Title
Towards Causally-Aware Machine Learning

Author
Kyono, Trent M

Publication Date
2021

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

Towards Causally-Aware Machine Learning

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Computer Science

by

Trent M. Kyono

2021

ABSTRACT OF THE DISSERTATION

Towards Causally-Aware Machine Learning

by

Trent M. Kyono
Doctor of Philosophy in Computer Science
University of California, Los Angeles, 2021
Professor Mihaela van der Schaar, Chair

The popularity of machine learning in both academia and industry has experienced unparalleled growth. This has been driven by many factors, including the proliferation and availability of digitized data, the recent growth of computational power available, such as graphical processing units, and the powerful machine learning software libraries that leverage them. The overwhelming majority of existing and current research focuses on learning correlations between data rather than leveraging cause-effect relationships.

In parallel to the machine learning revolution, the study of cause-effect relationships, causality has been well-studied but often overlooked in current practice. These two disciplines are often accepted as orthogonal approaches to data modeling. This dissertation focuses on the confluence of these two approaches in an attempt to advance current machine learning techniques with fundamental concepts from causality. Namely, we identify several strategies to leverage causal structure (in the form of a directed acyclic graph) to improve machine learning performance.

Our technical contributions touch several fundamental and widespread machine learning problems. We first present a regularization method, called CASTLE (Causal Structure Learning), that simultaneously learns the causal graph/structure in the input layers of a neural network, allowing for improved predictive performance on out-of-sample data. Next,

using a similar method, we develop a method for missing data imputation called MIRACLE (Missing data Imputation Refinement and Causal Learning). Similar to CASTLE, MIRACLE simultaneously learns the underlying causal structure to improve missing data imputation by refining its predictions in a unique "bootstrapping" manner. Next, we introduce a method, DECAF (Debiasing Causal Fairness), that introduces causal structure into Generative Adversarial Networks (GANs) to generate synthetic data that is fair for any downstream model. Next, we focus on the problem of unsupervised domain adaptation (UDA), where we leverage the invariance of causal structure to select models that best generalize to an unlabeled target domain. Lastly, we focus on extending our model selection method to individualized treatment effect (ITE) models, which are commonly used in the healthcare setting.

To demonstrate the utility of our models, we evaluate their performance on a variety of synthetic datasets, semi-synthetic datasets (for ITE models), and real-world datasets that include publicly available UCI datasets and healthcare datasets for heart failure, COVID-19, and prostate cancer among many others. We show that, compared to existing machine learning models that are agnostic to causality, our causally-aware models can improve regularization, missing data imputation, synthetic data quality, and UDA model selection.

The dissertation of Trent M. Kyono is approved.

Adnan Youssef Darwiche

Sriram Sankararaman

William Hsu

Mihaela van der Schaar, Committee Chair

University of California, Los Angeles

2021

To Ashley, Kai, Kamden, and Brooke ...

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

This dissertation would not be possible without the help and support of my family, my advisor, and my colleagues. I am beyond grateful!

First, I wish to thank my advisor, Professor Mihaela van der Schaar, for challenging me and pushing me beyond what I believed I was capable of. This dissertation would not have been possible without her guidance, support, and extraordinary research vision. Thank you, Mihaela, for challenging me to become more than an engineer, but a competent researcher. You have taught me the importance of digging deeper into problems outside of my comfort zone. For this I am eternally grateful. I will never take another step, without first asking "why?".

I would like to thank the other members of my dissertation committee, Professor Adnan Darwiche, Professor William Hsu, and Professor Sriram Sankararaman for their invaluable perspectives and thoughtful feedback.

I would like to thank all of my co-authors, collaborators, and labmates at UCLA: Professor Judea Pearl, Jinsung Yoon, Ahmed Alaa, and Changhee Lee. I would also like to thank my extended family of collaborators and teammates in the UK (at Cambridge and Oxford): Yao Zhang, Ioana Bica, Boris van Breugel, Jeroen Berrevoets, Alexis Bellot, Zhaozhi Qian, James Jordon, Fergus Imrie, Yuchao Qin, Jonathan Crabbe, Dan Jarrett, Alihan Huyuk, Alicia Curth, and Alex Chan. We had many thoughtful research discussion, and I thank you for your companionship and support over the years.

I would like to thank my family for helping with raising my children and supporting me my whole life. I would like to specifically thank my parents, Toni and Glen, my step-parents, Wayne and Suzie, my in-laws, Dean and Liane, and my brothers, Michael and Jaydon, for supporting me throughout this academic journey and unconditionally being in my corner. A special thanks goes to my grandparents, Fred and Akie, who I wish were around to see me at the finish line of this journey. You all have inspired me to never give up, try my best (no less than 100%), and encouraged me to believe in myself. Thank you!

Lastly, and most importantly, I would like thank my wife, Ashley, and my children, Kai, Kamden and Brooke. Ashley, you have taken care of me, the family, the home, essentially everything, giving me the time to pursue my dreams. I could not summarize everything you have done within all the pages of this dissertation. You are superhuman! Thank you!

Kai, Kamden and Brooke, you are too young to read now, but I hope one day when you can, you will see that if your silly Dad can accomplish this – there is nothing in the world you can't do!

CURRICULUM VITAE

2004–2008	B.S. in Mathematics/Computer Science, Pepperdine University.
2008–2010	M.S. in Computer Science, University of California, Los Angeles.
2010–2012	Trex Enterprises, Software Engineer.
2012–2021	The Boeing Company, Sr. Machine Learning Engineer.
2018–Present	Ph.D. Graduate Researcher in Computer Science, University of California, Los Angeles.
2021–Present	Facebook, Machine Learning Engineer.

PUBLICATIONS

Trent Kyono, Yao Zhang, Alexis Bellot, Mihaela van der Schaar, "MIRACLE: Causally-Aware Imputation via Learning Missing Data Mechanisms," (2021), Advances in Neural Information Processing Systems (NeurIPS).

Trent Kyono, Boris van Breugel, Jerroen Berrevoets, Mihaela van der Schaar, "DECAF: Generating Fair Synthetic Data Using Causally-Aware Generative Networks," (2021), Advances in Neural Information Processing Systems (NeurIPS).

Trent Kyono, Mihaela van der Schaar, "Exploiting Causal Structure for Robust Model Selection in Unsupervised Domain Adaptation," (2021), IEEE Transaction on Artificial Intelligence (TAI).

Trent Kyono, Ioana Bica, Mihaela van der Schaar, "Selecting Treatment Effects Models for Domain Adaptation Using Causal Knowledge," (2021), In Submission ACM Transaction on Computing for Healthcare.

Trent Kyono, Fiona Gilbert, Mihaela van der Schaar, "Triage of 2D Mammographic Images Using Multi-view Multi-task Convolutional Neural Networks," (2021), ACM Transaction on Computing for Healthcare.

Trent Kyono, Yao Zhang, Mihaela van der Schaar, "CASTLE: Regularization via Auxiliary Causal Graph Discovery," (2020), Advances in Neural Information Processing Systems (NeurIPS).

Trent Kyono, Fiona Gilbert, Mihaela van der Schaar, "Improving Workflow Efficiency for Mammography Using Machine Learning," (2019), Journal of the American College of Radiology (JACR).

Trent Kyono, Fiona Gilbert, Mihaela van der Schaar, "Multi-view Multi-task Learning for Improving Autonomous Mammogram Diagnosis," (2019), Machine Learning for Healthcare Conference (MLHC).

Trent Kyono, Jacob Lucas, Michael Werth, Justin Fletcher, Ian McQuaid, "Machine Learning for Quality Assessment of Ground-based Optical Images of Satellites," (2019), Optical Engineering.

Anton Nemchenko, **Trent Kyono**, Mihaela van der Schaar, "Siamese Survival Analysis with Competing Risks," (2018). International Conference of Artificial Neural Networks.

Trent Kyono "Commentator, A Front-end User-Interface Module for Graphical and Structural Equation Modeling," (2010), UCLA Master's Thesis.

CHAPTER 1

Introduction

Current advances in information technology have proliferated the availability of data for machine learning models to consume. Couple the large amounts of available data with recent hardware advancements and distributed scalability, and machine learning models are becoming overwhelmingly large and powerful. As a result, machine learning has infiltrated many fields of academia and has become a key driver of many technologies in countless industries.

Machine learning, apart from causal models, are often criticized for only learning correlations in data [177]. Because of this, we consider machine learning to be causally "unaware" or agnostic to underlying cause-effect relationships, and therefore incapable of causal reasoning [104]. While in some cases this may be sufficient, in general, there is invaluable information that can be leveraged from causality that can be used to make machine learning more causally "aware" and thus improve performance.

Although causality has been studied for years [104, 129], only recently with the growing popularity of machine learning, have causal researchers looked to inject causal notions into existing machine learning methods. These works include [177, 78, 45, 111, 172, 176, 178, 103, 87] to name just a few. This primary objective of this work is to introduce causality (or causal structure) – bring causal-awareness – to several well-studied areas in machine learning, including generalization/regularization, missing data imputation, fair synthetic data generation, and model selection for unsupervised domain adaptation.

1.1 Outline of the Dissertation

The rest of this dissertation is organized as follows. We first introduce a new regularization technique for improving machine learning generalization that leverages causal structure in Chapter 2. Next we focus on a new imputation method that simultaneously learns causal structure for improved imputation in Chapter 3. We then shift our focus in Chapter 4 to generating fair synthetic data using a generative neural network and causal structure. Lastly, in Chapters 5 and 6 we present model selection methods for unsupervised domain adaptation that leverage invariant causal graphs for predictive and treatment-effects models, respectively.

1.2 Summary of Technical Contributions

In the following sections, we present a brief summary of the technical contributions of each of the upcoming chapters. We address several well-studied problems in machine learning, including: regularization, missing data imputation, synthetic data fairness, and unsupervised domain adaptation.

1.2.1 Regularization

Regularization improves generalization of supervised models to out-of-sample data. Prior works have shown that prediction in the causal direction (effect from cause) results in lower testing error than the anti-causal direction. However, existing regularization methods are agnostic of causality. In Chapter 2 we introduce Causal Structure Learning (CASTLE) regularization and propose to regularize a neural network by jointly learning the causal relationships between variables. CASTLE learns the causal directed acyclical graph (DAG) as an adjacency matrix embedded in the neural network's input layers, thereby facilitating the discovery of optimal predictors. Furthermore, CASTLE efficiently reconstructs only the features in the causal DAG that have a causal neighbor, whereas reconstruction-based regularizers suboptimally reconstruct all input features. We provide a theoretical generalization bound for our approach and conduct experiments on a plethora of synthetic and real publicly

available datasets demonstrating that CASTLE consistently leads to better out-of-sample predictions as compared to other popular benchmark regularizers.

1.2.2 Missing data

Missing data is an important problem in machine learning practice. Starting from the premise that imputation methods should preserve the causal structure of the data, in Chapter 3, we develop a regularization scheme that encourages any baseline imputation method to be causally consistent with the underlying data generating mechanism. Our proposal is a causally-aware imputation algorithm (MIRACLE). MIRACLE iteratively refines the imputation of a baseline by simultaneously modeling the missingness generating mechanism, encouraging imputation to be consistent with the causal structure of the data. We conduct extensive experiments on synthetic and a variety of publicly available datasets to show that MIRACLE is able to consistently improve imputation over a variety of benchmark methods across all three missingness scenarios: at random, completely at random, and not at random.

1.2.3 Synthetic Data Fairness

Machine learning models have been criticized for reflecting unfair biases in the training data. Instead of solving for this by introducing fair learning algorithms directly, we focus on generating fair synthetic data, such that any downstream learner is fair. Generating fair synthetic data from unfair data— while remaining truthful to the underlying data-generating process (DGP) —is non-trivial. In Chapter 4, we introduce DECAF: a GAN-based fair synthetic data generator for tabular data. With DECAF we embed the DGP explicitly as a structural causal model in the input layers of the generator, allowing each variable to be reconstructed conditioned on its causal parents. This procedure enables inference-time debiasing, where biased edges can be strategically removed for satisfying user-defined fairness requirements. The DECAF framework is versatile and compatible with several popular definitions of fairness. In our experiments, we show that DECAF successfully removes undesired bias and— in contrast to existing methods —is capable of generating high-quality

synthetic data. Furthermore, we provide theoretical guarantees on the generator's convergence and the fairness of downstream models.

1.2.4 Predictive Model Selection for Unsupervised Domain Adaptation

In many real-world settings, such as healthcare, machine learning models are trained and validated on one labeled domain and tested or deployed on another where feature distributions differ, i.e., there is covariate shift. When annotations are costly or prohibitive, an unsupervised domain adaptation (UDA) regime can be leveraged requiring only unlabeled samples in the target domain. Existing UDA methods are unable to factor in a model's predictive loss based on predictions in the target domain and therefore suboptimally leverage density ratios of only the input covariates in each domain. In Chapter 5, we propose a model selection method for leveraging model predictions on a target domain without labels by exploiting the domain invariance of causal structure. We assume or learn a causal graph from the source domain, and select models that produce predicted distributions in the target domain that have the highest likelihood of fitting our causal graph. We thoroughly analyze our method under oracle knowledge using synthetic data. We then show on several real-world datasets, including several COVID-19 examples, that our method is able to improve on the state-of-the-art UDA algorithms for model selection.

1.2.5 Treatment-effect Model Selection for Unsupervised Domain Adaptation

While a large number of causal inference models for estimating individualized treatment effects (ITE) have been developed, selecting the best one poses a unique challenge since the counterfactuals are never observed. The problem is challenged further in the unsupervised domain adaptation (UDA) setting where we have access to labeled samples in the source domain, but desire selecting an ITE model that achieves good performance on a target domain where only unlabeled samples are available. Existing selection techniques for UDA are designed for predictive models and are sub-optimal for causal inference because they (1) do not account for the missing counterfactuals and (2) only examine the discriminative density

ratios between the input covariates in the source and target domain and do not factor in the model's predictions in the target domain. In Chapter 6, we leverage the invariance of causal structures across domains to introduce a novel model selection metric specifically designed for ITE models under UDA. We propose selecting models whose predictions of the effects of interventions satisfy invariant causal structures in the target domain. Experimentally, our method selects ITE models that are more robust to covariate shifts on a variety of datasets, including estimating the effect of ventilation in COVID-19 patients.

CHAPTER 2

CASTLE: Regularization via Auxiliary Causal Graph Discovery

2.1 Introduction

A primary concern of machine learning, and deep learning in particular, is generalization performance on out-of-sample data. Over-parameterized deep networks efficiently learn complex models and are, therefore, susceptible to overfit to training data. Common regularization techniques to mitigate overfitting include data augmentation [166, 73], dropout [56, 157, 131], adversarial training [82], label smoothing [51], and layer-wise strategies [18, 117, 57] to name a few. However, these methods are agnostic of the causal relationships between variables limiting their potential to identify optimal predictors based on graphical topology, such as the causal parents of the target variable. An alternative approach to regularization leverages supervised reconstruction, which has been proven theoretically and demonstrated empirically to improve generalization performance by obligating hidden bottleneck layers to reconstruct input features [152, 84]. However, supervised auto-encoders suboptimally reconstruct all features, including those without causal neighbors, i.e., adjacent cause or effect nodes. Naively reconstructing these variables does not improve regularization and representation learning for the predictive model. In some cases, it may be harmful to generalization performance, e.g., reconstructing a random noise variable.

Although causality has been a topic of research for decades, only recently has cause and effect relationships been incorporated into machine learning methodologies and research. Recently, researchers at the confluence of machine learning and causal modeling have advanced

causal discovery [172, 27], causal inference [134, 12], model explainability [135], domain adaptation [178, 103, 87] and transfer learning [111] among countless others. The existing synergy between these two disciplines has been recognized for some time [120], and recent work suggests that causality can improve and complement machine learning regularization [13, 114, 59]. Furthermore, many recent causal works have demonstrated and acknowledged the optimality of predicting in the causal direction, i.e., predicting effect from cause, which results in less test error than predicting in the anti-causal direction [111, 133, 83, 62].

Contributions. In this work, we introduce a novel regularization method called CASTLE (CAusal STructure LEarning) regularization. CASTLE regularization uses causal graph discovery as an auxiliary task when training a supervised model to improve the generalization performance of the primary prediction task. Specifically, CASTLE learns the causal directed acyclical graph (DAG) under continuous optimization as an adjacency matrix embedded in a feed-forward neural network's input layers. By jointly learning the causal graph, CASTLE can surpass the benefits provided by feature selection regularizers by identifying optimal predictors, such as the target variable's causal parents. Additionally, CASTLE further improves upon auto-encoder-based regularization [84] by reconstructing only the input features that have neighbors (adjacent nodes) in the causal graph. Regularization of a predictive model to satisfy the causal relationships among feature and target variables effectively guide the model towards the direction of better out-of-sample generalization guarantees. We provide a theoretical generalization bound for CASTLE and demonstrate improved performance against a variety of benchmark methods on a plethora of real and synthetic datasets.

2.2 Related Works

We compare to the related work in the simplest supervised learning setting where we desire learning a function from some features $\boldsymbol{X}$ to a target variable Y given some data of the variables $\boldsymbol{X}$ and Y to improve out-of-sample generalization within the same distribution. This is a significant departure from the branches of machine learning algorithms, such as in

Table 2.1: Comparison of related works.

Method	Feat. Sel.	Struct. Learning	Causal Pred.	Target Sel.
Capacity-based	✓	✗	✗	✗
SAE	✗	✓	✗	✗
CASTLE	✓	✓	✓	✓

semi-supervised learning and domain adaptation, where the regularizer is constructed with information other than variables $\boldsymbol{X}$ and Y.

Regularization controls model complexity and mitigates overfitting. ℓ_1 [145] and ℓ_2 [54] regularization are commonly used regularization approaches where the former is used when a sparse model is preferred. For deep neural networks, dropout regularization [56, 157, 131] has been shown to be superior in practice to ℓ_p regularization techniques. Other capacity-based regularization techniques commonly used in practice include early stopping [41], parameter sharing [41], gradient clipping [108], batch normalization [58], data augmentation [73], weight noise [101], and MixUp [173] to name a few. Norm-based regularizers with sparsity, e.g. Lasso [145], are used to guide feature selection for supervised models. The work of [84] on supervised auto-encoders (SAE) theoretically and empirically shows that adding a reconstruction loss of the input features functions as a regularizer for predictive models. However, this method does not select which features to reconstruct and therefore suffers performance degradation when tasked to reconstruct features that are noise or unrelated to the target variables.

