horizon T, transition function $F^+_{\tau,t}$, reward function r, constraint k, anchor set $\mathcal{S}_\dagger$

output : upper bounds $\hat{V}_\tau$

$\hat{V}_\tau(\boldsymbol{s}, l, T-1) \leftarrow \max_{a\in\mathcal{A}} r(\boldsymbol{s}, a)$ **for** $s \in \mathcal{S}_\dagger$, $l \in \{0, \dots, k-1\}$

$\hat{V}_\tau(\boldsymbol{s}, k, T-1) \leftarrow r(\boldsymbol{s}, a_t)$ **for** $s \in \mathcal{S}_\dagger$

for $t \leftarrow T-2$ **to** 0 **do**

  for $l \leftarrow k$ **to** 0 **do**

    available_actions $\leftarrow a_t$ **if** $l = k$ **else** $\mathcal{A}$

    for $\boldsymbol{s} \in \mathcal{S}_\dagger$ **do**

      bounds $\leftarrow \emptyset$

      for $a \in$ *available_actions* **do**

        /* Get the min bound for $V_\tau(\boldsymbol{s}_a, l_a, t+1)$ based on $\mathcal{S}_\dagger$ */

        $\boldsymbol{s}_a, l_a \leftarrow F^+_{\tau,t}((\boldsymbol{s}, l), a)$

        $V_a \leftarrow \min_{\boldsymbol{s}_\dagger\in\mathcal{S}_\dagger}\{\hat{V}_\tau(\boldsymbol{s}_\dagger, l_a, t+1) + L_{t+1}\|\boldsymbol{s}_\dagger - \boldsymbol{s}_a\|\}$

        bounds $\leftarrow$ bounds $\cup \{r(\boldsymbol{s}, a) + V_a\}$

      $\hat{V}_\tau(\boldsymbol{s}, l, t) \leftarrow \max($*bounds*$)$ // Get the max bound over the actions

return $\hat{V}_\tau(\boldsymbol{s}, l, t)$ **for** $\boldsymbol{s} \in \mathcal{S}_\dagger$, $l \in [k]$, $t \in [T-1]_0$

Based on this observation, our algorithm proceeds in a bottom-up fashion and computes valid upper bounds of the values $V_\tau(\boldsymbol{s}, l, t)$ for all $l \in [k]$, $t \in [T-1]_0$ and $\boldsymbol{s}$ in the anchor set $\mathcal{S}_\dagger$. To get the intuition, assume that, for a given t, the values $\hat{V}_\tau(\boldsymbol{s}, l, t+1)$ are already computed for all $\boldsymbol{s} \in \mathcal{S}_\dagger$, $l \in [k]$, and they are indeed valid upper bounds of the corresponding $V_\tau(\boldsymbol{s}, l, t+1)$. Then, let $(\boldsymbol{s}_a, l_a) = F^+_{\tau,t}((\boldsymbol{s}, l), a)$ for some $\boldsymbol{s} \in \mathcal{S}_\dagger$ and $l \in [k]$. Since $\boldsymbol{s}_a$ itself may not belong to the finite anchor set $\mathcal{S}_\dagger$, the algorithm uses the values $\hat{V}_\tau(\boldsymbol{s}_\dagger, l_a, t+1)$ of all anchors $\boldsymbol{s}_\dagger \in \mathcal{S}_\dagger$ in combination with their distance to $\boldsymbol{s}_a$, and it sets the value of $\hat{V}_\tau(\boldsymbol{s}, l, t)$ in way that it is also guaranteed to be a (maximally tight) upper bound of $V_\tau(\boldsymbol{s}, l, t)$. Fig. 4.6b illustrates the above operation. Algorithm 8 summarizes the overall procedure, which is guaranteed to return upper bounds, as shown by the following proposition:

Proposition 4.2.1. *For all $\boldsymbol{s} \in \mathcal{S}_\dagger$, $l \in [k], t \in [T-1]_0$, it holds that $\hat{V}_\tau(\boldsymbol{s}, l, t) \geq V_\tau(\boldsymbol{s}, l, t)$, where $\hat{V}_\tau(\boldsymbol{s}, l, t)$ are the values of the heuristic function computed by Algorithm 8.*

Next, we use the values $\hat{V}_\tau(\boldsymbol{s}, l, t)$ computed by Algorithm 8 to expand the definition of $\hat{V}_\tau$ over the entire domain as follows. For some $\boldsymbol{s} \in \mathcal{S}$, $a \in \mathcal{A}$, let $(\boldsymbol{s}_a, l_a) = F^+_{\tau,t}((\boldsymbol{s}, l), a)$, then, we have that

$$\hat{V}_\tau(\boldsymbol{s}, l, t) = \begin{cases} 0 & t = T \\ \max\limits_{a\in\mathcal{A}'} r(\boldsymbol{s}, a) & t = T-1 \\ \max\limits_{a\in\mathcal{A}'} \left\{ r(\boldsymbol{s}, a) + \min\limits_{\boldsymbol{s}_\dagger\in\mathcal{S}_\dagger} \left\{ \hat{V}_\tau(\boldsymbol{s}_\dagger, l_a, t+1) + L_{t+1}\|\boldsymbol{s}_\dagger - \boldsymbol{s}_a\| \right\} \right\} & \text{otherwise,} \end{cases} \tag{4.21}$$

where $\mathcal{A}' = \{a_t\}$ for $l = k$ and $\mathcal{A}' = \mathcal{A}$ for $l < k$. Finally, the following theorem shows that the resulting heuristic function $\hat{V}_\tau$ is consistent:

Theorem 4.2.3. *For any nodes $v = (\boldsymbol{s}, l, t), v_a = (\boldsymbol{s}_a, l_a, t+1)$ with $t < T-1$ connected with an edge associated with action a, it holds that $\hat{V}_\tau(\boldsymbol{s}, l, t) \geq r(\boldsymbol{s}, a) + \hat{V}_\tau(\boldsymbol{s}_a, l_a, t+1)$. Moreover, for any node $v = (\boldsymbol{s}, l, T-1)$ and edge connecting it to the goal node $v_T = (\boldsymbol{s}_\emptyset, k, T)$, it holds that $\hat{V}_\tau(\boldsymbol{s}, l, T-1) \geq r(\boldsymbol{s}, a) + \hat{V}_\tau(\boldsymbol{s}_\emptyset, k, T)$.*

Kick-starting the heuristic function computation with Monte Carlo anchor sets. For any $\boldsymbol{s} \notin \mathcal{S}_\dagger$, whenever we compute $\hat{V}_\tau(\boldsymbol{s}, l, t)$ using Eq. 4.21, the resulting value is set based on the value $\hat{V}_\tau(\boldsymbol{s}_\dagger, l_a, t+1)$ of some anchor $\boldsymbol{s}_\dagger$, increased by a *penalty* term $L_{t+1} \|\boldsymbol{s}_\dagger - \boldsymbol{s}_a\|$. Intuitively, this allows us to think of the heuristic function $\hat{V}_\tau$ as an upper bound of the function V_τ whose looseness depends on the magnitude of the penalty terms encountered during the execution of Algorithm 8 and each subsequent evaluation of Eq. 4.21. To speed up the A^* algorithm, note that, ideally, one would want all penalty terms to be zero, *i.e.*, an anchor set that includes all the states $\boldsymbol{s}$ of the nodes $v = (\boldsymbol{s}, l, t)$ that are going to appear in the search graph. However, as discussed in the beginning of Section 4.2.3, an enumeration of those states requires a runtime exponential in the number of actions.

To address this issue, we introduce a Monte Carlo simulation technique that adds to the anchor set the observed states $\{\boldsymbol{s}_0, \ldots, \boldsymbol{s}_{T-1}\}$ and all unique states $\{\boldsymbol{s}'_0, \ldots, \boldsymbol{s}'_{T-1}\}$ resulting by M randomly sampled counterfactual action sequences $a'_0, \ldots, a'_{T-1}$. Specifically, for each action sequence, we first sample a number k' of actions to be changed and what those actions are going to be, both uniformly at random from $[k]$ and $\mathcal{A}^{k'}$, respectively. Then, we sample from $[T-1]_0$ the k' time steps where the changes take place, with each time step t having a probability $L_t / \sum_{t'} L_{t'}$ to be selected. This biases the sampling towards earlier time steps, where the penalty terms are larger due to the higher Lipschitz constants. As we will see in the next section, this approach works well in practice, and it allows us to control the runtime of the A^* algorithm by appropriately adjusting the number of samples M. We experiment with additional anchor set selection strategies in Appendix C.5.

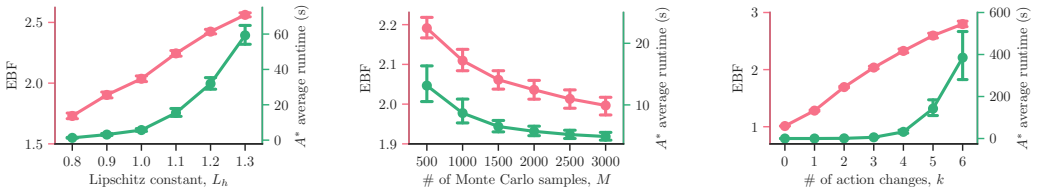
4.2.4 Experiments on real data

Here, we evaluate our method using real patient data from MIMIC-III [232], a freely accessible critical care dataset commonly used in reinforcement learning for healthcare [233–236].[8]

Experimental setup. We follow the preprocessing steps of Komorowski et al. [234] to identify a cohort of 20,926 patients treated for sepsis [237]. Each patient record contains vital signs and administered treatment information in time steps of 4-hour intervals. As an additional preprocessing step, we discard patient records whose associated time horizon T is shorter than 10, resulting in a final dataset of 15,992 patients with horizons between 10 and 20.

To form our state space $\mathcal{S} = \mathbb{R}^d$, we use $d = 13$ features. Four of these features are demographic or contextual and thus we always set their counterfactual values to the observed ones. The remaining $\tilde{d} = 9$ features are time-varying and include the SOFA score [238]—a standardized score of organ failure rate—along with eight vital signs that are required for its calculation. Since SOFA scores positively correlate with patient mortality [239], we assume that each $\boldsymbol{s} \in \mathcal{S}$ gives a reward $r(\boldsymbol{s})$ equal

[8] All experiments for Section 4.2 ran on an internal cluster of machines equipped with 16 Intel(R) Xeon(R) 3.20GHz CPU cores, 512GBs of memory and 2 NVIDIA A40 48GB GPUs.



(a) Efficiency vs. L_h (b) Efficiency vs. M (c) Efficiency vs. k

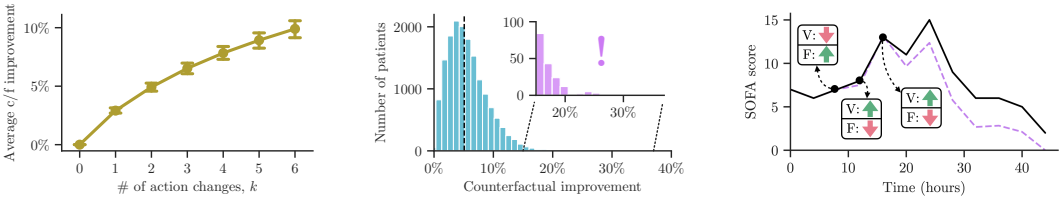
Figure 4.7: **Computational efficiency of our method.** Panels (a-c) show the effective branching factor (pink-left axis) and the runtime of the A^* algorithm (green-right axis) against the Lipschitz constant L_h, the number of Monte Carlo samples M and the number of action changes k, respectively. In Panel (a), we set $M = 2000$ and $k = 3$. In Panel (b), we set $L_h = 1.0$ and $k = 3$. In Panel (c), we set $L_h = 1.0$ and $M = 2000$. In all panels, we set $L_\phi = 0.1$ and error bars indicate 95% confidence intervals over 200 executions of the A^* algorithm for 200 patients with horizon $T = 12$.

to the negation of its SOFA value. Here, it is easy to see that this reward function is just a projection of $\boldsymbol{s}$, therefore, it is Lipschitz continuous with constant $C_a = 1$ for all $a \in \mathcal{A}$. Following related work [233, 234, 236], we consider an action space $\mathcal{A}$ that consists of 25 actions, which correspond to 5×5 levels of administered vasopressors and intravenous fluids. Refer to Appendix B.4.1 for additional details on the features and actions.

To model the transition dynamics of the time-varying features, we consider an SCM $\mathcal{C}$ whose transition mechanism takes a location-scale form $g_S(\boldsymbol{S}_t, A_t, \boldsymbol{U}_t) = h(\boldsymbol{S}_t, A_t) + \phi(\boldsymbol{S}_t, A_t) \odot \boldsymbol{U}_t$, where $h, \phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^{\tilde{d}}$, and $\odot$ denotes the element-wise multiplication [214, 216]. Notably, this model is element-wise bijective and hence it is counterfactually identifiable, as shown in Section 4.2.1. Moreover, we use neural networks to model the location and scale functions h and ϕ and enforce their Lipschitz constants to be L_h and L_ϕ, respectively. This results in a Lipschitz continuous SCM $\mathcal{C}$ with $K_{a,\boldsymbol{u}} = L_h + L_\phi \max_i |u_i|$. Further, we assume that the noise variable $\boldsymbol{U}_t$ follows a multivariate Gaussian distribution with zero mean and allow its covariance matrix to be a (trainable) parameter.

We jointly train the weights of the networks h and ϕ and the covariance matrix of the noise prior on the observed patient transitions using stochastic gradient descent with the negative log-likelihood of each transition as a loss. In our experiments, if not specified otherwise, we use an SCM with Lipschitz constants $L_h = 1.0$, $L_\phi = 0.1$ that achieves a log-likelihood only 6% lower to that of the best model trained without any Lipschitz constraint. Refer to Appendix B.4.2 for additional details on the network architectures, the training procedure and the way we enforce Lipschitz continuity.

Results. We start by evaluating the computational efficiency of our method against (i) the Lipschitz constant of the location network L_h, (ii) the number of Monte Carlo samples M used to generate the anchor set $\mathcal{S}_\dagger$, and (iii) the number of actions k that can differ from the observed ones. We measure efficiency using running time and the effective branching factor (EBF) [230]. The EBF is defined as a real number $b \geq 1$ such that the number of nodes expanded by A^* is equal to $1 + b + b^2 + \cdots + b^T$, where T is the horizon, and values close to 1 indicate that the heuristic function is the most efficient in guiding the search. Fig. 4.7 summarizes the results, which show that our



(a) Average counterfactual improvement vs. k

(b) Distribution of counterfactual improvement

(c) Observed vs. counterfactual episode

Figure 4.8: **Retrospective analysis of patients' episodes.** Panel (a) shows the average counterfactual improvement as a function of k for a set of 200 patients with horizon $T = 12$, where error bars indicate 95% confidence intervals. Panel (b) shows the distribution of counterfactual improvement across all patients for $k = 3$, where the dashed vertical line indicates the median. Panel (c) shows the observed (solid) and counterfactual (dashed) SOFA score across time for a patient who presents a 19.9% counterfactual improvement when $k = 3$. Upward (downward) arrows indicate action changes that suggest a higher (lower) dosage of vasopressors (V) and fluids (F). In all panels, we set $M = 2000$.

method maintains overall a fairly low running time that decreases with the number of Monte Carlo samples M used for the generation of the anchor set and increases with the Lipschitz constant L_h and the number of action changes k. That may not come as a surprise since, as L_h increases, the heuristic function becomes more loose, and as k increases, the size of the search space increases exponentially. To put things in perspective, for a problem instance with $L_h = 1.0$, $k = 3$ and horizon $T = 12$, the A^* search led by our heuristic function is effectively equivalent to an exhaustive search over a full tree with $2.1^{12} \approx 7{,}355$ leaves while the corresponding search space of our problem consists of more than 3 million action sequences—more than 3 million paths to reach from the root node to the goal node.

Next, we investigate to what extent the counterfactual action sequences generated by our method would have led the patients in our dataset to better outcomes. For each patient, we measure their counterfactual improvement—the relative decrease in cumulative SOFA score between the counterfactual and the observed episode. Figs. 4.8a and 4.8b summarize the results, which show that: (i) the average counterfactual improvement shows a diminishing increase as k increases; (ii) the median counterfactual improvement is only 5%, indicating that, the treatment choices made by the clinicians for most of the patients were close to optimal, even with the benefit of hindsight; and (iii) there are 176 patients for whom our method suggests that a different sequence of actions would have led to an outcome that is at least 15% better. That said, we view patients at the tail of the distribution as "interesting cases" that should be deferred to domain experts for closer inspection, and we present one such example in Fig. 4.8c. In this example, our method suggests that, had the patient received an early higher dosage of intravenous fluids while some of the later administered fluids where replaced by vasopressors, their SOFA score would have been lower across time. Although we present this case as purely anecdotal, the counterfactual episode is plausible, since there are indications of decreased mortality when intravenous fluids are administered at the early stages of a septic shock [240].

4.3 Chapter conclusions

In this chapter, we have explored the problem of counterfactually analyzing observed episodes of sequential decision making processes. To this end, we have built upon the frameworks of Markov decision processes and structural causal models, and we have introduced efficient algorithms to find counterfactually optimal action sequences in environments described by discrete and continuous state spaces. Through experiments with synthetic and real datasets, we have demonstrated that these methods offer valuable insights for decision making processes under uncertainty and can serve as a useful tool for domain experts in identifying key time steps and actions within an episode for further (manual) inspection.

Our work opens several avenues for future work. First, our modeling approach assumes no unobserved confounding in the decision making processes and that the Markov property holds. Developing methods to find approximately optimal counterfactual action sequences in SCMs representing partially observable MDPs [125] and general SCMs with unobserved confounding [129] would be a direction worth exploring.

Additionally, while we have learned SCMs from data for our experiments, SCMs are not (counterfactually) identifiable in general [197, 229]. A potential solution to this problem would be to design SCMs that incorporate knowledge from observational interventional data, along with domain knowledge from human experts, in the form of predictions about counterfactual outcomes from alternative action sequences for observed episodes. This approach would align (learned) SCMs with domain experts' intuition, thereby enhancing their reliability for making counterfactual predictions. Developing the modeling and methodological toolkit to enable this approach would be an exciting direction for future work.

Finally, an important next step is to evaluate our proposed methods through interventional experiments with human experts. Conducting empirical studies to assess the utility of counterfactually optimal action sequences as a learning signal in domains such as cognitive behavioral therapy and critical care would provide valuable insights into their effectiveness. Such studies could also explore how learning from counterfactual predictions impacts broader decision making pipelines, shedding light on the potential for collaboration between algorithmic tools and domain expertise.

Chapter 5

Understanding responsibility judgments in human-AI teams

AI systems are increasingly being used to help humans make better decisions in a variety of application areas such as healthcare, finance, and transportation. In healthcare, AI recommendations influence physician treatment decisions [241]; in finance, AI algorithms provide market predictions that inform critical business decisions [5]; in transportation, AI systems have an increasing effect on driver behavior through their route recommendations and semi-autonomous driving capabilities [3]. However, as this mixture of human and machine decisions becomes more common, it also becomes unclear who is responsible for the outcomes of those decisions. If a driver decides to handle control of their car to an AI system before a challenging intersection and the car is involved in an accident, who is responsible?

Questions about responsibility are ubiquitous in our daily lives, and humans make intuitive judgments about responsibility even in complex situations like the one described above. Cognitive scientists have developed and tested different theories about the cognitive process that underpins responsibility judgments [72, 73, 242, 243]. However, the increasing development of AI systems that assist and collaborate with humans, rather than replacing them [57, 60–62, 75–78], calls for more empirical and theoretical research to shed light on the way humans make responsibility judgments in situations involving human-AI teams [79]. Recent work in that area has identified several factors that influence responsibility judgments [145, 146, 244]. However, these works have not attempted to characterize the underlying cognitive process that supports such judgments. In this chapter, we take a step towards filling this gap by introducing a computational model to predict and understand responsibility judgments for human-AI teams in environments where the two agents collaborate, act sequentially, and influence each other's decisions.

Existing theories on the cognitive process of responsibility attribution have established strong ties to causality [71] and counterfactual reasoning [70, 192, 245]. Humans tend to consider an object, event, action, or agent as (causally) responsible for an outcome if they can mentally simulate an alternative reality where that outcome would have been different if the candidate cause had not existed or occurred in the first place [72, 74, 135–143]. In this chapter, we build upon recent work on the counterfactual simulation model (CSM) [193, 246], a computational model that accurately predicts the extent to which people perceive an object (*e.g.*, a moving billiard ball) as a cause of an observed outcome (*e.g.*, potting another ball).

Specifically, using a physics engine to approximate people's intuitive understanding of physics [247, 248], the model performs (stochastic) simulations of counterfactual situations where the candidate cause (*e.g.*, the moving billiard ball) is removed from the scene or slightly perturbed. Then, it predicts participants' causal judgments based on the estimated probability that the outcome would have been different had the respective intervention on the candidate cause taken place.

More recently, Wu et al. [140, 249] have explored extensions of the CSM in social settings using Markov decision processes (MDPs) [119] as generative models of agent behavior. Reminiscent of the results in the physical domain, they have shown that the CSM predicts people's judgments about the extent that a decision of an agent caused an outcome based on counterfactual simulations where that agent has made a different decision [249]. However, in the context of responsibility attribution, the shift of focus from physical objects to agents introduces additional complexity, since an agent's actions are conditioned on their epistemic state (*i.e.*, the knowledge and information they have) [137, 142, 250, 251]. To explore this further, Wu et al. [140] have experimented with a gridworld environment where an agent is trying to achieve an outcome in the presence of a second (potentially adversarial) agent. They have proposed an extension of the CSM that additionally models the first agent's *belief* about the second agent's intention and explains responsibility judgments by combining counterfactual simulations with intention inferences [252].

Here, we further extend the CSM by developing and experimenting with a stylized but rich semi-autonomous driving environment, where a (simulated) human and an AI agent collaborate towards a common objective. A distinctive feature of the setting we focus on is that the two agents share the same goal but have partial and differing knowledge about elements of the physical environment they operate in. As a result, they hold different beliefs about the state of the world, which they update either via direct observations or via inferences from each other's actions [253]. Moreover, the two agents take a series of interdependent actions, and their relationship is asymmetric, with the human having (some) control over the actions of the AI which, in turn, plays an assistive role. We start by formalizing this environment using decentralized partially observable Markov decision processes. Based on this formulation, we make the following contributions:

1. We propose a model of responsibility for the human and the AI that relies on counterfactual simulations to estimate how unexpected an agent's action was and what would have happened had each agent acted differently.

2. We conduct an (online) human-subject study to assess how well our proposed model predicts participants' responsibility judgments concerning the human and AI agents in various simulation scenarios from our driving environment.

Our analysis indicates that participants' responsibility judgments about the human are influenced by counterfactuals and are well-captured by our model. On the other hand, a simpler model, based solely on the actual contribution to the outcome, effectively captures responsibility judgments about the AI. The code for the simulation environment, the interface of the online study, and all data collected during the study described in Chapter 5 are available at https://github.com/cicl-stanford/responsibility_sequential.

5.1 Computational model

We develop a 2D gridworld environment that simulates and illustrates stylized cases of commute. Below, we start by providing a high-level description of our environment. Then, we formalize its main elements, and we introduce a generative model of agent behavior. Building upon that, we propose a model to predict responsibility judgments about the human and the AI agent in individual commutes.

5.1.1 Environment description

Consider the illustration in Fig. 5.1: The two agents (human & AI) are in a car, which is initially placed at the bottom left corner of an 8×8 grid consisted of black and white (road) tiles.[1] The grid is known to both agents a priori and they both share a common goal – to reach the human's workplace at the top right corner within a given time limit. The simulation proceeds in time steps and, at each time step, the car is controlled either by the AI or the human. The agent who is in control can move the car horizontally or vertically by one tile per time step. Moving to a tile is possible only if it is white (*i.e.*, a road) and it is not blocked by a road closure or an accident. The grid may also contain traffic spots that are either congested or not congested for the entire commute, with congested ones causing the car to remain idle for 10 time steps.

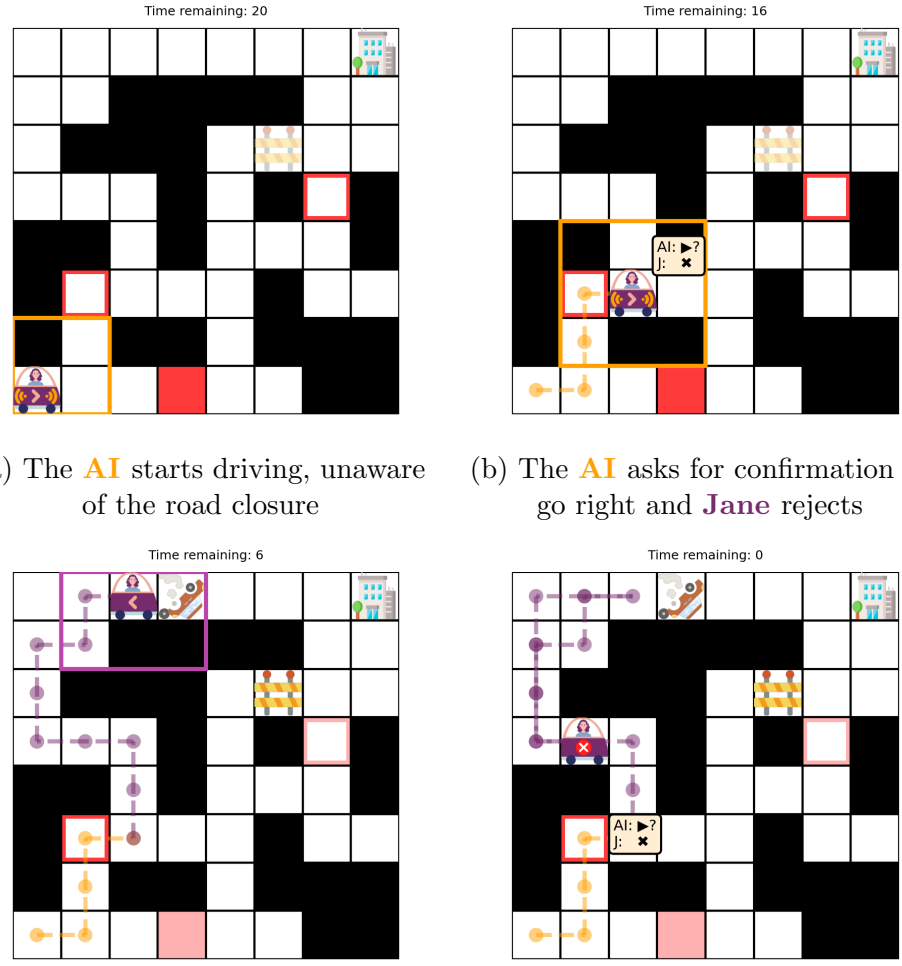
Each agent has only partial knowledge of potential obstacles in the environment. The human knows about road closures and the locations of the traffic spots but not about their congestion status. The AI knows everything about traffic spots but it is unaware of road closures. Lastly, accidents may appear randomly on any tile, and they are unknown to both of them. Each agent discovers a previously unknown obstacle only once it enters their field of view surrounding the car.

The two agents collaborate with each other by switching control of the car. One of them starts driving and, at a randomly chosen time step, the AI asks the human whether they want to switch control for the remainder of the commute. If the AI is driving, it requests confirmation to continue; if the human is driving, the AI asks whether it should take control of the car. The human decides based on the information they have about the environment at the time, and we refer to this decision as the *switching decision*. The agent who is in control after that point drives until they reach the workplace (success) or until time runs out (failure).

5.1.2 Formal framework

Our environment can be described using the framework of decentralized partially observable MDPs [139, 153, 154]. Therein, an episode unfolds over T time steps (here, the time limit to reach the workplace) and includes more than one agent (here, the human and the AI) who act independently. At each time step t, the process is characterized by a state $\boldsymbol{s}_t \in \mathcal{S}$ and, in our case, contains information about the world such as the location of the car and the identity of the current driver. The two agents take actions $a_{H,t} \in \mathcal{A}_H$, $a_{AI,t} \in \mathcal{A}_{AI}$, that correspond to doing nothing (NULL), moving on the grid (*e.g.*, LEFT), offering or accepting/rejecting

[1] We would like to give credit to creators Freepik, Creartive, Smashicons, surang and juicy_fish from flaticon.com whose icons we have used to design our experiment.



(a) The **AI** starts driving, unaware of the road closure

(b) The **AI** asks for confirmation to go right and **Jane** rejects

(c) **Jane** takes control of the car but encounters an accident

(d) Time runs out and they **fail** to reach the workplace

Figure 5.1: **Illustration of a commute in our semi-autonomous driving environment.** The human agent (**Jane**) and the **AI** are both in the same car and their goal is to reach the workplace within the time limit shown above the grid. The sign indicates that the AI is in control. The grid contains three traffic spots, one congested (■) and two non congested (□), whose status is initially known only to the AI. It also contains a road closure () which is known to the human but unknown to the AI. Obstacles that are unknown to the agent in control but known to the other agent appear faded. The arrow signs marked on the car (*e.g.*,) indicate the direction that the driver in control is planning to follow. The 3×3 rectangle around the car represents the agents' field of view via which they discover obstacles that are previously unknown to them. Here, the accident () present at the top row of the grid becomes visible only after the car goes next to it and it enters the agent's field of view.

to switch control and combinations thereof. For example, whenever the AI is in control, it can choose to move LEFT & ASK (for confirmation). The human either approves (¬SWITCH) or takes over (SWITCH) and drives themselves. A function $f_S : \mathcal{S} \times \mathcal{A}_H \times \mathcal{A}_{AI} \to \mathcal{S}$ controls the (deterministic) transitions between states and, at each time step, the agents receive a numerical reward – a positive value if the car has reached the workplace and -1 otherwise. Their goal is to maximize their total reward. Moreover, each agent is characterized by a belief P_{agent} about the state of the world and takes actions a sampled from a (stochastic) policy $\pi_{agent}(a \mid P_{agent})$. We dive deeper into agents' beliefs and policies next.

Beliefs & observations. Here, we focus on the agents' beliefs and their (partial) observability model, which form the basis for our generative model of agent behavior and the responsibility model we present next. The two agents start with their own prior beliefs, formalized as two distributions P_H, P_{AI} over all states in $\mathcal{S}$, where the uncertainty originates from their partial knowledge about obstacles (*i.e.*, traffic spots, road closures, accidents) that may be present on the grid.

Since the human is aware of road closures, their prior belief has zero probability on states $\boldsymbol{s}$ whose road closures do not match with the true state $\boldsymbol{s}_0$. Moreover, since accidents are unexpected, we set the prior probability of any state that contains an accident to a negligible amount close to zero.[2] To model the human's ignorance about the congestion status of K usual traffic spots in the grid, we set their prior uniformly over states corresponding to the 2^K different combinations of congestion status. The AI's prior is defined in a similar way, ensuring that the AI knows the true congestion status of traffic spots but ignores potential road closures and accidents.

At each time step, the two agents receive an observation $\boldsymbol{o}_t = \text{FOV}(\boldsymbol{s}_t)$ that includes all the obstacles within their field of view. Based on this observation, both agents update their beliefs about the state of the world by eliminating any state that would contradict their field of view, that is,

$$P_{agent}(\boldsymbol{s} \mid \boldsymbol{o}_t) \propto \mathbb{1}\left[\boldsymbol{o}_t = \text{FOV}(\boldsymbol{s})\right] \cdot P_{agent}(\boldsymbol{s}) \;\; \forall \boldsymbol{s} \in \mathcal{S},$$

where $\mathbb{1}[\cdot]$ denotes the indicator function. Moreover, whenever the AI is in control of the car, the human receives an enhanced observation $\boldsymbol{o}_t = (\text{FOV}(\boldsymbol{s}_t), a_{AI,t})$ that also includes the AI's action. Motivated by prior work that models action understanding as Bayesian inverse planning [253, 254], we assume that they update their belief about the congestion status of the traffic spots based on the direction that the AI intends to move. Let $a_{AI,t} = d$ denote a movement in direction d (*e.g.*, $d =$ LEFT) and π_H be the human's policy. The human performs a Bayesian update on their belief by considering the likelihood that they would have chosen direction d if they had the same belief as the AI. Formally, let $\tilde{P}_{AI}$ be a function that takes as input a state $\boldsymbol{s}$ and returns a belief (*i.e.*, a distribution over states) oblivious to any road closures in $\boldsymbol{s}$ that have not yet entered the agents' field of view. The human's Bayesian update, as described above, takes the form

$$P_H(\boldsymbol{s} \mid a_{AI,t} = d) \propto \pi_H\left(d \mid \tilde{P}_{AI}\left(\boldsymbol{s}\right)\right) \cdot P_H(\boldsymbol{s}) \;\; \forall \boldsymbol{s} \in \mathcal{S}.$$

Generative model of agent behavior. Similar to prior work, we consider the human and the AI to behave as approximate planners [140, 249], who tend to take

[2]This assumption implies that the human is aware that, in principle, an accident can exist in some part of the grid, but the possibility is so unlikely that it can be effectively ignored during planning.

the shortest path to the workplace whenever they are in control of the car. We assume that they choose a direction with a probability inversely proportional to $\text{ETA}(d \mid P_{agent})$, that is, the time they expect they will need to reach the workplace if their next movement is in direction d. To compute $\text{ETA}(d \mid P_{agent})$, we run Dijkstra's algorithm [255] on a graph whose nodes correspond to tiles of the grid and edge weights represent the time required to move from one tile to the other averaged over states following from the agent's belief P_{agent}. Then, an agent's policy is given by the softmax

$$\pi_{agent}(d \mid P_{agent}) \propto e^{-\tau \cdot \text{ETA}(d \mid P_{agent})}. \tag{5.1}$$

Whenever the AI is in control, it selects a movement direction (*e.g.*, LEFT) and, with a probability p_{switch}, it may also ask the human for confirmation (*e.g.*, LEFT & ASK). If the human is in control, the AI decides between asking the human to switch or doing nothing, again with probability p_{switch}.