Two existing works [59, 13] attempt to draw the connection between causality and regularization. Based on an analogy between overfitting and confounding in linear models, [59] proposed a method to determine the regularization hyperparameter in linear Ridge or Lasso regression models by estimating the strength of confounding. [13] use causality detectors [24, 83] to weight a sparsity regularizer, e.g. ℓ_1, for performing non-linear causality analysis and generating multivariate causal hypotheses. Neither of the works has the same

objective as us — improving the generalization performance of supervised learning models, nor do they overlap methodologically by using causal DAG discovery.

Causal discovery is an NP-hard problem that requires a brute-force search through a non-convex combinatorial search space, limiting the existing algorithms to reaching global optima for only small problems. Recent approaches have successfully accelerated these methods by using a novel acyclicity constraint and formulating the causal discovery problem as a continuous optimization over real matrices (avoiding combinatorial search) in the linear [170] and nonlinear [174, 78] cases. CASTLE incorporates these recent causal discovery approaches of [170, 174] to improve regularization for prediction problems in general.

As shown in Table 2.1, CASTLE regularization provides two additional benefits: causal prediction and target selection. First, CASTLE identifies causal predictors (e.g., causal parents if they exist) rather than correlated features. Furthermore, CASTLE improves upon reconstruction regularization by only reconstructing features that have neighbors in the underlying DAG. We refer to this advantage as "target selection". Collectively these benefits contribute to the improved generalization of CASTLE. Next we introduce our notation (Section 2.3.1) and provide more details of these benefits (Section 2.3.2).

2.3 Methodology

In this section, we provide a problem formulation with causal preliminaries for CASTLE. Then we provide a motivational discussion, regularizer methodology, and generalization theory for CASTLE.

2.3.1 Problem Formulation

In the standard supervised learning setting, we denote the input feature variables and target variable, by $\boldsymbol{X} = [X_1, ..., X_d] \in \mathcal{X}$ and $Y \in \mathcal{Y}$, respectively, where $\mathcal{X} \subseteq \mathbb{R}^d$ is a d-dimensional feature space and $\mathcal{Y} \subseteq \mathbb{R}$ is a one-dimensional target space. Let $P_{\boldsymbol{X},Y}$ denote the joint distribution of the features and target. Let $[N]$ denote the set $\{1, ..., N\}$. We observe a

dataset, $\mathcal{D} = \{(\boldsymbol{X}_i, Y_i), i \in [N]\}$, consisting of N i.i.d. samples drawn from $P_{\boldsymbol{X},Y}$. The goal of a supervised learning algorithm $\mathcal{A}$ is to find a predictive model, $f_Y : \mathcal{X} \to \mathcal{Y}$, in a hypothesis space $\mathcal{H}$ that can explain the association between the features and the target variable. In the learning algorithm $\mathcal{A}$, the predictive model $\hat{f}_Y$ is trained on a finite number of samples in $\mathcal{D}$, to predict well on the out-of-sample data generated from the same distribution $P_{\boldsymbol{X},Y}$. However, overfitting, a mismatch between training and testing performance of $\hat{f}_Y$, can occur if the hypothesis space $\mathcal{H}$ is too complex and the training data fails to represent the underlying distribution $P_{\boldsymbol{X},Y}$. This motivates the usage of regularization to reduce the hypothesis space's complexity $\mathcal{H}$ so that the learning algorithm $\mathcal{A}$ will only find the desired function to explain the data. Assumptions of the underlying distribution dictate regularization choice. For example, if we believe only a subset of features is associated with the label Y, then ℓ_1 regularization [145] can be beneficial in creating sparsity for feature selection.

CASTLE regularization is based on the assumption that a causal DAG exists among the input features and target variable. In the causal framework of [104], a causal structure of a set of variables $\boldsymbol{X}$ is a DAG in which each vertex $v \in V$ corresponds to a distinct element in $\boldsymbol{X}$, and each edge $e \in E$ represents direct functional relationships between two neighboring variables. Formally, we assume the variables in our dataset satisfy a nonparametric structural equation model (NPSEM) as defined in Definition 1. The word "nonparametric" means we do not make any assumption on the underlying functions f_i in the NPSEM. In this work, we characterize optimal learning by a predictive model as discovering the function $Y = f_Y(\text{Pa}(Y), u_Y)$ in NPSEM [104].

Definition 1. *(NPSEMs) Given a DAG $\mathcal{G} = (V = [d+1], E)$, the random variables $\tilde{\boldsymbol{X}} = [Y, \boldsymbol{X}]$ satisfy a NPSEM if*

$$X_i = f_i(Pa(X_i), u_i), \ \ i \in [d+1],$$

where $Pa(i)$ is the parents (direct causes) of X_i in $\mathcal{G}$ and $\boldsymbol{u}_{[d+1]}$ are some random noise variables.

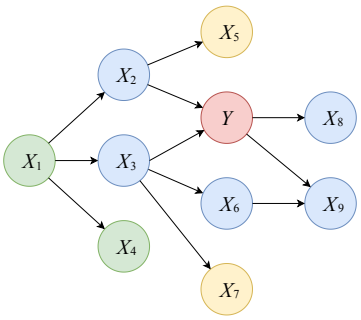


Figure 2.1: Example DAG.

2.3.2 Why CASTLE regularization matters

We now present a graphical example to explain the two benefits of CASTLE mentioned in Section 2.2, causal prediction and target selection. Consider Figure 2.1 where we are given nine feature variables $X_1, ..., X_9$ and a target variable Y.

Causal Prediction. The target variable Y is generated by a function $f_Y(\text{Pa}(Y), u_Y)$ from Definition 1 where the parents of Y are $\text{Pa}(Y) = \{X_2, X_3\}$. In CASTLE regularization, we train a predictive model $\hat{f}_Y$ jointly with learning the DAG among $\boldsymbol{X}$ and Y. The features that the model uses to predict Y are the causal parents of Y in the learned DAG. Such a model is sample efficient in uncovering the true function $f_Y(\text{Pa}(Y), u_Y)$ and generalizes well on the out-of-sample data. Our theoretical analysis in Section 2.3.4 validates this advantage when there exists a DAG structure among the variables $\boldsymbol{X}$ and Y. However, there may exist other variables that predict Y more accurately than the causal parents $\text{Pa}(Y)$. For example, if the function from Y to X_8 is a one-to-one linear mapping, we can predict Y trivially from the feature X_8. In our objective function introduced later, the prediction loss of Y will be weighted higher than the causal regularizer. Among the predictive models with a similar prediction loss of Y, our objective function still prefers to use the model, which minimizes the causal regularizer and uses the causal parents. However, it would favor the easier predictor if one exists and gives a much lower prediction loss of Y. In this case, the learned DAG may differ from the true DAG, but we reiterate that we are focused on the problem of generalization rather than causal discovery.

Target Selection. Consider the variables X_5, X_6 and X_7 which share parents (X_2 and X_3) with Y in Figure 2.1. The functions $X_5 = f_5(X_2, u_5)$, $X_6 = f_6(X_3, u_6)$, and $X_7 = f_7(X_3, u_7)$ may have some learnable similarity (e.g. basis functions and representations) with $Y = f_Y(X_2, X_3, u_Y)$, that we can exploit by training a shared predictive model of Y with the auxiliary task of predicting X_5, X_6 and X_7. From the causal graph topology, CASTLE discovers the optimal features that should act as the auxiliary task for learning f_Y. CASTLE learns the related functions jointly in a shared model, which is proven to improve the generalization performance of predicting Y by learning shared basis functions and representations [94].

2.3.3 CASTLE regularization

Let $\tilde{\mathcal{X}} = \mathcal{Y} \times \mathcal{X}$ denote the data space, $P_{(\boldsymbol{X},Y)} = P_{\tilde{\boldsymbol{X}}}$ the data distribution, and $\|\cdot\|_F$ the Frobenius norm. We define random variables $\tilde{\boldsymbol{X}} = [\tilde{X}_1, \tilde{X}_2, ..., \tilde{X}_{d+1}] := [Y, X_1, ..., X_d] \in \tilde{\mathcal{X}}$. Let $\mathbf{X} = \left[\mathbf{X}_1, ..., \mathbf{X}_d\right]$ denote the $N \times d$ input data matrix, $\mathbf{Y}$ the N-dimensional label vector, $\tilde{\mathbf{X}} = [\mathbf{Y}, \mathbf{X}]$ the $N \times (d+1)$ matrix that contains data of all the variables in the DAG.

To facilitate exposition, we first introduce CASTLE in the linear setting. Here, the parameters are a $(d+1) \times (d+1)$ adjacency matrix $\mathbf{W}$ with zero in the diagonal. The objective function is given as

$$\hat{\mathbf{W}} \in \min_{\mathbf{W}} \tfrac{1}{N}\|\mathbf{Y} - \tilde{\mathbf{X}}\mathbf{W}_{:,1}\|^2 + \lambda \mathcal{R}_{\text{DAG}}(\tilde{\mathbf{X}}, \mathbf{W}) \tag{2.1}$$

where $\mathbf{W}_{:,1}$ is the first column of $\mathbf{W}$. We define the DAG regularization loss $\mathcal{R}_{\text{DAG}}(\tilde{\mathbf{X}}, \mathbf{W})$ as

$$\mathcal{R}_{\text{DAG}}(\tilde{\mathbf{X}}, \mathbf{W}) = \mathcal{L}_{\mathbf{W}} + \mathcal{R}_{\mathbf{W}} + \beta \mathcal{V}_{\mathbf{W}}. \tag{2.2}$$

where $\mathcal{L}_{\mathbf{W}} = \frac{1}{N}\|\tilde{\mathbf{X}} - \tilde{\mathbf{X}}\mathbf{W}\|_F^2$, $R_{\mathbf{W}} = \left(\text{Tr}\left(e^{\mathbf{W}\odot\mathbf{W}}\right) - d - 1\right)^2$, $\mathcal{V}_{\mathbf{W}}$ is the ℓ_1 norm of $\mathbf{W}$, $\odot$ is the Hadamard product, and $e^{\mathbf{M}}$ is the matrix exponential of $\mathbf{M}$. The DAG loss $\mathcal{R}_{\text{DAG}}(\tilde{\mathbf{X}}, \mathbf{W})$ is introduced in [170] for learning linear DAG by continuous optimization. Here we use it as the regularizer for our linear regression model $\mathbf{Y} = \tilde{\mathbf{X}}\mathbf{W}_{:,1} + \boldsymbol{\epsilon}$. From Theorem 1 in [170], we know the graph given by $\mathbf{W}$ is a DAG if and only if $R_{\mathbf{W}} = 0$. The prediction $\hat{\mathbf{Y}} = \tilde{\mathbf{X}}\mathbf{W}_{:,1}$ is

the projection of $\mathbf{Y}$ onto the parents of Y in the learned DAG. This increases the stability of linear regression when issues pertaining to collinearity or multicollinearity among the input features appear.

Continuous optimization for learning nonparametric causal DAGs has been proposed in the prior work by [174]. In a similar manner, we also adapt CASTLE to nonlinear cases. Suppose the predictive model for Y and the function generating each feature X_k in the causal DAG are parameterized by an M-layer feed-forward neural network $f_\Theta : \tilde{\mathcal{X}} \to \tilde{\mathcal{X}}$ with ReLU activations and layer size h. Figure 2.2 shows the network architecture of f_Θ. This joint network can be instantiated as a $d+1$ sub-network f_k with shared hidden layers, where f_k is responsible for reconstructing the feature $\tilde{X}_k$. We let $\mathbf{W}_1^k$ denote the $h \times (d+1)$ weight matrix in the input layer of $f_k, k \in [d+1]$. We set the k-th column of $\mathbf{W}_1^k$ to zero such that f_k does not utilize $\tilde{X}_k$ in its prediction of $\tilde{X}_k$. We let $\mathbf{W}_m$, $m = 2, .., M-1$ denote the weight matrices in the network's shared hidden layers, and $\mathbf{W}_M = [\mathbf{W}_M^1, ..., \mathbf{W}_M^{d+1}]$ denotes the $h \times (d+1)$ weight matrix in the output layer. Explicitly, we define the sub-network f_k as

$$f_k(\tilde{\boldsymbol{X}}) = \phi\big(\cdots \phi\big(\phi\big(\tilde{\boldsymbol{X}}\mathbf{W}_1^k\big)\mathbf{W}_2\big)\cdots \mathbf{W}_{M-1}\big)\mathbf{W}_M^k, \tag{2.3}$$

where $\phi(\cdot)$ is the ReLU activation function. The function f_Θ is given as $f_\Theta(\tilde{\boldsymbol{X}}) = [f_1(\tilde{\boldsymbol{X}}), ..., f_{d+1}(\tilde{\boldsymbol{X}})]$. Let $f_\Theta(\tilde{\mathbf{X}})$ denote the prediction for the N samples matrix $\tilde{\mathbf{X}}$ where $[f_\Theta(\tilde{\mathbf{X}})]_{i,k} = f_k(\tilde{\boldsymbol{X}}_i)$, $i \in [N]$ and $k \in [d+1]$. All network parameters are collected into sets as

$$\Theta_1 = \{\mathbf{W}_1^k\}_{k=1}^{d+1},\ \Theta = \Theta_1 \cup \{\mathbf{W}_m\}_{k=2}^{M} \tag{2.4}$$

The training objective function of f_Θ is

$$\Theta \in \min_{\Theta} \tfrac{1}{N}\big\|\mathbf{Y} - [f_\Theta(\tilde{\mathbf{X}})]_{:,1}\big\|^2 + \lambda \mathcal{R}_{\text{DAG}}\big(\tilde{\mathbf{X}}, f_\Theta\big). \tag{2.5}$$

The DAG loss $\mathcal{R}_{\text{DAG}}\big(\tilde{\mathbf{X}}, f_\Theta\big)$ is given as

$$\mathcal{R}_{\text{DAG}}\big(\tilde{\mathbf{X}}, f_\Theta\big) = \mathcal{L}_N(f_\Theta) + \mathcal{R}_{\Theta_1} + \beta \mathcal{V}_{\Theta_1}. \tag{2.6}$$

Because the k-th column of the input weight matrix $\mathbf{W}_1^k$ is set to zero, $\mathcal{L}_N(f_\Theta) = \frac{1}{N}\big\|\tilde{\mathbf{X}} - f_\Theta(\tilde{\mathbf{X}})\big\|_F^2$ differs from the standard reconstruction loss in auto-encoders (e.g. SAE) by only

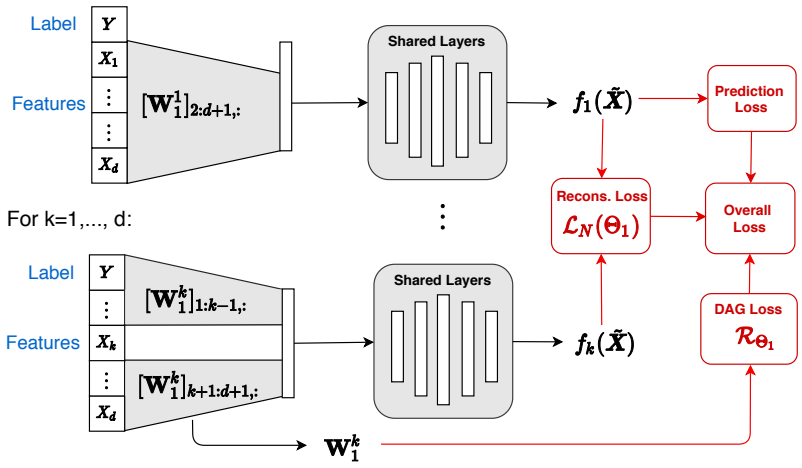


Figure 2.2: Schematic of CASTLE regularization. Our goal is to have the following tasks: (1) a prediction of a target variable Y, and (2) the discovered causal DAG for input features $\boldsymbol{X}$ and Y.

allowing the model to reconstruct each feature and target variable from the others. In contrast, auto-encoders reconstruct each feature using all the features including itself. $\mathcal{V}_{\Theta_1}$ is the ℓ_1 norm of the weight matrices $\mathbf{W}_1^k$ in Θ_1, and the term $\mathcal{R}_{\Theta_1}$ is given as,

$$\mathcal{R}_{\Theta_1} = (\mathrm{Tr}\big(e^{\mathbf{M}\odot\mathbf{M}}\big) - d - 1)^2, \tag{2.7}$$

where $\mathbf{M}$ is a $(d+1)\times(d+1)$ matrix such that $[\mathbf{M}]_{k,j}$ is the ℓ_2-norm of the k-th row of the matrix $\mathbf{W}_1^j$. When the acyclicity loss $\mathcal{R}_{\Theta_1}$ is minimized, the sub-networks $f_1,\dots f_{d+1}$ forms a DAG among the variables; $\mathcal{R}_{\Theta_1}$ obligates the sub-networks to reconstruct only the input features that have neighbors (adjacent nodes) in the learned DAG. We note that converting the nonlinear version of CASTLE into a linear form can be accomplished by removing all the hidden layers and output layers and setting the dimension h of the input weight matrices to be 1 in (2.3), i.e., $f_k(\tilde{\boldsymbol{X}}) = \tilde{\boldsymbol{X}}\mathbf{W}_1^k$ and $f_\Theta(\tilde{\boldsymbol{X}}) = [\tilde{\boldsymbol{X}}\mathbf{W}_1^1, ..., \tilde{\boldsymbol{X}}\mathbf{W}_1^{d+1}] = \tilde{\boldsymbol{X}}\mathbf{W}$, which is the linear model in (2.1-2.2).

Managing computational complexity. If the number of features is large, it is computationally expensive to train all the sub-networks simultaneously. We can mitigate this by sub-sampling. At each iteration of gradient descent, we randomly sample a subset of

features to reconstruct and only minimize the prediction loss and reconstruction loss on these sub-sampled features. Note that we do not have a hidden confounders issue here, since Y and the sub-sampled features are predicted by all the features except itself. The sparsity DAG constraint on the weight matrices is unchanged at each iteration. In this case, we keep the training complexity per iteration at a manageable level approximately around the computational time and space complexity of training a few networks jointly. We include experiments on CASTLE scalability with respect to input feature size in our Experimental section.

2.3.4 Generalization bound for CASTLE regularization

In this section, we analyze theoretically why CASTLE regularization can improve the generalization performance by introducing a generalization bound for our model in Figure 2.2. Our bound is based on the PAC-Bayesian learning theory in [85, 139, 88]. Here, we re-interpret the DAG regularizer as a special prior or assumption on the input weight matrices of our model and use existing PAC-Bayes theory to prove the generalization of our algorithm. Traditionally, PAC-Bayes bounds are only applied to randomized models, such as Bayesian or Gibbs classifiers. Here, our bound is applied to our deterministic model by using the recent derandomization formalism from [96, 98]. We acknowledge and note that developing tighter and non-vacuous generalization bounds for deep neural networks is still a challenging and evolving topic in learning theory. The bounds are often stated with many constants from different steps of the proof. For reader convenience, we provide the simplified version of our bound in Theorem 1. We begin with a few assumptions before stating our bound.