When the human encounters a prompt by the AI, they have to make a switching decision, that is, to decide whether they or the AI will drive the second half of the commute. We assume they behave rationally and they choose between the two options proportionally to their probability of a successful outcome S. Let $P(S \mid P_H, \text{SWITCH})$, $P(S \mid P_H, \neg\text{SWITCH})$ be the success probability estimates of the human for each option. We assume that the human estimates these via Monte Carlo simulations. For the option that corresponds to them driving the second half, they perform L simulations of their driving behavior using Eq. 5.1 and compute the total success rate. For the option involving the AI, they sample L possible states $\boldsymbol{s} \sim P_H$ and, for each sample, they simulate the AI's driving using Eq. 5.1 and the belief $\tilde{P}_{AI}(\boldsymbol{s})$ introduced earlier. Based on the estimated probabilities of success, the human makes a (stochastic) decision $a_{sw} \in \{\text{SWITCH}, \neg\text{SWITCH}\}$ using the softmax

$$\pi_H(a_{sw} \mid P_H) \propto e^{\theta \cdot P(S \mid P_H, a_{sw})}. \tag{5.2}$$

We consider a switching decision a_{sw} to be *right* when it is the one maximizing the probability of success from the point of view of the human, that is, $a_{sw} = \text{argmax}_{a \in \{\text{SWITCH}, \neg\text{SWITCH}\}} P(S \mid P_H, a)$ and *wrong* otherwise.

5.1.3 Responsibility model

Given a commute instance generated by our environment, we predict responsibility judgments as a function of probabilities estimated by performing counterfactual simulations that use the aforementioned generative model. In our experiment, we focus on failure instances and thus, the counterfactual probabilities we consider here focus on counterfactual successes.

Human responsibility. We predict that participants hold the human responsible for an observed failure relative to the extent that they would have succeeded had they made a different switching decision. Let a_{sw} denote the observed switching decision of the human and P_H be their belief at the moment the AI asked them to switch control. Then, we write the *counterfactual probability of success* as $P(S \mid a_{sw}, do[\neg a_{sw}])$, where $do[\cdot]$ denotes a counterfactual intervention [71]. Due to the multiplicity of counterfactual interventions in sequential decision-making and the varying sensitivity of responsibility to each intervention's expectancy [136, 256], our model also considers the extent to which the alternative switching decision was

expected. We will refer to this quantity as *counterfactual expectancy*, and we assume it is given by $\pi_H(\neg a_{sw} \mid P_H)$ and is proportional to the likelihood of success associated with the alternative decision (see Eq. 5.2). Our responsibility model considers the effects of the two factors both individually and jointly:

$$r_H = \alpha_1 + \alpha_2 \pi_H(\neg a_{sw} \mid P_H) + \alpha_3 P(S \mid a_{sw}, do[\neg a_{sw}]) \\ + \alpha_4 \pi_H(\neg a_{sw} \mid P_H) \cdot P(S \mid a_{sw}, do[\neg a_{sw}]) \quad (5.3)$$

AI responsibility. Our proposed model for the AI predicts that participants hold the AI responsible for an observed failure relative to the extent that the two agents would have succeeded if the AI had not assisted at all, and we write that counterfactual probability as $P(S \mid AI, do[\neg AI])$. Moreover, since the AI plays a more supportive role, we assume the participants' primary responsibility judgment is for the human, and the AI responsibility is complementary to the former. Let $\mathbb{1}[\text{AI}]$ denote the event that the AI drove for at least one tile. Then, our responsibility model takes the form

$$r_{AI} = \beta_1 + \beta_2 \mathbb{1}[\text{AI}] P(S \mid AI, do[\neg AI]) + \beta_3 (r_{max} - r_H). \quad (5.4)$$

5.2 Human-subject study

Our experiment asks participants to assign responsibility in a human-AI collaboration task (see Figure 5.1). We compare participants' responsibility judgments to the predictions of our responsibility model as well as a set of alternative models.

5.2.1 Methods

Participants. The experiment was preregistered[3] and conducted online via Prolific.[4] We recruited 50 participants (*age*: M = 37, SD = 12; *gender*: 31 female, 18 male, and 1 undisclosed; *race*: 5 Asian, 2 African American, 4 Multiracial, 38 White, and 1 undisclosed) who received $12/hour.

Procedure. Participants were introduced to the semi-autonomous driving environment and the behavior of the two agents within it. They were asked 6 comprehension questions that they had to answer correctly before proceeding to the main experiment. The experiment consisted of 16 trials where the agents failed to reach the target destination on time.

On each trial, participants first watched an interactive step-by-step illustration of the respective commute, and then, they were asked to provide responsibility judgments while watching a video replay of the commute. The two questions ("to what extent is the [human / AI] responsible for not reaching on time?") were presented separately, and participants provided their responses with two continuous sliders ranging from 0 ("Not at all") to 100 ("Very much"). The average completion time of the experiment was 21 minutes (SD = 10).

Design. The 16 trials of our experiment consist of 8 *twin trials*: pairs of trials where the observed commutes are exactly the same, but a small difference between

[3]The preregistration can be found at https://osf.io/5ajzd.

[4]https://www.prolific.com

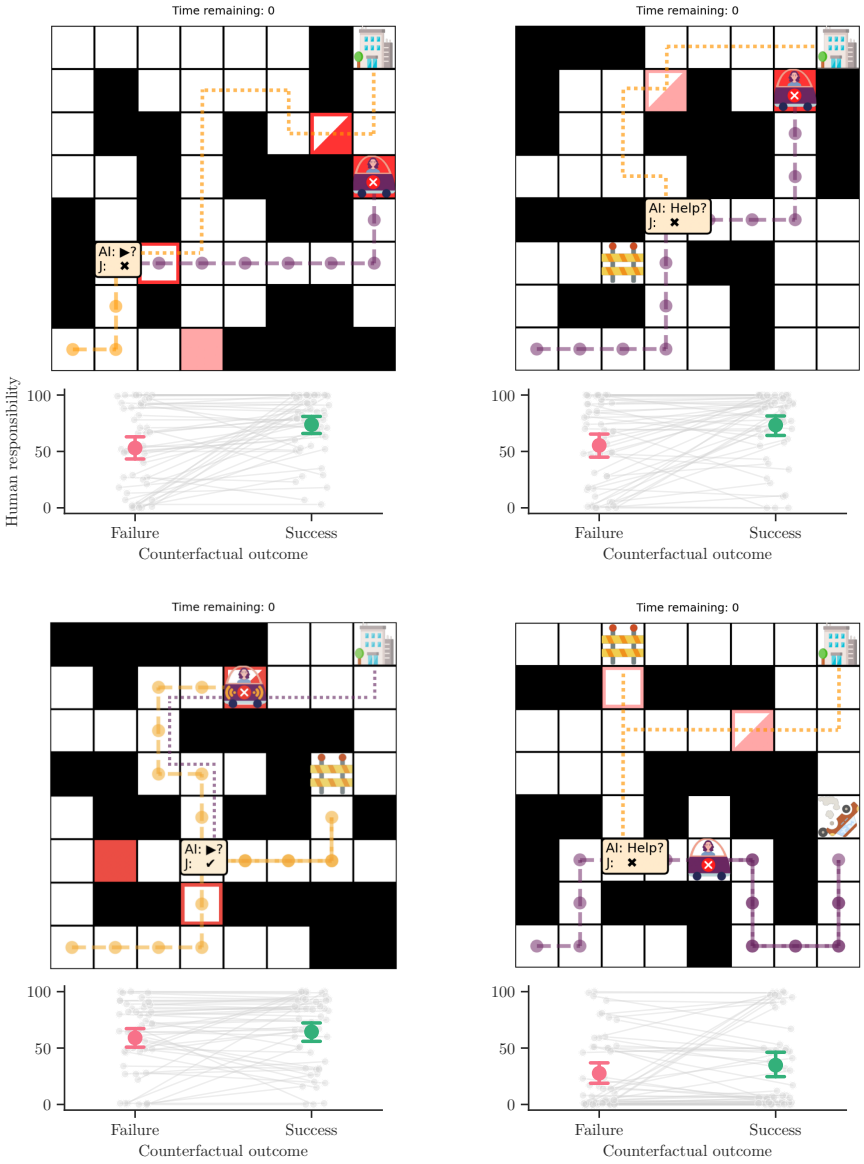


Figure 5.2: **Examples of twin trials and human responsibility judgments.** Each illustration shows a joint summary of two trials whose observed paths, outcomes, and decisions made by the agents are exactly the same. The grids of the two trials differ only in the congestion status of traffic spots illustrated as half colored (◪). In the trial where the traffic spot is not congested, had the human made a different switching decision, the agent who would have driven the second half would have reached the workplace on time following the dashed line. In the trial where the traffic spot is congested, the counterfactual outcome would have been a failure, same as the observed outcome. The figure below each illustration shows participants' judgments about the human's responsibility in the two twin trials. Colored points show means, and error bars show bootstrapped 95% confidence intervals. Each pair of gray points connected with a line shows the judgments of a single participant across the two twin trials.

the two grids alters the counterfactual outcome that would have occurred had the human made a different switching decision (see Fig. 5.2 for examples). To ensure participants do not recognize twin trials, we mirrored the twin gridworlds on the diagonal. The 8 twin trials manipulate 3 main factors: (i) whether the AI or the human is the initial driver, (ii) whether they switch control, and (iii) whether the decision of the human (not) to switch control was *right* or *wrong* at the moment that it was made. We will refer to that last factor as the human's *decision quality*, and we consider a decision to be right if the human believes that it leads to a higher probability of success (see Eq. 5.2). Across all trials, the path that each agent follows was sampled from our generative model given by Eq. 5.1. To manipulate factors (ii) and (iii), we generated switching decisions manually.

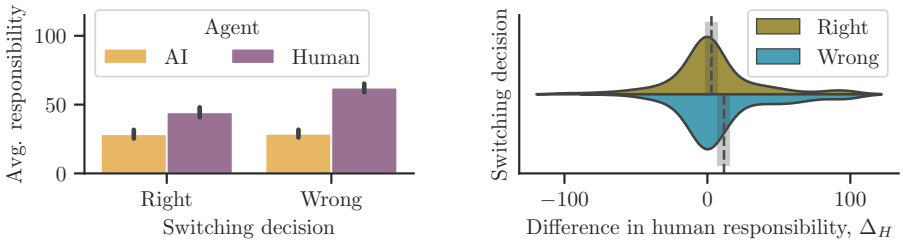
5.2.2 Results and discussion

Do counterfactual outcomes influence human responsibility judgments? We investigate to what extent the way participants assign responsibility to the human differs depending on whether they would have reached the workplace on time had they made a different switching decision. To this end, we focus on pairs of twin trials and perform the following analysis. Let $r_H(p, tw[S])$ and $r_H(p, tw[F])$ denote the responsibility that a participant p assigns to the human in two twin trials with a counterfactual success (S) and failure (F), respectively. We denote as $\Delta_H(p, tw) = r_H(p, tw[S]) - r_H(p, tw[F])$ their difference. To quantify the effect of counterfactual outcomes on responsibility judgments, we fit a Bayesian linear mixed effects model with a fixed global intercept and random coefficients for each participant and pair of trials (*i.e.*, $\Delta_H \sim 1 + (1 \mid p) + (1 \mid tw)$).[5] We observe that the global intercept's posterior mean is positive and equal to 6.48 (95% CI: $[-0.75, 13.78]$), which indicates that counterfactuals have a moderate effect on participants' judgments. To better understand this effect consider the examples in Fig. 5.2. Many (but not all) participants hold the human more responsible for failing to reach on time whenever a different switching decision would have made a difference in the outcome. However, participants' judgments vary considerably, with some of them assigning equal or slightly less responsibility to the human.

Does the human's decision quality make a difference to responsibility judgments? We first look at the average responsibility assigned to the human and the AI across trials where the human's switching decision is right and wrong, respectively. Fig. 5.3a shows that the AI's average responsibility remains the same independently of the human's decision quality, while the human's responsibility increases when their decision was wrong. Moreover, we observe that, across all trials, participants hold the human more responsible than the AI for not reaching the workplace on time.

Additionally, we explore whether the effect of counterfactual outcomes on human responsibility judgments Δ_H varies depending on the quality of the switching decision. To test this, we use a dummy variable called *decision*, and set its value to 0 if the human's switching decision was right and 1 if it was wrong. We fit a Bayesian linear mixed effects model that includes an additional coefficient measuring the effect of the new variable (*i.e.*, $\Delta_H \sim 1 + decision + (1 + decision \mid p) + (1 \mid tr)$). We observe

[5] We use the R formula notation to express mixed effects models concisely. For further details, refer to [257, 258].



(a) Average responsibility vs. decision quality

(b) Distribution of Δ_H vs. decision quality

Figure 5.3: **Effects of decision quality.** In panel (a), error bars indicate bootstrapped 95% confidence intervals. In panel (b), dashed lines show the means of the two distributions, and shaded areas illustrate 95% confidence intervals.

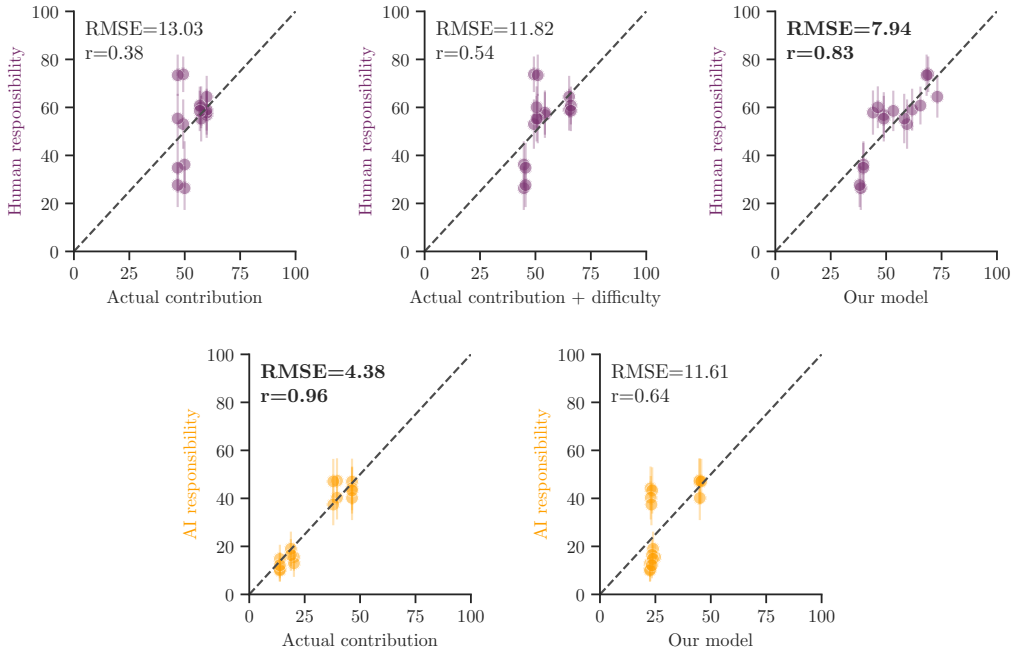


Figure 5.4: **Responsibility judgments and model predictions per trial.** Each point corresponds to one of our 16 trials, with the x-value showing the respective model prediction and the y-value showing the participants' average responsibility judgment. Different panels show results for the **human** and the **AI** under three models: (i) a simple model based on each agent's actual contribution to the outcome, (ii) an extension of the first model that also considers each trial's difficulty, and (iii) our proposed models given by Eqs. 5.3, 5.4. Across all panels, error bars indicate bootstrapped 95% confidence intervals.

that the mean for the posterior of the fixed coefficient of *decision* is positive and equal to 7.27 (95% CI: $[-5.67, 22.94]$). While its positive value indicates that participants may focus more on counterfactual outcomes whenever the observed switching decision was wrong, the effect is weak (the credible interval does not exclude 0). This can also be seen by looking directly at the distributions of Δ_H across pairs of twin trials with right and wrong decisions respectively (see Figure 5.3b). The two

distributions are concentrated around zero but, in the case of wrong decisions, the distribution has a relatively larger mass on the positive side.

How well do the responsibility models capture participants' judgments? We start by estimating the required probabilities $\pi_H(\neg a_{sw} \mid P_H)$, $P(S \mid a_{sw}, do[\neg a_{sw}])$ and $P(S \mid AI, do[\neg AI])$ that are associated with each trial. We fix the hyperparameters τ and θ to the values 2 and 8 respectively and perform 300 Monte Carlo simulations in each grid. Then, we use the estimated probabilities along with participants' responsibility judgments to fit two Bayesian linear mixed effects models that take the form of Eqs. 5.3, 5.4 while also including random intercepts for individual participants. Additionally, we fit two baseline models that use simple heuristics. The first one assigns responsibility proportional to the respective agent's actual contribution to the outcome, measured as the number of time steps that the agent was in control of the car. For the human, we fit a model of the form $r_H \sim 1 + T_H + (1 \mid p)$, where T_H denotes the number of time steps that the human was in control and p denotes an individual participant. Similarly, for the AI, we fit a model of the form $r_{AI} \sim 1 + T_{AI} + (1 \mid p)$, where T_{AI} is the number of time steps that the AI was in control. The second baseline model is an extension of the first that includes the difficulty of the respective grid as an additional term, measured as the total number of obstacles (*i.e.*, road closures, traffic spots, and accidents).

To evaluate the different models, we first compare their average predictions per trial. Fig. 5.4 shows the averaged model predictions per trial against participants' judgments. Our human responsibility model has the lowest RMSE and the highest correlation coefficient compared to the two baselines. In contrast, we observe that participants' judgments about the AI are best captured by the actual contribution model, although they didn't vary much across trials.

Because the models differ in their number of free parameters, we also compare them via approximate leave-one-out cross-validation [259] along with lesioned models that only contain individual components of our human responsibility model (*i.e.*, each additive term in Eq. 5.3). In total, we compare six models: (i) counterfactual expectancy, (ii) counterfactual probability of success, (iii) additive effect of (i, ii), (iv) multiplicative effect of (i, ii), (v) actual contribution, and (vi) our full model given by Eq. 5.3. Table 5.1 summarizes the results, which show that our model performs best overall. However, we observe that, when running cross-validations on individual participant responses, the actual contribution model best captures the most participants, followed by the model based on counterfactual expectancy.

Table 5.1: **Model comparison.** Δelpd shows the difference in expected log pointwise predictive density between each *global* model and the best performing model, with the values in parentheses indicating standard error. Lower values indicate worse performance. N-best shows the number of participants best captured by each model.

Model	Δelpd (se)	N-best
our model	0 (0)	3
additive effect	-2.4 (2.6)	7
counterfactual expectancy	-5.0 (3.6)	11
multiplicative effect	-27.5 (8.0)	5
actual contribution	-46.3 (11.1)	21
counterfactual prob. of success	-54.8 (10.5)	3

5.3 Chapter conclusions

In this chapter, we have studied responsibility judgments in sequential human-AI collaboration, using semi-autonomous driving as an example. We developed an environment that simulates commutes to work via a generative model of human-AI behavior and collaboration. Additionally, we introduced a model of responsibility based on counterfactual simulations sampled from this generative model. Through a human-subject study, we found that responsibility judgments are influenced by counterfactual considerations and unexpected actions. Our proposed model best captures how participants assign responsibility to the human agent, while a simple heuristic model better explains how they assign responsibility to the AI agent.

Our work opens up many interesting avenues for future research. Although our responsibility model performed best overall, there were large individual differences (see Table 5.1). Those may arise from varying conceptions of how responsibility should be determined for human-AI collaborations and from participants' varying levels of motivation to carefully reason through the different scenarios [30].

Moreover, since the actual contribution model best captured the participants' judgments about the AI, it would be interesting to explore the relative importance of actual and counterfactual contribution, as well as how this mixture differs when making judgments about humans and AI agents [141]. Additionally, while our experiments focused on settings where AI and human agents differ primarily in their knowledge, future work could investigate scenarios where the agents also differ in other aspects, such as their abilities.

To fit our responsibility model, we set fixed values for the hyperparameters that control the uncertainty of the model. In future work, it would be useful to conduct additional experiments to fit those hyperparameters by directly asking participants about counterfactual outcomes and the expectancy of the two agents' actions. Furthermore, while we focused on collaborations that feature a single control switch between the human and the AI agent, exploring settings with more frequent interactions between the two agents could offer additional insights into responsibility judgments in dynamic human-AI collaborations.

Chapter 6

General discussion

In this thesis, I have studied scenarios of AI-assisted decision making that involve strategic and counterfactual reasoning, ranging from decision making under transparency and counterfactual actions in sequential decision making to responsibility attribution. While working on these problems, I had the opportunity to engage with a rich body of work across various disciplines and research communities. Here, I highlight promising and broader directions for future work that go beyond the narrower directions discussed in the conclusions of Chapters 3, 4, 5.

Most of the current work on human-AI collaboration in machine learning focuses on assisting humans in relatively simple prediction tasks, such as classification and regression [51–61], with some works also studying more complex decision making settings such as screening [260, 261]. However, there is a variety of real-world decision making tasks that may involve a combination of machine learning predictions with decisions that require optimization over a combinatorial set of alternatives, similar to the problems discussed in Chapters 3 and 4. For example, scheduling operating rooms in a hospital must account for patient risk predictions and surgeon availability. Such tasks may be unethical to automate, but also too complex for a human to perform entirely on their own. Combining methods from operations research with machine learning approaches to effectively support such decision making pipelines is an interesting direction for future work.

In this context, it is also worth exploring the connection between cognitive science and computational complexity [32]. As also mentioned in Chapter 1, the approach of resource-rational analysis [29, 30] suggests that human decisions that fail to maximize a decision maker's utility often arise from the human mind's limited cognitive resources. A compelling direction for future work would be to examine how the scale of a decision problem affects human performance, and to relate these findings to a theoretical categorization of its complexity (*e.g.*, in terms of lower bounds on its runtime). This approach could yield a more comprehensive characterization of the problems that humans can (and cannot) solve effectively and efficiently, while also helping guide the design of AI systems that compensate for these human limitations.

With regards to strategic reasoning, it is worth noting that research on strategic machine learning has largely focused on strategic classification and variants of it. However, there are numerous application domains in which machine learning is used in the presence of (human) strategic behavior [262], opening many avenues for future work to develop alternative strategic machine learning frameworks. One such example is the ranking setting [263], which has received relatively limited attention.

A promising direction in this context is the development of strategic rankings on online platforms, where machine learning models are used to learn representations of online content, yet face strategic behavior from content creators who modify their content to increase their exposure [264, 265] and from users who attempt to steer recommendation algorithms toward specific types of content [266, 267].

As mentioned earlier in this thesis, there is evidence that counterfactual reasoning plays an important role in learning from past experience and using that knowledge to guide future actions [67–69]. This is a form of "learning by thinking" [268] that does not rely on acquiring new knowledge through additional observations of the world. Consequently, the computational steps by which an agent equipped with counterfactual reasoning can improve or accelerate its learning are not yet understood. In this context, a promising direction is to study foundational learning paradigms such as bandit learning [269], enriched with causal assumptions that allow the learning agent to make (approximately correct) counterfactual predictions. Developing learning algorithms for such settings and analyzing their regret compared to the regret of an agent without the capacity for counterfactual reasoning could shed light on the mechanisms by which this reasoning process enhances decision making.

Both strategic and counterfactual reasoning are central characteristics of human cognition. With the rapid emergence of large language models (LLMs) as general-purpose AI assistants, it is worth investigating to what extent they can emulate these processes. Recent evidence suggests that LLMs may exhibit certain reasoning capabilities [270], however, questions remain about how well they can handle strategic interactions or reason about causes and counterfactuals [271, 272]. A promising direction is to develop technical methodologies that explicitly equip such systems with the ability to perform these reasoning processes by design [273], thereby ensuring that their behavior is better aligned with that of the humans they assist.

Finally, an interesting direction for future research at the intersection of psychology and AI safety would be to look more closely into how humans reason about concepts such as responsibility, benefit, harm, and blame [72, 274] in the context of human-AI interaction. Addressing these questions requires both empirical studies and theory formation, as exemplified by the approach presented in Chapter 5. The ultimate goal would be to develop a formal understanding of these notions, which would allow AI systems to penalize or reward decisions based on the extent to which they contribute to their perception as responsible or harmful [274]. In this context, an especially promising direction involves modeling the incentives and objectives of AI systems through frameworks that combine elements of both game theory and causality [275, 276], ensuring that their behavior remains aligned with human values and ethical standards.

Bibliography

[1] Ravi Aggarwal, Viknesh Sounderajah, Guy Martin, Daniel SW Ting, Alan Karthikesalingam, Dominic King, Hutan Ashrafian, and Ara Darzi. Diagnostic accuracy of deep learning in medical imaging: a systematic review and meta-analysis. *NPJ digital medicine*, 4(1):65, 2021.

[2] Yogesh Kumar, Apeksha Koul, Ruchi Singla, and Muhammad Fazal Ijaz. Artificial intelligence in disease diagnosis: a systematic literature review, synthesizing framework and future research agenda. *Journal of ambient intelligence and humanized computing*, 14(7):8459–8486, 2023.

[3] Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of field robotics*, 37(3):362–386, 2020.

[4] Ekim Yurtsever, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. A survey of autonomous driving: Common practices and emerging technologies. *IEEE access*, 8:58443–58469, 2020.

[5] Longbing Cao. Ai in finance: challenges, techniques, and opportunities. *ACM Computing Surveys (CSUR)*, 55(3):1–38, 2022.

[6] Amir E Khandani, Adlar J Kim, and Andrew W Lo. Consumer credit-risk models via machine-learning algorithms. *Journal of Banking & Finance*, 34 (11):2767–2787, 2010.

[7] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017.

[8] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM computing surveys (CSUR)*, 54(6):1–35, 2021.

[9] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8340–8349, 2021.

[10] Thomas Grote and Philipp Berens. On the ethics of algorithmic decision-making in healthcare. *Journal of medical ethics*, 46(3):205–211, 2020.

[11] Manish Raghavan, Solon Barocas, Jon Kleinberg, and Karen Levy. Mitigating bias in algorithmic hiring: Evaluating claims and practices. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 469–481, 2020.

[12] Aleš Završnik. Algorithmic justice: Algorithms and big data in criminal justice settings. *European Journal of criminology*, 18(5):623–642, 2021.

[13] Antoni Roig. Safeguards for the right not to be subject to a decision based solely on automated processing (article 22 gdpr). *European Journal of Law and Technology*, 8(3), 2017.

[14] Pat Croskerry and Geoff Norman. Overconfidence in clinical decision making. *The American journal of medicine*, 121(5):S24–S29, 2008.

[15] Chloë FitzGerald and Samia Hurst. Implicit bias in healthcare professionals: a systematic review. *BMC medical ethics*, 18:1–18, 2017.

[16] Marianne Bertrand and Sendhil Mullainathan. Are emily and greg more employable than lakisha and jamal? a field experiment on labor market discrimination. *American economic review*, 94(4):991–1013, 2004.

[17] Jerry Kang, Mark Bennett, Devon Carbado, Pam Casey, and Justin Levinson. Implicit bias in the courtroom. *UCLa L. rev.*, 59:1124, 2011.

[18] Edgar E Kausel, Satoris S Culbertson, and Hector P Madrid. Overconfidence in personnel selection: When and why unstructured interview information can hurt hiring decisions. *Organizational Behavior and Human Decision Processes*, 137:27–44, 2016.

[19] John Stuart Mill. On the definition and method of political economy. *The philosophy of economics: An anthology*, 2:52–68, 1994.

[20] John von Neumann, Oskar Morgenstern, and Ariel Rubinstein. *Theory of Games and Economic Behavior (60th Anniversary Commemorative Edition)*. Princeton University Press, 1944. ISBN 9780691130613. URL http://www.jstor.org/stable/j.ctt1r2gkx.

[21] Amos Tversky and Daniel Kahneman. Judgment under uncertainty: Heuristics and biases: Biases in judgments reveal some heuristics of thinking under uncertainty. *science*, 185(4157):1124–1131, 1974.

[22] Herbert A Simon. A behavioral model of rational choice. *The quarterly journal of economics*, pages 99–118, 1955.

[23] Sendhil Mullainathan and Richard H Thaler. Behavioral economics, 2000.

[24] Xavier Gabaix, David Laibson, Guillermo Moloche, and Stephen Weinberg. Costly information acquisition: Experimental analysis of a boundedly rational model. *American Economic Review*, 96(4):1043–1068, 2006.

[25] Jennifer S Lerner, Ye Li, Piercarlo Valdesolo, and Karim S Kassam. Emotion and decision making. *Annual review of psychology*, 66(1):799–823, 2015.

[26] David S. Scharfstein and Jeremy C. Stein. Herd behavior and investment. *The American Economic Review*, 80(3):465–479, 1990. ISSN 00028282. URL http://www.jstor.org/stable/2006678.

[27] Daniel Kahneman and Amos Tversky. Prospect theory: An analysis of decision under risk. In *Handbook of the fundamentals of financial decision making: Part I*, pages 99–127. World Scientific, 2013.

[28] Samuel J Gershman, Eric J Horvitz, and Joshua B Tenenbaum. Computational rationality: A converging paradigm for intelligence in brains, minds, and machines. *Science*, 349(6245):273–278, 2015.

[29] Thomas L Griffiths, Falk Lieder, and Noah D Goodman. Rational use of cognitive resources: Levels of analysis between the computational and the algorithmic. *Topics in cognitive science*, 7(2):217–229, 2015.

[30] Falk Lieder and Thomas L Griffiths. Resource-rational analysis: Understanding human cognition as the optimal use of limited computational resources. *Behavioral and brain sciences*, 43:e1, 2020.

[31] Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.

[32] Iris Van Rooij. The tractable cognition thesis. *Cognitive science*, 32(6):939–984, 2008.

[33] Vijay V Vazirani. Approximation algorithms, 2001.

[34] Eric J Topol. High-performance medicine: the convergence of human and artificial intelligence. *Nature medicine*, 25(1):44–56, 2019.

[35] Wei Jiao, Gurnit Atwal, Paz Polak, Rosa Karlic, Edwin Cuppen, Alexandra Danyi, Jeroen de Ridder, Carla van Herpen, Martijn P Lolkema, et al. A deep learning system accurately classifies primary and metastatic cancers using passenger mutation patterns. *Nature communications*, 11(1):728, 2020.

[36] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[37] Bernhard H Korte, Jens Vygen, B Korte, and J Vygen. *Combinatorial optimization*, volume 1. Springer, 2011.

[38] Jianmei Guo, Edward Zulkoski, Rafael Olaechea, Derek Rayside, Krzysztof Czarnecki, Sven Apel, and Joanne M Atlee. Scaling exact multi-objective combinatorial optimization by parallelization. In *Proceedings of the 29th ACM/IEEE international conference on Automated software engineering*, pages 409–420, 2014.

[39] Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *SIAM review*, 60(2):223–311, 2018.

[40] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *nature*, 596(7873):583–589, 2021.

[41] Thomas B Sheridan. Human–robot interaction: status and challenges. *Human factors*, 58(4):525–532, 2016.

[42] Christoph Bartneck, Tony Belpaeme, Friederike Eyssel, Takayuki Kanda, Merel Keijsers, and Selma Šabanović. *Human-robot interaction: An introduction*. Cambridge University Press, 2020.

[43] Burr Settles and Brendan Meeder. A trainable spaced repetition model for language learning. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 1848–1858, 2016.

[44] Behzad Tabibian, Utkarsh Upadhyay, Abir De, Ali Zarezade, Bernhard Schölkopf, and Manuel Gomez-Rodriguez. Enhancing human learning via spaced repetition optimization. *Proceedings of the National Academy of Sciences*, 116(10):3988–3993, 2019.

[45] Jaeho Jeon and Seongyong Lee. Large language models in education: A focus on the complementary relationship between human teachers and chatgpt. *Education and Information Technologies*, 28(12):15873–15892, 2023.

[46] Mina Lee, Percy Liang, and Qian Yang. Coauthor: Designing a human-ai collaborative writing dataset for exploring language model capabilities. In *Proceedings of the 2022 CHI conference on human factors in computing systems*, pages 1–19, 2022.

[47] Jon Kleinberg, Jens Ludwig, Sendhil Mullainathan, and Ziad Obermeyer. Prediction policy problems. *American Economic Review*, 105(5):491–495, 2015.

[48] Sam Corbett-Davies, Emma Pierson, Avi Feller, Sharad Goel, and Aziz Huq. Algorithmic decision making and the cost of fairness. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 797–806, 2017.

[49] Jon Kleinberg, Himabindu Lakkaraju, Jure Leskovec, Jens Ludwig, and Sendhil Mullainathan. Human decisions and machine predictions. *The quarterly journal of economics*, 133(1):237–293, 2018.

[50] Niki Kilbertus, Manuel Gomez-Rodriguez, Bernhard Schölkopf, Krikamol Muandet, and Isabel Valera. Fair decisions despite imperfect predictions. In *AISTATS*, 2019.

[51] Eleni Straitouri, Lequn Wang, Nastaran Okati, and Manuel Gomez Rodriguez. Improving expert predictions with conformal prediction. In *International Conference on Machine Learning*, pages 32633–32653. PMLR, 2023.

[52] Gagan Bansal, Besmira Nushi, Ece Kamar, Walter S Lasecki, Daniel S Weld, and Eric Horvitz. Beyond accuracy: The role of mental models in human-ai

team performance. In *Proceedings of the AAAI conference on human computation and crowdsourcing*, volume 7, pages 2–11, 2019.

[53] Shengjia Zhao, Michael Kim, Roshni Sahoo, Tengyu Ma, and Stefano Ermon. Calibrating predictions to decisions: A novel approach to multi-class calibration. *Advances in Neural Information Processing Systems*, 34:22313–22324, 2021.

[54] Kailas Vodrahalli, Tobias Gerstenberg, and James Y Zou. Uncalibrated models can improve human-ai collaboration. *Advances in Neural Information Processing Systems*, 35:4004–4016, 2022.

[55] Nina Corvelo Benz and Manuel Rodriguez. Human-aligned calibration for ai-assisted decision making. *Advances in Neural Information Processing Systems*, 36, 2024.

[56] Vivian Lai, Chacha Chen, Alison Smith-Renner, Q Vera Liao, and Chenhao Tan. Towards a science of human-ai decision making: An overview of design space in empirical human-subject studies. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, pages 1369–1385, 2023.

[57] Nastaran Okati, Abir De, and Manuel Rodriguez. Differentiable learning under triage. *Advances in Neural Information Processing Systems*, 34:9140–9151, 2021.

[58] Hussein Mozannar and David Sontag. Consistent estimators for learning to defer to an expert. In *International Conference on Machine Learning*, pages 7076–7087. PMLR, 2020.

[59] Bryan Wilder, Eric Horvitz, and Ece Kamar. Learning to complement humans. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, IJCAI'20, 2021. ISBN 9780999241165.

[60] Abir De, Paramita Koley, Niloy Ganguly, and Manuel Gomez-Rodriguez. Regression under human assistance. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2611–2620, 2020.

[61] Abir De, Nastaran Okati, Ali Zarezade, and Manuel Gomez Rodriguez. Classification under human assistance. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 5905–5913, 2021.

[62] Eleni Straitouri, Adish Singla, Vahid Balazadeh Meresht, and Manuel Gomez-Rodriguez. Reinforcement learning under algorithmic triage. *arXiv preprint arXiv:2109.11328*, 2021.

[63] Vahid Balazadeh Meresht, Abir De, Adish Singla, and Manuel Gomez-Rodriguez. Learning to switch between machines and humans. *arXiv preprint arXiv:2002.04258*, 2020.

[64] Tim Roughgarden. Algorithmic game theory. *Communications of the ACM*, 53(7):78–86, 2010.

[65] Tim Roughgarden. Stackelberg scheduling strategies. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 104–113, 2001.

[66] Constantinos Daskalakis, Paul W Goldberg, and Christos H Papadimitriou. The complexity of computing a nash equilibrium. *Communications of the ACM*, 52(2):89–97, 2009.

[67] Aron K Barbey, Frank Krueger, and Jordan Grafman. Structured event complexes in the medial prefrontal cortex support counterfactual representations for future planning. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1521):1291–1300, 2009.

[68] Sara Aronowitz and Tania Lombrozo. Experiential explanation. *Topics in Cognitive Science*, 12(4):1321–1336, 2020.

[69] Kai Epstude and Neal J Roese. The functional theory of counterfactual thinking. *Personality and social psychology review*, 12(2):168–192, 2008.

[70] Daniel Kahneman, Paul Slovic, and Amos Tversky. *Judgment under uncertainty: Heuristics and biases*. Cambridge university press, 1982.

[71] Judea Pearl. *Causality*. Cambridge university press, 2009.

[72] Hana Chockler and Joseph Y Halpern. Responsibility and blame: A structural-model approach. *Journal of Artificial Intelligence Research*, 22:93–115, 2004.

[73] Tobias Gerstenberg and David A Lagnado. Spreading the blame: The allocation of responsibility amongst multiple agents. *Cognition*, 115(1):166–171, 2010.

[74] David A Lagnado, Tobias Gerstenberg, and Ro'i Zultan. Causal responsibility and counterfactuals. *Cognitive science*, 37(6):1036–1073, 2013.

[75] Maithra Raghu, Katy Blumer, Greg Corrado, Jon Kleinberg, Ziad Obermeyer, and Sendhil Mullainathan. The algorithmic automation problem: Prediction, triage, and human effort. *arXiv preprint arXiv:1903.12220*, 2019.

[76] Bryan Wilder, Eric Horvitz, and Ece Kamar. Learning to complement humans. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, IJCAI'20, 2021. ISBN 9780999241165.

[77] Hussein Mozannar, Gagan Bansal, Adam Fourney, and Eric Horvitz. Reading between the lines: Modeling user behavior and costs in ai-assisted programming. *arXiv preprint arXiv:2210.14306*, 2022.

[78] Vahid Balazadeh Meresht, Abir De, Adish Singla, and Manuel Gomez-Rodriguez. Learning to switch among agents in a team. *Transactions on Machine Learning Research*, 2022(7):1–30, 2022.

[79] José J Cañas. Ai and ethics when human beings collaborate with ai agents. *Frontiers in psychology*, 13:836650, 2022.

[80] Ling Huang, Anthony D Joseph, Blaine Nelson, Benjamin IP Rubinstein, and J Doug Tygar. Adversarial machine learning. In *Proceedings of the 4th ACM workshop on Security and artificial intelligence*, pages 43–58, 2011.

[81] Michael Brückner, Christian Kanzow, and Tobias Scheffer. Static prediction games for adversarial learning problems. *The Journal of Machine Learning Research*, 13(1):2617–2654, 2012.

[82] Michael Brückner and Tobias Scheffer. Stackelberg games for adversarial prediction problems. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 547–555, 2011.

[83] Nilesh Dalvi, Pedro Domingos, Sumit Sanghai, and Deepak Verma. Adversarial classification. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 99–108, 2004.

[84] Jinshuo Dong, Aaron Roth, Zachary Schutzman, Bo Waggoner, and Zhiwei Steven Wu. Strategic classification from revealed preferences. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pages 55–70, 2018.

[85] Moritz Hardt, Nimrod Megiddo, Christos Papadimitriou, and Mary Wootters. Strategic classification. In *Proceedings of the 2016 ACM conference on innovations in theoretical computer science*, pages 111–122, 2016.

[86] Alex Frankel and Navin Kartik. Improving information from manipulable data. *Journal of the European Economic Association*, 2019.

[87] Yiling Chen, Yang Liu, and Chara Podimata. Learning strategy-aware linear classifiers. In *Advances in Neural Information Processing Systems*, volume 33, pages 15265–15276, 2020.

[88] Mark Braverman and Sumegha Garg. The Role of Randomness and Noise in Strategic Classification. In *1st Symposium on Foundations of Responsible Computing (FORC 2020)*, volume 156, pages 9:1–9:20, 2020.

[89] Hanrui Zhang and Vincent Conitzer. Incentive-aware pac learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 5797–5804, 2021.

[90] Saba Ahmadi, Hedyeh Beyhaghi, Avrim Blum, and Keziah Naggita. The strategic perceptron. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, pages 6–25, 2021.

[91] Sagi Levanon and Nir Rosenfeld. Strategic classification made practical. In *International Conference on Machine Learning*, pages 6243–6253. PMLR, 2021.

[92] Lily Hu, Nicole Immorlica, and Jennifer Wortman Vaughan. The disparate effects of strategic manipulation. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 259–268, 2019.

[93] Smitha Milli, John Miller, Anca D Dragan, and Moritz Hardt. The social cost of strategic classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 230–239, 2019.

[94] Lydia T Liu, Ashia Wilson, Nika Haghtalab, Adam Tauman Kalai, Christian Borgs, and Jennifer Chayes. The disparate equilibria of algorithmic decision making when individuals invest rationally. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 381–391, 2020.

[95] Ganesh Ghalme, Vineet Nair, Itay Eilat, Inbal Talgam-Cohen, and Nir Rosenfeld. Strategic classification in the dark. In *International Conference on Machine Learning*, pages 3672–3681. PMLR, 2021.

[96] Yahav Bechavod, Chara Podimata, Steven Wu, and Juba Ziani. Information discrepancy in strategic learning. In *International Conference on Machine Learning*, pages 1691–1715. PMLR, 2022.

[97] Juan Perdomo, Tijana Zrnic, Celestine Mendler-Dünner, and Moritz Hardt. Performative prediction. In *International Conference on Machine Learning*, pages 7599–7609, 2020.

[98] John P Miller, Juan C Perdomo, and Tijana Zrnic. Outside the echo chamber: Optimizing the performative risk. In *International Conference on Machine Learning*, pages 7710–7720. PMLR, 2021.

[99] Zachary Izzo, Lexing Ying, and James Zou. How to learn when data reacts to your model: performative gradient descent. In *International Conference on Machine Learning*, pages 4641–4650. PMLR, 2021.

[100] Gavin Brown, Shlomi Hod, and Iden Kalemaj. Performative prediction in a stateful world. In *International Conference on Artificial Intelligence and Statistics*, pages 6045–6061. PMLR, 2022.

[101] Jon Kleinberg and Manish Raghavan. How do classifiers induce agents to invest effort strategically? *ACM Transactions on Economics and Computation (TEAC)*, 8(4):1–23, 2020.

[102] Tal Alon, Magdalen Dobson, Ariel Procaccia, Inbal Talgam-Cohen, and Jamie Tucker-Foltz. Multiagent evaluation mechanisms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 1774–1781, 2020.

[103] Nika Haghtalab, Nicole Immorlica, Brendan Lucier, and Jack Z. Wang. Maximizing welfare with incentive-aware evaluation mechanisms. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 160–166, 7 2020.

[104] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you? explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016.

[105] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, pages 1885–1894. PMLR, 2017.

[106] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, 2017.

[107] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841, 2017.

[108] Berk Ustun, Alexander Spangher, and Yang Liu. Actionable recourse in linear classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 10–19, 2019.

[109] Amir-Hossein Karimi, Gilles Barthe, Borja Belle, and Isabel Valera. Model-agnostic counterfactual explanations for consequential decisions. *arXiv preprint arXiv:1905.11190*, 2019.

[110] Ramaravind K Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 607–617, 2020.

[111] Solon Barocas, Andrew D Selbst, and Manish Raghavan. The hidden assumptions behind counterfactual explanations and principal reasons. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 80–89, 2020.

[112] Amir-Hossein Karimi, Bernhard Schölkopf, and Isabel Valera. Algorithmic recourse: from counterfactual explanations to interventions. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 353–362, 2021.

[113] Amir-Hossein Karimi, Gilles Barthe, Bernhard Schölkopf, and Isabel Valera. A survey of algorithmic recourse: contrastive explanations and consequential recommendations. *ACM Computing Surveys*, 55(5):1–29, 2022.

[114] Sahil Verma, John Dickerson, and Keegan Hines. Counterfactual explanations for machine learning: A review. *arXiv preprint arXiv:2010.10596*, 2020.

[115] Guido W Imbens and Donald B Rubin. *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press, 2015.

[116] Judea Pearl. A probabilistic calculus of actions. In *Uncertainty in artificial intelligence*, pages 454–462. Elsevier, 1994.

[117] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.

[118] Zhihong Deng, Jing Jiang, Guodong Long, and Chengqi Zhang. Causal reinforcement learning: A survey. *arXiv preprint arXiv:2307.01452*, 2023.

[119] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[120] Kun Zhang, Mingming Gong, and Bernhard Scholkopf. Multi-source domain adaptation: a causal view. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, page 3150–3157. AAAI Press, 2015. ISBN 0262511290.

[121] Mingming Gong, Kun Zhang, Tongliang Liu, Dacheng Tao, Clark Glymour, and Bernhard Schölkopf. Domain adaptation with conditional transferable components. In *International conference on machine learning*, pages 2839–2848. PMLR, 2016.

[122] Amy Zhang, Clare Lyle, Shagun Sodhani, Angelos Filos, Marta Kwiatkowska, Joelle Pineau, Yarin Gal, and Doina Precup. Invariant causal prediction for block mdps. In *International Conference on Machine Learning*, pages 11214–11224. PMLR, 2020.

[123] Junzhe Zhang and Elias Bareinboim. Transfer learning in multi-armed bandit: a causal approach. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, pages 1778–1780, 2017.

[124] Ishita Dasgupta, Jane Wang, Silvia Chiappa, Jovana Mitrovic, Pedro Ortega, David Raposo, Edward Hughes, Peter Battaglia, Matthew Botvinick, and Zeb Kurth-Nelson. Causal reasoning from meta-reinforcement learning. *arXiv preprint arXiv:1901.08162*, 2019.

[125] Guy Tennenholtz, Shie Mannor, and Uri Shalit. Off-policy evaluation in partially observable environments. In *AAAI Conference on Artificial Intelligence*, 2019. URL https://api.semanticscholar.org/CorpusID:202538757.

[126] Hongseok Namkoong, Ramtin Keramati, Steve Yadlowsky, and Emma Brunskill. Off-policy policy evaluation for sequential decisions under unobserved confounding. *Advances in Neural Information Processing Systems*, 33:18819–18831, 2020.

[127] Lingxiao Wang, Zhuoran Yang, and Zhaoran Wang. Provably efficient causal reinforcement learning with confounded observational data. *Advances in Neural Information Processing Systems*, 34:21164–21175, 2021.

[128] Junzhe Zhang and Elias Bareinboim. Near-optimal reinforcement learning in dynamic treatment regimes. *Advances in Neural Information Processing Systems*, 32, 2019.

[129] Junzhe Zhang and Elias Bareinboim. Designing optimal dynamic treatment regimes: A causal reinforcement learning approach. In *International Conference on Machine Learning*, pages 11012–11022. PMLR, 2020.

[130] Lars Buesing, Theophane Weber, Yori Zwols, Sebastien Racaniere, Arthur Guez, Jean-Baptiste Lespiau, and Nicolas Heess. Woulda, coulda, shoulda: Counterfactually-guided policy search. *arXiv preprint arXiv:1811.06272*, 2018.

[131] Michael Oberst and David Sontag. Counterfactual off-policy evaluation with gumbel-max structural causal models. In *International Conference on Machine Learning*, pages 4881–4890. PMLR, 2019.

[132] Chaochao Lu, Biwei Huang, Ke Wang, José Miguel Hernández-Lobato, Kun Zhang, and Bernhard Schölkopf. Sample-efficient reinforcement learning via counterfactual-based data augmentation. *arXiv preprint arXiv:2012.09092*, 2020.

[133] Prashan Madumal, Tim Miller, Liz Sonenberg, and Frank Vetere. Explainable reinforcement learning through a causal lens. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[134] Ioana Bica, Daniel Jarrett, Alihan Hüyük, and Mihaela van der Schaar. Learning" what-if" explanations for sequential decision-making. *arXiv preprint arXiv:2007.13531*, 2020.

[135] Ro'i Zultan, Tobias Gerstenberg, and David A Lagnado. Finding fault: causality and counterfactuals in group attributions. *Cognition*, 125(3):429–440, 2012.

[136] Tobias Gerstenberg, Tomer D Ullman, Jonas Nagel, Max Kleiman-Weiner, David A Lagnado, and Joshua B Tenenbaum. Lucky or clever? from expectations to responsibility judgments. *Cognition*, 177:122–141, 2018.

[137] Joseph Halpern and Max Kleiman-Weiner. Towards formal definitions of blameworthiness, intention, and moral responsibility. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[138] Antonia F Langenhoff, Alex Wiegmann, Joseph Y Halpern, Joshua B Tenenbaum, and Tobias Gerstenberg. Predicting responsibility judgments from dispositional inferences and causal attributions. *Cognitive Psychology*, 129: 101412, 2021.

[139] Stelios Triantafyllou, Adish Singla, and Goran Radanovic. Actual causality and responsibility attribution in decentralized partially observable markov decision processes. In *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*, pages 739–752, 2022.

[140] Sarah A Wu, Shruti Sridhar, and Tobias Gerstenberg. A computational model of responsibility judgments from counterfactual simulations and intention inferences. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 45, 2023.

[141] Yang Xiang, Jenna Landy, Fiery A. Cushman, Natalia Vélez, and Samuel J. Gershman. Actual and counterfactual effort contribute to responsibility attributions in collaborative tasks. *Cognition*, 241:105609, 2023. ISSN 0010-0277. doi: https://doi.org/10.1016/j.cognition.2023.105609. URL https://www.sciencedirect.com/science/article/pii/S0010027723002433.

[142] Sander Beckers. Moral responsibility for AI systems. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=jYIknUIgkd.

[143] Sarah A Wu and Tobias Gerstenberg. If not me, then who? responsibility and replacement. *Cognition*, 242:105646, 2024.

[144] Joseph Y Halpern. *Actual causality*. MiT Press, 2016.

[145] Edmond Awad, Sydney Levine, Max Kleiman-Weiner, Sohan Dsouza, Joshua B Tenenbaum, Azim Shariff, Jean-François Bonnefon, and Iyad Rahwan. Drivers are blamed more than their automated cars when both make mistakes. *Nature human behaviour*, 4(2):134–143, 2020.

[146] Gabriel Lima, Nina Grgić-Hlača, and Meeyoung Cha. Human perceptions on moral responsibility of ai: A case study in ai-assisted bail decision-making. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–17, 2021.

[147] Drew Fudenberg. *Game theory*. MIT press, 1991.

[148] Robert Aumann and Adam Brandenburger. Epistemic conditions for nash equilibrium. *Econometrica: Journal of the Econometric Society*, pages 1161–1180, 1995.

[149] John F Nash Jr. Equilibrium points in n-person games. *Proceedings of the national academy of sciences*, 36(1):48–49, 1950.

[150] Heinrich Von Stackelberg. *Market structure and equilibrium*. Springer Science & Business Media, 2010.

[151] Martin L Puterman. Markov decision processes. *Handbooks in operations research and management science*, 2:331–434, 1990.

[152] Dimitri Bertsekas. *Dynamic programming and optimal control: Volume I*, volume 4. Athena scientific, 2012.

[153] Frans A Oliehoek, Christopher Amato, et al. *A concise introduction to decentralized POMDPs*, volume 1. Springer, 2016.

[154] Daniel S Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. The complexity of decentralized control of markov decision processes. *Mathematics of operations research*, 27(4):819–840, 2002.

[155] Sagi Levanon and Nir Rosenfeld. Generalized strategic classification and the case of aligned incentives. In *International Conference on Machine Learning*, pages 12593–12618. PMLR, 2022.

[156] John Miller, Smitha Milli, and Moritz Hardt. Strategic classification is causal modeling in disguise. In *International Conference on Machine Learning*, pages 6917–6926, 2020.

[157] S. Coate and G. Loury. Will affirmative-action policies eliminate negative stereotypes? *The American Economic Review*, 1993.

[158] Roland G Fryer Jr and Glenn C Loury. Valuing diversity. *Journal of political Economy*, 121(4):747–774, 2013.

[159] Lily Hu and Yiling Chen. A short-term intervention for long-term fairness in the labor market. In *Proceedings of the 2018 World Wide Web Conference*, pages 1389–1398, 2018.

[160] Robert Kok. Six years of co2-based tax incentives for new passenger cars in the netherlands: Impacts on purchasing behavior trends and co2 effectiveness. *Transportation Research Part A: Policy and Practice*, 77:137–153, 2015.

[161] Dariush Mozaffarian, Kenneth S Rogoff, and David S Ludwig. The real cost of food: can taxes and subsidies improve public health? *Jama*, 312(9):889–890, 2014.

[162] WHO. *Using price policies to promote healthier diets*. World Health Organization. Regional Office for Europe, 2015.

[163] Frank J Chaloupka, Ayda Yurekli, and Geoffrey T Fong. Tobacco taxes as a tobacco control strategy. *Tobacco control*, 21(2):172–180, 2012.

[164] S Giblin and A McNabola. Modelling the impacts of a carbon emission-differentiated vehicle tax system on co2 emissions intensity from new vehicle purchases in ireland. *Energy Policy*, 37(4):1404–1411, 2009.

[165] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29, 2016.

[166] Lydia T Liu, Sarah Dean, Esther Rolf, Max Simchowitz, and Moritz Hardt. Delayed impact of fair machine learning. In *International Conference on Machine Learning*, pages 3150–3158, 2018.

[167] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer, 2009.

[168] Richard M Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

[169] Isabel Valera, Adish Singla, and Manuel Gomez Rodriguez. Enhancing the accuracy and fairness of human decision making. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[170] Niki Kilbertus, Manuel Gomez Rodriguez, Bernhard Schölkopf, Krikamol Muandet, and Isabel Valera. Fair decisions despite imperfect predictions. In *International Conference on Artificial Intelligence and Statistics*, pages 277–287, 2020.

[171] Yonadav Shavit, Benjamin Edelman, and Brian Axelrod. Causal strategic linear regression. In *International Conference on Machine Learning*, pages 8676–8686, 2020.

[172] Yahav Bechavod, Katrina Ligett, Steven Wu, and Juba Ziani. Gaming helps! learning from strategic interactions in natural dynamics. In *International Conference on Artificial Intelligence and Statistics*, pages 1234–1242, 2021.

[173] Dominique Feillet, Pierre Dejax, and Michel Gendreau. Traveling salesman problems with profits. *Transportation science*, 39(2):188–205, 2005.

[174] I-Cheng Yeh and Che-hui Lien. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications*, 36(2):2473–2480, 2009.

[175] Chirag Gupta, Aleksandr Podkopaev, and Aaditya Ramdas. Distribution-free binary classification: prediction sets, confidence intervals and calibration. *Advances in Neural Information Processing Systems*, 33:3711–3723, 2020.

[176] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 2013.

[177] Paul Voigt and Axel Von dem Bussche. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 2017.

[178] Sandra Wachter, Brent Mittelstadt, and Luciano Floridi. Why a right to explanation of automated decision-making does not exist in the general data protection regulation. *International Data Privacy Law*, 7(2):76–99, 2017.

[179] Supriyo Chakraborty, Richard Tomsett, Ramya Raghavendra, Daniel Harborne, Moustafa Alzantot, Federico Cerutti, Mani Srivastava, Alun Preece, Simon Julier, Raghuveer M Rao, et al. Interpretability of deep learning models: a survey of results. In *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, pages 1–6. IEEE, 2017.

[180] Ramaravind K. Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 2020.

[181] Shira Mitchell, Eric Potash, Solon Barocas, Alexander D'Amour, and Kristian Lum. Prediction-based decisions and fairness: A catalogue of choices, assumptions, and definitions. *arXiv preprint arXiv:1811.07867*, 2018.

[182] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14(1):265–294, 1978.

[183] Niv Buchbinder, Moran Feldman, Joseph Naor, and Roy Schwartz. Submodular maximization with cardinality constraints. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 1433–1452. SIAM, 2014.

[184] Credit score simulator. https://www.creditkarma.com/tools/credit-score-simulator/, 2024.

[185] Gruia Calinescu, Chandra Chekuri, Martin Pal, and Jan Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*, 40(6):1740–1766, 2011.

[186] Gabriele Tolomei, Fabrizio Silvestri, Andrew Haines, and Mounia Lalmas. Interpretable predictions of tree-based ensembles via actionable feature tweaking. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 465–474, 2017.

[187] Roberto Solis-Oba. Approximation algorithms for the k-median problem. In *Efficient Approximation and Online Algorithms*, pages 292–320. Springer, 2006.

[188] Chris Russell. Efficient search for diverse coherent explanations. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 20–28, 2019.

[189] Dorit S Hochbaum and Anu Pathria. Analysis of the greedy approach in problems of maximum k-coverage. *Naval Research Logistics (NRL)*, 45(6): 615–627, 1998.

[190] Lending club dataset. https://www.kaggle.com/wordsforthewise/lending-club/version/3, 2024.

[191] Mateo Rojas-Carulla, Bernhard Schölkopf, Richard Turner, and Jonas Peters. Invariant models for causal transfer learning. *The Journal of Machine Learning Research*, 19(1):1309–1342, 2018.

[192] Neal J Roese. Counterfactual thinking. *Psychological bulletin*, 121(1):133, 1997.

[193] Tobias Gerstenberg, Noah D Goodman, David A Lagnado, and Joshua B Tenenbaum. A counterfactual simulation model of causal judgments for physical events. *Psychological review*, 128(5):936, 2021.

[194] Vicky Arnold, Philip A Collier, Stewart A Leech, and Steve G Sutton. The effect of experience and complexity on order and recency bias in decision making by professional accountants. *Accounting & Finance*, 40(2):109–134, 2000.

[195] Clare R Walsh and Ruth MJ Byrne. Counterfactual thinking: The temporal order effect. *Memory & cognition*, 32(3):369–378, 2004.

[196] Paul Henne, Aleksandra Kulesza, Karla Perez, and Augustana Houcek. Counterfactual thinking and recency effects in causal judgment. *Cognition*, 212: 104708, 2021.

[197] Elias Bareinboim, Juan D. Correa, Duligur Ibeling, and Thomas Icard. *On Pearl's Hierarchy and the Foundations of Causal Inference*, page 507–556. Association for Computing Machinery, New York, NY, USA, 1 edition, 2022. ISBN 9781450395861. URL https://doi.org/10.1145/3501714.3501743.

[198] Chris J. Maddison and Danny Tarlow. Gumbel machinery, Jan 2017. URL https://cmaddis.github.io/gumbel-machinery.

[199] Scott Sussex, Caroline Uhler, and Andreas Krause. Near-optimal multi-perturbation experimental design for causal structure learning. *Advances in Neural Information Processing Systems*, 34:777–788, 2021.

[200] Ioana Bica, Daniel Jarrett, and Mihaela van der Schaar. Invariant causal imitation learning for generalizable policies. *Advances in Neural Information Processing Systems*, 34:3952–3964, 2021.

[201] Chris Cundy, Aditya Grover, and Stefano Ermon. Bcd nets: Scalable variational approaches for bayesian causal discovery. *Advances in Neural Information Processing Systems*, 34:7095–7110, 2021.

[202] Virginia Aglietti, Neil Dhir, Javier González, and Theodoros Damoulas. Dynamic causal bayesian optimization. *Advances in Neural Information Processing Systems*, 34:10549–10560, 2021.

[203] Xinshi Chen, Haoran Sun, Caleb Ellington, Eric Xing, and Le Song. Multi-task learning of order-consistent causal graphs. *Advances in Neural Information Processing Systems*, 34:11083–11095, 2021.

[204] Ilya Shpitser and Eli Sherman. Identification of personalized effects associated with causal pathways. In *Uncertainty in artificial intelligence: proceedings of the... conference. Conference on Uncertainty in Artificial Intelligence*, volume 2018. NIH Public Access, 2018.

[205] Juan Correa, Sanghack Lee, and Elias Bareinboim. Nested counterfactual identification from arbitrary surrogate experiments. *Advances in Neural Information Processing Systems*, 34:6856–6867, 2021.

[206] Junzhe Zhang, Jin Tian, and Elias Bareinboim. Partial counterfactual identification from observational and experimental data. In *International Conference on Machine Learning*, pages 26548–26558. PMLR, 2022.

[207] Kristina Fuhr, Cornelie Schweizer, Christoph Meisner, and Anil Batra. Efficacy of hypnotherapy compared to cognitive-behavioural therapy for mild-to-moderate depression: study protocol of a randomised-controlled rater-blind trial (wiki-d). *BMJ open*, 7(11):e016978, 2017.

[208] Kristina Fuhr, Christoph Meisner, Angela Broch, Barbara Cyrny, Juliane Hinkel, Joana Jaberg, Monika Petrasch, Cornelie Schweizer, Anette Stiegler, Christina Zeep, et al. Efficacy of hypnotherapy compared to cognitive behavioral therapy for mild to moderate depression-results of a randomized controlled rater-blind clinical trial. *Journal of Affective Disorders*, 286:166–173, 2021.

[209] Kurt Kroenke, Robert L Spitzer, and Janet BW Williams. The phq-9: validity of a brief depression severity measure. *Journal of general internal medicine*, 16(9):606–613, 2001.

[210] Malcolm Elliott and Alysia Coventry. Critical care: the eight vital signs of patient monitoring. *British Journal of Nursing*, 21(10):621–625, 2012.

[211] Arash Nasr-Esfahany, Mohammad Alizadeh, and Devavrat Shah. Counterfactual identifiability of bijective causal models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[212] Patrik Hoyer, Dominik Janzing, Joris M Mooij, Jonas Peters, and Bernhard Schölkopf. Nonlinear causal discovery with additive noise models. *Advances in neural information processing systems*, 21, 2008.

[213] Kun Zhang and Aapo Hyvärinen. On the identifiability of the post-nonlinear causal model. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, page 647–655, Arlington, Virginia, USA, 2009. AUAI Press. ISBN 9780974903958.

[214] Alexander Immer, Christoph Schultheiss, Julia E Vogt, Bernhard Schölkopf, Peter Bühlmann, and Alexander Marx. On the identifiability and estimation of causal location-scale noise models. In *International Conference on Machine Learning*, pages 14316–14332. PMLR, 2023.

[215] Nick Pawlowski, Daniel Coelho de Castro, and Ben Glocker. Deep structural causal models for tractable counterfactual inference. *Advances in Neural Information Processing Systems*, 33:857–869, 2020.

[216] Ilyes Khemakhem, Ricardo Monti, Robert Leech, and Aapo Hyvarinen. Causal autoregressive flows. In *International conference on artificial intelligence and statistics*, pages 3520–3528. PMLR, 2021.

[217] Pedro Sanchez and Sotirios A. Tsaftaris. Diffusion causal models for counterfactual estimation. In *First Conference on Causal Learning and Reasoning*, 2022.

[218] Dimitri Bertsekas. Convergence of discretization procedures in dynamic programming. *IEEE Transactions on Automatic Control*, 20(3):415–419, 1975.

[219] Karl Hinderer. Lipschitz continuity of value functions in markovian decision processes. *Mathematical Methods of Operations Research*, 62:3–22, 2005.

[220] Emmanuel Rachelson and Michail G. Lagoudakis. On the locality of action domination in sequential decision making. In *International Symposium on Artificial Intelligence and Mathematics*, 2010. URL https://api.semanticscholar.org/CorpusID:14029770.

[221] Jason Pazis and Ronald Parr. Pac optimal exploration in continuous space markov decision processes. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, page 774–781. AAAI Press, 2013.

[222] Matteo Pirotta, Marcello Restelli, and Luca Bascetta. Policy gradient in lipschitz markov decision processes. *Machine Learning*, 100:255–283, 2015.