Assumption 1. *For any sample* $\tilde{\boldsymbol{X}} = (Y, \boldsymbol{X}) \sim P_{\tilde{\boldsymbol{X}}}$, $\tilde{\boldsymbol{X}}$ *has bounded* ℓ_2 *norm s.t.* $\|\tilde{\boldsymbol{X}}\|_2 \leq B$, *for some* $B > 0$.

Assumption 2. *The loss function* $\mathcal{L}(f_\Theta) = \|f_\Theta(\tilde{\boldsymbol{X}}) - \tilde{\boldsymbol{X}}\|^2$ *is sub-Gaussian under the distribution* $P_{\tilde{\boldsymbol{X}}}$ *with a variance factor* s^2 *s.t.* $\forall t > 0$, $\mathbb{E}_{P_{\tilde{\boldsymbol{X}}}}\Big[\exp\Big(t\big(\mathcal{L}(f_\Theta) - \mathcal{L}_P(f_\Theta)\big)\Big)\Big] \leq \exp(\frac{t^2 s^2}{2})$.

Theorem 1. *Let $f_\Theta : \tilde{\mathcal{X}} \to \tilde{\mathcal{X}}$ be a M-layer ReLU feed-forward network with layer size h, and each of its weight matrices has the spectral norm bounded by κ. Then, under Assumptions 1 and 2, for any $\delta, \gamma > 0$, with probability $1-\delta$ over a training set of N i.i.d samples, for any Θ in (2.4), we have:*

$$\mathcal{L}_P(f_\Theta) \leq 4\mathcal{L}_N(f_\Theta) + \frac{1}{N}\left[\mathcal{R}_{\Theta_1} + C_1(\mathcal{V}_{\Theta_1} + \mathcal{V}_{\Theta_2}) + \log\left(\frac{8}{\delta}\right)\right] + C_3 \tag{2.8}$$

where $\mathcal{L}_P(f_\Theta)$ is the expected reconstruction loss of $\tilde{\boldsymbol{X}}$ under $P_{\tilde{\boldsymbol{X}}}$, $\mathcal{L}_N(f_\Theta)$, $\mathcal{V}_{\Theta_1}$ and $\mathcal{R}_{\Theta_1}$ are defined in (2.6-2.7), $\mathcal{V}_{\Theta_2}$ is the ℓ_2 norm of the network weights in the output and shared hidden layers, and C_1 and C_2 are some constants depending on γ, d, h, B, s and M.

Proof. Our proof consists of three steps: (1) We convert the existing PAC-Bayes bound for a randomized model $f_{\Theta_{\boldsymbol{u}}}$ to a deterministic model f_Θ; (2) We upper bound the KL divergence in the PAC-Bayes bound by the capability terms (i.e. the regularizers) of our model; (3) We discuss how to choose the constants in our bound to make our result universal.

Step 1. We let $\tilde{\Theta}_{\boldsymbol{u}}$ denote the Θ in which we perturb each parameter by a random perturbation u drawn from some Gaussian distribution. We collect all the random perturbation into one vector $\boldsymbol{u}$, and $\boldsymbol{u} \sim N(\boldsymbol{0}, \sigma^2 I)$. We let $Q_{\Theta_{\boldsymbol{u}}}$ denote the distribution of $\Theta_{\boldsymbol{u}}$, and $P_{\Theta_{\boldsymbol{u}}}$ denote our prior on $\Theta_{\boldsymbol{u}}$. For $\mathcal{L}_N(f_{\Theta_{\boldsymbol{u}}})$, we have

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{u}}\left[\mathcal{L}_N(f_{\Theta_{\boldsymbol{u}}})\right] &= \mathbb{E}_{\boldsymbol{u}}\left[\frac{1}{N}\sum_{i=1}^{N}\left\|f_{\Theta_{\boldsymbol{u}}}(\tilde{\boldsymbol{X}}_i) - f_\Theta(\tilde{\boldsymbol{X}}_i) + f_\Theta(\tilde{\boldsymbol{X}}_i) - \tilde{\boldsymbol{X}}_i\right\|^2\right] \\
&= \mathbb{E}_{\boldsymbol{u}}\left[\frac{1}{N}\sum_{i=1}^{N}\left\|f_{\Theta_{\boldsymbol{u}}}(\tilde{\boldsymbol{X}}_i) - f_\Theta(\tilde{\boldsymbol{X}}_i)\right\|^2\right] + \frac{1}{N}\sum_{i=1}^{N}\left\|f_\Theta(\tilde{\boldsymbol{X}}_i) - \tilde{\boldsymbol{X}}_i\right\|^2 \\
&\quad + \mathbb{E}_{\boldsymbol{u}}\left[\frac{2}{N}\sum_{i=1}^{N}\left(f_{\Theta_{\boldsymbol{u}}}(\tilde{\boldsymbol{X}}_i) - f_\Theta(\tilde{\boldsymbol{X}}_i)\right)\left(f_\Theta(\tilde{\boldsymbol{X}}_i) - \tilde{\boldsymbol{X}}_i\right)\right] \\
&\leq \mathbb{E}_{\boldsymbol{u}}\left[\frac{1}{N}\sum_{i=1}^{N}\left\|f_{\Theta_{\boldsymbol{u}}}(\tilde{\boldsymbol{X}}_i) - f_\Theta(\tilde{\boldsymbol{X}}_i)\right\|^2\right] + \frac{1}{N}\sum_{i=1}^{N}\left\|f_\Theta(\tilde{\boldsymbol{X}}_i) - \tilde{\boldsymbol{X}}_i\right\|^2 \\
&\quad + \mathbb{E}_{\boldsymbol{u}}\left[\frac{1}{N}\sum_{i=1}^{N}\left\|f_{\Theta_{\boldsymbol{u}}}(\tilde{\boldsymbol{X}}_i) - f_\Theta(\tilde{\boldsymbol{X}}_i)\right\|^2\right] + \frac{1}{N}\sum_{i=1}^{N}\left\|f_\Theta(\tilde{\boldsymbol{X}}_i) - \tilde{\boldsymbol{X}}_i\right\|^2 \\
&\leq 2\gamma + 2\mathcal{L}_N(f_\Theta)
\end{aligned} \tag{2.9}$$

Similarly, we have

$$\begin{aligned}\mathcal{L}_P(f_\Theta) &= \mathbb{E}_P\mathbb{E}_{\boldsymbol{u}}\left[\left\|f_\Theta(\tilde{\boldsymbol{X}}) - f_{\Theta_{\boldsymbol{u}}}(\tilde{\boldsymbol{X}}) + f_{\Theta_{\boldsymbol{u}}}(\tilde{\boldsymbol{X}}) - \tilde{\boldsymbol{X}}\right\|^2\right] \\ &\leq 2\gamma + 2\mathbb{E}_{\boldsymbol{u}}\left[L_P(f_{\Theta_{\boldsymbol{u}}})\right]\end{aligned} \tag{2.10}$$

where we let γ be a constant such that $\max_{\tilde{\boldsymbol{X}}\in\mathcal{X}} \mathbb{E}_{\boldsymbol{u}}\left[\|f_{\Theta_{\boldsymbol{u}}}(\tilde{\boldsymbol{X}}) - f_\Theta(\tilde{\boldsymbol{X}})\|^2\right] \leq \gamma$. It is the upper bound for the maximum expected change of the network output when the weights are perturbed, thereby the network's sharpness as defined in [70].

Using the Corollary 4 in [42] and Lemma 1 in [96], we have the following PAC Bayes bound for the randomized model $f_{\Theta_{\boldsymbol{u}}}$. Given a prior distribution $P_{\Theta_{\boldsymbol{u}}}$ over the set of predictors that is independent of the training data, the PAC-Bayes theorem states that with probability at least $1-\delta$, over N i.i.d training samples, the expected error of $f_{\Theta_{\boldsymbol{u}}}$ can be bounded as follows,

$$\mathbb{E}_{\boldsymbol{u}}\left[\mathcal{L}_P(f_{\Theta_{\boldsymbol{u}}})\right] \leq \mathbb{E}_{\boldsymbol{u}}\left[\mathcal{L}_N(f_{\Theta_{\boldsymbol{u}}})\right] + \tfrac{1}{N}\left[2\,\mathrm{KL}(Q_{\Theta_{\boldsymbol{u}}}\|P_{\Theta_{\boldsymbol{u}}}) + \log(\tfrac{8}{\delta})\right] + \tfrac{1}{2}s^2 \tag{2.11}$$

If we upper bound $\mathbb{E}_{\boldsymbol{u}}\left[\mathcal{L}_P(f_{\Theta_{\boldsymbol{u}}})\right]$ in (2.10) by (2.11), we have

$$\begin{aligned}\mathcal{L}_P(f_\Theta) &\leq 2\gamma + 2\mathbb{E}_{\boldsymbol{u}}\left[\mathcal{L}_N(f_{\Theta_{\boldsymbol{u}}})\right] + \tfrac{2}{N}\left[2\,\mathrm{KL}(Q_{\Theta_{\boldsymbol{u}}}\|P_{\Theta_{\boldsymbol{u}}}) + \log(\tfrac{8}{\delta})\right] + s^2 \\ &\leq 4\mathcal{L}_N(f_\Theta) + \tfrac{2}{N}\left[2\,\mathrm{KL}(Q_{\Theta_{\boldsymbol{u}}}\|P_{\Theta_{\boldsymbol{u}}}) + \log(\tfrac{8}{\delta})\right] + C_2\end{aligned} \tag{2.12}$$

where the last inequality is achieved by (2.9), and $C_2 = s^2 + 6\gamma$.

Step 2. For convenience, we restate the parameter set Θ in (2.4) here,

$$\Theta_1 = \{\mathbf{W}_1^k\}_{k=1}^{d+1},\ \Theta = \Theta_1 \cup \{\mathbf{W}_m\}_{k=2}^{M}$$

Now we write the distribution $Q_{\Theta_{\boldsymbol{u}}}$ and $P_{\Theta_{\boldsymbol{u}}}$ explicitly. Without loss of generality, we assume $Q_{\Theta_{\boldsymbol{u}}}$ and $P_{\Theta_{\boldsymbol{u}}}$ have the same standard deviation σ^2. First, $Q_{\Theta_{\boldsymbol{u}}}$ is given as $Q_{\Theta_{\boldsymbol{u}}} = Q^{(1)}_{\Theta_{\boldsymbol{u}}} Q^{(2)}_{\Theta_{\boldsymbol{u}}}$, where $Q^{(1)}_{\Theta_{\boldsymbol{u}}} = N(z_{\Theta_{\boldsymbol{u},1}}; z_{\Theta_1}, 1)$, and

$$Q^{(2)}_{\Theta_{\boldsymbol{u}}} = \prod_{k=1}^{d+1} N(\mathbf{W}^k_{\boldsymbol{u},1}; \mathbf{W}^k_1, \sigma^2 I) \prod_{m=2}^{M} N(\mathbf{W}_{\boldsymbol{u},m}; \mathbf{W}_m, \sigma^2 I).$$

And P_Θ is given as $P_{\Theta_{\boldsymbol{u}}} = P^{(1)}_{\Theta_{\boldsymbol{u}}} P^{(2)}_{\Theta_{\boldsymbol{u}}}$, where $P^{(1)}_{\Theta_{\boldsymbol{u}}} = N(z_{\Theta_{\boldsymbol{u},1}}; d+1, 1)$, and

$$P^{(2)}_{\Theta_{\boldsymbol{u}}} = \prod_{k=1}^{d+1} N(\mathbf{W}^k_{\boldsymbol{u},1}; \mathbf{0}, \sigma^2 I) \prod_{m=2}^{M} N(\mathbf{W}_{\boldsymbol{u},m}; \mathbf{0}, \sigma^2 I).$$

The variable $z_{\Theta_{\boldsymbol{u}},1}$ is given as,

$$z_{\Theta_{\boldsymbol{u}},1} = \mathrm{Tr}\big(e^{\mathbf{M}_{\boldsymbol{u}} \odot \mathbf{M}_{\boldsymbol{u}}}\big)$$

where $\mathbf{M}_{\boldsymbol{u}}$ is a $(d+1) \times (d+1)$ matrix such that $[\mathbf{M}_{\boldsymbol{u}}]_{k,j}$ is the ℓ_2-norm of the k-th row of the matrix $\mathbf{W}^j_{\boldsymbol{u},1}$. The variable z_{Θ_1} is defined in the same way as $z_{\Theta_{\boldsymbol{u}},1}$ but on the parameters without perturbations. Here, we use Gaussian distributions for z's for simplicity in our deterministic model. Formally, in Bayesian inference, we may consider using truncated normal or exponential priors for z's since we know $z_{\Theta_{\boldsymbol{u}},1} = \mathrm{Tr}(I) + \mathrm{Tr}(\mathbf{M}_{\boldsymbol{u}} \odot \mathbf{M}_{\boldsymbol{u}}) + \cdots \geq d+1$ using the power series of matrix exponential and the fact that each element of $\mathbf{M}_{\boldsymbol{u}}$ is non-negative. Now we upper bound the KL divergence as follows,

$$\begin{aligned}
\mathrm{KL}(Q_{\Theta_{\boldsymbol{u}}} \| P_{\Theta_{\boldsymbol{u}}}) &= \int Q^{(1)}_{\Theta_{\boldsymbol{u}}} Q^{(2)}_{\Theta_{\boldsymbol{u}}} \log\left(\frac{Q^{(1)}_{\Theta_{\boldsymbol{u}}} Q^{(2)}_{\Theta_{\boldsymbol{u}}}}{P^{(1)}_{\Theta_{\boldsymbol{u}}} P^{(2)}_{\Theta_{\boldsymbol{u}}}}\right) d\Theta_{\boldsymbol{u}} \\
&= \int Q^{(1)}_{\Theta_{\boldsymbol{u}}} Q^{(2)}_{\Theta_{\boldsymbol{u}}} \log\left(\frac{Q^{(1)}_{\Theta_{\boldsymbol{u}}}}{P^{(1)}_{\Theta_{\boldsymbol{u}}}}\right) d\Theta_{\boldsymbol{u}} + \int Q^{(1)}_{\Theta_{\boldsymbol{u}}} Q^{(2)}_{\Theta_{\boldsymbol{u}}} \log\left(\frac{Q^{(2)}_{\Theta_{\boldsymbol{u}}}}{P^{(2)}_{\Theta_{\boldsymbol{u}}}}\right) d\Theta_{\boldsymbol{u}} \\
&\leq \int Q^{(1)}_{\Theta_{\boldsymbol{u}}} \log\left(\frac{Q^{(1)}_{\Theta_{\boldsymbol{u}}}}{P^{(1)}_{\Theta_{\boldsymbol{u}}}}\right) d\Theta_{\boldsymbol{u}} + \int Q^{(2)}_{\Theta_{\boldsymbol{u}}} \log\left(\frac{Q^{(2)}_{\Theta_{\boldsymbol{u}}}}{P^{(2)}_{\Theta_{\boldsymbol{u}}}}\right) d\Theta_{\boldsymbol{u}} \\
&= \tfrac{1}{2}\big[z_{\Theta_1} - (d+1)\big]^2 + \tfrac{1}{2\sigma^2}\Big(\sum_{k=1}^{d+1} \|\mathbf{W}_1^k\|_F^2 + \sum_{m=2}^{M} \|\mathbf{W}_m\|_F^2\Big) \\
&\leq \tfrac{1}{2}\mathcal{R}_{\theta_1} + \tfrac{1}{2\sigma^2}(\mathcal{V}_{\Theta_1} + \mathcal{V}_{\Theta_2})
\end{aligned} \tag{2.13}$$

where the last inequality is achieved using the fact that the Euclidean norm of any vector is bounded by its ℓ_1-norm. Let $C_1 = \frac{1}{\sigma^2}$. Bounding the KL divergence in (2.12) with (2.13) gives that

$$\mathcal{L}_P(f_\Theta) \leq 4\mathcal{L}_N(f_\Theta) + \tfrac{2}{N}\Big[\mathcal{R}_{\theta_1} + C_1(\mathcal{V}_{\Theta_1} + \mathcal{V}_{\Theta_2}) + \log(\tfrac{8}{\delta})\Big] + C_2 \tag{2.14}$$

Step 3. Recall that γ is the upper bound for $\max_{\tilde{\boldsymbol{X}} \in \mathcal{X}} \mathbb{E}_{\boldsymbol{u}}\big[\|f_{\Theta_{\boldsymbol{u}}}(\tilde{\boldsymbol{X}}) - f_\Theta(\tilde{\boldsymbol{X}})\|^2\big]$, the expected maximum change of the network output when the weights are perturbed by $\boldsymbol{u} \sim N(\mathbf{0}, \sigma^2 I)$. We now derive the constant γ based on σ^2, the input upper bound B in Assumption 1. Our network uses ReLU activation functions in the hidden layers. The ReLU function $\phi(\cdot)$ is 1-Lipschitz. This proof is similar to Lemma 2 in [96]. Let $\|\cdot\|_2$ denote the

spectral norm. We define Δ_k^{M-1} as the output difference in the last hidden layer:

$$\Delta_k^{M-1} = \big\|\phi\big(\cdots\phi\big(\phi\big(\tilde{\boldsymbol{X}}[\mathbf{W}_1^k + \mathbf{U}_1^k]\big)[\mathbf{W}_2 + \mathbf{U}_2]\big)\cdots[\mathbf{W}_{M-1} + \mathbf{U}_{M-1}]\big) - \phi\big(\cdots\phi\big(\phi\big(\tilde{\boldsymbol{X}}\mathbf{W}_1^k\big)\mathbf{W}_2\big)\cdots\mathbf{W}_{M-1}\big)\big\|$$