[223] Felix Berkenkamp, Matteo Turchetta, Angela Schoellig, and Andreas Krause. Safe model-based reinforcement learning with stability guarantees. *Advances in neural information processing systems*, 30, 2017.

[224] Kavosh Asadi, Dipendra Misra, and Michael Littman. Lipschitz continuity in model-based reinforcement learning. In *International Conference on Machine Learning*, pages 264–273. PMLR, 2018.

[225] Ahmed Touati, Adrien Ali Taiga, and Marc G Bellemare. Zooming for efficient model-free reinforcement learning in metric spaces. *arXiv preprint arXiv:2003.04069*, 2020.

[226] Omer Gottesman, Kavosh Asadi, Cameron Allen, Sam Lobel, George Konidaris, and Michael Littman. Coarse-grained smoothness for rl in metric spaces. *arXiv preprint arXiv:2110.12276*, 2021.

[227] Cem Anil, James Lucas, and Roger Grosse. Sorting out lipschitz function approximation. In *International Conference on Machine Learning*, pages 291–301. PMLR, 2019.

[228] Henry Gouk, Eibe Frank, Bernhard Pfahringer, and Michael J Cree. Regularisation of neural networks by enforcing lipschitz continuity. *Machine Learning*, 110:393–416, 2021.

[229] Arash Nasr-Esfahany and Emre Kiciman. Counterfactual (non-)identifiability of learned structural causal models. *arXiv preprint arXiv:2301.09031*, 2023.

[230] Stuart Russel, Peter Norvig, et al. *Artificial intelligence: a modern approach*, volume 256. Pearson Education Limited London, 2013.

[231] Judea Pearl. *Heuristics: intelligent search strategies for computer problem solving*. Addison-Wesley Longman Publishing Co., Inc., 1984.

[232] Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016.

[233] Aniruddh Raghu, Matthieu Komorowski, Imran Ahmed, Leo Celi, Peter Szolovits, and Marzyeh Ghassemi. Deep reinforcement learning for sepsis treatment. *arXiv preprint arXiv:1711.09602*, 2017.

[234] Matthieu Komorowski, Leo A Celi, Omar Badawi, Anthony C Gordon, and A Aldo Faisal. The artificial intelligence clinician learns optimal treatment strategies for sepsis in intensive care. *Nature medicine*, 24(11):1716–1720, 2018.

[235] Siqi Liu, Kay Choong See, Kee Yuan Ngiam, Leo Anthony Celi, Xingzhi Sun, and Mengling Feng. Reinforcement learning for clinical decision support in critical care: comprehensive review. *Journal of medical Internet research*, 22 (7):e18477, 2020.

[236] Taylor W Killian, Haoran Zhang, Jayakumar Subramanian, Mehdi Fatemi, and Marzyeh Ghassemi. An empirical study of representation learning for reinforcement learning in healthcare. *arXiv preprint arXiv:2011.11235*, 2020.

[237] Mervyn Singer, Clifford S Deutschman, Christopher Warren Seymour, Manu Shankar-Hari, Djillali Annane, Michael Bauer, Rinaldo Bellomo, Gordon R Bernard, Jean-Daniel Chiche, Craig M Coopersmith, et al. The third international consensus definitions for sepsis and septic shock (sepsis-3). *Jama*, 315 (8):801–810, 2016.

[238] Simon Lambden, Pierre Francois Laterre, Mitchell M Levy, and Bruno Francois. The sofa score—development, utility and challenges of accurate assessment in clinical trials. *Critical Care*, 23(1):1–9, 2019.

[239] Flavio Lopes Ferreira, Daliana Peres Bota, Annette Bross, Christian Mélot, and Jean-Louis Vincent. Serial evaluation of the sofa score to predict outcome in critically ill patients. *Jama*, 286(14):1754–1758, 2001.

[240] Jason Waechter, Anand Kumar, Stephen E Lapinsky, John Marshall, Peter Dodek, Yaseen Arabi, Joseph E Parrillo, R Phillip Dellinger, Allan Garland, Cooperative Antimicrobial Therapy of Septic Shock Database Research Group, et al. Interaction between fluids and vasoactive agents on mortality in septic shock: a multicenter, observational study. *Critical care medicine*, 42(10):2158–2168, 2014.

[241] Venkatesh Sivaraman, Leigh A Bukowski, Joel Levin, Jeremy M Kahn, and Adam Perer. Ignore, trust, or negotiate: understanding clinician acceptance of ai-based treatment recommendations in health care. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–18, 2023.

[242] Mark D Alicke. Culpable control and the psychology of blame. *Psychological bulletin*, 126(4):556, 2000.

[243] Kelly G Shaver. *The attribution of blame: Causality, responsibility, and blameworthiness*. Springer Science & Business Media, 2012.

[244] Louis Longin, Bahador Bahrami, and Ophelia Deroy. Intelligence brings responsibility-even smart ai assistants are held responsible. *Iscience*, 26(8), 2023.

[245] Ruth MJ Byrne. Counterfactual thought. *Annual review of psychology*, 67(1): 135–157, 2016.

[246] Liang Zhou, Kevin A Smith, Joshua B Tenenbaum, and Tobias Gerstenberg. Mental jenga: A counterfactual simulation model of causal judgments about physical support. *Journal of Experimental Psychology: General*, 152(8):2237, 2023.

[247] Tobias Gerstenberg and Joshua B. Tenenbaum. 515Intuitive Theories. In *The Oxford Handbook of Causal Reasoning*. Oxford University Press, 06 2017. ISBN 9780199399550. doi: 10.1093/oxfordhb/9780199399550.013.28. URL https://doi.org/10.1093/oxfordhb/9780199399550.013.28.

[248] Tomer D. Ullman, Elizabeth Spelke, Peter Battaglia, and Joshua B. Tenenbaum. Mind games: Game engines as an architecture for intuitive physics. *Trends in Cognitive Sciences*, 21(9):649–665, 2017. doi: 10.1016/j.tics.2017.05.012. URL https://doi.org/10.1016%2Fj.tics.2017.05.012.

[249] Sarah A Wu, Shruti Sridhar, and Tobias Gerstenberg. That was close! a counterfactual simulation model of causal judgments about decisions. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 44, 2022.

[250] Lara Kirfel and David Lagnado. Causal judgments about atypical actions are influenced by agents' epistemic states. *Cognition*, 212:104721, 2021.

[251] Matija Franklin, Hal Ashton, Edmond Awad, and David Lagnado. Causal framework of artificial autonomous agent responsibility. In *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*, pages 276–284, 2022.

[252] Max Kleiman-Weiner, Tobias Gerstenberg, Sydney Levine, and Joshua B Tenenbaum. Inference of intention and permissibility in moral decision making. In *CogSci*, 2015.

[253] Chris L Baker, Rebecca Saxe, and Joshua B Tenenbaum. Action understanding as inverse planning. *Cognition*, 113(3):329–349, 2009.

[254] Chris L Baker, Julian Jara-Ettinger, Rebecca Saxe, and Joshua B Tenenbaum. Rational quantitative attribution of beliefs, desires and percepts in human mentalizing. *Nature Human Behaviour*, 1(4):0064, 2017.

[255] Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[256] John V Petrocelli, Elise J Percy, Steven J Sherman, and Zakary L Tormala. Counterfactual potency. *Journal of personality and social psychology*, 100(1): 30, 2011.

[257] Paul-Christian Bürkner. brms: An r package for bayesian multilevel models using stan. *Journal of statistical software*, 80:1–28, 2017.

[258] Violet A Brown. An introduction to linear mixed-effects modeling in r. *Advances in Methods and Practices in Psychological Science*, 4(1): 2515245920960351, 2021.

[259] Aki Vehtari, Andrew Gelman, and Jonah Gabry. Practical bayesian model evaluation using leave-one-out cross-validation and waic. *Statistics and computing*, 27:1413–1432, 2017.

[260] Lequn Wang, Thorsten Joachims, and Manuel Gomez Rodriguez. Improving screening processes via calibrated subset selection. In *International Conference on Machine Learning*, pages 22702–22726. PMLR, 2022.

[261] Nastaran Okati, Stratis Tsirtsis, and Manuel Gomez Rodriguez. On the within-group fairness of screening classifiers. In *International Conference on Machine Learning*, pages 26495–26516. PMLR, 2023.

[262] Jessie Finocchiaro, Roland Maio, Faidra Monachou, Gourab K Patro, Manish Raghavan, Ana-Andreea Stoica, and Stratis Tsirtsis. Bridging machine learning and mechanism design towards algorithmic fairness. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 489–503, 2021.

[263] Lydia T Liu, Nikhil Garg, and Christian Borgs. Strategic ranking. In *International Conference on Artificial Intelligence and Statistics*, pages 2489–2518. PMLR, 2022.

[264] Jiri Hron, Karl Krauth, Michael I Jordan, Niki Kilbertus, and Sarah Dean. Modeling content creator incentives on algorithm-curated platforms. *arXiv preprint arXiv:2206.13102*, 2022.

[265] Meena Jagadeesan, Nikhil Garg, and Jacob Steinhardt. Supply-side equilibria in recommender systems. *Advances in Neural Information Processing Systems*, 36:14597–14608, 2023.

[266] Andreas Haupt, Dylan Hadfield-Menell, and Chara Podimata. Recommending to strategic users. *arXiv preprint arXiv:2302.06559*, 2023.

[267] Sarah H Cen, Andrew Ilyas, Jennifer Allen, Hannah Li, and Aleksander Madry. Measuring strategization in recommendation: Users adapt their behavior to shape future content. *arXiv preprint arXiv:2405.05596*, 2024.

[268] Tania Lombrozo. Learning by thinking in natural and artificial minds. *Trends in Cognitive Sciences*, 2024.

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

[270] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

[271] Zhijing Jin, Yuen Chen, Felix Leeb, Luigi Gresele, Ojasv Kamal, LYU Zhiheng, Kevin Blin, Fernando Gonzalez Adauto, Max Kleiman-Weiner, Mrinmaya Sachan, et al. Cladder: Assessing causal reasoning in language models. In *Thirty-seventh conference on neural information processing systems*, 2023.

[272] Yadong Zhang, Shaoguang Mao, Tao Ge, Xun Wang, Adrian de Wynter, Yan Xia, Wenshan Wu, Ting Song, Man Lan, and Furu Wei. Llm as a mastermind: A survey of strategic reasoning with large language models. *arXiv preprint arXiv:2404.01230*, 2024.

[273] Ivi Chatzi, Nina Corvelo Benz, Eleni Straitouri, Stratis Tsirtsis, and Manuel Gomez-Rodriguez. Counterfactual token generation in large language models. *arXiv preprint arXiv:2409.17027*, 2024.

[274] Jonathan Richens, Rory Beard, and Daniel H Thompson. Counterfactual harm. In *Advances in Neural Information Processing Systems*, volume 35, pages 36350–36365, 2022.

[275] Tom Everitt, Ryan Carey, Eric D. Langlois, Pedro A. Ortega, and Shane Legg. Agent incentives: A causal perspective. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(13):11487–11495, May 2021. doi: 10.1609/aaai.v35i13.17368. URL https://ojs.aaai.org/index.php/AAAI/article/view/17368.

[276] Lewis Hammond, James Fox, Tom Everitt, Ryan Carey, Alessandro Abate, and Michael Wooldridge. Reasoning about causality in games. *Artificial Intelligence*, 320:103919, 2023.

Appendix A

Omitted proofs

A.1 Proofs for Section 3.1

Theorem 3.1.1. *The problem of finding the optimal decision policy π^* that maximizes utility in a strategic setting is NP-hard.*

Proof. We will reduce any given instance of the SAT problem [168], which is known to be NP-complete, to a particular instance of our problem. In a SAT problem, the goal is finding the value of a set of boolean variables $\{v_1, v_2, \ldots, v_l\}$, and their logical complements $\{\bar{v}_1, \bar{v}_2, \ldots, \bar{v}_l\}$, that satisfy s number of *OR* clauses, which we label as $\{k_1, k_2, \ldots, k_s\}$.

First, we start by representing our problem, as defined in Eq. 3.8, using a directed weighted bipartite graph, whose nodes can be divided into two disjoint sets $\mathcal{U}$ and $\mathcal{V}$. In each of these sets, there are m nodes with labels $\{x_1, \ldots, x_m\}$. We characterize each node x_i in $\mathcal{U}$ with $P(x_i)$ and each node x_j in $\mathcal{V}$ with $\pi(x_j)$ and $u(x_j) = P(Y = 1 \,|\, x_j) - \gamma$. Then, we connect each node x_i in U to each node x_j in V and characterize each edge with a weight $w(x_i, x_j) = b(x_j) - c(x_i, x_j) = \pi(x_j) - c(x_i, x_j)$ and a utility

$$u(x_i, x_j) = \pi(x_j) u(x_j) P(x_i) \mathbb{1}[x_j = \operatorname*{argmax}_{x_k} w(x_i, x_k)],$$

which, for each node x_i in $\mathcal{U}$, is only nonzero for the edge with maximum weight (solving ties at random). Under this representation, the problem reduces to finding the values of $\pi(x_j)$ such that the sum of the utilities of all edges in the graph is maximized.

Next, given an instance of the SAT problem with variables $\{v_1, v_2, \ldots, v_l\}$ and clauses $\{k_1, k_2, \ldots, k_s\}$, we use the above representation to build an instance of our problem. More specifically, consider $\mathcal{U}$ and $\mathcal{V}$ have $m = 7l + s$ nodes each with labels

$$\{y_1, \ldots, y_l, \bar{y}_1, \ldots, \bar{y}_l, a_1, \ldots, a_l, b_1, \ldots, b_l, \\ z_{11}, \ldots, z_{1l}, z_{21}, \ldots, z_{2l}, z_{31}, \ldots, z_{3l}, k_1, k_2, \ldots, k_s\}$$

For the set $\mathcal{U}$, characterize each node u with $P(u)$, where

$$P(z_{1i}) = \frac{3(s+1)}{3l + 3(s+1)l}, \; P(z_{2i}) = P(z_{3i}) = \frac{1}{3l + 3(s+1)l},$$
$$P(k_j) = \frac{1}{3l + 3(s+1)l}, \text{ and } P(y_i) = P(\bar{y}_i) = P(a_i) = P(b_i) = 0,$$

for all $i = 1, \ldots, l$ and $j = 1, \ldots, s$. For the set $\mathcal{V}$, characterize each node v with $\pi(v)$ and $u(v)$, where

$$u(y_i) = u(\bar{y}_1) = \frac{1}{2l + 4(s+1)l}, \; u(a_i) = u(b_i) = \frac{2(s+1)}{2l + 4(s+1)l}, \text{ and}$$
$$u(z_{1i}) = u(z_{2i}) = u(z_{3i}) = u(k_j) = 0,$$

for all $i = 1, \ldots, l$ and $j = 1, \ldots, s$. Then, connect each node u in $\mathcal{U}$ to each node v in $\mathcal{V}$ and set each edge weights to $w(u, v) = \pi(v) - c(u, v)$, where:

(i) $c(z_{1i}, y_j) = c(z_{1i}, \bar{y}_j) = 0$ and $c(z_{2i}, y_j) = c(z_{3i}, y_j) = c(k_q, y_j) = c(z_{2i}, \bar{y}_j) = c(z_{3i}, \bar{y}_j) = c(k_q, \bar{y}_j) = 2$ for each $i, j = 1, \ldots, l$ and $q = 1, \ldots, s$.

(ii) $c(z_{2i}, y_j) = 0$, $c(z_{2i}, a_j) = 1 - \epsilon$ and $c(z_{1i}, y_j) = c(z_{3i}, y_j) = c(k_q, y_j) = c(z_{1i}, a_j) = c(z_{3i}, a_j) = c(k_q, a_j) = 2$ for each $i, j = 1, \ldots, l$ and $q = 1, \ldots, s$.

(iii) $c(z_{3i}, \bar{y}_j) = 0$, $c(z_{3i}, b_j) = 1 - \epsilon$ and $c(z_{1i}, \bar{y}_j) = c(z_{2i}, \bar{y}_j) = c(k_q, \bar{y}_j) = c(z_{1i}, b_j) = c(z_{2i}, b_j) = c(k_q, b_j) = 2$ for each $i, j = 1, \ldots, l$ and $q = 1, \ldots, s$.

(iv) $c(k_i, y_j) = 0$ if the clause k_i contains y_j, $c(k_i, \bar{y}_j) = 0$ if the clause k_i contains $\bar{y}_j$, and $c(k_i, a_j) = c(k_i, b_j) = 2$ for all $i = 1, \ldots, s$ and $j = 1, \ldots, l$.

(v) For all remaining edge weights, set $c(\cdot, \cdot) = \infty$.

Note that, in this particular instance, finding the optimal values of $\pi(v)$ such that the sum of the utilities of all edges in the graph is maximized reduces to first solving l independent problems, one per pair y_j and $\bar{y}_j$, since whenever $c(u, v) = 2$, the edge will never be active, and each optimal π value will be always either zero or one. Moreover, the maximum utility due to the nodes k_z will be always smaller than the utility due to y_j and $\bar{y}_j$ and we can exclude them by the moment. In the following, we fix j and compute the sum of utilities for all possible values of y_j and $\bar{y}_j$:

- For $\pi(y_j) = \pi(\bar{y}_j) = 0$, the maximum sum of utilities is $\frac{4(s+1)}{(3l+3(s+1)l)\cdot(2l+4(s+1)l)}$ whenever $\pi(a_j) = \pi(b_j) = 1$.
- For $\pi(y_j) = \pi(\bar{y}_j) = 1$, the sum of utilities is $\frac{3(s+1)+2}{(3l+3(s+1)l)\cdot(2l+4(s+1)l)}$ for any value of $\pi(a_j)$ and $\pi(b_j)$.
- For $\pi(y_j) = 1 - \pi(\bar{y}_j)$, the maximum sum of utilities is $\frac{5(s+1)}{(3l+3(s+1)l)\cdot(2l+4(s+1)l)}$.

Therefore, the maximum sum of utilities $\frac{5(s+1)}{(3+3(s+1))\cdot(2l+4(s+1)l)}$ occurs whenever $\pi(y_j) = 1 - \pi(\bar{y}_j)$ for all $j = 1, \ldots, l$ and the solution that maximizes the overall utility, including the utility due to the nodes k_z, gives us the solution of the SAT problem. □

Proposition 3.1.1. *Let π^* be an optimal policy that maximizes utility. If the cost $c(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is outcome monotonic then there exists an outcome monotonic policy π such that $u(\pi, \gamma) = u(\pi^*, \gamma)$.*

Proof. This proposition can be easily proven by contradiction. More specifically, assume that all outcome monotonic policies π are suboptimal, *i.e.*, $u(\pi, \gamma) < u(\pi^*, \gamma)$, where π^* is an optimal policy that maximizes utility, and sort the values of the optimal policy in decreasing order, *i.e.*, $1 = \pi^*(\boldsymbol{x}_{l_1}) \geq \pi^*(\boldsymbol{x}_{l_2}) \geq ... \geq \pi^*(\boldsymbol{x}_{l_n})$. Here,

note that there is a state $\boldsymbol{x}_k$ such that $l_k \neq k$, otherwise, the policy π^* would be outcome monotonic. Now, define the index $r = \text{argmin}_k\, l_k \neq k$ and build a policy π' such that $\pi'(\boldsymbol{x}_l) = \pi^*(\boldsymbol{x}_{l_r})$ for all $r - 1 < l < l_r$ and $\pi'(\boldsymbol{x}_l) = \pi^*(\boldsymbol{x}_l)$ otherwise. Then, it is easy to see that the policy π' has greater or equal utility than π^* and it holds that $\pi'(\boldsymbol{x}_l) \geq \pi'(\boldsymbol{x}_t) \Leftrightarrow P(Y = 1 \,|\, \boldsymbol{x}_l) \geq P(Y = 1 \,|\, \boldsymbol{x}_t)$ for all $\boldsymbol{x}_l, \boldsymbol{x}_t$ such that $l, t \leq l_r$. If the policy π' satisfies outcome monotonicity, we are done. Otherwise, we repeat the procedure starting from π' and continue building increasingly better policies until we eventually build one that satisfies outcome monotonicity. By construction, this last policy will achieve equal or greater utility than the policy π^*, leading to a contradiction. □

Theorem 3.1.2. *Let π^* be an optimal policy that maximizes utility. If the cost $c(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is additive and outcome monotonic then there exists an outcome monotonic binary policy π such that $u(\pi, \gamma) = u(\pi^*, \gamma)$.*

Proof. We prove this theorem by contradiction. More specifically, assume that all outcome monotonic binary policies π are suboptimal, *i.e.*, $u(\pi, \gamma) < u(\pi^*, \gamma)$, where π^* is an optimal policy. According to Proposition 3.1.1, under outcome monotonic costs, there is always an optimal outcome monotonic policy. Now, assume there is an optimal outcome monotonic policy π^* such that $\pi^*(\boldsymbol{x}_{i-1}) > \pi^*(\boldsymbol{x}_i) > \pi^*(\boldsymbol{x}_{i-1}) - c(\boldsymbol{x}_i, \boldsymbol{x}_{i-1}) \vee \pi^*(\boldsymbol{x}_i) < \pi^*(\boldsymbol{x}_{i-1}) - c(\boldsymbol{x}_i, \boldsymbol{x}_{i-1})$ for some $i > 1$. Moreover, if there are more than one i, consider the one with the highest outcome $P(Y = 1 \,|\, \boldsymbol{x}_i)$. Then, we analyze each case separately.

If $\pi^*(\boldsymbol{x}_{i-1}) > \pi^*(\boldsymbol{x}_i) > \pi^*(\boldsymbol{x}_{i-1}) - c(\boldsymbol{x}_i, \boldsymbol{x}_{i-1})$, we can show that the policy π' with $\pi'(\boldsymbol{x}_j) = \pi^*(\boldsymbol{x}_j)\ \forall j \neq i$ and $\pi'(\boldsymbol{x}_i) = \pi^*(\boldsymbol{x}_{i-1})$ has greater or equal utility than π^*. More specifically, consider an individual with initial feature values $\boldsymbol{x}_k$. Then, it is easy to see that, if $k < i$, the best-response under π^* and π' will be the same and, if $k \geq i$, the best-response will be either the same or change to $\boldsymbol{x}_i$ under π'. In the latter case, it also holds that $P(Y = 1 \,|\, x_i) > P(Y = 1 \,|\, x_j)$, where x_j is the best-response under π^*, otherwise, we would have a contradiction. Therefore, we can conclude that π' provides higher utility than π^*.

If $\pi^*(\boldsymbol{x}_i) < \pi^*(\boldsymbol{x}_{i-1}) - c(\boldsymbol{x}_i, \boldsymbol{x}_{i-1})$, we can show that the policy π' with $\pi'(\boldsymbol{x}_j) = \pi^*(\boldsymbol{x}_j)\ \forall j \neq i$ and $\pi'(\boldsymbol{x}_i) = \pi^*(\boldsymbol{x}_{i-1}) - c(\boldsymbol{x}_i, \boldsymbol{x}_{i-1})$ has greater or equal utility than π^*. More specifically, consider an individual with initial feature values $\boldsymbol{x}_k$ and denote the individual's best-response under π^* as $\boldsymbol{x}_j$. Then, it is easy to see that the individual's best-response is the same under π^* and π', however, if $\boldsymbol{x}_j = \boldsymbol{x}_i$, the term in the utility corresponding to the individual does increase under π'. Therefore, we can conclude that π' provides higher utility than π^*.

In both cases, if the policy π' is an outcome monotonic binary policy, we are done, otherwise, we repeat the procedure starting from the corresponding π' and continue building increasingly better policies until we eventually build one that is an outcome monotonic binary policy. By construction, this last policy will achieve equal or greater utility than the policy π^*, leading to a contradiction. □

Proposition 3.1.2. *Let π be an outcome monotonic binary policy, $c(\boldsymbol{x}_i, \boldsymbol{x}_j)$ be an additive and outcome monotonic cost, $\boldsymbol{x}_i$ be an individual's initial set of features, and define $j = \max\{k \,|\, k \leq i, \pi(\boldsymbol{x}_k) = 1 \vee \pi(\boldsymbol{x}_k) = \pi(\boldsymbol{x}_{k-1})\}$. If $P(Y = 1 \,|\, \boldsymbol{x}_i) > \gamma$, the individual's best-response is $\boldsymbol{x}_j$ and, if $P(Y = 1 \,|\, \boldsymbol{x}_i) \leq \gamma$, the individual's best-response is $\boldsymbol{x}_j$ if $\pi(\boldsymbol{x}_j) \geq c(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $\boldsymbol{x}_i$ otherwise.*

Proof. Consider an individual with initial features $\boldsymbol{x}_i$ such that $P(Y = 1 \,|\, \boldsymbol{x}_i) > \gamma$. As argued just after Proposition 3.1.1, given an individual with a set of features $\boldsymbol{x}_i$, any outcome monotonic policy always induces a best-response $\boldsymbol{x}_l$ such that $P(Y = 1 \,|\, \boldsymbol{x}_l) \geq P(Y = 1 \,|\, \boldsymbol{x}_i)$, that means, $l < i$. Then, we just need to prove that the best-response $\boldsymbol{x}_l$ cannot satisfy that $P(Y = 1 \,|\, \boldsymbol{x}_l) > P(Y = 1 \,|\, \boldsymbol{x}_j)$ nor satisfy that $P(Y = 1 \,|\, \boldsymbol{x}_j) > P(Y = 1 \,|\, \boldsymbol{x}_l) \geq P(Y = 1 \,|\, \boldsymbol{x}_i)$, where $j = \max\{k \,|\, k \leq i, \pi(\boldsymbol{x}_k) = 1 \vee \pi(\boldsymbol{x}_k) = \pi(\boldsymbol{x}_{k-1})\}$. Without loss of generality, we assume that $j < i$, however, in case $j = i$ the main idea of the proof is the same.

First, assume that $P(Y = 1 \,|\, \boldsymbol{x}_l) > P(Y = 1 \,|\, \boldsymbol{x}_j)$. Then, using the additivity and outcome monotonicity of the cost and the fact that the policy is an outcome monotonic binary policy, it should hold that $\pi(\boldsymbol{x}_j) - c(\boldsymbol{x}_i, \boldsymbol{x}_j) = \pi(\boldsymbol{x}_{j-1}) - c(\boldsymbol{x}_i, \boldsymbol{x}_j) > \pi(\boldsymbol{x}_{j-1}) - c(\boldsymbol{x}_i, \boldsymbol{x}_{j-1}) \geq \pi(\boldsymbol{x}_{j-2}) - c(\boldsymbol{x}_i, \boldsymbol{x}_{j-2}) \geq \cdots \geq \pi(\boldsymbol{x}_l) - c(\boldsymbol{x}_i, \boldsymbol{x}_l)$. This implies that $\boldsymbol{x}_j$ is a strictly better response for the individual than $\boldsymbol{x}_l$, which is a contradiction. Now, assume that $P(Y = 1 \,|\, \boldsymbol{x}_j) > P(Y = 1 \,|\, \boldsymbol{x}_l) \geq P(Y = 1 \,|\, \boldsymbol{x}_i)$. Then, using the additivity of the cost, the definition of $\boldsymbol{x}_j$ and the fact that $\boldsymbol{x}_l$ is the best-response, it should hold that $\pi(\boldsymbol{x}_j) - c(\boldsymbol{x}_i, \boldsymbol{x}_j) < \pi(\boldsymbol{x}_l) - c(\boldsymbol{x}_i, \boldsymbol{x}_l) = \pi(\boldsymbol{x}_j) - c(\boldsymbol{x}_l, \boldsymbol{x}_j) - c(\boldsymbol{x}_i, \boldsymbol{x}_l) = \pi(\boldsymbol{x}_j) - c(\boldsymbol{x}_i, \boldsymbol{x}_j)$, which is clearly a contradiction. Therefore, $\boldsymbol{x}_j$ is a best-response.

Now, consider an individual with initial features $\boldsymbol{x}_i$ such that $P(Y = 1 \,|\, \boldsymbol{x}_i) \leq \gamma$ and $\pi(\boldsymbol{x}_j) \geq c(\boldsymbol{x}_i, \boldsymbol{x}_j)$. The argument for proving that $P(Y = 1 \,|\, \boldsymbol{x}_l) > P(Y = 1 \,|\, \boldsymbol{x}_j)$ is a contradiction remains as is. Assume that $P(Y = 1 \,|\, \boldsymbol{x}_j) > P(Y = 1 \,|\, \boldsymbol{x}_l) \geq P(Y = 1 \,|\, \boldsymbol{x}_i)$. Then $\pi(\boldsymbol{x}_l) = \pi(\boldsymbol{x}_j) - c(\boldsymbol{x}_l, \boldsymbol{x}_j)$ or $\pi(\boldsymbol{x}_l) = 0$, meaning that $\pi(\boldsymbol{x}_j) - c(\boldsymbol{x}_l, \boldsymbol{x}_j) > \pi(\boldsymbol{x}_l)$ since $\pi(\boldsymbol{x}_j) - c(\boldsymbol{x}_l, \boldsymbol{x}_j) > \pi(\boldsymbol{x}_j) - c(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0$. Therefore, it should hold that $\pi(\boldsymbol{x}_j) - c(\boldsymbol{x}_i, \boldsymbol{x}_j) < \pi(\boldsymbol{x}_l) - c(\boldsymbol{x}_i, \boldsymbol{x}_l) \leq \pi(\boldsymbol{x}_j) - c(\boldsymbol{x}_l, \boldsymbol{x}_j) - c(\boldsymbol{x}_i, \boldsymbol{x}_l) = \pi(\boldsymbol{x}_j) - c(\boldsymbol{x}_i, \boldsymbol{x}_j)$, which is clearly a contradiction. As a result, $\boldsymbol{x}_j$ is a best-response.

Now, consider an individual with initial features $\boldsymbol{x}_i$ such that $P(Y = 1 \,|\, \boldsymbol{x}_i) \leq \gamma$ and $\pi(\boldsymbol{x}_j) < c(\boldsymbol{x}_i, \boldsymbol{x}_j)$. The argument for proving that $P(Y = 1 \,|\, \boldsymbol{x}_l) > P(Y = 1 \,|\, \boldsymbol{x}_j)$ is a contradiction remains as is. For all $\boldsymbol{x}_l$ such that $P(Y = 1 \,|\, \boldsymbol{x}_j) \geq P(Y = 1 \,|\, \boldsymbol{x}_l) > P(Y = 1 \,|\, \boldsymbol{x}_i)$ we have $\pi(\boldsymbol{x}_l) = \pi(\boldsymbol{x}_j) - c(\boldsymbol{x}_l, \boldsymbol{x}_j)$ meaning that $\pi(\boldsymbol{x}_l) - c(\boldsymbol{x}_i, \boldsymbol{x}_l) = \pi(\boldsymbol{x}_j) - c(\boldsymbol{x}_i, \boldsymbol{x}_j) < 0$ or $\pi(\boldsymbol{x}_l) = 0$ meaning that $\pi(\boldsymbol{x}_l) - c(\boldsymbol{x}_i, \boldsymbol{x}_l) < 0$. In both cases, because $\pi(\boldsymbol{x}_i) = 0$, we get that $\boldsymbol{x}_i$ is a best-response. □

Proposition 3.1.4. *Algorithm 2 terminates after at most* $m^{1+\frac{1}{\bar{u}}} - 1$ *steps, where* $\bar{u}$ *is the greatest common denominator of all elements in the set* $A = \{c(\boldsymbol{x}_i, \boldsymbol{x}_j) - c(\boldsymbol{x}_i, \boldsymbol{x}_k) \,|\, \boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_k \in \{1, \ldots, m\}\} \cup 1$.[1]

Proof. We prove that $\bar{u}$ is a denominator of $\pi(\boldsymbol{x}_j)$ $\forall \boldsymbol{x}_j \in \{1, \ldots, m\}$ after each step in the iterative algorithm. We prove this claim by induction. The induction basis is obvious as we initialize the values of $\pi(\boldsymbol{x}_j) = 0$ for all $\boldsymbol{x}_j$. For the induction step, suppose that we are going to update $\pi(\boldsymbol{x}_j)$ in our iterative algorithm. According to the induction hypothesis we know that $\frac{\pi(\boldsymbol{x}_k)}{\bar{u}} \in \mathcal{Z}$ $\forall \boldsymbol{x}_k \in \{1, \ldots, m\}$. Then, it can be shown that the new value of $\pi(\boldsymbol{x}_j)$ will be chosen among the elements of the following set (these are the thresholds that might change the transfer of masses):

$$\pi_{new}(\boldsymbol{x}_j) \in \{0\} \cup \{1\} \cup \{max_k(\pi(\boldsymbol{x}_k) - c(\boldsymbol{x}_i, \boldsymbol{x}_k)) + c(\boldsymbol{x}_i, \boldsymbol{x}_j) \,|\, \boldsymbol{x}_i \in \{0, \ldots, m\}\}$$

In the above, it is clear that all these possible values are divisible by $\bar{u}$, so the new value of $\pi(\boldsymbol{x}_j)$ will be divisible by $\bar{u}$ too. Then, since $0 \leq \pi(\boldsymbol{x}_j) \leq 1$ and

[1] The common denominator $\bar{u}$ satisfies that $\frac{a}{\bar{u}} \in \mathbb{Z}$ $\forall a \in A \cup \{1\}$. Such $\bar{u}$ exists if and only if $\frac{a}{b}$ is rational $\forall a, b \in A$.

$\frac{\pi(\boldsymbol{x}_j)}{\bar{u}} \in \mathcal{Z}$ for all $\boldsymbol{x}_j \in \{1, \ldots, m\}$, there are $1 + \frac{1}{\bar{u}}$ possible values for each $\pi(\boldsymbol{x}_j)$, *i.e.*, $0, \bar{u}, 2\bar{u}, \ldots, 1$. As a result, there are $m^{1+\frac{1}{\bar{u}}}$ different decision policies π. Finally, since the total utility increases after each step, the decision policy π at each step must be different. As a result, the algorithm will terminate after at most $m^{1+\frac{1}{\bar{u}}} - 1$ steps. □

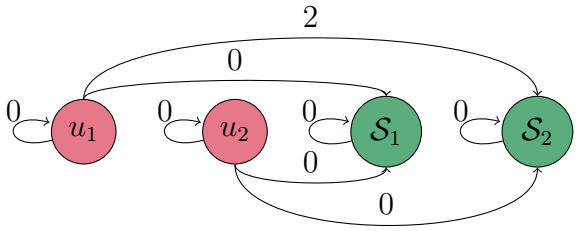


Figure A.1: **Reduction for Theorem 3.2.1.** Consider that $\mathcal{U} = \{u_1, u_2\}$ and $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2\}$ with $\mathcal{S}_1 = \{u_1, u_2\}$, $\mathcal{S}_2 = \{u_2\}$. The pink feature values have initial population $P(\boldsymbol{x}) = 1/2$, $\pi(\boldsymbol{x}) = 0$ and $P(Y = 1 \,|\, \boldsymbol{x}) = \gamma$, while for the green feature values it is $P(\boldsymbol{x}) = 0$, $\pi(\boldsymbol{x}) = 1$ and $P(Y = 1 \,|\, \boldsymbol{x}) = 1$. The edges represent the cost between feature values corresponding to sets and their respective elements while all the non-visible pairwise costs are equal to 2.

A.2 Proofs for Section 3.2

Theorem 3.2.1. *The problem of finding the optimal set of counterfactual explanations that maximizes utility under a cardinality constraint is NP-Hard.*

Proof. Consider an instance of the Set Cover problem with a set of elements $\mathcal{U} = \{u_1, \ldots, u_n\}$ and a collection $\mathcal{S} = \{\mathcal{S}_1, \ldots, \mathcal{S}_m\} \subseteq 2^{\mathcal{U}}$ such that $\bigcup_{i \in [m]} \mathcal{S}_i = \mathcal{U}$. In the decision version of the problem, given a constant k, we need to answer the question whether there are at most k sets from the collection $\mathcal{S}$ such that their union is equal to $\mathcal{U}$ or not. With the following procedure, we show that any instance of that problem can be transformed to an instance of the problem of finding the optimal set of counterfactual explanations, defined in Eq. 3.15, in polynomial time.

Consider $n+m$ feature values corresponding to the n elements of $\mathcal{U}$ and the m sets of $\mathcal{S}$. Moreover, denote the first n feature values as $\boldsymbol{x}_{u_1}, \ldots, \boldsymbol{x}_{u_n}$ and the remaining m as $\boldsymbol{x}_{\mathcal{S}_1}, \ldots, \boldsymbol{x}_{\mathcal{S}_m}$. We set the decision maker's parameter γ to some positive constant less than 1. Then, we set the outcome probabilities $P(Y = 1|\boldsymbol{x}_{u_i}) = \gamma \ \forall i \in [n]$ and $P(Y = 1|\boldsymbol{x}_{\mathcal{S}_i}) = 1 \ \forall i \in [m]$ and the policy values $\pi(\boldsymbol{x}_{u_i}) = 0 \ \forall i \in [n]$ and $\pi(\boldsymbol{x}_{\mathcal{S}_i}) = 1 \ \forall i \in [m]$. This way, the portion of utility the decision-maker obtains from the first n feature values is zero, while the portion of utility they obtain from the remaining m is proportional to $1 - \gamma$. Regarding the cost function, we set $c(\boldsymbol{x}_{u_i}, \boldsymbol{x}_{\mathcal{S}_j}) = 0 \ \forall (\boldsymbol{x}_{u_i}, \boldsymbol{x}_{\mathcal{S}_j}) : u_i \in S_j$, $c(\boldsymbol{x}_{u_i}, \boldsymbol{x}_{u_i}) = 0 \ \forall i \in [n]$, and all the remaining values of the cost function to 2. Finally, we set the initial feature value distribution to $P(\boldsymbol{x}_{u_i}) = \frac{1}{n} \ \forall i \in [n]$ and $P(\boldsymbol{x}_{\mathcal{S}_i}) = 0 \ \forall i \in [m]$. A toy example of this transformation is presented in Fig. A.1.

In this setting, it easy to observe that an individual with initial feature value $\boldsymbol{x}_{u_i}$ is always rejected at first and has the ability to move to a new feature value $\boldsymbol{x}_{\mathcal{S}_j}$ recommended to them iff $c(\boldsymbol{x}_{u_i}, \boldsymbol{x}_{\mathcal{S}_j}) \leq 1 \Leftrightarrow u_i \in \mathcal{S}_j$. Also, we can easily see that the transformation of instances can be done in $\mathcal{O}((m+n)^2)$ time.

Now, assume there exists an algorithm that optimally solves the problem of finding the optimal set of counterfactual explanations in polynomial time. Given the aforementioned instance and a maximum number of counterfactual explanations k, the utility $u(\pi, \mathcal{A})$ achieved by the set of counterfactual explanations $\mathcal{A}$ the algorithm returns can fall into one of the following two cases:

1. $u(\pi, \mathcal{A}) = 1 - \gamma$. This can happen only if all individuals, according to the

induced distribution $P(\boldsymbol{x}; \pi, \mathcal{A})$, have moved to some of the feature values $\boldsymbol{x}_{\mathcal{S}_j}$, *i.e.*, for all $\boldsymbol{x}_{u_i}$ with $i \in [n]$, there exists $\boldsymbol{x}_{\mathcal{S}_j}$ with $j \in [m]$ such that $\boldsymbol{x}_{\mathcal{S}_j} \in \mathcal{A} \wedge c(\boldsymbol{x}_{u_i}, \boldsymbol{x}_{\mathcal{S}_j}) \leq 1$ with $|\mathcal{A}| \leq k$. As a consequence, if we define $\mathcal{S}' = \{\mathcal{S}_j : \boldsymbol{x}_{\mathcal{S}_j} \in \mathcal{A}\}$, it holds that for all u_i with $i \in [n]$, there exists $\mathcal{S}_j$ with $j \in [m]$ such that $\mathcal{S}_j \in \mathcal{S}' \wedge u_i \in \mathcal{S}_j$ and therefore $\mathcal{S}'$ is a set cover with $|\mathcal{S}'| = |\mathcal{A}| \leq k$.

2. $u(\pi, \mathcal{A}) < 1 - \gamma$. This can happen only if every possible set of k counterfactual explanations leaves the individuals of at least one feature value $\boldsymbol{x}_{u_i}$ with a best-response of not following the counterfactual explanation they were given, *i.e.*, for all $\mathcal{A} \subseteq \mathcal{P}_\pi$ such that $|\mathcal{A}| \leq k$, there exists $\boldsymbol{x}_{u_i}$ with $i \in [n]$ such that, for all $\boldsymbol{x}_{\mathcal{S}_j} \in \mathcal{A}$, it holds that $c(\boldsymbol{x}_{u_i}, \boldsymbol{x}_{\mathcal{S}_j}) > 1$. Equivalently, it holds that for all $\mathcal{S}' \subseteq \mathcal{S}$ such that $|\mathcal{S}'| \leq k$, there exists u_i with $i \in [n]$ such that for all $\mathcal{S}_j \in \mathcal{S}'$, it holds that $u_i \notin \mathcal{S}_j$ and therefore there does not exist a set cover of size less or equal than k.

The above directly implies that we can have a decision about any instance of the Set Cover problem in polynomial time, which is a contradiction unless $P = NP$. This concludes the reduction and proves that the problem of finding the optimal set of counterfactual explanations for a given policy is NP-Hard. □

Proposition 3.2.1. *The function f is non-negative, submodular and monotone. Formally, all three of the following conditions are satisfied:*

1. *$f(\mathcal{A}) \geq 0$ for all $\mathcal{A} \subseteq \mathcal{P}_\pi$.*

2. *For all $\mathcal{A}, \mathcal{B} \subseteq \mathcal{P}_\pi : \mathcal{A} \subseteq \mathcal{B}$ and $\boldsymbol{x} \in \mathcal{P}_\pi \setminus \mathcal{B}$, it holds that $f(\mathcal{A} \cup \{\boldsymbol{x}\}) - f(\mathcal{A}) \geq f(\mathcal{B} \cup \{\boldsymbol{x}\}) - f(\mathcal{B})$.*

3. *For all $\mathcal{A} \subseteq \mathcal{P}_\pi$ and $\boldsymbol{x} \in \mathcal{P}_\pi$, it holds that $f(\mathcal{A} \cup \{\boldsymbol{x}\}) \geq f(\mathcal{A})$.*

Proof. It readily follows that the function f is non-negative from the fact that, if the decision maker is rational, it holds that $\pi(\boldsymbol{x}) = 0$ for all $\boldsymbol{x} \in \mathcal{X}$ such that $P(Y = 1 \,|\, \boldsymbol{x}) < \gamma$.

Now, consider two sets $\mathcal{A}, \mathcal{B} \subseteq \mathcal{P}_\pi : \mathcal{A} \subseteq \mathcal{B}$ and a feature value $\boldsymbol{x} \in \mathcal{P}_\pi \setminus \mathcal{B}$. Also, let $\mathsf{e}_{\mathcal{S}}(\boldsymbol{x}_i)$ be the counterfactual explanation given to the individuals with initial feature value $\boldsymbol{x}_i$ under a set of counterfactual explanations $\mathcal{S}$. It is easy to see that the marginal difference $f(\mathcal{S} \cup \{\boldsymbol{x}\}) - f(\mathcal{S})$ can only be affected by individuals with initial features $\boldsymbol{x}_i$ such that $\boldsymbol{x}_i \notin \mathcal{P}_\pi$, $\boldsymbol{x} \in \mathcal{R}(\boldsymbol{x}_i)$ and $\boldsymbol{x} = \mathsf{e}_{S \cup \{\boldsymbol{x}\}}(\boldsymbol{x}_i)$. Moreover, we can divide all of these individuals into two cases:

1. $\mathcal{R}(\boldsymbol{x}_i) \cap \mathcal{A} = \emptyset$: in this case, the addition of $\boldsymbol{x}$ to $\mathcal{A}$ causes a change in their best-response from $\boldsymbol{x}_i$ to $\boldsymbol{x}$ contributing to the marginal difference of f by a factor $P(\boldsymbol{x}_i)[P(Y = 1 \,|\, \boldsymbol{x}) - \gamma - \pi(\boldsymbol{x}_i)(P(Y = 1 \,|\, \boldsymbol{x}_i) - \gamma)]$. However, considering the marginal difference of f under the set of counterfactual explanations $\mathcal{B}$, three subcases are possible:

   (a) $\mathsf{e}_{\mathcal{B}}(\boldsymbol{x}_i) \in \mathcal{R}(\boldsymbol{x}_i) \wedge P(Y = 1 \,|\, \mathsf{e}_{\mathcal{B}}(\boldsymbol{x}_i)) > P(Y = 1 \,|\, \boldsymbol{x})$: the contribution to the marginal difference of f is zero.

(b) $\mathsf{e}_{\mathcal{B}}(\boldsymbol{x}_i) \in \mathcal{R}(\boldsymbol{x}_i) \wedge P(Y = 1 \,|\, \mathsf{e}_{\mathcal{B}}(\boldsymbol{x}_i)) \leq P(Y = 1 \,|\, \boldsymbol{x})$: the contribution to the marginal difference of f is $P(\boldsymbol{x}_i)[P(Y = 1 \,|\, \boldsymbol{x}) - P(Y = 1 \,|\, \mathsf{e}_{\mathcal{B}}(\boldsymbol{x}_i))]$. Since π is outcome monotonic, $\mathsf{e}_{\mathcal{B}}(\boldsymbol{x}_i) \in \mathcal{P}_\pi$ and $\boldsymbol{x}_i \notin \mathcal{P}_\pi$, it holds that

$$P(Y = 1 \,|\, \mathsf{e}_{\mathcal{B}}(\boldsymbol{x}_i)) \geq P(Y = 1 \,|\, \boldsymbol{x}_i) \Rightarrow$$
$$P(Y = 1 \,|\, \mathsf{e}_{\mathcal{B}}(\boldsymbol{x}_i)) - \gamma \geq P(Y = 1 \,|\, \boldsymbol{x}_i) - \gamma > \pi(\boldsymbol{x}_i)[P(Y = 1 \,|\, \boldsymbol{x}_i) - \gamma].$$

Therefore, it readily follows that

$$P(\boldsymbol{x}_i)[P(Y = 1 \,|\, \boldsymbol{x}) - P(Y = 1 \,|\, \mathsf{e}_{\mathcal{B}}(\boldsymbol{x}_i))] <$$
$$P(\boldsymbol{x}_i)[P(Y = 1 \,|\, \boldsymbol{x}) - \gamma - \pi(\boldsymbol{x}_i)(P(Y = 1 \,|\, \boldsymbol{x}_i) - \gamma)].$$

(c) $\mathcal{R}(\boldsymbol{x}_i) \cap \mathcal{B} = \emptyset$: the contribution to the marginal difference of f is $P(\boldsymbol{x}_i)[P(Y = 1 \,|\, \boldsymbol{x}) - \gamma - \pi(\boldsymbol{x}_i)(P(Y = 1 \,|\, \boldsymbol{x}_i) - \gamma)]$.

2. $\mathcal{R}(\boldsymbol{x}_i) \cap \mathcal{A} \neq \emptyset \wedge P(Y = 1 \,|\, \boldsymbol{x}) > P(Y = 1 \,|\, \mathsf{e}_{\mathcal{A}}(\boldsymbol{x}_i))$: In this case, the addition of $\boldsymbol{x}$ to $\mathcal{A}$ causes a change in their best-response from $\mathsf{e}_{\mathcal{A}}(\boldsymbol{x}_i)$ to $\boldsymbol{x}$ contributing to the marginal difference of f by a factor $P(\boldsymbol{x}_i)[P(Y = 1 \,|\, \boldsymbol{x}) - P(Y = 1 \,|\, \mathsf{e}_{\mathcal{A}}(\boldsymbol{x}_i))]$. Considering the marginal difference of f under the set of counterfactual explanations $\mathcal{B}$, two subcases are possible:

   (a) $\mathsf{e}_{\mathcal{B}}(\boldsymbol{x}_i) \in \mathcal{R}(\boldsymbol{x}_i) \wedge P(Y = 1 \,|\, \mathsf{e}_{\mathcal{B}}(\boldsymbol{x}_i)) > P(Y = 1 \,|\, \boldsymbol{x})$: the contribution to the marginal difference of f is zero.

   (b) $\mathsf{e}_{\mathcal{B}}(\boldsymbol{x}_i) \in \mathcal{R}(\boldsymbol{x}_i) \wedge P(Y = 1 \,|\, \mathsf{e}_{\mathcal{B}}(\boldsymbol{x}_i)) \leq P(Y = 1 \,|\, \boldsymbol{x})$. Then, the contribution of those individuals to the marginal difference of f is $P(\boldsymbol{x}_i)[P(Y = 1 \,|\, \boldsymbol{x}) - P(Y = 1 \,|\, \mathsf{e}_{\mathcal{B}}(\boldsymbol{x}_i))]$. Since $\mathcal{A} \subseteq \mathcal{B}$ and $\mathcal{R}(\boldsymbol{x}_i) \cap \mathcal{A} \neq \emptyset$, it readily follows that

$$P(Y = 1 \,|\, \mathsf{e}_{\mathcal{B}}(\boldsymbol{x}_i)) \geq P(Y = 1 \,|\, \mathsf{e}_{\mathcal{A}}(\boldsymbol{x}_i)) \Rightarrow$$
$$P(\boldsymbol{x}_i)[P(Y = 1 \,|\, \boldsymbol{x}) - P(Y = 1 \,|\, \mathsf{e}_{\mathcal{A}}(\boldsymbol{x}_i))] \geq$$
$$P(\boldsymbol{x}_i)[P(Y = 1 \,|\, \boldsymbol{x}) - P(Y = 1 \,|\, \mathsf{e}_{\mathcal{B}}(\boldsymbol{x}_i))].$$

Finally, because $\mathcal{A} \subseteq \mathcal{B}$, we can conclude that $f(\mathcal{B} \cup \{\boldsymbol{x}\}) - f(\mathcal{B}) \neq 0 \Rightarrow f(\mathcal{A} \cup \{\boldsymbol{x}\}) - f(\mathcal{A}) \neq 0$ and therefore the aforementioned cases are sufficient. Combining all cases, we can see that the contribution of each individual to the marginal difference of f is always greater or equal under the set of counterfactual explanations $\mathcal{A}$ than under the set of counterfactual explanations $\mathcal{B}$. As a direct consequence, it follows that f is submodular. Additionally, we can easily see that this contribution is always greater or equal than zero, leading to the conclusion that f is also monotone. □

Proposition 3.2.2. *Let $\mathcal{Y} = \{\boldsymbol{x} \in \mathcal{X} : P(Y = 1 \,|\, \boldsymbol{x}) \geq \gamma\}$. Given a set of counterfactual explanations $\mathcal{A} \subseteq \mathcal{Y}$, the policy $\pi^*_{\mathcal{A}} = \operatorname{argmax}_{\pi:\mathcal{A}\subseteq\mathcal{P}_\pi} u(\pi, \gamma, \mathcal{A})$ is deterministic, can be found in polynomial time, and is given by*

$$\pi^*_{\mathcal{A}}(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \Big(\{\boldsymbol{x}' \in \mathcal{A} : P(Y = 1 \,|\, \boldsymbol{x}') > P(Y = 1 \,|\, \boldsymbol{x}) \wedge c(\boldsymbol{x}, \boldsymbol{x}') \leq 1\} \\ & \qquad = \emptyset \wedge \boldsymbol{x} \in \mathcal{Y}\Big) \vee \boldsymbol{x} \in \mathcal{A} \\ 0 & \text{otherwise.} \end{cases}$$

Proof. By definition, since $\mathcal{A} \subseteq \mathcal{P}_{\pi^*_{\mathcal{A}}}$, it readily follows that $\pi^*_{\mathcal{A}}(\boldsymbol{x}) = 1$ for all $\boldsymbol{x} \in \mathcal{A}$. To find the remaining values of the decision policy, we first observe that, for each $\boldsymbol{x} \notin \mathcal{A}$, the value of the decision policy $\pi^*_{\mathcal{A}}(\boldsymbol{x})$ does not affect the best-responses of the individuals with initial feature values $\boldsymbol{x}' \neq \boldsymbol{x}$. As a result, we can just set $\pi^*_{\mathcal{A}}(\boldsymbol{x})$ for all $\boldsymbol{x} \notin \mathcal{A}$ independently for each feature value $\boldsymbol{x}$ such that the best-response of the respective individuals is the one that contributes maximally to the overall utility.

First, it is easy to see that, for all $\boldsymbol{x} \notin \mathcal{A}$ such that $P(Y = 1 \,|\, \boldsymbol{x}) < \gamma$, we should set $\pi^*_{\mathcal{A}}(\boldsymbol{x}) = 0$. Next, consider the feature values $\boldsymbol{x} \notin \mathcal{A}$ such that $P(Y = 1 \,|\, \boldsymbol{x}) \geq \gamma$. Here, we distinguish two cases. If there exists $\boldsymbol{x}' \in \mathcal{A}$ such that $c(\boldsymbol{x}, \boldsymbol{x}') \leq 1 \wedge P(Y = 1 \,|\, \boldsymbol{x}') > P(Y = 1 \,|\, \boldsymbol{x})$, then, if the individuals move to that $\boldsymbol{x}'$, the corresponding contribution to the utility will be higher. Moreover, the value of the decision policy that maximizes their region of adaption (and thus increases their chances of moving to $\boldsymbol{x}'$) is clearly $\pi^*_{\mathcal{A}}(\boldsymbol{x}) = 0$. If there does not exist $\boldsymbol{x}' \in \mathcal{A}$ such that $c(\boldsymbol{x}, \boldsymbol{x}') \leq 1 \wedge P(Y = 1 \,|\, \boldsymbol{x}') > P(Y = 1 \,|\, \boldsymbol{x})$, then, the contribution of the corresponding individuals to the utility will be higher if they keep their initial feature values. Moreover, the value of the decision policy that will maximize this contribution will be clearly $\pi^*_{\mathcal{A}}(\boldsymbol{x}) = 1$. □

Proposition 3.2.3. *The function h is non-negative, submodular and non-monotone.*

Proof. It readily follows that the function h is non-negative from the fact that, if the decision maker is rational, $\pi(\boldsymbol{x}) = 0$ for all $\boldsymbol{x} \in \mathcal{X}$ such that $P(Y = 1 \,|\, \boldsymbol{x}) < \gamma$.

Next, consider two sets $\mathcal{A}, \mathcal{B} \subseteq \mathcal{Y}$ such that $\mathcal{A} \subseteq \mathcal{B}$ and a feature value $\boldsymbol{x} \in \mathcal{Y} \setminus \mathcal{B}$. Also, let $\mathsf{e}_{\mathcal{S}}(\boldsymbol{x}_i)$ be the counterfactual explanation given to the individuals with initial feature value $\boldsymbol{x}_i$ under a set of counterfactual explanations $\mathcal{S}$. Then, it is clear that the marginal difference $h(\mathcal{S} \cup \{\boldsymbol{x}\}) - h(\mathcal{S})$ only depends on individuals with initial features $\boldsymbol{x}_i$ such that either $1 - c(\boldsymbol{x}_i, \boldsymbol{x}) \geq 0$ and $\boldsymbol{x} = \mathsf{e}_{\mathcal{S} \cup \{\boldsymbol{x}\}}(\boldsymbol{x}_i)$ or $\boldsymbol{x}_i = \boldsymbol{x}$. Moreover, if $1 - c(\boldsymbol{x}_i, \boldsymbol{x}) \geq 0$ and $\boldsymbol{x} = \mathsf{e}_{\mathcal{S} \cup \{\boldsymbol{x}\}}(\boldsymbol{x}_i)$, the contribution to the marginal difference is positive and, if $\boldsymbol{x}_i = \boldsymbol{x}$, the contribution to the marginal difference is negative.

Consider first the individuals with initial features $\boldsymbol{x}_i$ such that $1 - c(\boldsymbol{x}_i, \boldsymbol{x}) \geq 0$ and $\boldsymbol{x} = \mathsf{e}_{\mathcal{A} \cup \{\boldsymbol{x}\}}(\boldsymbol{x}_i)$. We can divide all of these individuals into three cases:

1. $\pi_{\mathcal{B}}(\boldsymbol{x}_i) = 0$: in this case, $\boldsymbol{x}_i \notin \mathcal{B}$ and the individuals change their best-response from $\mathsf{e}_{\mathcal{B}}(\boldsymbol{x}_i)$ to $\boldsymbol{x}$. Moreover, under the set of counterfactual explanations $\mathcal{A}$, their best-response is either $\boldsymbol{x}_i$ or $\mathsf{e}_{\mathcal{A}}(\boldsymbol{x}_i)$ and it changes to $\boldsymbol{x}$. Then, using a similar argument as in the proof of proposition 3.2.1, we can conclude that the contribution of the individuals to the marginal difference is greater or equal under the set of counterfactual explanations $\mathcal{A}$ than under $\mathcal{B}$.

2. $\pi_{\mathcal{B}}(\boldsymbol{x}_i) = 1 \wedge \pi_{\mathcal{A}}(\boldsymbol{x}_i) = 0$: in this case, $\boldsymbol{x}_i \notin \mathcal{A}$ and $\boldsymbol{x}_i \in \mathcal{B}$. Therefore, under the set of counterfactual explanations $\mathcal{A}$, the individuals' best-response changes from $\mathsf{e}_{\mathcal{A}}(\boldsymbol{x}_i)$ to $\boldsymbol{x}$ and there is a positive contribution to the marginal difference while, under $\mathcal{B}$, the individuals' best-response does not change and the contribution to the marginal difference is zero.

3. $\pi_{\mathcal{B}}(\boldsymbol{x}_i) = 1 \wedge \pi_{\mathcal{A}}(\boldsymbol{x}_i) = 1$: in this case, $\boldsymbol{x}_i \notin \mathcal{B}$. Therefore, the best-response changes from $\boldsymbol{x}_i$ to $\boldsymbol{x}$ under both sets of counterfactual explanations and there is an equal positive contribution to the marginal difference.

Now, consider the individuals with initial features $\boldsymbol{x}_i$ such that $\boldsymbol{x}_i = \boldsymbol{x}$. We can divide all of these individuals also into three cases:

1. $\pi_{\mathcal{A}}(\boldsymbol{x}) = \pi_{\mathcal{B}}(\boldsymbol{x}) = 0$: in this case, under both sets of counterfactual explanations, the counterfactual explanation $\boldsymbol{x}$ changes the value of the decision policy to $\pi_{\mathcal{A}\cup\{\boldsymbol{x}\}}(\boldsymbol{x}) = \pi_{\mathcal{B}\cup\{\boldsymbol{x}\}}(\boldsymbol{x}) = 1$. Moreover, the contribution to the marginal difference is less negative under the set of counterfactual explanations $\mathcal{A}$ than under $\mathcal{B}$ since $P(Y = 1 \,|\, \mathsf{e}_{\mathcal{A}}(\boldsymbol{x})) \leq P(Y = 1 \,|\, \mathsf{e}_{\mathcal{B}}(\boldsymbol{x}))$ and thus $P(\boldsymbol{x})[P(Y = 1 \,|\, \boldsymbol{x}) - P(Y = 1 \,|\, \mathsf{e}_{\mathcal{A}}(\boldsymbol{x}))] \geq P(\boldsymbol{x})[P(Y = 1 \,|\, \boldsymbol{x}) - P(Y = 1 \,|\, \mathsf{e}_{\mathcal{B}}(\boldsymbol{x}))]$.

2. $\pi_{\mathcal{A}}(\boldsymbol{x}) = 1 \wedge \pi_{\mathcal{B}}(\boldsymbol{x}) = 0$: in this case, under the set of counterfactual explanations $\mathcal{A}$, the individuals' best-response does not change and thus the contribution to the marginal difference is zero and, under the set of counterfactual explanations $\mathcal{B}$, their best-response changes from $\mathsf{e}_{\mathcal{B}}(\boldsymbol{x})$ to $\boldsymbol{x}$ and thus there is a negative contribution to the marginal difference *i.e.*, $P(\boldsymbol{x})[P(Y = 1 \,|\, \boldsymbol{x}) - P(Y = 1 \,|\, \mathsf{e}_{\mathcal{B}}(\boldsymbol{x}))] < 0$.

3. $\pi_{\mathcal{A}}(\boldsymbol{x}) = \pi_{\mathcal{B}}(\boldsymbol{x}) = 1$: in this case, under both sets of counterfactual explanations, the individuals' best-response does not change and thus the contribution to the marginal difference is zero.

As a direct consequence of the above observations, it readily follows that $h(\mathcal{A} \cup \{\boldsymbol{x}\}) - h(\mathcal{A}) \geq h(\mathcal{B} \cup \{\boldsymbol{x}\}) - h(\mathcal{B})$ and therefore the function h is submodular.

However, in contrast with Section 3.2.2, the function h is non-monotone since it can happen that the negative marginal contribution exceeds the positive one. For example, consider the following instance of the problem, where $\boldsymbol{x} \in \{1, 2, 3\}$ with $\gamma = 0.1$:

$$P(\boldsymbol{x}) = 0.1\,\mathbb{1}[\boldsymbol{x} = 1] + 0.8\,\mathbb{1}\boldsymbol{x} = 2] + 0.1\,\mathbb{1}[\boldsymbol{x} = 3],$$
$$P(Y = 1 \,|\, \boldsymbol{x}) = 1.0\,\mathbb{1}[\boldsymbol{x} = 1] + 0.5\,\mathbb{1}[\boldsymbol{x} = 2] + 0.4\,\mathbb{1}[\boldsymbol{x} = 3],$$

and

$$c(\boldsymbol{x}_i, \boldsymbol{x}_j) = \begin{bmatrix} 0.0 & 0.2 & 0.3 \\ 0.3 & 0.0 & 0.7 \\ 0.4 & 0.5 & 0.0 \end{bmatrix}.$$

Assume there is a set of counterfactual explanations $\mathcal{A} = \{1\}$. Then, the optimal policy is given by $\pi^*_{\mathcal{A}}(1) = 1, \pi^*_{\mathcal{A}}(2) = 0, \pi^*_{\mathcal{A}}(3) = 0$ inducing a movement from feature values $2, 3$ to feature value 1, giving a utility equal to 0.9. Now, add $\boldsymbol{x} = 2$ to the set of counterfactual explanations *i.e.*, $\mathcal{A} = \{1, 2\}$. Then, the optimal policy is given by $\pi^*_{\mathcal{A}}(1) = 1, \pi^*_{\mathcal{A}}(2) = 1, \pi^*_{\mathcal{A}}(3) = 0$ inducing a movement from feature value 3 to feature value 1, giving a lower utility, equal to 0.5. Therefore, the function h is non-monotone. □

A.3 Proofs for Section 4.1

Proposition 4.1.1. *The counterfactual policy π^*_τ returned by Algorithm 6 is the solution to the optimization problem defined by Eq. 4.8.*

Proof. Using induction, we will prove that the policy value $\pi_\tau((s,l),t')$ set by Algorithm 6 is optimal for every $s \in \mathcal{S}$, $l \in \{0,\dots,k\}$, $t' \in \{0,\dots,T-1\}$ in the sense that following this policy maximizes the average cumulative reward $h(s,q,c)$ that one could have achieved in the last $q = T - t'$ steps of the decision making process, starting from state $S_{T-q} = s$, if at most $c = k - l$ actions had been different to the observed ones in those last steps. Formally:

$$h(s,q,c) = \max_{\pi} \mathbb{E}_{\{((s'_t,l_t),a'_t)\}_{t=t'}^{T-1} \sim P_\tau^+ \,|\, S^+_{t'}=(s,l)} \left[\sum_{t=t'}^{T-1} r^+\left(\left(s'_t,l_t\right),a'_t\right) \right] \tag{A.1}$$

$$\text{subject to} \sum_{t=t'}^{T-1} \mathbb{1}[a_t \neq a'_t] \leq c \quad \forall \{((s'_t,l_t),a'_t)\}_{t=t'}^{T-1} \sim P_\tau^+ \tag{A.2}$$

Recall that, $a_1,\dots,a_{T-1}$ are the observed actions and the counterfactual realizations $a'_1,\dots,a'_{T-1}$ are induced by the counterfactual transition probability P_τ^+ and the policy π.

We start by proving the induction basis. Assume that a realization has reached a state $s^+_{T-1} = (s,l)$ at time $T-1$, one time step before the end of the process. If $c = 0$ (*i.e.*, $l = k$), following Eq. 4.10, the algorithm will choose the observed action $\pi_\tau((s,l),t') = a_{T-1}$ and return an average cumulative reward $h(s,1,0) = r(s,a_{T-1}) + \sum_{s'\in\mathcal{S}} P_{\tau,T-1}(s' \,|\, s,a_{T-1})\, h(s',0,0) = r(s,a_{T-1})$, where $h(s',0,0) = 0$ for all $s' \in \mathcal{S}$. Since no more action changes can be performed at this stage, this is the only feasible solution and therefore it is optimal.