We have

$$\begin{aligned}
\Delta_k^M &= \big([f_\Theta(\tilde{\boldsymbol{X}})]_k - [f_{\Theta_{\boldsymbol{u}}}(\tilde{\boldsymbol{X}})]_k\big)^2 \\
&= \Delta_k^{M-1}\big[\|\mathbf{W}_M\|_2 + \|\mathbf{U}_M\|_2\big] + \|\tilde{\boldsymbol{X}}\|\|\mathbf{U}_M\|_2\|\mathbf{W}_1^k\|_2\prod_{m=2}^{M-1}\|\mathbf{W}_m\|_2 \\
&\le (1+\tfrac{1}{M})\|\mathbf{W}_M\|_2\Delta_k^{M-1} + \tfrac{\|\mathbf{U}_M\|_2}{\|\mathbf{W}_M\|_2}\|\tilde{\boldsymbol{X}}\|\|\mathbf{W}_1^k\|_2\prod_{m=2}^{M}\|\mathbf{W}_m\|_2 \\
&\le (1+\tfrac{1}{M})\|\mathbf{W}_M\|_2\Big((1+\tfrac{1}{M})\|\mathbf{W}_{M-1}\|_2\Delta_k^{M-2} + \tfrac{\|\mathbf{U}_{M-1}\|_2}{\|\mathbf{W}_{M-1}\|_2}\|\tilde{\boldsymbol{X}}\|\|\mathbf{W}_1^k\|_2\prod_{m=2}^{M-1}\|\mathbf{W}_m\|_2\Big) \\
&\quad + \tfrac{\|\mathbf{U}_M\|_2}{\|\mathbf{W}_M\|_2}\|\tilde{\boldsymbol{X}}\|\|\mathbf{W}_1^k\|_2\prod_{m=2}^{M}\|\mathbf{W}_m\|_2 \\
&\le (1+\tfrac{1}{M})^2\Delta_k^{M-2}\prod_{m=M-1}^{M}\|\mathbf{W}_m\|_2 + \sum_{m=0}^{1}(1+\tfrac{1}{M})^m\tfrac{\|\mathbf{U}_{M-m}\|_2}{\|\mathbf{W}_{M-m}\|_2}\|\tilde{\boldsymbol{X}}\|\|\mathbf{W}_1^k\|_2\prod_{m=2}^{M}\|\mathbf{W}_m\|_2 \\
&\le (1+\tfrac{1}{M})^M\|\tilde{\boldsymbol{X}} - \tilde{\boldsymbol{X}}\|F_k + (1+\tfrac{1}{M})^{M-1}\tfrac{\|\mathbf{U}_1^k\|_2}{\|\mathbf{W}_1^k\|_2}\|\tilde{\boldsymbol{X}}\|F_k \\
&\quad + \sum_{m=0}^{M-2}(1+\tfrac{1}{M})^m\tfrac{\|\mathbf{U}_{M-m}\|_2}{\|\mathbf{W}_{M-m}\|_2}\|\tilde{\boldsymbol{X}}\|F_k \\
&\le eBF_k\left(\tfrac{\|\mathbf{U}_1^k\|_2}{\|\mathbf{W}_1^k\|_2} + \sum_{m=2}^{M}\tfrac{\|\mathbf{U}_m\|_2}{\|\mathbf{W}_m\|_2}\right)
\end{aligned}$$

where $F_k = \|\mathbf{W}_1^k\|_2\prod_{m=2}^{M}\|\mathbf{W}_m\|_2$, the last inequality is achieved by $(1+\frac{1}{m})^M \le e$ for $m \le M$, and $\|\tilde{\boldsymbol{X}}\| \le B$ in Assumption 1. Then $\mathbb{E}_{\boldsymbol{u}}\big[\|f_{\Theta_{\boldsymbol{u}}}(\tilde{\boldsymbol{X}}) - f_\Theta(\tilde{\boldsymbol{X}})\|^2\big]$ is given as

$$\begin{aligned}
\sum_{k=1}^{d+1}\mathbb{E}_{\boldsymbol{u}}\Big[\Delta_k^M\Big] &\le \sigma reB\sum_{k=1}^{d+1}F_k\left(\|\mathbf{W}_1^k\|_2^{-1} + \sum_{m=2}^{M}\|\mathbf{W}_m\|_2^{-1}\right) \\
&\le \sigma reB\sum_{k=1}^{d+1}\left(\prod_{m=2}^{M}\|\mathbf{W}_m\|_2 + \sum_{m=2}^{M}\tfrac{F_k}{\|\mathbf{W}_m\|_2}\right) \\
&\le \sigma reB(d+1)M\kappa^{M-1}
\end{aligned}$$

where $r = \big[2\log(2eh)\big]^{1/2}$, and the first inequality is achieved bounding the spectral norm of the random matrices $\mathbf{U}$'s using random matrix theory (See Section 4.4 in [147]). Hence,

setting $\sigma = (reB(d+1)M\kappa^{M-1})^{-1}\gamma$, then we have

$$\max_{\tilde{\boldsymbol{X}} \in \mathcal{X}} \mathbb{E}_{\boldsymbol{u}} \left[\| f_{\Theta_{\boldsymbol{u}}}(\tilde{\boldsymbol{X}}) - f_{\Theta}(\tilde{\boldsymbol{X}}) \|^2 \right] < \gamma.$$

Given any ReLU network satisfying the Assumptions 1 and 2 and with bounded spectral norm on its weights, we can upper bound its expected loss using the network sharpness, measured by some perturbations on the network parameters. □

The statistical properties of the reconstruction loss in learning linear DAGs, e.g. $\mathcal{L}_{\mathbf{W}} = \frac{1}{N}\|\tilde{\mathbf{X}} - \mathbf{W}\tilde{\mathbf{X}}\|_F^2$, have been well studied in the literature: the loss minimizer provably recovers a true DAG with high probability on finite-samples, and hence is consistent for both Gaussian SEM [77] and non-Gaussian SEM [1, 150]. Note also that the regularizer $\mathcal{R}_{\mathbf{W}}$ or $\mathcal{R}_{\Theta_1}$ are not a part of the results in [77, 1, 150]. However, the works of [170, 174] empirically show that using $\mathcal{R}_{\mathbf{W}}$ or $\mathcal{R}_{\Theta_1}$ on top of the reconstruction loss leads to more efficient and more accurate DAG learning than existing approaches. Our theoretical result on the reconstruction loss explains the benefit of $\mathcal{R}_{\mathbf{W}}$ or $\mathcal{R}_{\Theta_1}$ for the generalization performance of predicting Y. This provides theoretical support for our CASTLE regularizer in supervised learning. However, the objectives of DAG discovery, e.g., identifying the Markov Blanket of Y, is beyond the scope of our analysis.

The bound in (2.8) justifies $\mathcal{R}_{\Theta_1}$ in general, including linear or nonlinear cases, if the underlying distribution $P_{\tilde{\boldsymbol{X}}}$ is factorized according to some causal DAG. We note that the expected loss $\mathcal{L}_P(f_\Theta)$ is upper bounded by the empirical loss $\mathcal{L}_N(f_\Theta)$, $\mathcal{V}_{\Theta_1}$, $\mathcal{V}_{\Theta_1}$ and $\mathcal{R}_{\Theta_1}$ which measures how close (via acyclicity constraint) the model is to a DAG. From (2.8) it is obvious that not minimizing $\mathcal{R}_{\Theta_1}$ is an acceptable strategy asymptotically or in the large samples limit (large N) because $^{\mathcal{R}_{\Theta_1}}/_N$ becomes negligible. This aligns with the consistency theory in [77, 1, 150] for linear models. However for small N, a preferred strategy is to train a model f_Θ by minimizing $\mathcal{L}_N(f_\Theta)$ and $\mathcal{R}_{\Theta_1}$ jointly. This would be trivial because the samples are generated under the DAG structure in $P_{\tilde{\boldsymbol{X}}}$. Minimizing $\mathcal{R}_{\Theta_1}$ can decrease the upper bound of $\mathcal{L}_P(f_\Theta)$ in (2.8), improve the generalization performance of f_Θ, as well as facilitate the convergence of f_Θ to the true model.

If $P_{\tilde{\boldsymbol{X}}}$ does not correspond to any causal DAG, such as image data, then there will be a trade-off between minimizing $\mathcal{R}_{\Theta_1}$ and $\mathcal{L}_N(f_\Theta)$. In this case, $\mathcal{R}_{\Theta_1}$ becomes harder to minimize, and generalization may not benefit from adding CASTLE. However, this is a rare case since causal structure exists in most datasets inherently. Our experiments demonstrate that CASTLE regularization outperforms popular regularizers on a variety of datasets in the next section.

2.4 Experiments

In this section, we empirically evaluate CASTLE as a regularization method for improving generalization performance. We present our benchmark methods and training architecture, followed by our synthetic and publicly available data results.

Benchmarks. We benchmark CASTLE against common regularizers that include: early stopping (Baseline) [41], L1 [145], L2 [54], dropout [56] with drop rate of 20% and 50% denoted as DO(0.2) and DO(0.5) respectively, SAE [84], batch normalization (BN) [58], data augmentation or input noise (IN) [73], and MixUp (MU) [173], in no particular order. For each regularizer with tunable hyperparameters we performed a standard grid search. For the weight decay regularizers L1 and L2 we searched for $\lambda_{\ell_p} \in \{0.1, 0.01, 0.001\}$, and for input noise we use a Gaussian noise with mean of 0 and standard deviation $\sigma \in \{0.1, 0.01, 0.01\}$. L1 and L2 were applied at every dense layer. BN and DO were applied after every dense layer and active only during training. Because each regularization method converges at different rates, we use early stopping on a validation set to terminate each benchmark training, which we refer to as our Baseline.

Network architecture and training. We implemented CASTLE in `Tensorflow`. Our proposed architecture is comprised of $d+1$ sub-networks with shared hidden layers, as shown in Figure 2.2. In the linear case, $\mathcal{V}_{\mathbf{W}}$ is the ℓ_1 norm of $\mathbf{W}$. In the nonlinear case, $\mathcal{V}_{\Theta_1}$ is the ℓ_1 norm of the input weight matrices $\mathbf{W}_1^k, k \in [d+1]$. To make a clear comparison with L2

regularization, we exclude the capacity term $\mathcal{V}_{\Theta_2}$ from CASTLE, although it is a part of our generalization bound in (2.8). Since we predict the target variable as our primary task, we benchmark CASTLE against this common network architecture. Specifically, we use a network with two hidden layers of $d + 1$ neurons with ReLU activation. Each benchmark method is initialized and seeded identically with the same random weights. For dataset preprocessing, all continuous variables are standardized with a mean of 0 and a variance of 1. Each model is trained using the Adam optimizer with a learning rate of 0.001 for up to a maximum of 200 epochs. An early stopping regime halts training with a patience of 30 epochs.

2.5 Synthetic details

In this section, we cover details regarding our synthetic data generation process and experiments. We first provide an overview of our data generation, and then we will cover a supplementary linear example.

2.5.1 Synthetic data generating process

Here we describe our synthetic data generation process in detail. We enumerated all nodes in G randomly. We generated random DAG instantiations with a randomly sampled branching factor up to the number of nodes in the DAG for our synthetic DAG generation. Edges were randomly added to the graph until either the branching factor was met or no more edges can be added without violating graphical acyclicity. We provide pseudocode for our synthetic DGP in Algorithm 1. For each random DAG in our experiment we randomly chose a σ between 0.3 and 1, and we set $\mu = 0$ and $w = 1$.

For our experiments in the main paper, we use the following settings. In the linear case, each variable is equal to the sum of its parents plus noise. For the nonlinear case, each variable is equal to the sum of the sigmoid function of each parent plus noise.

Algorithm 1 Synthetic Data Generating Process (DGP)

Input: A Graphical structure G, a mean μ, standard deviation σ, edge weights w, a dataset size n.

Output: A dataset according to G with n samples.

Function: `gen_data`(G, μ, σ, w, n)**:**

$e \leftarrow$ edges of G

$G_{sorted} \leftarrow$ `topological_graph_sort`(G)

$ret \leftarrow$ empty list

for $node \in G$ **do**

Append to $ret[node]$ a list of Gaussian (μ and σ) randomly sampled list of size n.

end for

for $node \in G_{sorted}$ **do**

for $par \in \{parents(node)\}$ **do**

$ret[node] \mathrel{+}= ret[par] * w(par, node)$, where $w(par, node)$ is the edge weight from par to $node$. Note that a non-linear function can be applied to $ret[par]$ to convert this into a non-linear data generator.

end for

end for

return ret.

2.5.2 Regularization on Synthetic Data

Table 2.2: Experiments on nonlinear synthetic data of size n generated according to Fig. 2.1 in terms of MSE ($\pm$ standard deviation)

Regularizer	$n = 500$	$n = 1000$	$n = 5000$
Baseline	0.83 ± 0.03	0.80 ± 0.04	0.73 ± 0.02
L1	0.81 ± 0.05	0.79 ± 0.03	0.71 ± 0.02
L2	0.81 ± 0.05	0.77 ± 0.02	0.71 ± 0.01
DO(0.2)	0.80 ± 0.04	0.79 ± 0.01	0.70 ± 0.02
DO(0.5)	0.79 ± 0.02	0.78 ± 0.04	0.70 ± 0.02
SAE	0.79 ± 0.03	0.77 ± 0.04	0.69 ± 0.02
BN	0.81 ± 0.04	0.79 ± 0.03	0.72 ± 0.02
IN	0.82 ± 0.05	0.78 ± 0.04	0.71 ± 0.02
MU	0.79 ± 0.05	0.78 ± 0.04	0.72 ± 0.08
CASTLE	$\mathbf{0.77 \pm 0.02}$	$\mathbf{0.75 \pm 0.04}$	$\mathbf{0.68 \pm 0.02}$

Given a DAG G, we generate functional relationships between each variable and its respective parent(s) with additive Gaussian noise applied to each variable with a mean of 0 and variance of 1. In the linear case, each variable is equal to the sum of its parents plus noise. For the nonlinear case, each variable is equal to the sum of the sigmoid of its parents plus noise. Consider Table 2.2, using our nonlinear DGP we generated 1000 test samples according to the DAG in Figure 2.1. We then used 10-fold cross-validation to train and validate each benchmark on varying training sets of size n. Each model was evaluated on the test set from weights saved at the lowest validation error. Table 2.2 shows that CASTLE improves over all experimental benchmarks.

Dissecting CASTLE. In the synthetic environment, we know the causal relationships with certainty. We analyze three aspects of CASTLE regularization using synthetic data. Because we are comparing across randomly simulated DAGs with differing functional relationships,

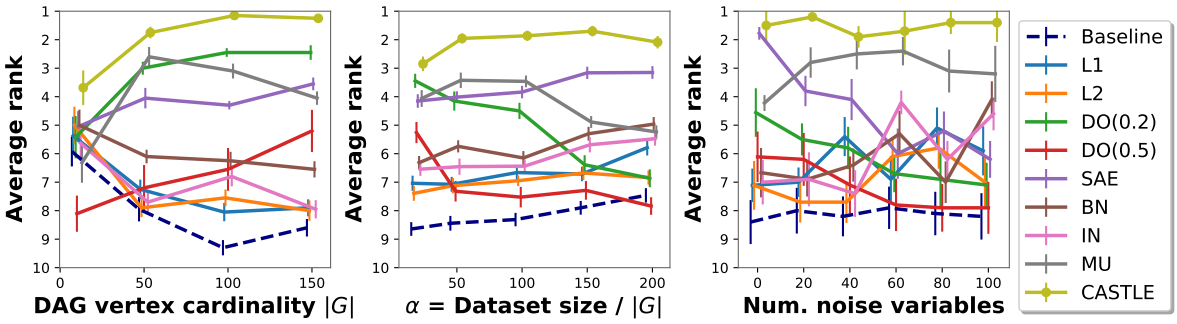
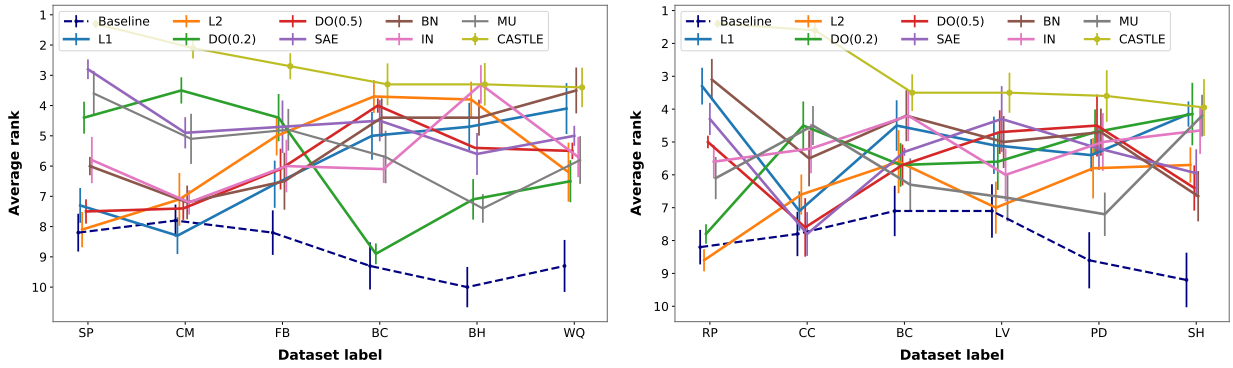


Figure 2.3: Experiments on synthetic data. The y-axis is the average rank ($\pm$ standard deviation) of each regularizer on the test set over each synthetic DAG. We show the average rank as we increase the number of features or vertex cardinality $|G|$ (**left**), increase the dataset size normalized by the vertex cardinality $|G|$ (**center**), and as we increase the number of noise (neighborless) variables (**right**).

the magnitude of regression testing error will vary between runs. We examine the model performance in terms of each model's average rank over each fold to normalize this. If we have r regularizers, the best and worst possible rank is one and r, respectively (i.e., the higher the rank the better). We used 10-fold cross-validation to terminate model training and tested each model on a held-out test set of 1000 samples.

First, we examine the impact of increasing the feature size or DAG vertex cardinality $|G|$. We do this by randomly generating a DAG of size $|G| \in \{10, 50, 100, 150\}$ with $50|G|$ training samples. We repeat this ten times for each DAG cardinality. On the left-hand side of Fig. 2.3, CASTLE has the highest rank of all benchmarks and does not degrade with increasing $|G|$. Second, we analyze the impact of increasing dataset size. We randomly generate DAGs of size $|G| \in \{10, 50, 100, 150\}$, which we use to create datasets of $\alpha|G|$ samples, where $\alpha \in \{20, 50, 100, 150, 200\}$. We repeat this ten times for each dataset size. In the middle plot of Figure 2.3, we see that CASTLE has superior performance for all dataset sizes, and as expected, all benchmark methods (except for SAE) start to converge about the average rank at large data sizes ($\alpha = 200$). Third, we analyze our method's sensitivity to noise variables, i.e., variables disconnected to the target variable in G. We randomly generate

DAGs of size $|G| = 50$ to create datasets with $50|G|$ samples. We randomly add $v \in \{20i\}_{i=0}^{5}$ noise variables normally distributed with 0 mean and unit variance. We repeat this process for ten different DAG instantiations. The results on the right-hand side of Figure 2.3 show that our method is not sensitive to the existence of disconnected noise variables, whereas SAE performance degrades with the increase of uncorrelated input features. This highlights the benefit of target selection based on the DAG topology.



(a) Regression

(b) Classification

Figure 2.4: Comparison of CASTLE against benchmark regularization methods in terms or average rank across each fold (10-fold cross-validation) for regression (a) and classification (b) tasks. For clarity, we have sorted the datasets by average rank of CASTLE in decreasing order. In comparison to the other benchmarks, CASTLE maintains stable performance across datasets. Higher rank is better.

2.5.3 Weight characterization

In this subsection, we provide a characterization of the input weights that are learned during the CASTLE regularization. We performed synthetic experiments using the same setup for generating Figure 2.3. We investigated two different scenarios. In the first scenario, we randomly generated DAGs where the target must have causal parents. We examine the average weight value of the learned DAG adjacency matrix in comparison to the truth adjacency matrix for the parents, children, spouses, and siblings of the target variable. The

Table 2.3: Complete table of benchmark regularizers on regression in terms of test MSE ($\pm$ standard deviation) for experiments on real datasets using 10-fold cross-validation. Bold denotes lowest test MSE. For readability we split the table into two.

$\mathcal{D}$	Baseline	L1	L2	Dropout 0.2	Dropout 0.5
BH	0.141 ± 0.023	0.137 ± 0.025	0.131 ± 0.014	0.168 ± 0.032	0.389 ± 0.106
WQ	0.747 ± 0.038	0.747 ± 0.043	0.746 ± 0.039	0.738 ± 0.029	0.850 ± 0.068
FB	0.758 ± 1.017	0.663 ± 0.796	1.341 ± 1.069	0.429 ± 0.449	0.597 ± 0.313
BC	0.359 ± 0.061	0.342 ± 0.037	0.370 ± 0.142	0.334 ± 0.030	0.434 ± 0.080
SP	0.416 ± 0.108	0.421 ± 0.181	0.550 ± 0.291	0.285 ± 0.042	0.482 ± 0.128
CM	0.536 ± 0.103	0.574 ± 0.125	0.527 ± 0.060	0.327 ± 0.025	0.519 ± 0.064

$\mathcal{D}$	SAE	Batch Norm	Input Noise	MixUp	CASTLE
BH	0.148 ± 0.027	0.139 ± 0.021	0.137 ± 0.018	0.194 ± 0.064	$\mathbf{0.123 \pm 0.016}$
WQ	0.727 ± 0.030	0.723 ± 0.039	0.771 ± 0.036	0.712 ± 0.018	$\mathbf{0.708 \pm 0.030}$
FB	0.372 ± 0.168	0.705 ± 0.396	0.609 ± 0.511	0.385 ± 0.208	$\mathbf{0.246 \pm 0.153}$
BC	0.322 ± 0.021	0.325 ± 0.024	0.319 ± 0.022	0.322 ± 0.030	$\mathbf{0.318 \pm 0.036}$
SP	0.228 ± 0.022	0.318 ± 0.062	0.389 ± 0.095	0.267 ± 0.072	$\mathbf{0.200 \pm 0.020}$
CM	0.387 ± 0.034	0.470 ± 0.047	0.495 ± 0.081	0.376 ± 0.030	$\mathbf{0.326 \pm 0.031}$

results are shown in Figure 2.5. As expected, the results show that when causal parents exist, CASTLE prefers to predict in the causal direction, rather than the anti-causal direction (from children).

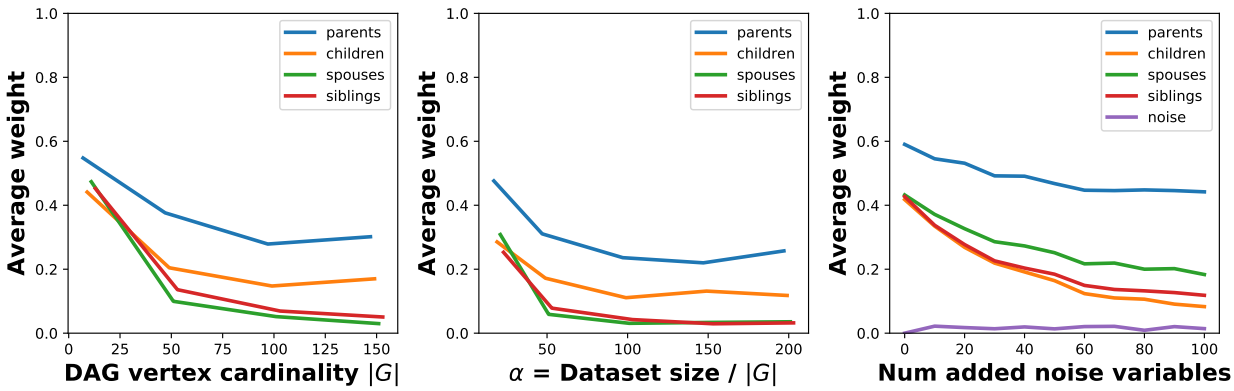


Figure 2.5: Weight values on synthetic data when true causal structure is known. Our method favors using the parents of the target when available.

As a secondary experiment, we ran the same sets of experiments, except for DAGs without parents of the target variable. Results are shown in Figure 2.6. The results show that when parents are not available that CASTLE finds the children as predictors rather than spouses. Note that in this experiment, there will be no siblings of the target variable, since the target variable has no parents.

Lastly, CASTLE does not reconstruct features that do not have causal neighbors in the discovered DAG. To highlight this, in our noise variable experiment, we show the average weighting of the input layers. In the right-most figures of Figure 2.5 and Figure 2.6, it is evident that the weighting is much lower (near zero) for the noise variables in comparison to the other variables in the DAG. This highlights the advantages of CASTLE over SAE, which naively reconstructs all variables.

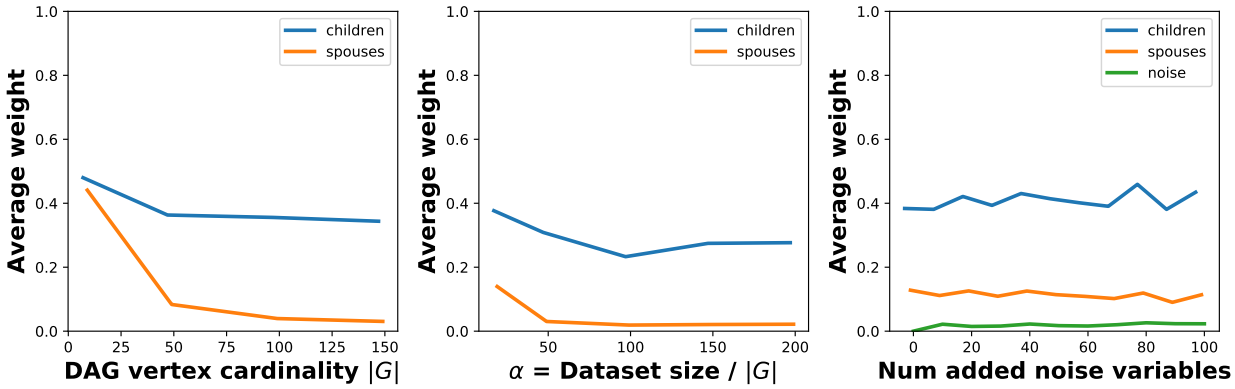


Figure 2.6: Weight values on synthetic data when true causal structure is known. This simulation was run with target variables not having any causal parents (and therefore no siblings as well). Our method favors using the children rather than spouses of the target.

2.5.4 Sensitivity analysis and hyperparameter optimization

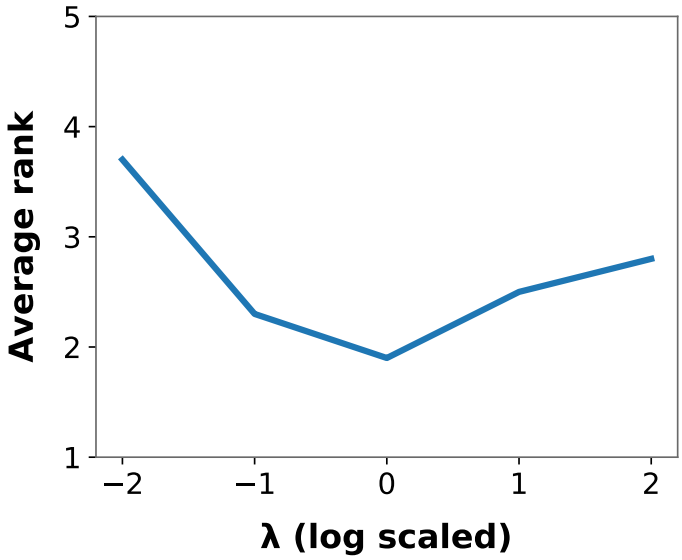


Figure 2.7: Sensitivity analysis on λ.

Before we present further results, we first provide a sensitivity analysis on λ from (2.5). We use our synthetic DGP to synthesize a random DAG with between 10 and 150 nodes. We

generated 2000 test samples and a training set with between 1000 and 5000 samples. We repeated this 50 times. Using 10-fold cross-validation we show a sensitivity analysis over $\lambda \in \{0.01, 0.1, 1, 10, 100\}$ in Figure 2.7 in terms of average rank. We compare using average rank since each experimental run (random DAG) will vary significantly in the magnitude of errors. Based on these results, for all of our experiments in this paper we use $\lambda = 1$, i.e., $\log(\lambda) = 0$. After fixing λ, our model has only one hyperparameter β to tune. For β in (2.6), we performed a standard grid search for the hyperparameter $\beta \in \{0.001, 0.01, 0.1, 1\}$.

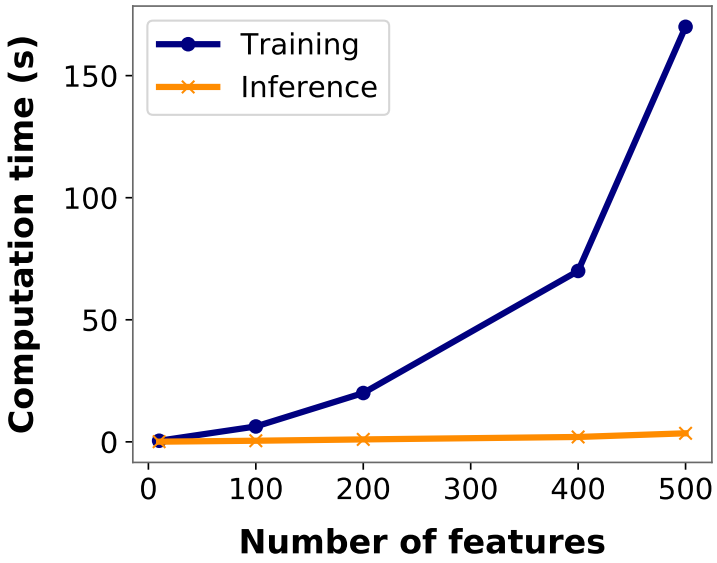


Figure 2.8: CASTLE scalability analysis

2.5.5 Scalability analysis

We perform an analysis of the scalability of CASTLE. Using our synthetic DAG and dataset generator, we synthesized datasets of 1000 samples. We used the same experimental setup used for the synthetic experiments. We present the computational timing results for CASTLE as we increase the number of input features on inference and training time in Figure 2.8. We see that the time to train 1000 samples grows exponentially with the feature size; however, the inference time remains linear as expected. Inference time on 1000 samples with 400 features

takes approximately 2 seconds, while training time takes nearly 70 seconds. Computational time scales linearly with increasing the number of input samples. Experiments were conducted on an Ubuntu 18.04 OS using 6 Intel i7-6850K CPUs.

2.5.6 Regularization on Real Data

We perform regression and classification experiments on a spectrum of publicly available datasets from [31] including Boston Housing (BH), Wine Quality (WQ), Facebook Metrics (FB), Bioconcentration (BC), Student Performance (SP), Community (CM), Contraception Choice (CC), Pima Diabetes (PD), Las Vegas Ratings (LV), Statlog Heart (SH), and Retinopathy (RP). For each dataset, we randomly reserve 20% of the samples for a testing set. We perform 10-fold cross-validation on the remaining 80%. As the results show in Table 2.3, CASTLE provides improved regularization across all datasets for both regression and classification tasks. Additionally, CASTLE consistently ranks as the top regularizer, with no definitive benchmark method coming in as a consensus runner-up. This emphasizes the stability of CASTLE as a reliable regularizer.

2.5.7 Dataset details

In Table 2.4, we provide details of the real world datasets used in this paper. We demonstrated improved performance by CASTLE across a diverse collection of datasets in terms of sample and feature size.

2.5.8 CASTLE ablation study

We provide an ablation study on CASTLE to understand the sources of gain of our methodology. Here we execute this experiment on our real datasets used in the main manuscript. We show the results of our ablation on our CASTLE regularizer to highlight our sources of gain in Table 2.5.

Table 2.4: Real-world dataset details.

Dataset	Sample size	Feature size
Boston Housing (BH)	506	14
Wine Quality (WQ)	4894	12
Facebook Metrics (FB)	500	19
Bioconcentration (BC)	779	14
Student Performance (SP)	649	33
Community and Crime (CM)	1994	128
Contraceptive Choice (CC)	1472	9
Pima Diabetes (PD)	768	9
Las Vegas Ratings (LV)	504	20
Statlog Heart (SH)	270	13
Retinopathy (RP)	1151	20
Medical Expenditure Panel Survey (ME)	15786	139
Meta-analysis Global Group in Chronic (MG)	40367	33

Table 2.5: Ablation study of CASTLE on real datasets to highlight sources of gain.

Dataset	$\mathcal{L}_N(f_\Theta) + \mathcal{V}_{\Theta_1}$	$\mathcal{R}_{\Theta_1} + \mathcal{V}_{\Theta_1}$	$\mathcal{L}_N(f_\Theta) + \mathcal{R}_{\Theta_1}$	$\mathcal{L}_N(f_\Theta) + \mathcal{R}_{\Theta_1} + \mathcal{V}_{\Theta_1}$
		Regression (MSE)		
BH	0.162 ± 0.018	0.226 ± 0.158	0.174 ± 0.025	$\mathbf{0.123 \pm 0.016}$
WQ	0.711 ± 0.035	0.753 ± 0.013	0.713 ± 0.019	$\mathbf{0.708 \pm 0.030}$
FB	0.265 ± 0.045	0.327 ± 0.088	0.451 ± 0.032	$\mathbf{0.246 \pm 0.150}$
BC	0.362 ± 0.040	0.416 ± 0.009	0.373 ± 0.016	$\mathbf{0.318 \pm 0.036}$
SP	0.338 ± 0.181	0.212 ± 0.018	0.572 ± 0.340	$\mathbf{0.200 \pm 0.020}$
CM	0.347 ± 0.016	0.334 ± 0.007	0.478 ± 0.078	$\mathbf{0.326 \pm 0.031}$
		Classification (AUROC)		
CC	0.778 ± 0.006	0.780 ± 0.008	0.768 ± 0.011	$\mathbf{0.787 \pm 0.007}$
PD	0.795 ± 0.012	0.792 ± 0.012	0.766 ± 0.012	$\mathbf{0.817 \pm 0.004}$
BC	0.712 ± 0.018	0.722 ± 0.008	0.712 ± 0.020	$\mathbf{0.731 \pm 0.010}$
LV	0.562 ± 0.033	0.586 ± 0.023	0.566 ± 0.027	$\mathbf{0.595 \pm 0.032}$
SH	0.895 ± 0.006	0.889 ± 0.011	0.890 ± 0.010	$\mathbf{0.929 \pm 0.007}$
RP	0.801 ± 0.012	0.802 ± 0.014	0.791 ± 0.012	$\mathbf{0.814 \pm 0.014}$

2.6 Conclusion

We have introduced CASTLE regularization, a novel regularization method that jointly learns the causal graph to improve generalization performance in comparison to existing capacity-based and reconstruction-based regularization methods. We used existing PAC-Bayes theory to provide a theoretical generalization bound for CASTLE. We have shown experimentally that CASTLE is insensitive to increasing feature dimensionality, dataset size, and uncorrelated noise variables. Furthermore, we have shown that CASTLE regularization improves performance on a plethora of real datasets and, in the worst case, never degrades performance. We hope that CASTLE will play a role as a general-purpose regularizer that can be leveraged by the entire machine learning community.

CHAPTER 3

MIRACLE: Causally-Aware Imputation via Learning Missing Data Mechanisms

3.1 Introduction

Missing data is an unavoidable byproduct of collecting data in most practical domains. In medicine, for example, doctors may choose to omit what they deem to be irrelevant information (e.g., some patients may be asked to get comprehensive blood tests while others don't), data may be explicitly omitted by the patient (e.g., avoiding questions on smoking status precisely because of their smoking habit) or simply misrecorded in electronic health systems (see e.g., [15, 60, 140]).

Imputation algorithms can be used to estimate missing values based on data that was recorded, but their correctness depends on the type of missingness. For instance, expanding on the example above, younger patients may also be more likely to omit their smoking status. As illustrated in Figure 3.1, the challenge is that implicitly conditioning inference on observed data introduces a spurious path of correlation between age and the prevalence of smoking that wouldn't exist with complete data.

Missing data creates a shift between the available missing data distribution and the target complete data distribution. It is a shift that may be explicitly modeled as missingness indicators in an underlying causal model (i.e., a missingness graph as proposed by Mohan et al. [95]) as shown in Figure 3.1. The learning problem is one of *extrapolation*, learning with access to a missing data distribution for prediction and inference on the complete data distribution – that is, generated from a model where all missingness indicators have been

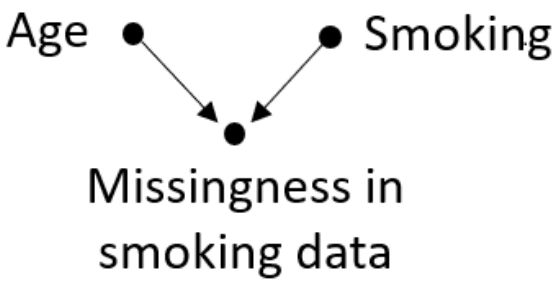


Figure 3.1: Missingness may introduce spurious dependencies.

intervened on (interventions interpreted in the sense of Pearl [104]) thus graphically removing the dependence between missingness and its causes, and any spurious correlations among its ancestors.