If $c > 0$, since $h(s',0,c) = h(s',0,c-1) = 0$ for all $s' \in \mathcal{S}$ it is easy to verify that Eq. 4.9 reduces to $h(s,1,c) = \max_{a\in\mathcal{A}} r(s,a)$ and $\pi_\tau((s,l),t') = \operatorname{argmax}_{a\in\mathcal{A}} r(s,a)$ is obviously the optimal choice for the last time step.

Now, we will prove that, for a counterfactual realization being at state $s^+_{t'} = (s,l)$ at a time step $t' < T-1$ (*i.e.*, $r = T - t'$, $c = k - l$), the maximum average cumulative reward $h(s,q,c)$ given by Algorithm 6 is optimal, under the inductive hypothesis that the values of $h(s',q',c')$ already computed for $q' < q$, $c' < c$ and all $s' \in \mathcal{S}$ are optimal. Assume that the algorithm returns an average cumulative reward $h(s,q,c)$ by choosing action $\pi_\tau((s,l),t') = a$ while the optimal solution gives an average cumulative reward $OPT_{s,q,c} > h(s,q,c)$ by choosing an action $a^* \neq a$. Here, by $\tau'_{t'} = \{((s'_t,l_t),a'_t)\}_{t=t'}^{T-1}$ we will denote realizations starting from time t' with $a'_t = \pi_\tau\left(\left(s'_t,l_t\right),t\right)$ where π_τ is the policy given by Algorithm 6 and we will use $\tau^*_{t'}$ if the policy is optimal. Also, we will denote a possible next state at time $t'+1$, after performing action a, as (s',l') where $l' = l+1$ if $a \neq a_t$, $l' = l$ otherwise and, $c' = k - l'$. Similarly, after performing action a^*, we will denote a possible next state as (s',l^*) where $l^* = l+1$ if $a^* \neq a_t$, $l^* = l$ otherwise and, $c^* = k - l^*$. Then, we get:

$$h(s,q,c) < OPT_{s,q,c}$$

$$\implies \mathbb{E}_{\tau'_{t'}\sim P_\tau^+ \,|\, S^+_{t'}=(s,l)} \left[\sum_{t=t'}^{T-1} r^+\left(\left(s_t,l_t\right),a_t\right) \right] < \mathbb{E}_{\tau^*_{t'}\sim P_\tau^+ \,|\, S^+_{t'}=(s,l)} \left[\sum_{t=t'}^{T-1} r^+\left(\left(s_t,l_t\right),a_t\right) \right]$$

$$
\begin{aligned}
&\overset{(a)}{\Longrightarrow} \sum_{s'} P_{\tau,T-q}(s' \mid s,a)\mathbb{E}_{\tau'_{t'+1}\sim P^+_\tau \mid S^+_{t'+1}=(s',l')}\left[\sum_{t=t'}^{T-1} r^+\left((s_t,l_t),a_t\right)\right] \\
&\quad < \sum_{s'} P_{\tau,T-q}(s' \mid s,a^*)\mathbb{E}_{\tau^*_{t'+1}\sim P^+_\tau \mid S^+_{t'+1}=(s',l^*)}\left[\sum_{t=t'}^{T-1} r^+\left((s_t,l_t),a_t\right)\right] \\
&\Longrightarrow \sum_{s'} P_{\tau,T-q}(s' \mid s,a)\left[r^+\left((s,l),a\right) + \mathbb{E}_{\tau'_{t'+1}\sim P^+_\tau \mid S^+_{t'+1}=(s',l')}\left[\sum_{t=t'+1}^{T-1} r^+\left((s_t,l_t),a_t\right)\right]\right] \\
&\quad < \sum_{s'} P_{\tau,T-q}(s' \mid s,a^*)\Bigg[r^+\left((s,l),a^*\right) \\
&\qquad + \mathbb{E}_{\tau^*_{t'+1}\sim P^+_\tau \mid S^+_{t'+1}=(s',l^*)}\left[\sum_{t=t'+1}^{T-1} r^+\left((s_t,l_t),a_t\right)\right]\Bigg] \\
&\overset{(b)}{\Longrightarrow} \sum_{s'} P_{\tau,T-r}(s' \mid s,a)r(s,a) + \sum_{s'} P_{\tau,T-q}(s' \mid s,a)h(s',q-1,c') \\
&\quad < \sum_{s'} P_{\tau,T-q}(s' \mid s,a^*)r(s,a^*) + \sum_{s'} P_{\tau,T-q}(s' \mid s,a^*)OPT_{s',q-1,c^*} \\
&\overset{(c)}{\Longrightarrow} r(s,a) + \sum_{s'} P_{\tau,T-q}(s' \mid s,a)h(s',q-1,c') \\
&\quad < r(s,a^*) + \sum_{s'} P_{\tau,T-q}(s' \mid s,a^*)h(s',q-1,c^*),
\end{aligned}
$$

where, in (a), we expand the expectation for one time step, in (b), we replace the average cumulative reward starting from time step $t'+1$ with $h(s',q-1,c')$ and $OPT_{s',q-1,c^*}$ for the policy of Algorithm 6 and the optimal one respectively and, in (c), we replace $OPT_{s',q-1,c^*}$ with $h(s',q-1,c^*)$ due to the inductive hypothesis.

It is easy to see that, it can either be $a^* = a_t$ with $c^* = c$ or $a^* \in \mathcal{A} \setminus a_t$ with $c^* = c-1$. If $c = 0$, following Eq. 4.10, the algorithm will choose the observed action (*i.e.*, $a = a_t$). This is the only feasible solution, since $a^* \neq a_t$ would give $c^* = -1$ and $l^* = k - c^* = k+1$, which is not a valid state. Therefore, we get $a = a^* = a_t$, which is a a contradiction. If $c > 0$, because of the max operator in Eq. 4.9, for the action a chosen by Algorithm 6, it necessarily holds that:

$$
r(s,a) + \sum_{s'} P_{\tau,T-q}(s' \mid s,a)h(s',q-1,c') \geq r(s,a^*) + \sum_{s'} P_{\tau,T-q}(s' \mid s,a^*)h(s',q-1,c^*),
$$

which is clearly a contradiction.

Therefore, the average cumulative reward $h(s,q,c)$ computed by Algorithm 6 and its associated policy value $\pi_\tau((s,l),t')$ are optimal for every $s \in \mathcal{S}$, $l \in \{0,\ldots,k\}$, $t' \in \{0,\ldots,T-1\}$ and $h(s_0,T,k)$ is the solution to the optimization problem defined by Eq. 4.8. □

A.4 Proofs for Section 4.2

Theorem 4.2.1. *Let $\mathcal{C}$ and $\mathcal{C}'$ be two element-wise bijective SCMs with transition mechanisms g_S and h_S, respectively, and, for any observed transition $(\boldsymbol{s}_t, a_t, \boldsymbol{s}_{t+1})$, let $\boldsymbol{u}_t = g_S^{-1}(\boldsymbol{s}_t, a_t, \boldsymbol{s}_{t+1})$ and $\tilde{\boldsymbol{u}}_t = h_S^{-1}(\boldsymbol{s}_t, a_t, \boldsymbol{s}_{t+1})$. Moreover, given any $\boldsymbol{s} \in \mathcal{S}, a \in \mathcal{A}$, let $\boldsymbol{s}' = g_S(\boldsymbol{s}, a, \boldsymbol{u}_t)$ and $\boldsymbol{s}'' = h_S(\boldsymbol{s}, a, \tilde{\boldsymbol{u}}_t)$. If $P^{\mathcal{C}}(\boldsymbol{S}_{t+1} \,|\, \boldsymbol{S}_t = \boldsymbol{s}, A_t = a) = P^{\mathcal{C}'}(\boldsymbol{S}_{t+1} \,|\, \boldsymbol{S}_t = \boldsymbol{s}, A_t = a)$ for all $\boldsymbol{s} \in \mathcal{S}, a \in \mathcal{A}$, it must hold that $\boldsymbol{s}' = \boldsymbol{s}''$.*

Proof. We prove the theorem by induction, starting by establishing the base case $s'_1 = s''_1$. Without loss of generality, assume that both $g_{S,1}$ and $h_{S,1}$ are strictly increasing with respect to their third argument. Since the two SCMs entail the same transition distributions, we have that

$$P^{\mathcal{C}}\left(S_{t+1,1} \leq s_{t+1,1} \,|\, \boldsymbol{S}_t = \boldsymbol{s}_t, A_t = a_t\right) = P^{\mathcal{C}'}\left(S_{t+1,1} \leq s_{t+1,1} \,|\, \boldsymbol{S}_t = \boldsymbol{s}_t, A_t = a_t\right) \overset{(*)}{\Rightarrow}$$
$$P^{\mathcal{C}}\left(g_{S,1}\left(\boldsymbol{s}_t, a_t, U_{t,1}\right) \leq g_{S,1}\left(\boldsymbol{s}_t, a_t, u_{t,1}\right)\right) = P^{\mathcal{C}'}\left(h_{S,1}\left(\boldsymbol{s}_t, a_t, U_{t,1}\right) \leq h_{S,1}\left(\boldsymbol{s}_t, a_t, \tilde{u}_{t,1}\right)\right) \overset{(**)}{\Rightarrow}$$
$$P^{\mathcal{C}}\left(U_{t,1} \leq u_{t,1}\right) = P^{\mathcal{C}'}\left(U_{t,1} \leq \tilde{u}_{t,1}\right),$$

where $(*)$ holds because both SCMs are element-wise bijective, and $(**)$ holds because $g_{S,1}$ and $h_{S,1}$ are increasing with respect to their third argument. Similarly, we have that

$$\begin{aligned}
P^{\mathcal{C}}\left(S_{t+1,1} \leq s'_1 \,|\, \boldsymbol{S}_t = \boldsymbol{s}, A_t = a\right) &= P^{\mathcal{C}}\left(g_{S,1}\left(\boldsymbol{s}, a, U_{t,1}\right) \leq g_{S,1}\left(\boldsymbol{s}, a, u_{t,1}\right)\right) \\
&\overset{(\star)}{=} P^{\mathcal{C}}\left(U_{t,1} \leq u_{t,1}\right) \\
&\overset{(\star\star)}{=} P^{\mathcal{C}'}\left(U_{t,1} \leq \tilde{u}_{t,1}\right) \\
&\overset{(\dagger)}{=} P^{\mathcal{C}'}\left(h_{S,1}\left(\boldsymbol{s}, a, U_{t,1}\right) \leq h_{S,1}\left(\boldsymbol{s}, a, \tilde{u}_{t,1}\right)\right) \\
&= P^{\mathcal{C}'}\left(S_{t+1,1} \leq s''_1 \,|\, \boldsymbol{S}_t = \boldsymbol{s}, A_t = a\right) \\
&= P^{\mathcal{C}}\left(S_{t+1,1} \leq s''_1 \,|\, \boldsymbol{S}_t = \boldsymbol{s}, A_t = a\right),
\end{aligned}$$

where in $(\star), (\dagger)$ we have used the monotonicity of g_S and h_S, and $(\star\star)$ follows from the previous result. The last equality implies that s'_1 and s''_1 correspond to the same quantile of the distribution for $S_{t+1,1} \,|\, \boldsymbol{S}_t = \boldsymbol{s}, A_t = a$. Therefore, it is easy to see that $s'_1 = s''_1$ since the opposite would be in contradiction to $g_{S,1}$ being bijective. Note that, we can reach that conclusion irrespective of the direction of monotonicity of $g_{S,1}$ and $h_{S,1}$, since any change in the direction of the inequalities happening at step $(**)$ is reverted at steps $(\star)$ and $(\dagger)$.

Now, starting from the inductive hypothesis that $s'_i = s''_i$ for all $i \in \{1, \ldots, n\}$ with $n < d$, we show the inductive step, *i.e.*, $s'_{n+1} = s''_{n+1}$. Again, without loss of generality, assume that both $g_{S,n+1}$ and $h_{S,n+1}$ are strictly increasing with respect to their last argument. Note that, the two SCMs entail the same transition distributions, *i.e.*, the same joint distributions for $\boldsymbol{S}_{t+1} \,|\, \boldsymbol{S}_t, A_t$. Following from the law of total probability, they also entail the same conditional distributions for $S_{t+1,n+1} \,|\, \boldsymbol{S}_{t+1,\leq n}, \boldsymbol{S}_t, A_t$, where we use the notation $\boldsymbol{x}_{\leq n}$ to refer to a vector that contains the first n elements of a d-dimensional vector $\boldsymbol{x}$. Therefore, we have that

$$P^{\mathcal{C}}\left(S_{t+1,n+1} \leq s_{t+1,n+1} \,|\, \boldsymbol{S}_{t+1,\leq n} = \boldsymbol{s}_{t+1,\leq n}, \boldsymbol{S}_t = \boldsymbol{s}_t, A_t = a_t\right) =$$
$$P^{\mathcal{C}'}\left(S_{t+1,n+1} \leq s_{t+1,n+1} \,|\, \boldsymbol{S}_{t+1,\leq n} = \boldsymbol{s}_{t+1,\leq n}, \boldsymbol{S}_t = \boldsymbol{s}_t, A_t = a_t\right) \overset{(*)}{\Rightarrow}$$

$$
\begin{aligned}
&P^{\mathcal{C}}\left(g_{S,n+1}\left(\boldsymbol{s}_t, a_t, U_{t,n+1}\right) \leq g_{S,n+1}\left(\boldsymbol{s}_t, a_t, u_{t,n+1}\right) \,|\, \boldsymbol{U}_{t,\leq n} = \boldsymbol{u}_{t,\leq n}\right) \\
&\qquad = P^{\mathcal{C}'}\left(h_{S,n+1}\left(\boldsymbol{s}_t, a_t, U_{t,n+1}\right) \leq h_{S,n+1}\left(\boldsymbol{s}_t, a_t, \tilde{u}_{t,n+1}\right) \,|\, \boldsymbol{U}_{t,\leq n} = \tilde{\boldsymbol{u}}_{t,\leq n}\right) \overset{(**)}{\Rightarrow} \\
&P^{\mathcal{C}}\left(U_{t,n+1} \leq u_{t,n+1} \,|\, \boldsymbol{U}_{t,\leq n} = \boldsymbol{u}_{t,\leq n}\right) = P^{\mathcal{C}'}\left(U_{t,n+1} \leq \tilde{u}_{t,n+1} \,|\, \boldsymbol{U}_{t,\leq n} = \tilde{\boldsymbol{u}}_{t,\leq n}\right),
\end{aligned}
$$

where for the first equality we have used the inductive hypothesis, $(*)$ holds because both SCMs are element-wise bijective, and $(**)$ holds because $g_{S,n+1}$ and $h_{S,n+1}$ are increasing with respect to their third argument. Similarly, we get that

$$
\begin{aligned}
&P^{\mathcal{C}}\left(S_{t+1,n+1} \leq s'_{n+1} \,|\, \boldsymbol{S}_{t+1,\leq n} = \boldsymbol{s}_{t+1,\leq n}, \boldsymbol{S}_t = \boldsymbol{s}, A_t = a\right) \\
&= P^{\mathcal{C}}\Big(g_{S,n+1}\left(\boldsymbol{s}, a, U_{t,n+1}\right) \leq g_{S,n+1}\left(\boldsymbol{s}, a, u_{t,n+1}\right) \\
&\qquad\qquad | \, g_{S,\leq n}\left(\boldsymbol{s}, a, \boldsymbol{U}_{t,\leq n}\right) = g_{s,\leq n}\left(\boldsymbol{s}, a, \boldsymbol{u}_{t,\leq n}\right)\Big) \\
&\overset{(\star)}{=} P^{\mathcal{C}}\left(U_{t,n+1} \leq u_{t,n+1} \,|\, \boldsymbol{U}_{t,\leq n} = \boldsymbol{u}_{t,\leq n}\right) \\
&\overset{(\star\star)}{=} P^{\mathcal{C}'}\left(U_{t,n+1} \leq \tilde{u}_{t,n+1} \,|\, \boldsymbol{U}_{t,\leq n} = \tilde{\boldsymbol{u}}_{t,\leq n}\right) \\
&\overset{(\dagger)}{=} P^{\mathcal{C}'}\Big(h_{S,n+1}\left(\boldsymbol{s}, a, U_{t,1}\right) \leq h_{S,n+1}\left(\boldsymbol{s}, a, \tilde{u}_{t,1}\right) \\
&\qquad\qquad | \, h_{S,\leq n}\left(\boldsymbol{s}, a, \boldsymbol{U}_{t,\leq n}\right) = h_{s,\leq n}\left(\boldsymbol{s}, a, \tilde{\boldsymbol{u}}_{t,\leq n}\right)\Big) \\
&= P^{\mathcal{C}'}\left(S_{t+1,n+1} \leq s''_{n+1} \,|\, \boldsymbol{S}_{t+1,\leq n} = \boldsymbol{S}_{t+1,\leq n}, \boldsymbol{S}_t = \boldsymbol{s}, A_t = a\right) \\
&= P^{\mathcal{C}}\left(S_{t+1,n+1} \leq s''_{n+1} \,|\, \boldsymbol{S}_{t+1,\leq n} = \boldsymbol{s}_{t+1,\leq n}, \boldsymbol{S}_t = \boldsymbol{s}, A_t = a\right),
\end{aligned}
$$

where in $(\star), (\dagger)$ we have used the invertibility and monotonicity of g_S and h_S, and $(\star\star)$ follows from the previous result. With the same argument as in the base case, the last equality implies that $s'_{n+1} = s''_{n+1}$. That concludes the proof. □

Theorem 4.2.2. *The problem defined by Eq. 4.19. is NP-Hard.*

Proof. We prove the hardness of our problem as defined in Eq. 4.19 by performing a reduction from the partition problem [168], which is known to be NP-Complete. In the partition problem, we are given a multiset of B positive integers $\mathcal{V} = \{v_1, \ldots, v_B\}$ and the goal is to decide whether there is a partition of $\mathcal{V}$ into two subsets $\mathcal{V}_1, \mathcal{V}_2$ with $\mathcal{V}_1 \cap \mathcal{V}_2 = \emptyset$ and $\mathcal{V}_1 \cup \mathcal{V}_2 = \mathcal{V}$, such that their sums are equal, *i.e.*, $\sum_{v_i \in \mathcal{V}_1} v_i = \sum_{v_j \in \mathcal{V}_2} v_j$.

Consider an instance of our problem where $\mathcal{S} = \mathcal{U} = \mathbb{R}^2$, $\mathcal{A}$ contains 2 actions $a_{\text{diff}}, a_{\text{null}}$ and the horizon is $T = B + 1$. Let $\mathcal{C}$ be an element-wise bijective SCM with arbitrary prior distributions $P^{\mathcal{C}}(\boldsymbol{U}_t)$ such that their support is on $\mathbb{R}^2$ and a transition mechanism g_S such that

$$
g_S(\boldsymbol{S}_t, a_{\text{diff}}, \boldsymbol{U}_t) = \begin{bmatrix} S_{t,1} - S_{t,2} \\ 0 \end{bmatrix} + \boldsymbol{U}_t \quad \text{and} \quad g_S(\boldsymbol{S}_t, a_{\text{null}}, \boldsymbol{U}_t) = \begin{bmatrix} S_{t,1} \\ 0 \end{bmatrix} + \boldsymbol{U}_t. \tag{A.3}
$$

Moreover, assume that the reward function is given by

$$
\begin{aligned}
r(\boldsymbol{S}_t, a_{\text{diff}}) = r(\boldsymbol{S}_t, a_{\text{null}}) = &- \max\left(0, S_{t,1} - \frac{sum(\mathcal{V})}{2} - S_{t,2}\frac{sum(\mathcal{V})}{2}\right) \\
&- \max\left(0, \frac{sum(\mathcal{V})}{2} - S_{t,1} - S_{t,2}\frac{sum(\mathcal{V})}{2}\right),
\end{aligned} \tag{A.4}
$$

where $sum(\mathcal{V})$ is the sum of all elements $\sum_{i=1}^{B} v_i$. Note that, the SCM $\mathcal{C}$ defined above is Lipschitz-continuous as suggested by the following lemma.

Lemma A.4.1. *The SCM $\mathcal{C}$ defined by Eqs. A.3, A.4 is Lipschitz-continuous according to Definition 4.2.1.*

Proof. It is easy to see that, for all $\boldsymbol{u} \in \mathcal{U}$ and for all $\boldsymbol{s}, \boldsymbol{s}' \in \mathcal{S}$, the function $g_S(\boldsymbol{S}_t, a_{\text{null}}, \boldsymbol{u})$ satisfies $\|g_S(\boldsymbol{s}, a_{\text{null}}, \boldsymbol{u}) - g_S(\boldsymbol{s}', a_{\text{null}}, \boldsymbol{u})\| \leq \|\boldsymbol{s} - \boldsymbol{s}'\|$, and therefore $K_{a_{\text{null}}, \boldsymbol{u}} = 1$ satisfies Definition 4.2.1. For the case of $A_t = a_{\text{diff}}$, we have that

$$\begin{aligned}
&\|g_S(\boldsymbol{s}, a_{\text{diff}}, \boldsymbol{u}) - g_S(\boldsymbol{s}', a_{\text{diff}}, \boldsymbol{u})\| \\
&\quad = \left\| \begin{bmatrix} s_1 - s_2 \\ 0 \end{bmatrix} - \begin{bmatrix} s_1' - s_2' \\ 0 \end{bmatrix} \right\| \\
&\quad = \left\| \begin{bmatrix} (s_1 - s_1') + (s_2' - s_2) \\ 0 \end{bmatrix} \right\| \\
&\quad = |(s_1 - s_1') + (s_2' - s_2)| \leq |s_1 - s_1'| + |s_2 - s_2'|,
\end{aligned}$$

and therefore $\|g_S(\boldsymbol{s}, a_{\text{diff}}, \boldsymbol{u}) - g_S(\boldsymbol{s}', a_{\text{diff}}, \boldsymbol{u})\|^2 \leq (s_1 - s_1')^2 + (s_2 - s_2')^2 + 2|s_1 - s_1'||s_2 - s_2'|$. We also have that

$$\sqrt{2}\,\|\boldsymbol{s} - \boldsymbol{s}'\| = \sqrt{2}\sqrt{(s_1 - s_1')^2 + (s_2 - s_2')^2} \Rightarrow 2\,\|\boldsymbol{s} - \boldsymbol{s}'\|^2 = 2(s_1 - s_1')^2 + 2(s_2 - s_2')^2.$$

By combining these, we get

$$\begin{aligned}
&2\,\|\boldsymbol{s} - \boldsymbol{s}'\|^2 - \|g_S(\boldsymbol{s}, a_{\text{diff}}, \boldsymbol{u}) - g_S(\boldsymbol{s}', a_{\text{diff}}, \boldsymbol{u})\|^2 \\
&\qquad \geq (s_1 - s_1')^2 + (s_2 - s_2')^2 - 2|s_1 - s_1'||s_2 - s_2'| \Rightarrow \\
&2\,\|\boldsymbol{s} - \boldsymbol{s}'\|^2 - \|g_S(\boldsymbol{s}, a_{\text{diff}}, \boldsymbol{u}) - g_S(\boldsymbol{s}', a_{\text{diff}}, \boldsymbol{u})\|^2 \\
&\qquad \geq (|s_1 - s_1'| - |s_2 - s_2'|)^2 \geq 0.
\end{aligned}$$

Hence, we can easily see that $K_{a_{\text{diff}}, \boldsymbol{u}} = \sqrt{2}$ satisfies Definition 4.2.1.

Next, we need to show that, for all $a \in \mathcal{A}$ there exists a $C_a \in \mathbb{R}_+$ such that, for all $\boldsymbol{s}, \boldsymbol{s}' \in \mathcal{S}$, it holds $|r(\boldsymbol{s}, a) - r(\boldsymbol{s}', a)| \leq C_a \|\boldsymbol{s} - \boldsymbol{s}'\|$. Note that, to show that a function of the form $\max(0, f(\boldsymbol{s}))$ with $f : \mathbb{R}^2 \to \mathbb{R}$ is Lipschitz continuous, it suffices to show that $f(\boldsymbol{s})$ is Lipschitz continuous, since the function $\max(0, x)$ with $x \in \mathbb{R}$ has a Lipschitz constant equal to 1.

We start by showing that the function $f(\boldsymbol{s}) = s_1 - \alpha - s_2 \cdot \alpha$ is Lipschitz continuous, where $\alpha = sum(\mathcal{V})/2$ is a positive constant. For an arbitrary pair $\boldsymbol{s}, \boldsymbol{s}' \in \mathcal{S}$, we have that

$$\begin{aligned}
&|f(\boldsymbol{s}) - f(\boldsymbol{s}')| = |s_1 - s_1' - \alpha(s_2 - s_2')| \leq |s_1 - s_1'| + \alpha|s_2 - s_2'| \Rightarrow \\
&|f(\boldsymbol{s}) - f(\boldsymbol{s}')|^2 \leq (s_1 - s_1')^2 + (s_2 - s_2')^2 + 2\alpha|s_1 - s_1'||s_2 - s_2'|.
\end{aligned}$$

We also have that

$$\begin{aligned}
&\sqrt{1 + \alpha}\,\|\boldsymbol{s} - \boldsymbol{s}'\| = \sqrt{1 + \alpha}\sqrt{(s_1 - s_1')^2 + (s_2 - s_2')^2} \Rightarrow \\
&(1 + \alpha)\,\|\boldsymbol{s} - \boldsymbol{s}'\|^2 = (1 + \alpha)(s_1 - s_1')^2 + (1 + \alpha)(s_2 - s_2')^2
\end{aligned}$$

By combining these, we get

$$\begin{aligned}
&(1 + \alpha)\,\|\boldsymbol{s} - \boldsymbol{s}'\|^2 - |f(\boldsymbol{s}) - f(\boldsymbol{s}')|^2 \\
&\qquad \geq \alpha(s_1 - s_1')^2 + \alpha(s_2 - s_2')^2 - 2\alpha|s_1 - s_1'||s_2 - s_2'| \Rightarrow \\
&(1 + \alpha)\,\|\boldsymbol{s} - \boldsymbol{s}'\|^2 - |f(\boldsymbol{s}) - f(\boldsymbol{s}')|^2 \geq \alpha\,(|s_1 - s_1'| - |s_2 - s_2'|)^2 \geq 0.
\end{aligned}$$

Hence, we arrive to $|f(\boldsymbol{s}) - f(\boldsymbol{s}')| \leq \sqrt{1+\alpha}\,\|\boldsymbol{s} - \boldsymbol{s}'\|$, and the function f is Lipschitz continuous. It is easy to see that the function $\phi(\boldsymbol{s}) = \alpha - s_1 - s_2 \cdot \alpha$ is also Lipschitz continuous with the proof being almost identical. As a direct consequence, the reward function given in Eq. A.4 satisfies Definition 4.2.1 with $C_{a_{\text{null}}} = C_{a_{\text{diff}}} = 2\sqrt{1 + \frac{sum(\mathcal{V})}{2}}$. This concludes the proof of the lemma. □

Now, assume that the counterfactual action sequence can differ in an arbitrary number of actions from the action sequence in the observed episode τ, *i.e.*, $k = T$ and, let the observed action sequence be such that $a_t = a_{\text{null}}$ for $t \in \{0, \ldots, T-1\}$. Lastly, let the initial observed state be $\boldsymbol{s}_0 = [0, v_1]$, the observed states $\{\boldsymbol{s}_t\}_{t=1}^{T-2}$ be such that $\boldsymbol{s}_t = \left[\sum_{i=1}^{t} v_i, v_{t+1}\right]$ for $t \in \{1, \ldots, T-2\}$ and the last observed state be $\boldsymbol{s}_{T-1} = [sum(\mathcal{V}), 0]$. Then, it is easy to see that the noise variables $\boldsymbol{U}_t$ have posterior distributions with a point mass on the respective values

$$\boldsymbol{u}_t = \begin{bmatrix} v_{t+1} \\ v_{t+2} \end{bmatrix} \text{ for } t \in \{0, \ldots, T-3\} \qquad \text{and} \qquad \boldsymbol{u}_{T-2} = \begin{bmatrix} v_{T-1} \\ 0 \end{bmatrix}.$$

Note that, for all $t \in \{1, \ldots, T-2\}$, we have $0 \leq s_{t,1} < sum(\mathcal{V})$ and $s_{t,2} \geq 1$, hence the immediate reward according to Eq. A.4 is equal to 0. Consequently, the outcome of the observed episode τ is $o^+(\tau) = r(\boldsymbol{s}_{T-1}, a_{\text{null}}) = -\max(0, \frac{sum(\mathcal{V})}{2}) - \max(0, -\frac{sum(\mathcal{V})}{2}) = -\frac{sum(\mathcal{V})}{2}$.

Next, we will characterize the counterfactual outcome $o(\tau')$ of a counterfactual episode τ' with a sequence of states $\{\boldsymbol{s}'_t\}_{t=0}^{T-1}$ resulting from an alternative sequence of actions $\{a'_t\}_{t=0}^{T-1}$. Let $\mathcal{D}'_t$, $\mathcal{N}'_t$ denote the set of time steps until time t, where the actions taken in a counterfactual episode τ' are a_{diff} and a_{null} respectively. Formally, $\mathcal{D}'_t = \{t' \in \{0, \ldots, t\} : a'_{t'} = a_{\text{diff}}\}$, $\mathcal{N}'_t = \{t' \in \{0, \ldots, t\} : a'_{t'} = a_{\text{null}}\}$. Then, as an intermediate result, we get the following lemma.

Lemma A.4.2. *It holds that* $s'_{t,1} = \sum_{t' \in \mathcal{N}'_{t-1}} v_{t'+1}$ *for all* $t \in \{1, \ldots T-1\}$.

Proof. We will prove the lemma by induction. For the base case of $t = 1$, we distinguish between the cases $a'_0 = a_{\text{diff}}$ and $a'_0 = a_{\text{null}}$. In the first case, we have $s'_{1,1} = u_{0,1} + s_{0,1} - s_{0,2} = v_1 + 0 - v_1 = 0$ and $\mathcal{N}'_0 = \emptyset$ and, therefore, the statement holds. In the second case, we have $s'_{1,1} = u_{0,1} + s_{0,1} = v_1 + 0 = v_1$, $\mathcal{N}'_0 = \{0\}$ and $\sum_{t' \in \mathcal{N}'_0} v_{t'+1} = v_1$. Therefore, the statement also holds.

For the inductive step ($t > 1$), we assume that $s'_{t-1,1} = \sum_{t' \in \mathcal{N}'_{t-2}} v_{t'+1}$ and we will show that $s'_{t,1} = \sum_{t' \in \mathcal{N}'_{t-1}} v_{t'+1}$. Again, we distinguish between the cases $a'_{t-1} = a_{\text{diff}}$ and $a'_{t-1} = a_{\text{null}}$. However, note that, in both cases, $s'_{t-1,2} = u_{t-2,2} + 0 = v_t$. Therefore, in the case of $a'_{t-1} = a_{\text{diff}}$, we get

$$s'_{t,1} = u_{t-1,1} + s'_{t-1,1} - s'_{t-1,2} = v_t + \sum_{t' \in \mathcal{N}'_{t-2}} v_{t'+1} - v_t = \sum_{t' \in \mathcal{N}'_{t-2}} v_{t'+1} = \sum_{t' \in \mathcal{N}'_{t-1}} v_{t'+1},$$

where the last equation holds because $a'_{t-1} = a_{\text{diff}}$ and, therefore, $\mathcal{N}'_{t-1} = \mathcal{N}'_{t-2}$. In the case of $a'_{t-1} = a_{\text{null}}$, we get

$$s'_{t,1} = u_{t-1,1} + s'_{t-1,1} = v_t + \sum_{t' \in \mathcal{N}'_{t-2}} v_{t'+1} = \sum_{t' \in \mathcal{N}'_{t-1}} v_{t'+1},$$

where the last equation holds because $a'_{t-1} = a_{\text{null}}$ and, therefore, $\mathcal{N}'_{t-1} = \mathcal{N}'_{t-2} \cup \{t-1\}$. □

Following from that, we get that $0 \leq s'_{t,1} \leq sum(\mathcal{V})$ for all $t \in \{1, \ldots, T-1\}$. Moreover, we can observe that the transition mechanism given in Eq. A.3 is such that $g_{S,2}(\boldsymbol{S}_t, A_t, U_{t,2}) = U_{t,2}$ for all $t \in \{0, \ldots, T-2\}$, independently of $\boldsymbol{S}_T$ and A_t. Therefore, it holds that $s'_{t,2} = u_{t-1,2} \geq 1$ for $t \in \{1, \ldots, T-2\}$, and $s'_{0,2} = s_{0,2} = v_1 \geq 1$. As a direct consequence, it is easy to see that $r(\boldsymbol{s}'_t, a'_t) = 0$ for all $t \in \{0, \ldots, T-2\}$, and the counterfactual outcome is given by

$$o^+(\tau') = r(\boldsymbol{s}'_{T-1}, a'_{T-1}), \tag{A.5}$$

In addition to that, we have that $u_{T-2,2} = 0$, hence

$$\boldsymbol{s}'_{T-1} = \begin{bmatrix} \sum_{t \in \mathcal{N}'_{T-2}} v_{t+1} \\ 0 \end{bmatrix} \tag{A.6}$$

Now, we will show that, if we can find the action sequence $\{a^*_t\}_{t=0}^{T-1}$ that gives the optimal counterfactual outcome $o^+(\tau^*)$ for the aforementioned instance in polynomial time, then we can make a decision about the corresponding instance of the partition problem, also in polynomial time. To this end, let $\{\boldsymbol{s}^*_t\}_{t=0}^{T-1}$ be the sequence of states in the optimal counterfactual realization and, let $\mathcal{D}^*_{T-2} = \{t \in \{0, \ldots, T-2\} : a^*_t = a_{\text{diff}}\}$, $\mathcal{N}^*_{T-2} = \{t' \in \{0, \ldots, T-2\} : a^*_{t'} = a_{\text{null}}\}$.