With this causal interpretation, imputation of missing data on a given variable Y from other observed variables X is formulated as a problem of robust optimization,

$$\underset{\theta\in\Theta}{\text{minimize}} \sup_{P\in\mathcal{P}} \mathbb{E}_{(X,Y)\sim P}\left[(f_\theta(X)-Y)^2\right], \tag{3.1}$$

simultaneously optimizing over the set of distributions $\mathcal{P}$ arising from interventions on missingness indicators. Causal solutions – i.e. imputation using functions of causal parents of each missing variable in the underlying causal graph – are a closely-related version of this problem with an uncertainty set $\mathcal{Q}$ defined as any distribution arising from interventions on observed variables and variable indicators (see e.g. sections 3.2 and 3.3 in [89]),

$$\sup_{P\in\mathcal{P}} \mathbb{E}_{(X,Y)\sim P}\left[(f_\theta(X)-Y)^2\right] \leq \sup_{P\in\mathcal{Q}} \mathbb{E}_{(X,Y)\sim P}\left[(f_\theta(X)-Y)^2\right], \tag{3.2}$$

since $\mathcal{P}\subset\mathcal{Q}$. Our premise is that causal solutions, i.e. minimizing the right-hand-side of (3.2), are expected to correct for spurious correlations introduced by distribution shift due to missing data and preserve the dependencies of the complete data for downstream analysis.

3.1.1 Contributions

In this paper, we propose to impute while preserving the causal structure of the data. Missing values in a given variable are replaced with their conditional expectation given the realization

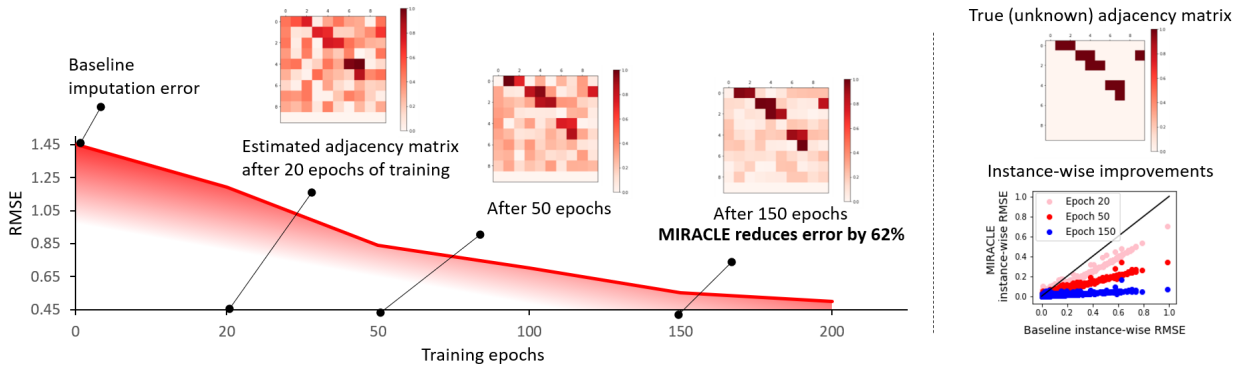


Figure 3.2: MIRACLE refines baseline imputation by simultaneously learning an m-graph using a bootstrap imputation loop that serves to incrementally regularize predictions with a learned causal graph. We plot average testing error and estimated causal graph as a function of training epochs on a synthetic data experiment described in Section 3.4. The true causal structure (as an adjacency matrix) and imputation improvements for each missing value separately (each missing value with a corresponding dot) is shown in the right-most panel.

of its causal parents instead of the more common conditional expectation given all other observed variables, which absorbs spurious correlations.

We propose a novel imputation method called Missing data Imputation Refinement And Causal LEarning (MIRACLE). MIRACLE is a general framework for imputation that operates on any baseline (existing) imputation method. A visual description is given in Figure 3.2: given some initial imputation from a baseline method, MIRACLE refines its imputations iteratively by learning a missingness graph (m-graph) [95] and regularizing the imputation function such that it is consistent with the causal graph generating the data, substantially improving performance. In experiments, we apply MIRACLE to improve six popular imputation methods as baselines. We present detailed simulations to demonstrate on synthetic and a variety of publicly available datasets from the UCI Machine Learning Repository [31] that MIRACLE can improve imputation in almost every scenario and never degrades performance across all imputation methods.

3.1.2 Related work

The literature on imputation is large and varied. Still, most imputation algorithms work with the prior assumption that the missing data mechanism is ignorable, in the sense that imputation does not require explicitly modeling the distribution of missing values for imputation [128]. Accordingly, classical imputation methods impute using a joint distribution over the incomplete data that is either explicit or implicitly defined through a set of conditional distributions. For example, explicit joint modeling methods include generative methods based on Generative Adversarial Networks [162, 168, 7], matrix completion methods [91], and parametric models for the joint distribution of the data. Missing values are then imputed by drawing from their predictive distribution. The conditional modeling approach [149] consists of specifying one model for each variable and iteratively imputing using estimated conditional distributions. Examples of discriminative methods are random forests [121], autoencoders [43, 50, 90], graph neural networks [167], distribution matching via optimal transport [92], and multiple imputation using chained equations [16].

In a different line of research, Mohan et al., in a series of papers, see e.g. [95, 93], explicitly considered missing data within the underlying causal mechanisms of the data. Subsequently, a range of related problems has been studied, including identifiability of distributions and causal effects in the presence of missing data, see e.g. [19, 138, 97], testable implications relating to the causal structure using missing data [93], and causal discovery in the presence of missing data [47, 148]. Our focus, in contrast, is algorithmic in nature. We aim to develop an algorithm that improves imputation quality by leveraging causal insights represented as an estimated missingness graph learned from data.

3.2 Background

The basic semantic framework of our analysis rests on structural causal models (SCMs) (see e.g. Chapter 7 in [104] for more details) explicitly introducing missingness indicators and their functional relationship with other variables, using in part the notation of [95]. We define

an SCM $\mathcal{M}$ as a tuple $(\boldsymbol{X}, \boldsymbol{R}, \boldsymbol{U}, \mathcal{F}, P)$ where $\boldsymbol{X}$ is a vector of d endogenous variables and $\boldsymbol{U}$ is a vector of exogenous variables.[1] $\boldsymbol{R}$ is the vector of missingness indicators that represent the status of missingness of the endogenous variables $\boldsymbol{X}$. Precisely, R_j is responsible for the value of a proxy variable Z_j of X_j, i.e., the observed version of X_j. For example, Z_j is equal to X_j if the corresponding record is observed ($R_j = 1$), otherwise Z_j is missing ($R_j = 0$). $\mathcal{F}$ is a set of functions where each $f_X, f_R \in \mathcal{F}$ decide the values of an endogenous variable X and a missingness indicator variable R, respectively. The function f_X takes two separate arguments as parts of $\boldsymbol{X}$ (except X itself) and $\boldsymbol{U}$, termed as Pa_X and U_X. That is, $X \leftarrow f_X(\mathrm{Pa}_X, U_X)$ and $R \leftarrow f_R(\mathrm{Pa}_R, U_R)$.

The randomness in SCMs comes from the exogenous distribution $P_{\boldsymbol{U}}(\boldsymbol{u})$ where the exogenous variables in $\boldsymbol{U}$ are generated independently and are mutually independent. Naturally, through the functions in $\mathcal{F}$, the SCM $\mathcal{M}$ induces a joint distribution $P_{\boldsymbol{X}}(\boldsymbol{x})$ over the endogenous variables $\boldsymbol{X}$, called the endogenous distribution. An intervention on some arbitrary random variables $\boldsymbol{V}$ in $\boldsymbol{X}$ and $\boldsymbol{R}$, denoted by $do(\boldsymbol{v})$, is an operation which sets the value of $\boldsymbol{V}$ to be $\boldsymbol{v}$, regardless of how they are ordinarily determined. For an SCM $\mathcal{M}$, let $\mathcal{M}_{\boldsymbol{v}}$ denote a submodel of $\mathcal{M}$ induced by intervention $do(\boldsymbol{v})$. The interventional distribution $P_{\boldsymbol{X}}(\boldsymbol{x}|do(\boldsymbol{v}))$ induced by $do(\boldsymbol{v})$ is defined as the distribution over $\boldsymbol{X}$ in the submodel $\mathcal{M}_{\boldsymbol{v}}$, namely, $P_{\boldsymbol{X},\mathcal{M}_{\boldsymbol{v}}}(\boldsymbol{x}) = P_{\boldsymbol{X}}(\boldsymbol{x}|do(\boldsymbol{v}))$.

Each SCM in the context of missingness is associated with a m-graph $\mathcal{G}$ (e.g., Fig. 1a), which is a directed acyclic graph (DAG) where nodes represent endogenous variables $\boldsymbol{X}$ and missingness indicators $\boldsymbol{R}$, and arrows represent the arguments Pa_X and Pa_R of each function f_X and f_R respectively. By convention, exogenous variables $\boldsymbol{U}$ are often not shown explicitly in the graph.

Assumption 3 (Missingness indicators are not causes)**.** *No missingness indicator in $\boldsymbol{R}$ can be the cause of the endogenous variables $\boldsymbol{X}$, i.e., the arguments of the functions generating $\boldsymbol{X}$.*

[1] Essentially, $\boldsymbol{X}$ is the ground-truth features; $\boldsymbol{U}$ is the random noise in the data generating process.

Assumption 4 (Causal sufficiency)**.** *Exogeneous variables* $\boldsymbol{U}$ *are mutually independent, i.e., all common parents of the endogenous variables are included in* $\boldsymbol{X}$*.*

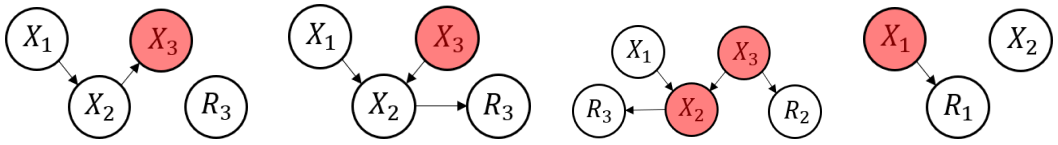
Assumption 5 (No self-masking missingness)**.** *Self-masking missingness refers to missingness in a variable that is caused by itself. In the m-graph this is depicted by an edge from* X_j *to* R_j *(as shown in Figure 3.3 (d)). We assume that there is no such edges in the m-graph.*

Assumption 6 (Observed root nodes)**.** *The endogenous variables* X_j *such that* $Pa_{X_j} = \emptyset$ *(i.e., the root nodes) in the m-graph are always observed (*$R_j = 1$ *with probability 1).*

We make the four assumptions above throughout the following sections. Assumption 3 and 4 are employed in most related works using m-graphs (see e.g. [93, 95]). Assumption 3 is valid, for example, if $\boldsymbol{R}$ is generated in the data collection process after the variable values are assigned. Consequently, under this assumption, if two endogenous variables of interest X_1 and X_2 are not d-separated by some variable X_3, they are not d-separated by X_3 together with their missingness indicators R_1 and R_2. We denote an independent relation in a data distribution by "$\perp\!\!\!\perp$" and d-separation in a m-graph by "$\perp\!\!\!\perp_d$". We assume the data distribution is faithful to a m-graph, meaning that the two independencies are equivalent. As shown in Figure 3.3, data is missing completely at random (MCAR) if $\boldsymbol{X} \perp\!\!\!\perp_d \boldsymbol{R}$ holds in the m-graph, missing at random (MAR) if for any endogenous variable X_j, $R_j \perp\!\!\!\perp_d X_j \mid \boldsymbol{X}_{-j}$ holds, and missing not at random (MNAR) otherwise, as stated in [95]. If Assumption 5 is violated, we are unable to learn the missingness for self-masked variables. Assumption 6 is necessary for imputing all the missing variables from their causal parents. These assumptions are imperative for MIRACLE to provide improved imputations by leveraging the causal structure of the underlying data generating process. In our experiments (Section 3.4), we apply MIRACLE to real-world datasets where these assumptions are not guaranteed.

3.2.1 Why is imputation prone to bias?

The reason for considering the causal structure of the underlying system is that when learning an imputation model from observed data, implicitly conditioning on some missingness



(a) A MCAR graph. (b) A MAR graph. (c) A MNAR graph. (d) Self-masking missingness.

Figure 3.3: Example graphs. $\boldsymbol{X} = (X_1, X_2, X_3)$ are endogenous variables and $\boldsymbol{R} = (R_1, R_2, R_3)$ are missing data indicators. Red shaded variables are not always observed while white shaded variables are always observed.

indicators in $\boldsymbol{R}$ induces spurious dependencies that would not otherwise exist. For example, in a graph $X_1 \rightarrow R_3 \leftarrow X_2$, conditioning on $R_3 = 1$ induces a dependence between X_1 and X_2. In general, the distributions $P_{\boldsymbol{X}}(\boldsymbol{x}|\boldsymbol{R} = \boldsymbol{r})$ and $P_{\boldsymbol{X}}(\boldsymbol{x}|do(\boldsymbol{r}))$ differ unless missingness occurs completely at random, and motivates an interpretation of the problem as domain generalization, training on data from one distribution ultimately to be applied on data from a different distribution that, in our case, arises from missing data (i.e., interventions on missingness indicators). This shift is not addressed in the imputation methods that only use the feature correlations.

3.3 MIRACLE

In this section, we propose to correct for the shift in distribution due to missing data by searching for causal solutions and explicitly refining imputation methods using a penalty on the induced causal structure. In practice, we have n *i.i.d.* realizations of the observed version of $\boldsymbol{X} \in \mathbb{R}^d$, concatenated as a incomplete data matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$, together with missingness indicators concatenated in a matrix $\mathbf{R} \in \{0, 1\}^{n \times d}$. We use here the same bold uppercase notation for sets of variables and matrices of observations but their meaning should be clear from the context. Our goal is to impute the unobserved values in $\mathbf{X}$ using each variable's

causal parents. We define the *imputed* data $\tilde{\mathbf{X}} \in \mathbb{R}^{n \times d}$,

$$\tilde{\mathbf{X}} = \mathbf{R} \odot \mathbf{X} + (1 - \mathbf{R}) \odot \hat{\mathbf{X}}$$

where $\odot$ is the element-wise product of matrices and $\hat{\mathbf{X}}$ is an estimate of the complete data matrix.

3.3.1 Network architecture

In this section, we describe our approach for estimating $\hat{\mathbf{X}}$. Let $d_S \leq d$ be the number of partially observed features, i.e., missing for at least one realization. S is the set of missing features indices. The imputation network is defined as a function $f : \mathbb{R}^d \to \mathbb{R}^d \times [0, 1]^{d_S}$ that takes an initially imputed dataset $\tilde{\mathbf{X}}^{(0)}$ (using an existing baseline imputation method), and returns two quantities:

1. A refined imputation $\hat{\mathbf{X}}$.

2. An estimation of the probabilities of features X_{ij} being missing, $i = 1..., n$ and $j \in S$.

A depiction of the network architecture and optimization algorithm is shown in Figure 3.4. The architecture is constructed with respect to the assumptions shown in Section 3.2. Our model f is decomposed into two sub-networks, $f = (f^{(imp)}, f^{(miss)})$, responsible for imputing the unobserved data and estimate the probabilities of missingness, respectively. The imputation network has d components, $f^{(imp)} = (f_1^{(imp)}, \ldots, f_d^{(imp)})$, one for each variable, and the missingness network has d_S components, $f^{(miss)} = (f_1^{(miss)}, \ldots, f_{d_S}^{(miss)})$. Each component, for both networks, has separate input and output layers but shared hidden layers (of size h). Let $\mathbf{W}_{1,j}^{(imp)}$ and $\mathbf{W}_{1,j}^{(miss)}$ denote the $h \times d$ weight matrix (we omit biases for clarity) in the input layer of $f_j^{(imp)}$ and $f_j^{(miss)}$ respectively. The j-th column of $\mathbf{W}_{1,j}^{(imp)}$ and $\mathbf{W}_{1,j}^{(miss)}$ is set to $\mathbf{0}$. Let $\mathbf{W}_m \in \mathbb{R}^{h \times h}$, for $m = 2, \ldots, M-1$, denote the weight matrix of each hidden layer and let $\mathbf{W}_{M,j}^{(imp)}$ and $\mathbf{W}_{M,j}^{(miss)}$, be the $1 \times h$ dimensional output layers of each sub-network. The imputation network prediction is given by,

$$f_j^{(imp)}(\mathbf{x}) := \mathbf{W}_{M,j}^{(imp)} \phi(\cdots \phi(\mathbf{W}_2 \phi(\mathbf{W}_{1,j}^{(imp)} \mathbf{x}))),$$

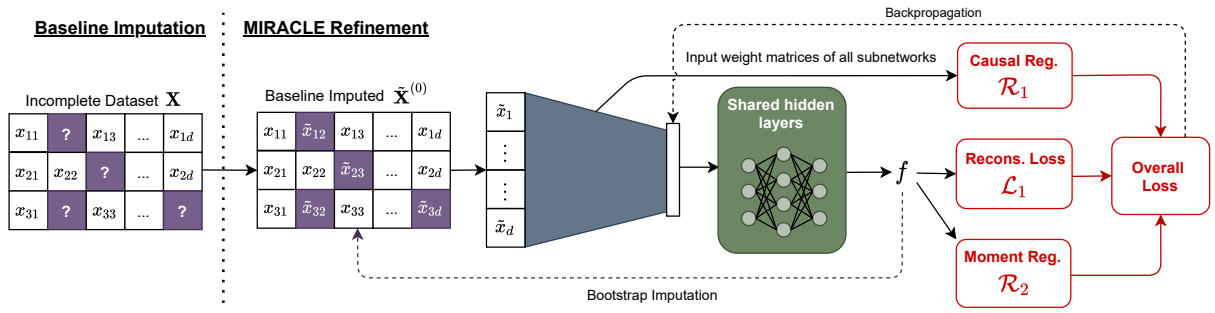


Figure 3.4: Network and optimization diagram for MIRACLE.

for $j = 1, \ldots, d$. And similarly, the missingness network prediction is given by,

$$f_j^{(miss)}(\mathbf{x}) := \sigma\big(\mathbf{W}_{M,j}^{(miss)}\phi(\cdots\phi(\mathbf{W}_2\phi(\mathbf{W}_{1,j}^{(miss)}\mathbf{x})))\big),$$

for $j = 1, \ldots, d_S$, where $\phi(\cdot)$ is the ELU activation function and σ is the sigmoid function. Our network is optimized with respect to three objectives. First, to accurately predict missing values, second, to faithfully encode the causal relationships given by the underlying m-graph, and third to satisfy a moment constraint of the missing data mechanism on the imputed values.

3.3.2 Reconstruction loss

The first objective is to train f to correctly reconstruct each feature from the observed data using a reconstruction loss,

$$\mathcal{L}_1 = \frac{1}{n}\left(\sum_{i=1}^{n}\left\|\mathbf{x}_i \odot \mathbf{r}_i - f^{(imp)}(\tilde{\mathbf{x}}_i^{(0)}) \odot \mathbf{r}_i\right\|^2 + \sum_{i=1}^{n}\text{CrossEntropy}\left[\tilde{\mathbf{r}}_i, f^{(miss)}(\tilde{\mathbf{x}}_i^{(0)})\right]\right),$$

where $\mathbf{x}_i$ and $\tilde{\mathbf{x}}_i^{(0)}$ are the realized and imputed feature vector of the i-th instance, $\tilde{\mathbf{r}}_i$ are the d_S components of $\mathbf{r}_i$ that are missing for at least one instance. The first loss term is for reconstructing the observed features, and the second loss term is for estimating the probabilities of missingness.

3.3.3 Causal regularizer

The second objective is to ensure that the dependencies defined by f correspond to a DAG over the features X and the missing indicators R_S, which enforces that the learned functional dependencies recover a DAG in the equivalence class of causal graphs over the observed data. Enforcing the acyclicity of the dependencies induced by a continuous function f is originally proposed in [170, 174]. Define a binary adjacency matrix $\mathbf{B} \in \{0,1\}^{(d+d_S)\times(d+d_S)}$; $[\mathbf{B}]_{k,j} = 0$ (i.e., the l_2-norm of the k-th column of the matrix $\mathbf{W}_{1,j}^{(imp)}$ or $\mathbf{W}_{1,j}^{(miss)}$ is 0) is a realistic and sufficient condition for achieving $\partial_k f_j = 0$. The adjacency matrix $\mathbf{B}$ of the graph induced by the learned f is acyclic if and only if,

$$\mathcal{R}_1 = \tfrac{1}{2}h^2(\mathbf{B}) + h(\mathbf{B}), \tag{3.3}$$

is equal to zero, where $h(B) := \mathrm{Tr}(\exp\{\mathbf{B}\odot\mathbf{B}\}) - (d+d_S)$ and $\exp(\cdot)$ is the matrix exponential.

Remark 1. Existing imputation methods based on feature correlations essentially assume an undirected (non-causal) graph between the features. Further, acyclicity is a realistic and practical assumption to make on the static datasets collected by human experts. In nature, most data distributions generate their features in some order. In a directed graph, a cycle means a path starts and ends at the same node. This is unlikely to happen in the data generating process if not considering variables over time, i.e., time-series data. By enforcing acyclicity, MIRACLE only uses the causal parents for imputation, which is less biased by spurious dependencies that only exist in the observed data.

3.3.4 Moment regularizer

The third objective leverages a set of moment constraints in the missingness pattern to improve imputation. Assume $\xi_j = P\big(R_j = 1 \mid \mathrm{Pa}_{R_j}\big) \in (\delta, 1-\delta)$, for some $\delta > 0$. The following derivation holds for MAR or MCAR missingness patterns only. It holds that,

$$\mathbb{E}\left\{\frac{R_j X_j}{\xi_j}\right\} = \mathbb{E}\left\{\mathbb{E}\left[\frac{R_j X_j}{\xi_j} \mid X_j, \mathrm{Pa}_{R_j}\right]\right\} = \mathbb{E}\left\{\frac{X_j}{\xi_j}\mathbb{E}\left[R_j \mid X_j, \mathrm{Pa}_{R_j}\right]\right\} = \mathbb{E}\left\{X_j\right\}, \tag{3.4}$$

where the third equality follows from the MAR assumption ($R_j \perp\!\!\!\perp X_j \mid \boldsymbol{X}_{-j}$). Under the MCAR assumption, this derivation holds trivially since in that case $R_j \perp\!\!\!\perp X_j$.

We can use the missingness and imputation networks to enforce the above equality algorithmically, ensuring the left hand side equals the right hand side in the empirical version of (3.4) as follows,

$$\mathcal{R}_2 = \sum_{j=1}^{d_S} \left[\hat{\tau}_{j,\text{SIPW}} - \hat{\tau}_{j,\text{mean}}\right]^2 = \sum_{j=1}^{d_S} \left[\left(\sum_{i=1}^{n} e_{ij} r_{ij} \right)^{-1} \sum_{i=1}^{n} e_{ij} r_{ij} x_{ij} - \frac{1}{n} \sum_{i=1}^{n} f_{S[j]}^{(imp)}(\tilde{\mathbf{x}}_i^{(0)}) \right]^2,$$

where $e_{ij} = 1/f_j^{(miss)}(\tilde{\mathbf{x}}_i^{(0)})$, and $S[j]$ is the j-th element of S, i.e., the index of the j-th missing feature. Minimizing $\mathcal{R}_2$ forces the two estimators of $\mathbb{E}\{X_j\}$ to match, the stabilized inverse propensity score weighting (SIPW) estimator $\hat{\tau}_{j,\text{SIPW}}$ [118] using the missingness network $f_j^{(miss)}$ (in e_{ij}) and the mean estimator $\hat{\tau}_{j,\text{mean}}$ using the imputation network $f_{S[j]}^{(imp)}$.

Remark 2. We hypothesize this mechanism can improve performance for two reasons. First, the missing data mechanism $P(R_j = 1 \mid \text{Pa}_{R_j})$ can be a simpler function that takes less samples to learn than the function that generates the feature j, $\mathbb{E}\big[X_j|\text{Pa}_{X_j}\big]$. Then the SIPW estimator based on $f_j^{(miss)}$ will converge to the true mean faster than the estimator based on $f_{S[j]}^{(imp)}$. Second, in $\mathcal{R}_2$, the mean estimator using $f_{S[j]}^{(imp)}$ is based on all the samples; $f_{S[j]}^{(imp)}$ is trained to produce predictions on the samples with missing feature j for the sake of matching the SIPW estimator. By contrast, without the regularizer $\mathcal{R}_2$, $f_{S[j]}^{(imp)}$ is solely trained on the samples with observed feature j, and its performance may fail to generalize to data with missing feature j.

3.3.5 Bootstrap Imputation

Discovering a causal graph requires complete data. However, this is not the case for missing data problems. Because of this, we require that MIRACLE be seeded by another imputation method. Imputed values are iteratively refined by MIRACLE, hence "bootstrapping", to potentially converge to a new imputation that minimizes MIRACLE's objective (including

causal and moment regularizers). MIRACLE's objective for optimization is,

$$\mathcal{L} = \mathcal{L}_1 + \beta_1 \mathcal{R}_1 + \beta_2 \mathcal{R}_2, \tag{3.5}$$

where β_1 and β_2 are hyperparameters that define the strength of regularization. We iteratively update the baseline matrix $\tilde{\mathbf{X}}^{(0)}$ with a new imputed matrix $\tilde{\mathbf{X}}$ given by MIRACLE every ten epochs in training. With increasing epochs, stochastic optimization minimizes the loss for the imputed matrices that respect the causal and moment regularization. In theory, this is analogous to supervised training a denoising autoencoder (DAE) [153, 23, 107], but differs only by the fact that "noise" comes from prior or previous imputations. In training DAE, the input samples are corrupted by independent noise with each epoch, yet convergence is still guaranteed [4]. In our experiments, we demonstrate that bootstrap imputation indeed converges on multiple datasets and baseline methods.

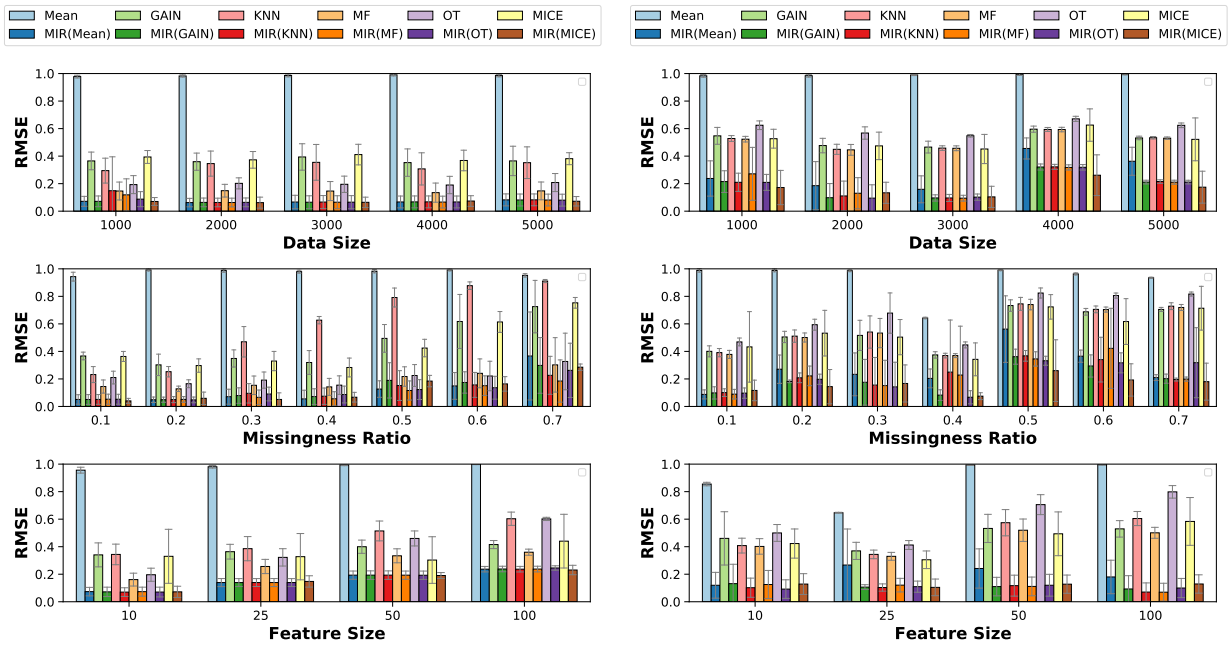
3.4 Experiments

In this section, we validate the performance of MIRACLE using both synthetic and a variety of real-world datasets.

1. In the first set of experiments, we quantitatively analyze MIRACLE on synthetic data generated from a known causal structure.

2. In the second set of experiments, we quantitatively evaluate the imputation performance of MIRACLE using various publicly available UCI datasets [31].

General set-up. We conduct each experiment five times under random instantiations of missingness. We report the RMSE along with standard deviation across each of the five experiments. Unless otherwise specified, missingness is applied at a rate of 30% per feature. For MCAR, this is applied uniformly across all features. For MAR, we randomly select 30% of the features to have missingness caused by another disjoint and randomly chosen set of features. Similarly, we randomly select 30% of features to be MNAR. We induce MAR and MNAR missingness using the methods outlined in [162].



(a) MAR (b) MNAR

Figure 3.5: Experiments on MAR (left) and MNAR (right) synthetic data in terms of RMSE over varying *dataset sizes* **(top)**, *missingness rates* **(middle)**, and *feature sizes* **(bottom)**. Note that we show the average error over a variety of DAG instantiations and target variables, thus the magnitude and standard deviation of errors vary significantly between runs.

We use an 80-20 train-test split. We performed a hyperparameter sweep (log-based) for β_1 and β_2 with ranges between 1e-3 and 100. By default we have β_1 and β_2 set to 0.1 and 1, respectively.

Evaluating imputation. For each subsection below, we present three model evaluations in terms of missingness imputation performance, label prediction performance of a prediction algorithm trained on imputed data and the congeniality of imputation models.

- **Missingness imputation performance** is evaluated with the root mean squared error comparing the imputed missing values with their actual unobserved values.
- **Label prediction performance** of an imputation model is its ability to improve the post-imputation prediction. By post-imputation, we refer to using the imputed data to

perform a downstream prediction task. To be fair to all benchmark methods, we use the same model (support vector regression) in all cases.

- **The congeniality** of an imputation model is its ability to impute values that respect the feature-label relationship post imputation. Specifically, we compare, support vector parameters, w, learned from the complete dataset with the parameters $\hat{w}$, learned from the imputed dataset. We report root mean square error $(||w - \hat{w}||^2)^{1/2}$ for each method. Lower values imply better congeniality [162].

Baseline imputation methods. We apply MIRACLE imputation over a variety of six commonly used imputation baseline methods: (1) mean imputation using the feature-wise mean, (2) a deep generative adversarial network for imputation using GAIN [162] (3) k-nearest neighbors (KNN) [143] using the Euclidean distance as a distance metric of each missing sample to observed samples, (4) a tree-based algorithm using MissForest (MF) [121], (5) a deep neural distribution matching method based on optimal transport (OT) [92], and (6) Multivariate Imputation by Chained Equations (MICE) [16]. For each of the baseline imputation methods with tunable hyperparameters, we used the published values. We implement MIRACLE using the `tensorflow`[2] library.

We used the following network architecture for MIRACLE. Our proposed architecture consists of d sub-networks with shared hidden layers, as shown in Figure 3.4. Each network is constructed with two hidden layers of d neurons with ELU activation. Each benchmark method is initialized and seeded identically with the same random weights. For dataset preprocessing, all continuous variables are standardized with a mean of 0 and a variance of 1. We train each model using the Adam optimizer with a learning rate of 0.0005 for up to a maximum of 300 epochs.

[2]Source code at https://github.com/vanderschaarlab/MIRACLE.

3.4.1 Generating missingness.

The following explains how we constructed synthetic datasets that satisfy MCAR, MAR and MNAR patterns of missingness. We apply a modification to the missingness generation from [162].

- **MCAR**. Missing completely at random was introduced by randomly removing 30% of the observations in each feature.

- **MAR**. We sequentially define the probability that the i-th component of the n-th sample is observed conditional on the missingness and values (if observed) of the previous $i-1$ components to be,

$$P^m(i) = \frac{p^m(i) \cdot N \cdot \exp(\sum_{j<i} w_j m_j(n) x_j(n) + b_j(1 - m_j(n)))}{\sum_{l=1}^{N} \exp(\sum_{j<i} w_j m_j(l) x_j(l) + b_j(1 - m_j(l)))} \tag{3.6}$$

  where $p^m(i)$ corresponds to the average missing rate of the i-th feature, and w_j, b_j are sampled from $\mathcal{U}(0,1)$ (but are only sampled once for the entire dataset).

- **MNAR.** Missing not at random was introduced by defining the probability of the i-th component of the n-th sample to be observed by,

$$P^m(i) = \frac{p^m(i) \cdot N \cdot \exp(-w_i x_i(n))}{\sum_{l=1}^{N} \exp(-w_i x_i(l))} \tag{3.7}$$

  with the same notation as above. Here, the missingness of a data point is directly dependent on its value (with dependence determined by the weight w_i, sampled from $\mathcal{U}(0,1)$).

3.4.2 Synthetic data

In this subsection, we evaluate MIRACLE on synthetic data. In doing so, we can control aspects of the underlying data generating process and possess oracle knowledge of the DAG structure.

3.4.3 Synthetic data generation

In each synthetic experiment, we generated a p-dimensional random graph G from a Erdös–Rényi random graph model with p edges on average. Given G, we assigned uniformly random edge weights to obtain a weighted adjacency matrix $W \in \mathbb{R}^{p \times p}$. Given W, we sampled $X = WX + E$ repeatedly from a Gaussian noise model for $E \in \mathbb{R}^p$ (each dimension sampled independently) to generate independent observations from this system.

3.4.4 Data generating process.

We generate random Erdos-Renyi graphs with functional relationships from parent to children nodes. At each node, we add Gaussian noise with mean 0 and variance 1.

3.4.5 Synthetic results.

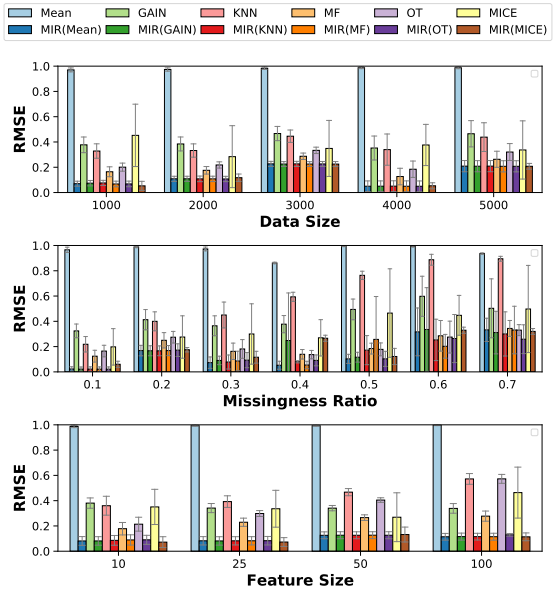
In Figure 3.5, we show experiments of MIRACLE on synthetic MAR data in terms of RMSE. Our experiments show that MIRACLE is able to significantly improve imputation over each of the baseline imputation methods. Figure 3.5 shows MIRACLE improves performance over each baseline method across various **dataset sizes, missingness ratios, and feature sizes (DAG sizes)**.

Note that the error bars are large for some of the plots with predictive error and congeniality. This is because the y-axis of these plots are min-max normalized between 0 and 1, so the high variance (large error bars) shows that the improvement by MIRACLE may be minimal for the mentioned datasets. Additionally, this could be caused by the fact that the missing features aren't predictive of a target variable, i.e., better imputation does not necessarily lead to any performance gain for the predicting the target variable.