From Eq. A.5, we get that the optimal counterfactual outcome is $o^+(\tau^*) = r(\boldsymbol{s}^*_{T-1}, a^*_{T-1})$, and it is easy to see that the reward function given in Eq. A.4 is always less or equal than zero. If $o(\tau^*) = 0$, it has to hold that

$$\max\left(0, s^*_{T-1,1} - \frac{sum(\mathcal{V})}{2} - s^*_{T-1,2}\frac{sum(\mathcal{V})}{2}\right) = \max\left(0, \frac{sum(\mathcal{V})}{2} - s^*_{T-1,1} - s^*_{T-1,2}\frac{sum(\mathcal{V})}{2}\right) = 0 \overset{(*)}{\Rightarrow}$$

$$\left(\sum_{t \in \mathcal{N}^*_{T-2}} v_{t+1}\right) - \frac{sum(\mathcal{V})}{2} \leq 0 \quad \text{and} \quad \frac{sum(\mathcal{V})}{2} - \left(\sum_{t \in \mathcal{N}^*_{T-2}} v_{t+1}\right) \leq 0 \Rightarrow$$

$$\sum_{t \in \mathcal{N}^*_{T-2}} v_{t+1} = \frac{sum(\mathcal{V})}{2},$$

where $(*)$ follows from Eq. A.6. As a consequence, the subsets $\mathcal{V}_1 = \{v_i : i-1 \in \mathcal{N}^*_{T-2}\}$ and $\mathcal{V}_2 = \{v_i : i-1 \in \mathcal{D}^*_{T-2}\}$ partition $\mathcal{V}$ and their sums are equal.

On the other hand, if $o^+(\tau^*) < 0$, as we will show, there is no partition of $\mathcal{V}$ into two sets with equal sums. For the sake of contradiction, assume there are two sets $\mathcal{V}_1, \mathcal{V}_2$ that partition $\mathcal{V}$, with $sum(\mathcal{V}_1) = sum(\mathcal{V}_2) = sum(\mathcal{V})/2$, and let $\mathcal{N}'_{T-2} = \{t \in \{0, \ldots, T-2\} : v_{t+1} \in \mathcal{V}_1\}$ and $\mathcal{D}'_{T-2} = \{t \in \{0, \ldots, T-2\} : v_{t+1} \in \mathcal{V}_2\}$. Then, consider the counterfactual episode τ' with an action sequence $\{a'_t\}_{t=0}^{T-1}$ such that its elements take values a_{null} and a_{diff} based on the sets $\mathcal{N}'_{T-2}, \mathcal{D}'_{T-2}$ respectively, with a'_{T-1} taking an arbitrary value. It is easy to see that

$$o^+(\tau') = r(\boldsymbol{s}'_{T-1}, a'_{T-1}) = r\left(\begin{bmatrix} \sum_{t \in \mathcal{N}'_{T-2}} v_{t+1} \\ 0 \end{bmatrix}, a'_{T-1}\right)$$

$$
\begin{aligned}
&= -\max\left(0, \sum_{t\in\mathcal{N}'_{T-2}} v_{t+1} - \frac{sum(\mathcal{V})}{2}\right) - \max\left(0, \frac{sum(\mathcal{V})}{2} - \sum_{t\in\mathcal{N}'_{T-2}} v_{t+1}\right)\\
&= -\max\left(0, \frac{sum(\mathcal{V})}{2} - \frac{sum(\mathcal{V})}{2}\right) - \max\left(0, \frac{sum(\mathcal{V})}{2} - \frac{sum(\mathcal{V})}{2}\right)\\
&= 0 > o^+(\tau^*),
\end{aligned}
$$

which is a contradiction. This step concludes the reduction and, therefore, the problem given in Eq. 4.19 cannot be solved in polynomial time, unless $P = NP$. $\square$

Lemma 4.2.1. *Let* $\boldsymbol{u}_t = g_S^{-1}(\boldsymbol{s}_t, a_t, \boldsymbol{s}_{t+1})$, $K_{\boldsymbol{u}_t} = \max_{a\in\mathcal{A}} K_{a,\boldsymbol{u}_t}$, $C = \max_{a\in\mathcal{A}} C_a$ *and the sequence* $L_0, \ldots, L_{T-1} \in \mathbb{R}_+$ *be such that* $L_{T-1} = C$ *and* $L_t = C + L_{t+1}K_{\boldsymbol{u}_t}$ *for* $t \in [T-2]$. *Then, it holds that* $|V_\tau(\boldsymbol{s}, l, t) - V_\tau(\boldsymbol{s}', l, t)| \leq L_t \|\boldsymbol{s} - \boldsymbol{s}'\|$, *for all* $t \in [T-1]_0$, $l \in [k]$ *and* $\boldsymbol{s}, \boldsymbol{s}' \in \mathcal{S}$.

Proof. We will prove the proposition by induction, starting from the base case, where $t = T-1$. First, Let $t = T-1$ and $l = k$. It is easy to see that, if the process is at a state $\boldsymbol{s} \in \mathcal{S}$ in the last time step with no action changes left, the best reward that can be achieved is $r(\boldsymbol{s}, a_{T-1})$, as already discussed after Eq. 4.20. Therefore, it holds that $|V_\tau(\boldsymbol{s}, k, T-1) - V_\tau(\boldsymbol{s}', k, T-1)| = |r(\boldsymbol{s}, a_{T-1}) - r(\boldsymbol{s}', a_{T-1})| \leq C_{a_{T-1}} \|\boldsymbol{s} - \boldsymbol{s}'\| \leq C \|\boldsymbol{s} - \boldsymbol{s}'\|$, where the last step holds because $C = \max_{a\in\mathcal{A}} C_a$. Now, consider the case of $t = T-1$ with l taking an arbitrary value in $\{0, \ldots, k-1\}$. Let $\boldsymbol{s}, \boldsymbol{s}'$ be two states in $\mathcal{S}$ and a^* be the action that gives the maximum immediate reward at state $\boldsymbol{s}$, that is, $a^* = \text{argmax}_{a\in\mathcal{A}}\{r(\boldsymbol{s}, a)\}$. Then, we get

$$
\begin{aligned}
|V_\tau(\boldsymbol{s}, l, T-1) - V_\tau(\boldsymbol{s}', l, T-1)| &= |\max_{a\in\mathcal{A}}\{r(\boldsymbol{s}, a)\} - \max_{a\in\mathcal{A}}\{r(\boldsymbol{s}', a)\}|\\
&\overset{(*)}{\leq} |r(\boldsymbol{s}, a^*) - r(\boldsymbol{s}', a^*)| \leq C_{a^*} \|\boldsymbol{s} - \boldsymbol{s}'\| \leq C \|\boldsymbol{s} - \boldsymbol{s}'\|,
\end{aligned}
$$

where $(*)$ follows from the fact that $r(\boldsymbol{s}', a^*) \leq \max_{a\in\mathcal{A}}\{r(\boldsymbol{s}', a)\}$. Therefore, for any $l \in \{0, \ldots, k\}$ and $\boldsymbol{s}, \boldsymbol{s}' \in \mathcal{S}$, it holds that $|V_\tau(\boldsymbol{s}, l, T-1) - V_\tau(\boldsymbol{s}', l, T-1)| \leq L_{T-1} \|\boldsymbol{s} - \boldsymbol{s}'\|$, where $L_{T-1} = C$.

Now, we will proceed to the induction step. Let $t < T-1$, $l < k$ and, as an inductive hypothesis, assume that $L_{t+1} \in \mathbb{R}_+$ as defined in Lemma 4.2.1 is such that, for all $l \in \{0, \ldots, k\}$ and $\boldsymbol{s}, \boldsymbol{s}' \in \mathcal{S}$, it holds that $|V_\tau(\boldsymbol{s}, l, t+1) - V_\tau(\boldsymbol{s}', l, t+1)| \leq L_{t+1} \|\boldsymbol{s} - \boldsymbol{s}'\|$. Additionally, let $(\boldsymbol{s}_a, l_a), (\boldsymbol{s}'_a, l_a)$ denote the enhanced states that follow from $(\boldsymbol{s}, l), (\boldsymbol{s}', l)$ after taking an action a, *i.e.*, $(\boldsymbol{s}_a, l_a) = F^+_{\tau,t}((\boldsymbol{s}, l), a)$ and $(\boldsymbol{s}'_a, l_a) = F^+_{\tau,t}((\boldsymbol{s}', l), a)$. Lastly, let a^* be the action that maximizes the future total reward starting from state $\boldsymbol{s}$, *i.e.*, $a^* = \text{argmax}_{a\in\mathcal{A}}\{r(\boldsymbol{s}, a) + V_\tau(\boldsymbol{s}_a, l_a, t+1)\}$. Then, we have that

$$
\begin{aligned}
&|V_\tau(\boldsymbol{s}, l, t) - V_\tau(\boldsymbol{s}', l, t)|\\
&\quad= |\max_{a\in\mathcal{A}}\{r(\boldsymbol{s}, a) + V_\tau(\boldsymbol{s}_a, l_a, t+1)\} - \max_{a\in\mathcal{A}}\{r(\boldsymbol{s}', a) + V_\tau(\boldsymbol{s}'_a, l_a, t+1)\}|\\
&\quad\overset{(*)}{\leq} |r(\boldsymbol{s}, a^*) + V_\tau(\boldsymbol{s}_{a^*}, l_{a^*}, t+1) - r(\boldsymbol{s}', a^*) - V_\tau(\boldsymbol{s}'_{a^*}, l_{a^*}, t+1)|\\
&\quad\leq |r(\boldsymbol{s}, a^*) - r(\boldsymbol{s}', a^*)| + |V_\tau(\boldsymbol{s}_{a^*}, l_{a^*}, t+1) - V_\tau(\boldsymbol{s}'_{a^*}, l_{a^*}, t+1)|
\end{aligned}
$$

$$
\begin{aligned}
&\overset{(**)}{\leq} C_{a^*} \left\| \boldsymbol{s} - \boldsymbol{s}' \right\| + L_{t+1} \left\| \boldsymbol{s}_{a^*} - \boldsymbol{s}'_{a^*} \right\| \\
&\leq C_{a^*} \left\| \boldsymbol{s} - \boldsymbol{s}' \right\| + L_{t+1} K_{a^*, \boldsymbol{u}_t} \left\| \boldsymbol{s} - \boldsymbol{s}' \right\| \\
&\overset{(***)}{\leq} C \left\| \boldsymbol{s} - \boldsymbol{s}' \right\| + L_{t+1} K_{\boldsymbol{u}_t} \left\| \boldsymbol{s} - \boldsymbol{s}' \right\| \\
&= (C + L_{t+1} K_{\boldsymbol{u}_t}) \left\| \boldsymbol{s} - \boldsymbol{s}' \right\| = L_t \left\| \boldsymbol{s} - \boldsymbol{s}' \right\|.
\end{aligned}
$$

In the above, $(*)$ holds due to $r(\boldsymbol{s}', a^*) + V_\tau\left(\boldsymbol{s}'_{a^*}, l_{a^*}, t+1\right) \leq \max_{a \in \mathcal{A}} \{r(\boldsymbol{s}', a) + V_\tau\left(\boldsymbol{s}'_a, l_a, t+1\right)\}$, $(**)$ follows from the inductive hypothesis, and $(***)$ holds because $C = \max_{a \in \mathcal{A}} C_a$ and $K_{\boldsymbol{u}_t} = \max_{a \in \mathcal{A}} K_{a, \boldsymbol{u}_t}$. It is easy to see that, similar arguments hold for the simple case of $l = k$, therefore, we omit the details. This concludes the inductive step and the proof of Lemma 4.2.1. □

Proposition 4.2.1. *For all $\boldsymbol{s} \in \mathcal{S}_\dagger$, $l \in [k], t \in [T-1]_0$, it holds that $\hat{V}_\tau(\boldsymbol{s}, l, t) \geq V_\tau(\boldsymbol{s}, l, t)$, where $\hat{V}_\tau(\boldsymbol{s}, l, t)$ are the values of the heuristic function computed by Algorithm 8.*

Proof. We will prove the proposition by induction, starting from the base case, where $t = T-1$. If $t = T-1$, the algorithm initializes $\hat{V}_\tau(\boldsymbol{s}, l, T-1)$ to $\max_{a \in \mathcal{A}} r(\boldsymbol{s}, a)$ for all $\boldsymbol{s} \in \mathcal{S}_\dagger$, $l \in \{0, \ldots, k-1\}$ and $\hat{V}_\tau(\boldsymbol{s}, k, T-1)$ to $r(\boldsymbol{s}, a_{T-1})$. It is easy to see that those values are optimal, as already discussed after Eq. 4.20. Therefore, the base case $\hat{V}_\tau(\boldsymbol{s}, l, T-1) \geq V_\tau(\boldsymbol{s}, l, T-1)$ follows trivially.

Now, we will proceed to the induction step. Let $t < T-1$ and, as an inductive hypothesis, assume that $\hat{V}_\tau(\boldsymbol{s}, l, t+1) \geq V_\tau(\boldsymbol{s}, l, t+1)$ for all $\boldsymbol{s} \in \mathcal{S}_\dagger$, $l \in \{0, \ldots, k\}$. Our goal is to show that $\hat{V}_\tau(\boldsymbol{s}, l, t) \geq V_\tau(\boldsymbol{s}, l, t)$ for all $\boldsymbol{s} \in \mathcal{S}_\dagger$, $l \in \{0, \ldots, k\}$. First, let $l < k$. For a given point $\boldsymbol{s} \in \mathcal{S}_\dagger$, Algorithm 8 finds the next state $\boldsymbol{s}_a$ that would have occurred by taking each action a, *i.e.*, $(\boldsymbol{s}_a, l_a) = F^+_{\tau,t}\left((\boldsymbol{s}, l), a\right)$, and it computes the associated value $V_a = \min_{\boldsymbol{s}_\dagger \in \mathcal{S}_\dagger} \{\hat{V}_\tau(\boldsymbol{s}_\dagger, l_a, t+1) + L_{t+1} \left\| \boldsymbol{s}_\dagger - \boldsymbol{s}_a \right\|\}$. Then, it simply sets $\hat{V}_\tau(\boldsymbol{s}, l, t)$ equal to $\max_{a \in \mathcal{A}} \{r(\boldsymbol{s}, a) + V_a\}$. We have that

$$
\begin{aligned}
V_a &= \min_{\boldsymbol{s}_\dagger \in \mathcal{S}_\dagger} \{\hat{V}_\tau(\boldsymbol{s}_\dagger, l_a, t+1) + L_{t+1} \left\| \boldsymbol{s}_\dagger - \boldsymbol{s}_a \right\|\} \\
&\overset{(*)}{\geq} \min_{\boldsymbol{s}_\dagger \in \mathcal{S}_\dagger} \{V_\tau(\boldsymbol{s}_\dagger, l_a, t+1) + L_{t+1} \left\| \boldsymbol{s}_\dagger - \boldsymbol{s}_a \right\|\} \\
&\overset{(**)}{\geq} \min_{\boldsymbol{s}_\dagger \in \mathcal{S}_\dagger} \{V_\tau(\boldsymbol{s}_a, l_a, t+1)\} \\
&= V_\tau(\boldsymbol{s}_a, l_a, t+1),
\end{aligned}
$$

where $(*)$ follows from the inductive hypothesis, and $(**)$ is a consequence of Lemma 4.2.1. Then, we get

$$
\hat{V}_\tau(\boldsymbol{s}, l, t) = \max_{a \in \mathcal{A}} \{r(\boldsymbol{s}, a) + V_a\} \geq \max_{a \in \mathcal{A}} \{r(\boldsymbol{s}, a) + V_\tau(\boldsymbol{s}_a, l_a, t+1)\} = V_\tau(\boldsymbol{s}, l, t).
$$

Additionally, when $l = k$, we have $\hat{V}_\tau(\boldsymbol{s}, k, t) = r(\boldsymbol{s}, a_t) + \min_{\boldsymbol{s}_\dagger \in \mathcal{S}_\dagger} \{\hat{V}_\tau(\boldsymbol{s}_\dagger, k, t+1) + L_{t+1} \left\| \boldsymbol{s}_\dagger - \boldsymbol{s}_{a_t} \right\|\}$ and $V_\tau(\boldsymbol{s}, k, t) = r(\boldsymbol{s}, a_t) + V_\tau(\boldsymbol{s}_{a_t}, k, t+1)$. Therefore, the proof for $\hat{V}_\tau(\boldsymbol{s}, k, t) \geq V_\tau(\boldsymbol{s}, k, t)$ is almost identical. □

Theorem 4.2.3. *For any nodes $v = (\boldsymbol{s}, l, t), v_a = (\boldsymbol{s}_a, l_a, t+1)$ with $t < T-1$ connected with an edge associated with action a, it holds that $\hat{V}_\tau(\boldsymbol{s}, l, t) \geq r(\boldsymbol{s}, a) + \hat{V}_\tau(\boldsymbol{s}_a, l_a, t+1)$. Moreover, for any node $v = (\boldsymbol{s}, l, T-1)$ and edge connecting it to the goal node $v_T = (\boldsymbol{s}_\emptyset, k, T)$, it holds that $\hat{V}_\tau(\boldsymbol{s}, l, T-1) \geq r(\boldsymbol{s}, a) + \hat{V}_\tau(\boldsymbol{s}_\emptyset, k, T)$.*

Proof. We start from the case where $t = T - 1$. Let $v = (\boldsymbol{s}, l, T-1)$ and, consider an edge associated with action a^* connecting v to the goal node $v_T = (\boldsymbol{s}_\emptyset, k, T)$ that carries a reward $r(\boldsymbol{s}, a^*)$. Then, we have

$$\hat{V}_\tau(\boldsymbol{s}, l, T-1) = \max_{a \in \mathcal{A}'} r(\boldsymbol{s}, a) \geq r(\boldsymbol{s}, a^*) + 0 = r(\boldsymbol{s}, a^*) + \hat{V}_\tau(\boldsymbol{s}_\emptyset, k, T),$$

and the base case holds.

For the more general case, where $t < T - 1$, we first establish the following intermediate result:

Lemma A.4.3. *For every $\boldsymbol{s}, \boldsymbol{s}' \in \mathcal{S}$, $l \in \{0, \ldots, k\}$, $t \in \{0, \ldots, T-1\}$, it holds that $|\hat{V}_\tau(\boldsymbol{s}, l, t) - \hat{V}_\tau(\boldsymbol{s}', l, t)| \leq L_t \|\boldsymbol{s} - \boldsymbol{s}'\|$, where L_t is as defined in Lemma 4.2.1.*

Proof. Without loss of generality, we will assume that $l < k$, since the proof for the case of $l = k$ is similar and more straightforward. We start from the case where $t = T - 1$ and, for two states $\boldsymbol{s}, \boldsymbol{s}' \in \mathcal{S}$ we have

$$\begin{aligned}|\hat{V}_\tau(\boldsymbol{s}, l, T-1) - \hat{V}_\tau(\boldsymbol{s}', l, T-1)| &= \left|\max_{a \in \mathcal{A}} r(\boldsymbol{s}, a) - \max_{a \in \mathcal{A}} r(\boldsymbol{s}', a)\right| \\ &= |V_\tau(\boldsymbol{s}, l, T-1) - V_\tau(\boldsymbol{s}', l, T-1)| \leq C \|\boldsymbol{s} - \boldsymbol{s}'\| = L_{T-1} \|\boldsymbol{s} - \boldsymbol{s}'\|,\end{aligned}$$

where the last inequality follows from Lemma 4.2.1.

Now, consider the case $t < T-1$, and let $(\boldsymbol{s}_a, l_a)$ denote the enhanced state that follows from $(\boldsymbol{s}, l)$ after taking an action a at time t, *i.e.*, $(\boldsymbol{s}_a, l_a) = F^+_{\tau,t}((\boldsymbol{s}, l), a)$. Then, we have

$$\begin{aligned}&|\hat{V}_\tau(\boldsymbol{s}, l, t) - \hat{V}_\tau(\boldsymbol{s}', l, t)| \\ &\quad= \left| \max_{a \in \mathcal{A}} \left\{ r(\boldsymbol{s}, a) + \min_{\boldsymbol{s}_\dagger \in \mathcal{S}_\dagger} \left\{ \hat{V}_\tau(\boldsymbol{s}_\dagger, l_a, t+1) + L_{t+1} \|\boldsymbol{s}_\dagger - \boldsymbol{s}_a\| \right\} \right\} \right. \\ &\qquad \left. - \max_{a \in \mathcal{A}} \left\{ r(\boldsymbol{s}', a) + \min_{\boldsymbol{s}_\dagger \in \mathcal{S}_\dagger} \left\{ \hat{V}_\tau(\boldsymbol{s}_\dagger, l_a, t+1) + L_{t+1} \|\boldsymbol{s}_\dagger - \boldsymbol{s}'_a\| \right\} \right\} \right|. \quad \text{(A.7)}\end{aligned}$$

Let a^* be the action $a \in \mathcal{A}$ that maximizes the first part of the above subtraction, *i.e.*,

$$a^* = \operatorname*{argmax}_{a \in \mathcal{A}} \left\{ r(\boldsymbol{s}, a) + \min_{\boldsymbol{s}_\dagger \in \mathcal{S}_\dagger} \left\{ \hat{V}_\tau(\boldsymbol{s}_\dagger, l_a, t+1) + L_{t+1} \|\boldsymbol{s}_\dagger - \boldsymbol{s}_a\| \right\} \right\}$$

Then, Eq. A.7 implies that

$$\begin{aligned}|\hat{V}_\tau(\boldsymbol{s}, l, t) - \hat{V}_\tau(\boldsymbol{s}', l, t)| &\leq \left| r(\boldsymbol{s}, a^*) + \min_{\boldsymbol{s}_\dagger \in \mathcal{S}_\dagger} \left\{ \hat{V}_\tau(\boldsymbol{s}_\dagger, l_{a^*}, t+1) + L_{t+1} \|\boldsymbol{s}_\dagger - \boldsymbol{s}_{a^*}\| \right\} \right. \\ &\qquad \left. - r(\boldsymbol{s}', a^*) - \min_{\boldsymbol{s}_\dagger \in \mathcal{S}_\dagger} \left\{ \hat{V}_\tau(\boldsymbol{s}_\dagger, l_{a^*}, t+1) + L_{t+1} \|\boldsymbol{s}_\dagger - \boldsymbol{s}'_{a^*}\| \right\} \right| \\ &\leq |r(\boldsymbol{s}, a^*) - r(\boldsymbol{s}', a^*)| \\ &\qquad + \left| \min_{\boldsymbol{s}_\dagger \in \mathcal{S}_\dagger} \left\{ \hat{V}_\tau(\boldsymbol{s}_\dagger, l_{a^*}, t+1) + L_{t+1} \|\boldsymbol{s}_\dagger - \boldsymbol{s}_{a^*}\| \right\} \right. \\ &\qquad\quad \left. - \min_{\boldsymbol{s}_\dagger \in \mathcal{S}_\dagger} \left\{ \hat{V}_\tau(\boldsymbol{s}_\dagger, l_{a^*}, t+1) + L_{t+1} \|\boldsymbol{s}_\dagger - \boldsymbol{s}'_{a^*}\| \right\} \right| \quad \text{(A.8)}\end{aligned}$$

Now, let $\tilde{\boldsymbol{s}}$ be the $\boldsymbol{s}_\dagger \in \mathcal{S}_\dagger$ that minimizes the second part of the above subtraction, *i.e.*,

$$\tilde{\boldsymbol{s}} = \underset{\boldsymbol{s}_\dagger \in \mathcal{S}_\dagger}{\operatorname{argmin}} \left\{ \hat{V}_\tau(\boldsymbol{s}_\dagger, l_{a^*}, t+1) + L_{t+1} \left\| \boldsymbol{s}_\dagger - \boldsymbol{s}'_{a^*} \right\| \right\}.$$

As a consequence and in combination with Eq. A.8, we get

$$\begin{aligned}
|\hat{V}_\tau(\boldsymbol{s}, l, t) - \hat{V}_\tau(\boldsymbol{s}', l, t)| &\leq |r(\boldsymbol{s}, a^*) - r(\boldsymbol{s}', a^*)| \\
&\quad + \Big| \hat{V}_\tau(\tilde{\boldsymbol{s}}, l_{a^*}, t+1) + L_{t+1} \left\| \tilde{\boldsymbol{s}} - \boldsymbol{s}_{a^*} \right\| \\
&\quad\quad - \hat{V}_\tau(\tilde{\boldsymbol{s}}, l_{a^*}, t+1) + L_{t+1} \left\| \tilde{\boldsymbol{s}} - \boldsymbol{s}'_{a^*} \right\| \Big| \\
&= |r(\boldsymbol{s}, a^*) - r(\boldsymbol{s}', a^*)| + L_{t+1} \left| \left\| \tilde{\boldsymbol{s}} - \boldsymbol{s}_{a^*} \right\| - \left\| \tilde{\boldsymbol{s}} - \boldsymbol{s}'_{a^*} \right\| \right| \\
&\overset{(*)}{\leq} C_{a^*} \left\| \boldsymbol{s} - \boldsymbol{s}' \right\| + L_{t+1} \left\| \boldsymbol{s}_{a^*} - \boldsymbol{s}'_{a^*} \right\| \\
&\overset{(**)}{\leq} C \left\| \boldsymbol{s} - \boldsymbol{s}' \right\| + L_{t+1} K_{\boldsymbol{u}_t} \left\| \boldsymbol{s} - \boldsymbol{s}' \right\| = L_t \left\| \boldsymbol{s} - \boldsymbol{s}' \right\|,
\end{aligned}$$

where in $(*)$ we use the triangle inequality and the fact that the SCM $\mathcal{C}$ is Lipschitz-continuous, and $(**)$ follows from Lemma 4.2.1. □

That said, consider an edge associated with an action a^* connecting node $v = (\boldsymbol{s}, l, t)$ to node $v_{a^*} = (\boldsymbol{s}_{a^*}, l_{a^*}, t+1)$. Then, we have

$$\begin{aligned}
\hat{V}_\tau(\boldsymbol{s}, l, t) &= \max_{a \in \mathcal{A}'} \left\{ r(\boldsymbol{s}, a) + \min_{\boldsymbol{s}_\dagger \in \mathcal{S}_\dagger} \left\{ \hat{V}_\tau(\boldsymbol{s}_\dagger, l_a, t+1) + L_{t+1} \left\| \boldsymbol{s}_\dagger - \boldsymbol{s}_a \right\| \right\} \right\} \\
&\geq r(\boldsymbol{s}, a^*) + \min_{\boldsymbol{s}_\dagger \in \mathcal{S}_\dagger} \left\{ \hat{V}_\tau(\boldsymbol{s}_\dagger, l_{a^*}, t+1) + L_{t+1} \left\| \boldsymbol{s}_\dagger - \boldsymbol{s}_{a^*} \right\| \right\} \\
&\geq r(\boldsymbol{s}, a^*) + \min_{\boldsymbol{s}_\dagger \in \mathcal{S}_\dagger} \left\{ \hat{V}_\tau(\boldsymbol{s}_{a^*}, l_{a^*}, t+1) \right\} \\
&= r(\boldsymbol{s}, a^*) + \hat{V}_\tau(\boldsymbol{s}_{a^*}, l_{a^*}, t+1).
\end{aligned}$$

That concludes the proof and, therefore, the heuristic function $\hat{V}_\tau$ is consistent.

□

Appendix B

Omitted details of the experimental setups

B.1 Additional details for Section 3.1

B.1.1 Raw features in the credit dataset

Each credit card holder has a label which indicates whether they will default during the next month ($Y = 0$) or not ($y = 1$) and the features $\boldsymbol{x}$ are:

- Marital status: whether the person is married or single.
- Age group: group depending on the person's age ($< 25, 25-39, 40-59, > 60$).
- Education level: the level of education the individual has acquired (1-4).
- Maximum bill amount over last 6 months
- Maximum payment amount over last 6 Months
- Months with zero balance over last 6 Months
- Months with low spending over last 6 Months
- Months with high spending over last 6 Months
- Most recent bill amount
- Most recent payment amount
- Total overdue counts
- Total months overdue

We consider all features except marital status, age group and education level to be actionable and, among the actionable features, we assume that total overdue counts and total months overdue can only increase.

B.1.2 Information about the processed data and the trained classifier

Table B.1 summarizes the experimental setup for the credit card dataset, that is, the number of samples, the pair of classifier - number of clusters k picked through cross-validation, the accuracy achieved by the corresponding classifier, the resulting number of feature values m and the parameter γ.

B.1.3 Details regarding modeling unobserved confounding

We set the parameters $\alpha(\lambda)$ and $\beta(\lambda)$ as follows:

$$\alpha(\lambda) = \begin{cases} 1.5 & \text{if } \lambda \in (0, 0.5] \\ 0.1 \cdot (2\lambda - 1) + 1.5 \cdot (2 - 2\lambda) & \text{if } \lambda \in (0.5, 1) \end{cases}$$

$$\beta(\lambda) = \begin{cases} 0.1 \cdot (1 - 2\lambda) + 1.5 \cdot 2\lambda & \text{if } \lambda \in (0, 0.5] \\ 1.5 & \text{if } \lambda \in (0.5, 1) \end{cases}$$

Table B.1: Credit: dataset and classifier details

Dataset	# of samples	Classifier	k	Accuracy	m	γ
credit	30000	Logistic Regression	100	80.49%	3200	0.85

B.2 Additional details for Section 3.2

For each applicant in the lending dataset, the label Y indicates whether an applicant fully pays a loan ($Y = 1$) or ends up to a default/charge-off ($Y = 0$) and the features $\boldsymbol{X}$ are:

- Loan Amount: The amount that the applicant initially requested.
- Employment Length: How long the applicant has been employed.
- Debt to Income Ratio: The ratio between the applicant's financial debts and their average income.
- FICO Score: The applicant's FICO score, which is a credit score based on consumer credit files. The FICO scores are in the range of 300-850 and the average of the high and low range for the FICO score of each applicant has been used for this study.
- Annual Income: The declared annual income of the applicant.

Here, we assume that all of the aforementioned features are *actionable*, meaning that an individual denied a loan can change their values in order to get a positive decision. Note that the actionable features are numerical, however, our methodology only allows for discrete valued features. Therefore, to discretize the features $\boldsymbol{X}$, estimate the conditional distribution $P(Y \mid \boldsymbol{X})$, we follow the same procedure as in Section 3.1. Finally, we set γ equal to the 50-th percentile of all the individuals' $P(Y = 1 \mid \boldsymbol{x})$ values causing a 50% acceptance rate by the optimal threshold policy in the non strategic setting. Table B.2 summarizes the experimental setup for the lending dataset, that is, the number of samples, the pair of classifier - number of clusters picked through cross-validation, the accuracy achieved by the corresponding classifier, the resulting number of feature values m and the parameter γ.

Table B.2: Lending: dataset and classifier details

Dataset	# of samples	Classifier	k	Accuracy	m	γ
lending	1266817	Logistic Regression	400	89.9%	400	0.97

B.3 Additional details for Section 4.1

Each patient's severity of depression is measured using the standardized questionnaire PHQ-9 [209], which consists of 9 questions regarding the frequency of depressive symptoms (*e.g.*, "Feeling tired or having little energy?") manifested over a period of two weeks. The patient has to answer each question by placing themselves on a scale ranging from 0 ("Not at all") to 3 ("Nearly every day"). The sum of those answers, ranging from 0 to 27, reflects the overall depression severity and it is usually discretized into five categories, corresponding to no depression $(0-4)$, mild depression $(5-9)$, moderate depression $(10-14)$, moderately severe depression $(15-19)$, severe depression $(20-27)$[1]. In our experiments, the states $\mathcal{S} = \{0, \ldots, 4\}$ correspond to these five categories.

Each session of cognitive behavioral therapy contains information about the topic of discussion between the patient and the therapist, among 24 pre-defined topics [207], with some of the topics having similar content. For example, there were 4 topics about "cognitive restructuring techniques" which, we observed that, some therapists merged and covered in 2 sessions. Here, we grouped the above topics into the following eleven broader themes:

- STR – First session: Introduction, discussing expectations, getting to know each other, discussing the current symptoms / problems, current life situation.
- BIO – Biography: A look at biography, family and social frame of reference, school and professional development, emotional development, partnerships, important turning points or crises.
- PSE – Psychoeducation: Discuss symptoms of depression, recognize and understand connections between feelings, thoughts and behavior (depression triangle) based on a situation analysis from the current / last episode, causes of depression, develop a disease model, explain the treatment approach in relation to the model.
- BHA – Behavioural activation: Focus on behaviour, discuss the vicious circle (depression spiral), discuss list of pleasant activities, attention to life balance, if necessary improve the daily structure, recognizing and eliminating obstacles and problems.
- REV – Review: Review of the last sessions, collection of strategies learned so far, find suitable strategies for typical situations, draft a personal strategy plan, plan further steps.
- CRT – Cognitive restructuring techniques: Discuss influence of thoughts on feelings and actions, identify thought patterns, discuss influence of automatic thoughts / basic assumptions, check the validity of automatic thoughts.
- INR – Interactional competence: Self-assessment of your own self-confidence, discuss current interpersonal issues and derive goals, carry out role plays, transfer into everyday life.

[1]The full version of the questionnaire can be found at https://patient.info/doctor/patient-health-questionnaire-phq-9.

- THP – Re-evaluation of thought patterns: Review, evaluate and rename basic assumptions, schemes and general plans.
- RLP – Relapse prevention: Explain the risk of relapse, discuss early warning symptoms, recognize risk situations, develop suitable strategies.
- END – Closing session: Finding a good end to the therapy, looking back on the last 5-6 months, parting ritual.
- EXT – Extra material: Sleep disorders, problem-solving skills, brooding module "When thinking doesn't help", discuss the influence of rumination on mood and impairments in everyday life, progressive muscle relaxation.

In our experiments, the actions $\mathcal{A}$ correspond to these broader themes. However, since the themes STR and END appeared only in the first ($t = 0$) and last ($t = T-1$) time steps of each realization, we kept them fixed and we did not allow these themes to be used as action changes during the time steps $t = \{1, \ldots, T-2\}$.

B.4 Additional details for Section 4.2

B.4.1 Features and actions in the sepsis management dataset

As mentioned in Section 4.2.4, our state space is $\mathcal{S} = \mathbb{R}^d$, where $d = 13$ is the number of features. We distinguish between three types of features: (i) demographic features, whose values remain constant across time, (ii) contextual features, for which we maintain their observed (and potentially varying) values throughout all counterfactual episodes and, (iii) time-varying features, whose counterfactual values are given by the SCM $\mathcal{C}$. The list of features is as follows:

- Gender (demographic)
- Re-admission (demographic)
- Age (demographic)
- Mechanical ventilation (contextual)
- FiO_2 (time-varying)
- PaO_2 (time-varying)
- Platelet count (time-varying)
- Bilirubin (time-varying)
- Glasgow Coma Scale (time-varying)
- Mean arterial blood pressure (time-varying)
- Creatinine (time-varying)
- Urine output (time-varying)
- SOFA score (time-varying)

To define our set of actions $\mathcal{A}$ we follow related work [233, 234, 236], and we consider 25 actions corresponding to 5×5 levels of administered vasopressors and intravenous fluids. Specifically, for both vasopressors and fluids, we find all non-zero values appearing in the data, and we divide them into 4 intervals based on the

Table B.3: Levels of vasopressors and intravenous fluids corresponding to the 25 actions in $\mathcal{A}$

Vasopressors (mcg/kg/min)	**Intravenous fluids (mL/4 hours)**
0.00	0
0.04	30
0.113	80
0.225	279
0.788	850

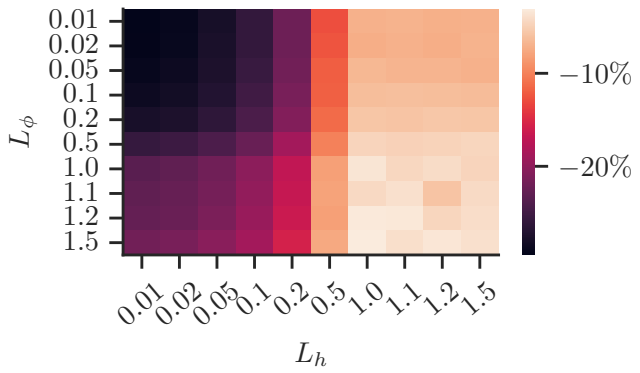


Figure B.1: **Goodness of fit of the Lipschitz-continuous SCM $\mathcal{C}$.** The heatmap shows the percentage decrease in log likelihood of the data in comparison with an SCM trained without Lipschitz-continuity constraints. The x and y axes correspond to different enforced values for the Lipschitz constants L_h, L_ϕ of the location and scale networks h and ϕ, respectively. Darker values indicate that the learned SCM achieves a significantly lower log likelihood than the unconstrained SCM.

quartiles of the observed values. Then, we set the 5 levels to be the median values of the 4 intervals and 0. Table B.3 shows the resulting values of vasopressors and fluids.

B.4.2 Additional details on the network architecture & training

We represent the location and scale functions h and ϕ of the SCM $\mathcal{C}$ using neural networks with 1 hidden layer, 200 hidden units and *tanh* activation functions. The mapping from a state $\boldsymbol{s}$ and an action a to the hidden vector $\boldsymbol{z}$ takes the form $\boldsymbol{z} = tanh(W_s\boldsymbol{s} + W_a\boldsymbol{a})$, where $\boldsymbol{a}$ is a 2-D vector representation of the respective action. The mapping from the hidden vector $\boldsymbol{z}$ to the network's output is done via a fully connected layer with weights W_z. To enforce a network to have a Lipschitz constant L with respect to the state input, we apply spectral normalization to the weight matrices W_s and W_z, so that their spectral norms are $\|W_s\|_2 = \|W_z\|_2 = 1$. Additionally, we add 2 intermediate layers between the input and the hidden layer and between the hidden layer and the output layer, each one multiplying its respective input by a constant $\sqrt{L}$. Since it is known that the *tanh* activation function has a Lipschitz constant of 1, it is easy to see that, by function composition, the resulting network is guaranteed to be Lipschitz continuous with respect to its state input with constant L. Note that, since the matrix W_a is not normalized, the network's Lipschitz constant with respect to the action input can be arbitrary.

To train the SCM $\mathcal{C}$, for each sample, we compute the negative log-likelihood of the observed transition under the SCM's current parameter values (*i.e.*, network weight matrices & covariance matrix of the multivariate Gaussian prior), and we use that as a loss. Subsequently, we optimize those parameters using the Adam optimizer with a learning rate of 0.001, a batch size of 256, and we train the model for 100 epochs.

We train the model under multiple values of the Lipschitz constants L_h, L_ϕ of

the location and scale networks, and we evaluate the log-likelihood of the data under each model using 5-fold cross-validation. Specifically, for each configuration of L_h and L_ϕ, we randomly split the dataset into a training and a validation set (with a size ratio 4-to-1), we train the corresponding SCM using the training set, and we evaluate the log-likelihood of the validation set based on the trained SCM. This results in the log-likelihood always being measured on a different set of data points than the one used for training. For each configuration of L_h and L_ϕ, we repeat the aforementioned procedure 5 times and we report the average log-likelihood achieved on the validation set. In addition, we train an unconstrained model without spectral normalization, which can have an arbitrary Lipschitz constant.

Fig. B.1 shows the decrease in log-likelihood of the respective constrained model as a percentage of the log-likelihood achieved by the unconstrained model, under various values of the Lipschitz constants L_h, L_ϕ. We observe that, the model's performance is predominantly affected by the Lipschitz constant of the location network L_h, and its effect is more pronounced when L_h takes values smaller than 1. Additionally, we can see that the scale network's Lipschitz constant L_ϕ has a milder effect on performance, especially when L_h is greater or equal than 1. Since we are interested in constraining the overall Lipschitz constant of the SCM $\mathcal{C}$, in our experiments in Section 4.2.4, we set $L_h = 1$ and $L_\phi = 0.1$, which achieves a log-likelihood only 6% lower to that of the best model trained without any Lipschitz constraint.

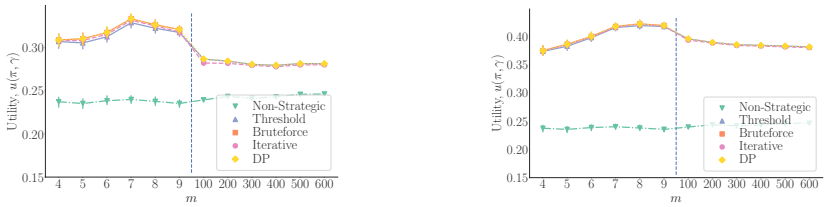
Appendix C

Additional experimental results

C.1 Section 3.1: Additional results on synthetic data

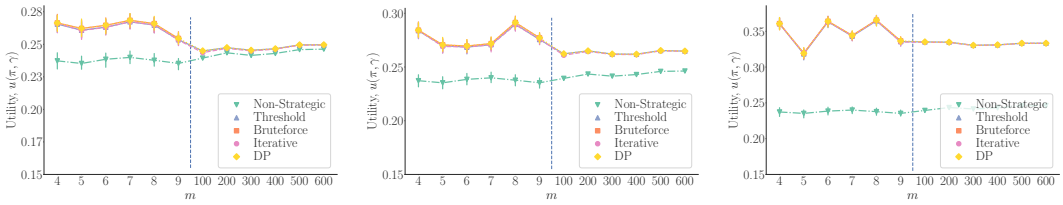
Figs. C.1-C.4 present experimental results similar to the ones presented in Fig. 3.3 under additional cost functions and values of the parameter κ.

We consider the following additional cost functions. For instances with outcome monotonic additive costs, we initially set $c(\boldsymbol{x}_i, \boldsymbol{x}_j) = 0 \; \forall \boldsymbol{x}_i, \boldsymbol{x}_j : P(Y = 1 \,|\, \boldsymbol{x}_j) \leq P(Y = 1 \,|\, \boldsymbol{x}_i)$. Then, we assign to the costs $\{c(\boldsymbol{x}_m, \boldsymbol{x}_i)$ for $i \in \{1, \ldots, m-1\}\}$ $m-1$ samples from an exponential distribution $\text{Exp}(\lambda = 1)$, sorted in increasing order, and we scale them such that individuals with feature values $\boldsymbol{x}_m$ can move to at most κm *better* states, that is, $c(\boldsymbol{x}_m, \boldsymbol{x}_i) \leq 1 \; \forall \boldsymbol{x}_i : i \geq m - \kappa m$. Finally, we set the remaining values $c(\boldsymbol{x}_i, \boldsymbol{x}_j)$, in decreasing order of i and j such that $c(\boldsymbol{x}_i, \boldsymbol{x}_j) = c(\boldsymbol{x}_{i-1}, \boldsymbol{x}_j) - c(\boldsymbol{x}_{i-1}, \boldsymbol{x}_i)$. For instances with general costs, we sample the cost between feature values $c(\boldsymbol{x}_i, \boldsymbol{x}_j) \sim \text{Exp}(\lambda = 1)$ for a fraction κ of all pairs, we scale them such that the maximum cost is equal to 1 and, we set $c(\boldsymbol{x}_i, \boldsymbol{x}_j) = \infty$ for the remaining pairs.



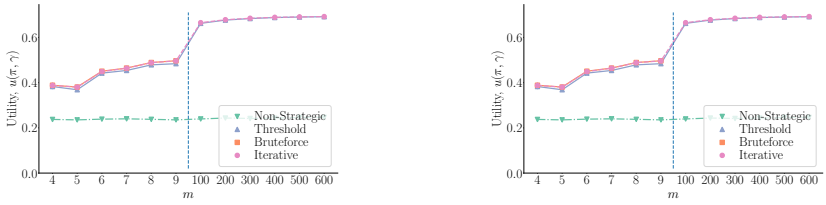
(a) $\kappa = 0.25$ (b) $\kappa = 0.5$

Figure C.1: Performance evaluation on synthetic data with additive outcome monotonic costs using the cost function defined in Section 3.1.5.



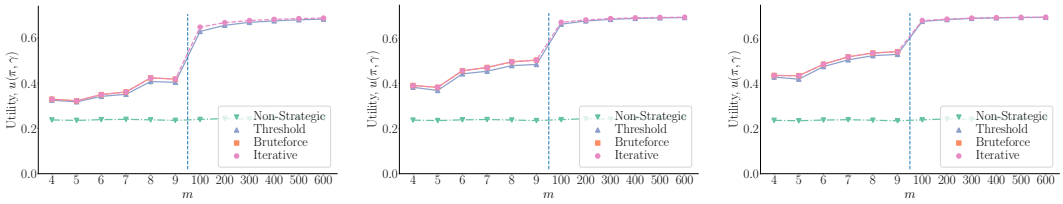
(a) $\kappa = 0.1$ (b) $\kappa = 0.25$ (c) $\kappa = 0.5$

Figure C.2: Performance evaluation on synthetic data with additive outcome monotonic costs using the additional cost function defined in Appendix C.1.



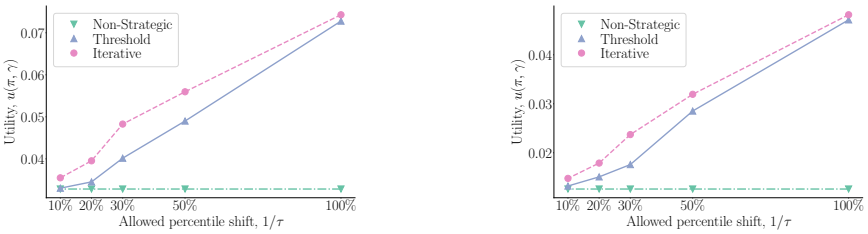
(a) $\kappa = 0.5$ (b) $\kappa = 0.75$

Figure C.3: Performance evaluation on synthetic data with general costs using the cost function defined in Section 3.1.5.



(a) $\kappa = 0.25$ (b) $\kappa = 0.5$ (c) $\kappa = 0.75$

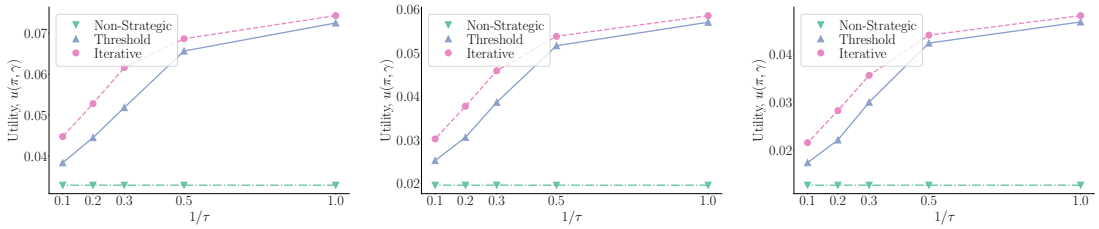
Figure C.4: Performance evaluation on synthetic data with general costs using the additional cost function defined in Appendix C.1.



(a) $\gamma = 40$-th percentile of $\{P(Y = 1 \mid \boldsymbol{x})\}$

(b) $\gamma = 60$-th percentile of $\{P(Y = 1 \mid \boldsymbol{x})\}$

Figure C.5: Performance evaluation on credit card data using the cost function proportional to the *maximum percentile shift* defined in Section 3.1.6.



(a) $\gamma = 40$-th percentile of $\{P(Y = 1 \mid \boldsymbol{x})\}$

(b) $\gamma = 50$-th percentile of $\{P(Y = 1 \mid \boldsymbol{x})\}$

(c) $\gamma = 60$-th percentile of $\{P(Y = 1 \mid \boldsymbol{x})\}$

Figure C.6: Performance evaluation on credit card data using the cost function proportional to the euclidean distance defined in Appendix C.2.

C.2 Section 3.1: Additional results on real data

Figs. C.5-C.6 present experimental results similar to the ones presented in Fig. 3.6 under additional values of the parameter γ and one additional cost function. Here, the additional cost function $c(\boldsymbol{x}_i, \boldsymbol{x}_j)$ we consider is proportional to the euclidean distance between the feature values $\boldsymbol{x}_i, \boldsymbol{x}_j$. More specifically, let $\mathcal{L}$ be the set of actionable (numerical) features and $\bar{\mathcal{L}}$ be the set of non-actionable (discrete-valued) features. Then, for each pair of feature values, we define an intermediate cost function

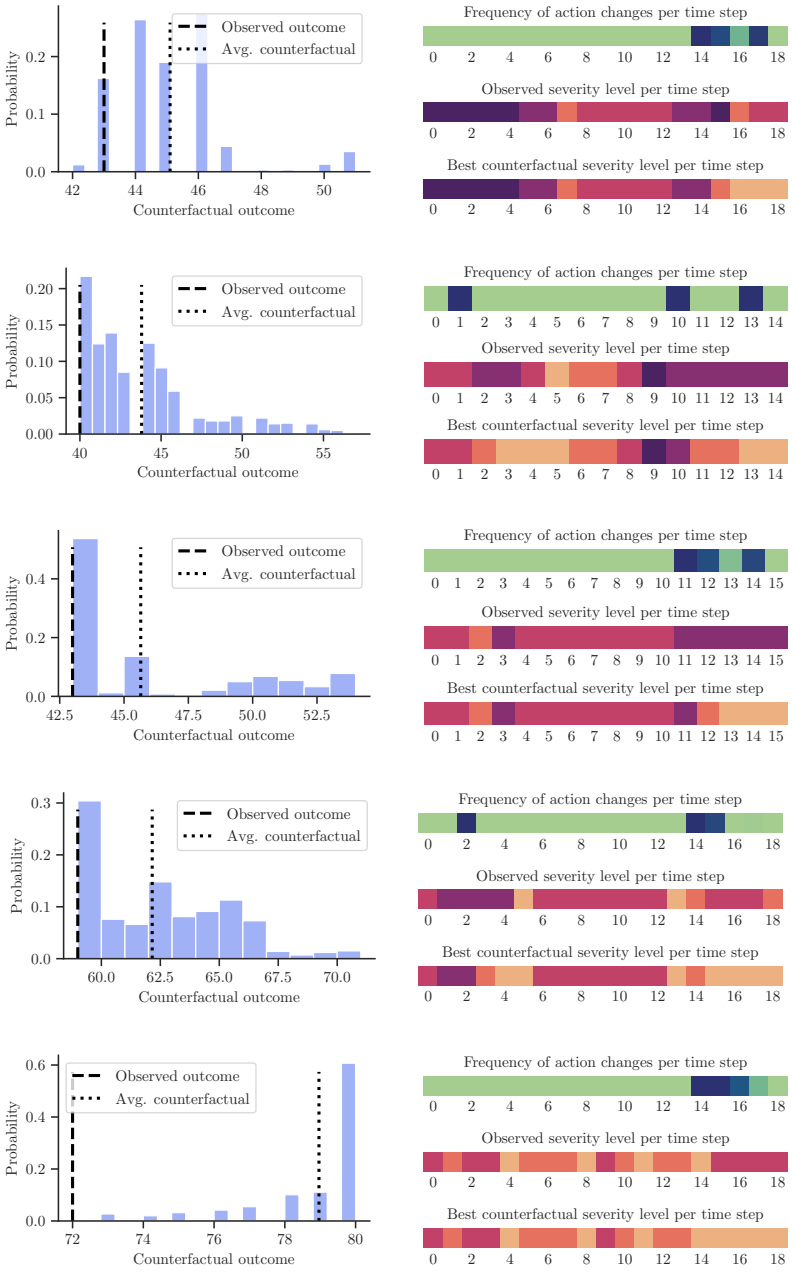
$$c'(\boldsymbol{x}_i, \boldsymbol{x}_j) = \begin{cases} \|\boldsymbol{x}_i - \boldsymbol{x}_j\| & \text{if } x_{i,l} = x_{j,l}\ \forall l \in \bar{\mathcal{L}} \\ \infty & \text{otherwise,} \end{cases} \tag{C.1}$$

where $x_{j,l}$ is the value of the l-th feature for the feature value $\boldsymbol{x}_j$, and $\|\cdot\|$ denotes the Euclidean distance. As an exception, we always set the cost $c'(\boldsymbol{x}_i, \boldsymbol{x}_j)$ between two feature values to ∞ if $x_{j,l} < x_{i,l}$ for $l \in \{\text{Total overdue counts, Total months overdue}\}$, not allowing the history of overdue payments to be erased. For consistency with the cost function defined in Section 3.1.6, we scale the values $c'(\boldsymbol{x}_i, \boldsymbol{x}_j)$ such that the maximum of all non-infinite values is equal to 1 and, we set the final cost function $c(\boldsymbol{x}_i, \boldsymbol{x}_j)$ equal to $\tau \cdot c'(\boldsymbol{x}_i, \boldsymbol{x}_j)$, where $\tau \geq 1$ is a scaling factor which controls the difficulty of changing features.

Finally, as additional values for the parameter γ, we consider the 40-th and 60-th percentile of all the individuals' $P(Y = 1 \mid \boldsymbol{x})$, such that 40% and 60% of the population is accepted by the optimal threshold policy in the non strategic setting, respectively.

C.3 Section 4.1: Insights about individual patients

In this section, we provide insights about additional patients in the dataset. For each of these additional patients, we follow the same procedure in Section 4.1.5, that is, we use Algorithm 5 with the policy π^*_τ, with $k = 3$, to sample multiple counterfactual explanations τ' and look at the corresponding counterfactual outcomes $o(\tau')$. Fig. C.7 summarizes the results, where each row corresponds to a different patient. The results reveal several interesting insights. For most of the patients, all of the counterfactual realizations lead to counterfactual outcomes greater or equal than the observed outcome (left column), however, the difference between the average counterfactual outcome and the observed outcome is relatively small. Notable exceptions are a few patients for whom there is a small probability that the counterfactual outcome is worse than the observed one (top row) as well as patients for whom the difference between the average counterfactual outcome and the observed outcome is high (bottom row). Additionally, we also find that the actual action changes suggested by the optimal counterfactual policies π^*_τ are typically concentrated in a few time steps across counterfactual realizations (right column), usually at the beginning or the end of the realizations.



$\mathbb{P}[o(\tau')]$ Action changes, severity vs. t

Figure C.7: **Insights provided by the optimal counterfactual policy π^*_τ for five real patient who received cognitive behavioral therapy.** Each row corresponds to a different patient with an observed realization τ. The panels in the left column show the distribution of the counterfactual outcomes $o(\tau')$ for the counterfactual realizations τ' induced by π^*_τ and P_τ. The panels in the right column show, for each time step, how frequently a counterfactual explanation changes the observed action as well as the observed severity level and the severity level in the counterfactual realization with the highest counterfactual outcome. Here, darker colors correspond to higher frequencies and higher severities. In all panels, we set $d = 1{,}000$, $k = 3$, and the results are estimated using 1,000 counterfactual realizations.

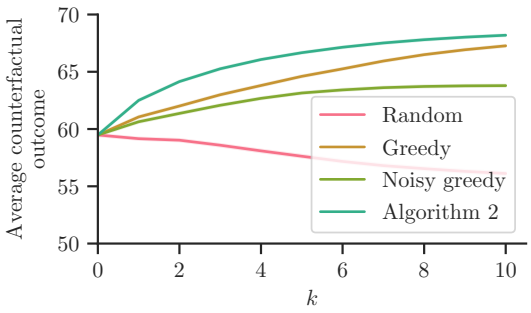


Figure C.8: **Performance achieved by the optimal counterfactual policy π^*_τ given by Algorithm 6 and the baseline policies.** The plot shows the average counterfactual outcome $\frac{1}{\mathcal{T}} \sum_{\tau \in \mathcal{T}} \bar{o}_{\pi_\tau}(\tau)$ achieved by π^*_τ and the baseline policies, averaged over the set of observed realizations $\mathcal{T}$, against the number of actions k differing from the observed ones. For each observed realization, the average counterfactual outcome is estimated using 1,000 counterfactual realizations. Here, we set $d = 1{,}000$ and use data from $|\mathcal{T}| = 73$ patients. Shaded regions correspond to 95% confidence intervals.

C.4 Section 4.1: Performance comparison with baseline policies

Experimental setup. In this section, we compare the average counterfactual outcome achieved by the optimal counterfactual policy, given by Algorithm 6, with that achieved by several baseline policies. To this end, we use the same experimental setup as in Section 4.1.5, however, instead of setting $r(s, a) = -\infty$ for every unobserved pair (s, a), we set $r(s, a) = 5 - s \in \{1, \ldots, 5\}$, similarly as for the observed pairs. This is because, otherwise, we observed that there were always realizations under the baselines policies for which the counterfactual outcome was $-\infty$. In our experiments, we consider with the following baselines policies:

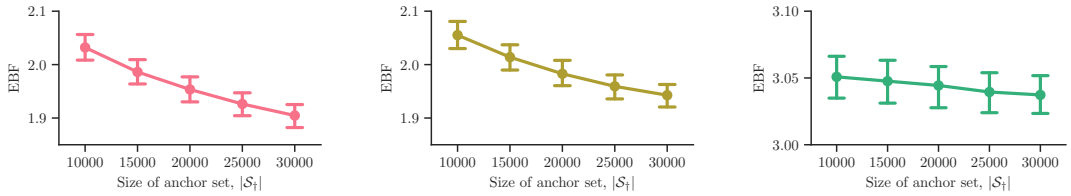
- **Random**: At each time step t, the policy chooses the next action a^* uniformly at random if $l_t < k$ and it chooses $a^* = a_t$ otherwise.
- **Greedy**: At each time step t, being at state (s'_t, l_t), the policy chooses the next action a^* greedily, *i.e.*, if $l_t < k$, then

$$a^* = \operatorname*{argmax}_{a \in \mathcal{A}} \; r(s, a) + \sum_{s' \in \mathcal{S}} P_{\tau,t}(S_{t+1} = s' \,|\, S_t = s'_t, A_t = a) r(s', a'), \tag{C.2}$$

  and, if $l_t = k$, $a^* = a_t$.
- **Noisy greedy**: At each time step t, being at state (s'_t, l_t), it chooses the next action a^* as follows. If $l_t < k$, a^* is given by Eq. C.2 with probability 0.5 and $a^* = a_t$ otherwise. If $l_t = k$, $a^* = a_t$.

Results. Fig. C.8 shows the average counterfactual outcomes achieved by the optimal policy, as given by Algorithm 6, and the above baselines for different k values. The results show that, as expected, the optimal policy outperforms all the baselines across the entire range of k values and, moreover, the competitive advantage

is greater for smaller k values. In addition, we also find that the performance of the random baseline policy drops significantly as k increases, since, as discussed in Section 4.1.5, the observed trajectories are close to optimal in retrospect and, differing from them causes the random policy to worsen the counterfactual outcome.



(a) MC-Lipschitz (b) MC-Uniform (c) Facility-Location

Figure C.9: **Computational efficiency of our methodunder three different anchor set selection strategies.** Panels (a-c) show the Effective Branching Factor (EBF), under three different anchor set selection strategies against the size of the anchor set $\mathcal{S}_\dagger$. In all panels, we set $L_h = 1.0$, $L_\phi = 0.1$ and $k = 3$. Error bars indicate 95% confidence intervals over 200 executions of the A^* algorithm for 200 patients with horizon $T = 12$.

C.5 Section 4.2: Experimental evaluation of anchor set selection strategies

In this section, we benchmark the anchor set selection strategy presented in Section 4.2.3 against two alternative competitive strategies using the sepsis management dataset and the same experimental setup as in Section 4.2.4. More specifically, we consider the following anchor set selection strategies:

(i) *MC-Lipschitz*: This is the strategy described in depth in Section 4.2.3, based on Monte Carlo simulations of counterfactual episodes under randomly sampled counterfactual action sequences. Notably, the time steps where each counterfactual action sequence differs from the observed one are sampled proportionally to the respective Lipschitz constant L_t of the SCM's transition mechanism. To ensure a fair comparison with other strategies, instead of controlling the number of sampled action sequences M, we fix the desired size of the anchor set $\mathcal{S}_\dagger$, and we repeatedly sample counterfactual action sequences until the specified size is met.

(ii) *MC-Uniform*: This strategy is a variant of the previous strategy where we sample the time steps where each counterfactual action sequence differs from the observed one uniformly at random, rather than biasing the sampling towards time steps with higher Lipschitz constants L_t.

(iii) *Facility-Location*: Under this strategy, the anchor set is the solution to a minimax facility location problem defined using the observed available data. Let $\mathcal{S}_o$ be the union of all state vectors observed in all episodes τ in a given dataset. Then, we choose an anchor set $\mathcal{S}_\dagger \subset \mathcal{S}_o$ of fixed size $|\mathcal{S}_\dagger| = b$, such that the maximum distance of any point in $\mathcal{S}_o$ to its closest point in $\mathcal{S}_\dagger$ is minimized. Here, the rationale is that counterfactual states resulting from counterfactual action sequences for one observed episode are likely to be close to the observed states of some other episode in the data. Formally,

$$\mathcal{S}_\dagger = \underset{\mathcal{S}' \subset \mathcal{S}_o : |\mathcal{S}'| = b}{\operatorname{argmin}} \left\{ \max_{\boldsymbol{s} \in \mathcal{S}_o} \min_{\boldsymbol{s}' \in \mathcal{S}'} \{||\boldsymbol{s} - \boldsymbol{s}'||\} \right\}. \tag{C.3}$$

Although the above problem is known to be NP-Complete, we find a solution using the farthest-point clustering algorithm, which is known to have an approximation factor equal to 2 and runs in polynomial time. The algorithm starts by adding one point from $\mathcal{S}_o$ to $\mathcal{S}_\dagger$ at random. Then, it proceeds iteratively and, at each iteration, it adds to $\mathcal{S}_\dagger$ the point from $\mathcal{S}_o$ that is the furthest from all points already in $\mathcal{S}_\dagger$, *i.e.*, $\mathcal{S}_\dagger = \mathcal{S}_\dagger \cup \boldsymbol{s}$, where $\boldsymbol{s} = \max_{\boldsymbol{s}' \in \mathcal{S}_o} \left\{ \min_{\boldsymbol{s}_\dagger \in \mathcal{S}_\dagger} ||\boldsymbol{s}' - \boldsymbol{s}_\dagger|| \right\}$. The algorithm terminates after b iterations.

Results. We compare the computational efficiency of our method under each of the above anchor set selection strategies for various values of the size of the anchor set $|\mathcal{S}_\dagger|$. Fig. C.9 summarizes the results. We observe that the Facility-Location selection strategy performs rather poorly compared to the other two strategies, achieving an effective branching factor (EBF) higher than 3. In contrast, the MC-Lipschitz and MC-Uniform strategies achieve an EBF close to 2, which decreases rapidly as the size of the anchor set increases. Among these two strategies, the MC-Lipschitz strategy, which we use in our experiments in Section 4.2.4, achieves the lowest EBF.

Curriculum vitae

Research interests

Machine learning, decision making, causal inference, game theory, optimization, computational cognitive science.

Education

2019-2025 **PhD in computer science**
Max Planck Institute for Software Systems &
University of Kaiserslautern-Landau

2013-2019 **Diploma in electrical & computer engineering**
National Technical University of Athens

Employment

2025- **Postdoctoral researcher**
Hasso Plattner Institute, Potsdam, Germany

2023 **AI research scientist intern**
Meta AI (FAIR), Paris, France

2019-2025 **Doctoral researcher**
Max Planck Institute for Software Systems, Kaiserslautern, Germany

2018 **Research engineering intern**
National Centre for Scientific Research "Demokritos", Athens, Greece