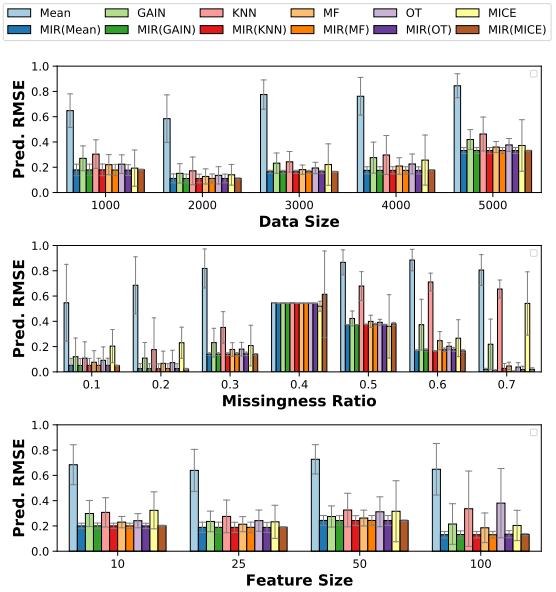
3.4.6 MCAR Results

Using our synthetic experimental setup used in the main paper, we show the performance of MIRACLE in terms of RMSE, predictive error, and congeniality in Figure 3.6 for each of our

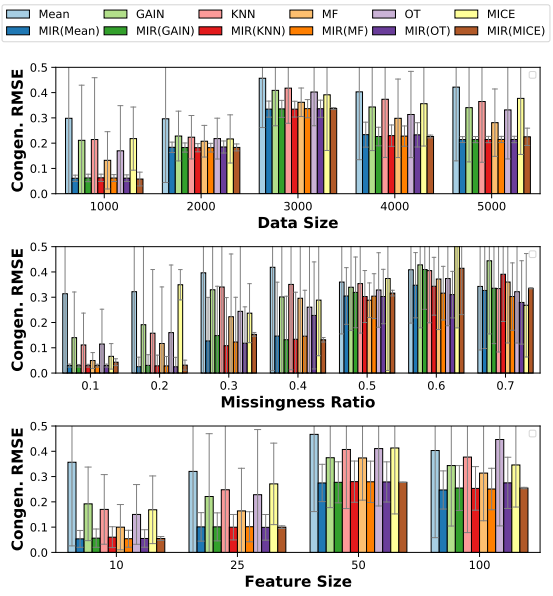
baseline methods with MCAR.



(a) Performance in terms of RMSE.



(b) Performance in terms of prediction RMSE.



(c) MCAR congeniality (in terms of RMSE).

Figure 3.6: Experiments on MCAR synthetic data as a function of dataset sizes **(top)**, missingness rates **(middle)**, and feature sizes **(bottom)** of each subfigure: **(a)** RMSE, **(b)** machine learning predictive error of a random variable, and **(c)** congeniality.

3.4.7 MAR Results

Using our synthetic experimental setup used in the main paper, we show the performance of MIRACLE in terms of RMSE, predictive error, and congeniality in Figure 3.7 for each of our baseline methods with MAR.

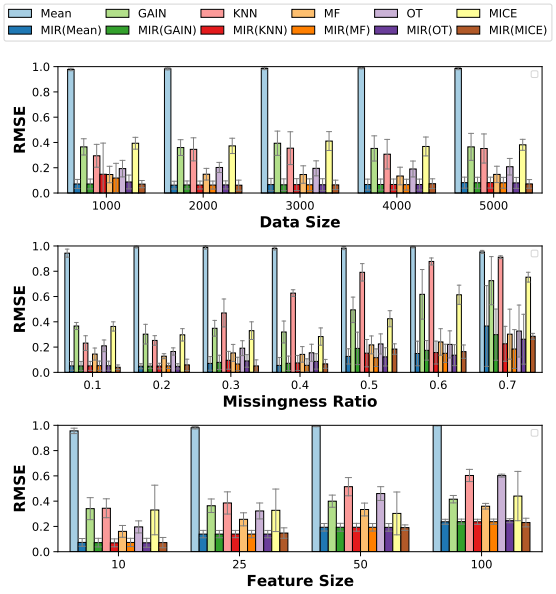
3.4.8 MNAR Results

Using our synthetic experimental setup used in the main paper, we show the performance of MIRACLE in terms of RMSE, predictive error, and congeniality in Figure 3.8 for each of our baseline methods with MNAR.

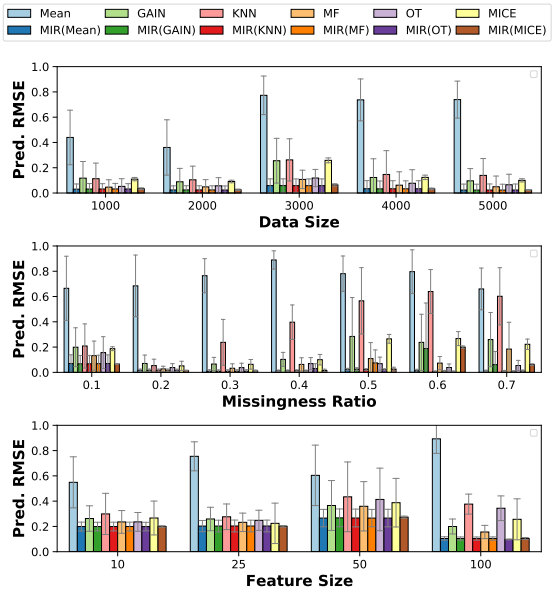
3.4.9 Experiments on real data

In Figure 3.9 we show experiments of MIRACLE on real data. We perform experiments on several UCI datasets used in [162, 167, 92]: Autism, Life expectancy, Energy, Abalone, Protein Structure, Communities and Crime, Yeast, Mammographic Masses, Wine Quality, and Facebook Metrics. In Figure 3.9, the improvements of MIRACLE are minimal for MCAR (except for mean imputation). This agrees with our discussion in Section 3.2.1, because the baseline imputations are not biased in the MCAR setting where $\boldsymbol{X} \perp\!\!\!\perp_d \boldsymbol{R}$ holds in the m-graph. Conversely for the MAR and MNAR settings, as expected, we observe MIRACLE has an significant improvement on some of the datasets, such as Abalone, Autism, Energy and Protein Structure. As discussed in Section 3.2, MIRACLE can improve the baseline imputation under Assumptions 3-6, which may not hold in these real-world datasets. Nevertheless, we observe that MIRACLE never degrades performance relative to its baseline imputation on any dataset. Furthermore, no baseline imputer is optimal across the datasets. In almost all cases, applying MIRACLE to any baseline results in the lowest error.

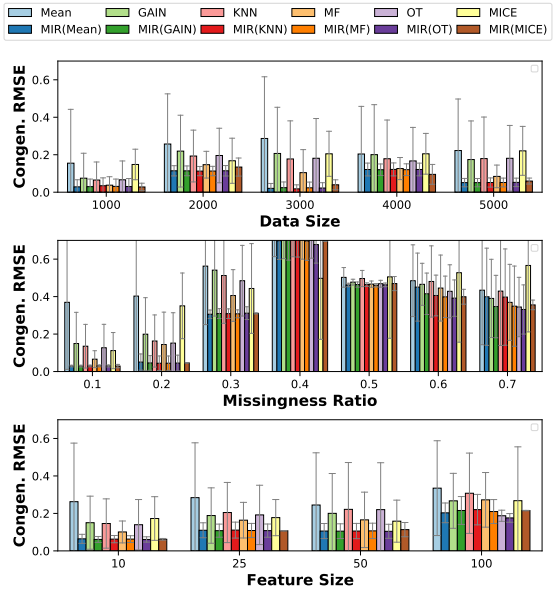
Prediction error and congeniality. We include additional plots for the real data experiments for prediction error and congeniality in Figures 3.11 and 3.12, respectively.



(a) Performance in terms of RMSE.

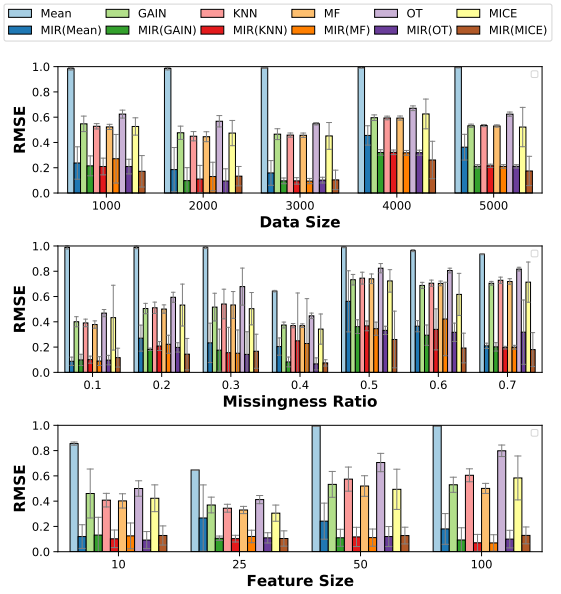


(b) Performance in terms of prediction RMSE.

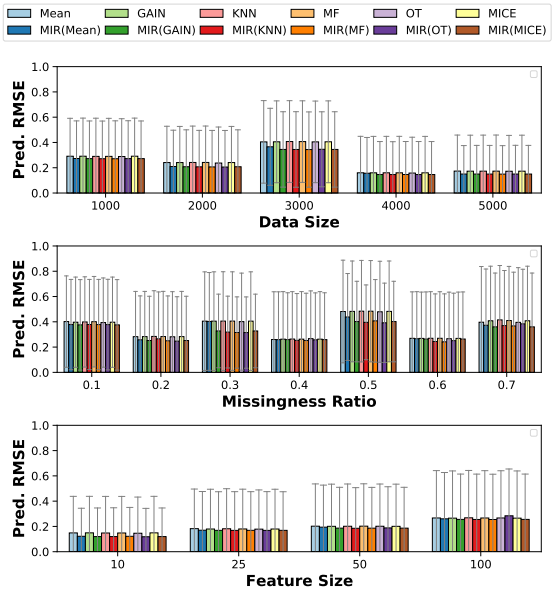


(c) Congeniality (in terms of RMSE).

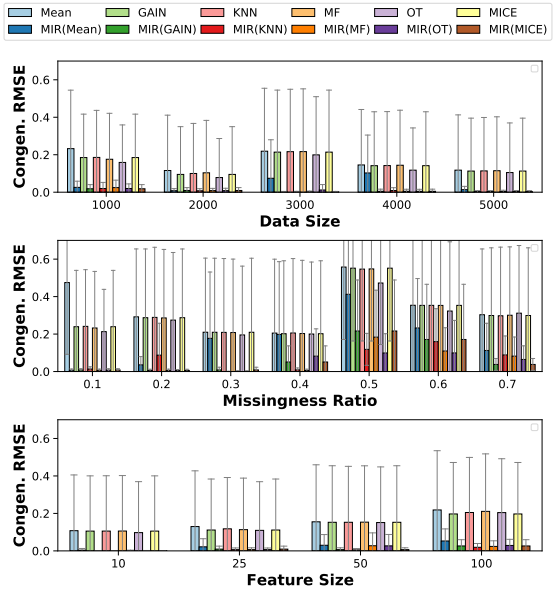
Figure 3.7: Experiments on MAR synthetic data as a function of dataset sizes **(top)**, missingness rates **(middle)**, and feature sizes **(bottom)** of each subfigure: **(a)** RMSE, **(b)** machine learning predictive error of a random variable, and **(c)** congeniality.



(a) Performance in terms of RMSE.



(b) Performance in terms of prediction RMSE.



(c) Congeniality (in terms of RMSE).

Figure 3.8: Experiments on MNAR synthetic data as a function of dataset sizes **(top)**, missingness rates **(middle)**, and feature sizes **(bottom)** of each subfigure: **(a)** RMSE, **(b)** machine learning predictive error of a random variable, and **(c)** congeniality.

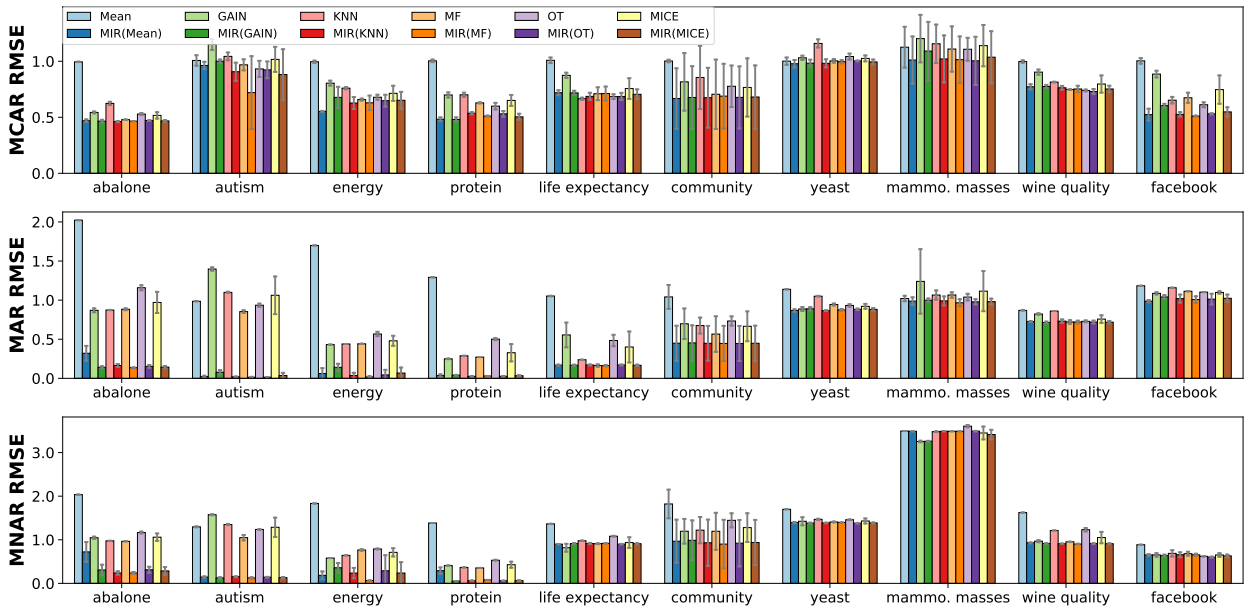


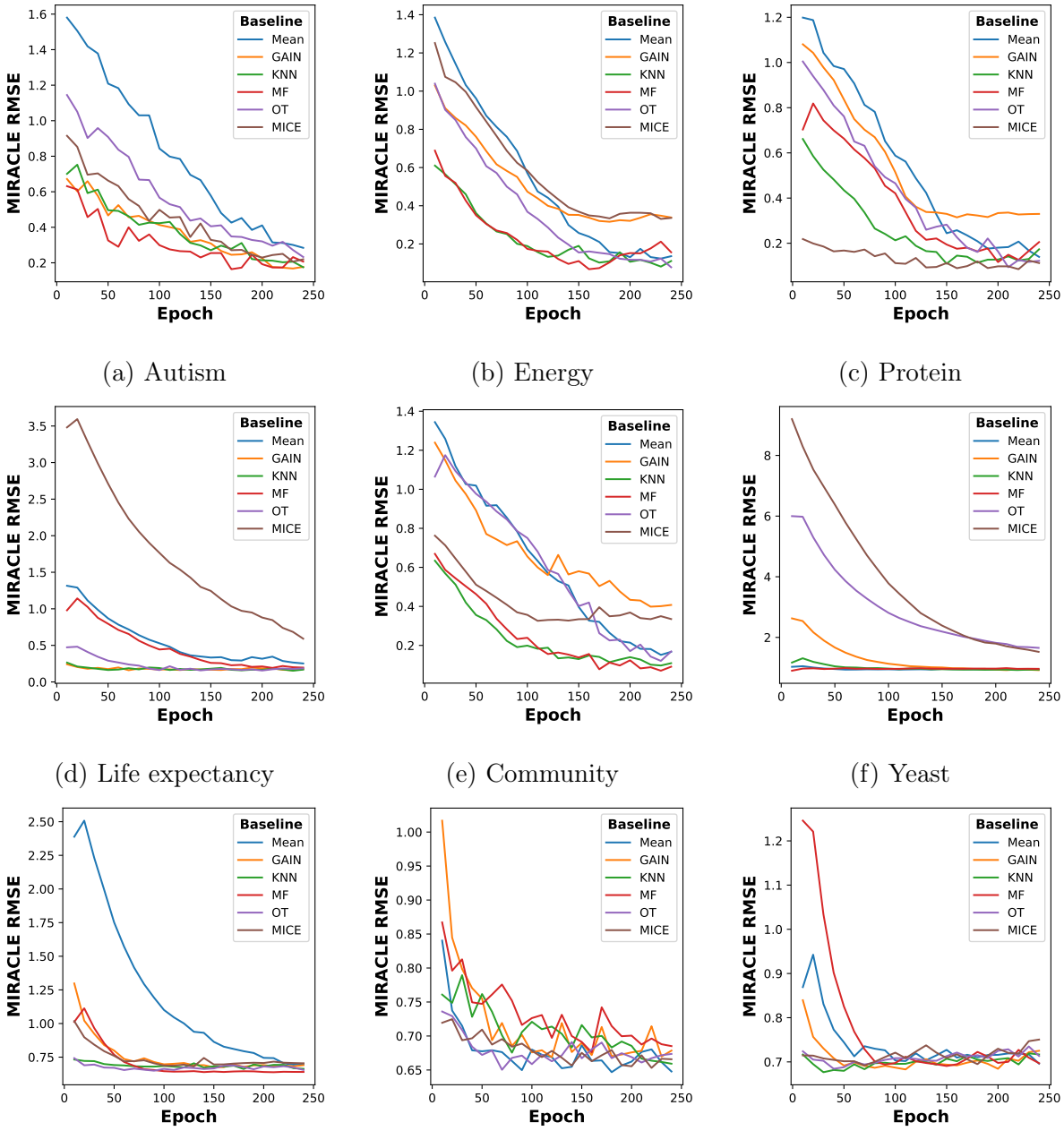
Figure 3.9: MIRACLE on real data. MIRACLE improves all baselines across MCAR **(top)**, MAR **(middle)**, and MNAR **(bottom)**. In the worst-case, MIRACLE never harms performance.

Additional convergence plots. We include additional convergence plots on real datasets in Figure 3.10. We use the same experimental setup used in Figure 3.15 in Section 3.4. We observe that MIRACLE is able to converge regardless of baseline imputation used.

3.4.10 Understanding missingness location

An important consideration is how well does predicting with the causal parents work when down-selecting features. Consider missingness in X_5 in the DAG in Figure 3.14. The first column with the causal parents $\text{Pa}(X_5)$ mean that only the parents of features were used for imputation. X_9 represents a variable that is not causally linked to anything.

Using our synthetic data generating process, we synthesized a dataset according to Figure 3.14. The goal here was to impute the missing values in X_5, using each variable in Figure 3.14 to induce the missingness. Each of the missingness causes is categorized as MAR, except for X_5, which is MNAR (since missingness caused by itself), and for X_9, which is



(a) Autism (b) Energy (c) Protein

(d) Life expectancy (e) Community (f) Yeast

(g) Mammo. masses (h) Wine quality (i) Facebook

Figure 3.10: Convergence plots for real datasets.

MCAR, since it is an external noise variable. The results provide several interesting findings.

1. Using MissForest as a baseline imputer, the results in Table 3.1 show that MIRACLE performs as well as $\text{Pa}(X_5)$, and has better performance than the baseline imputer.

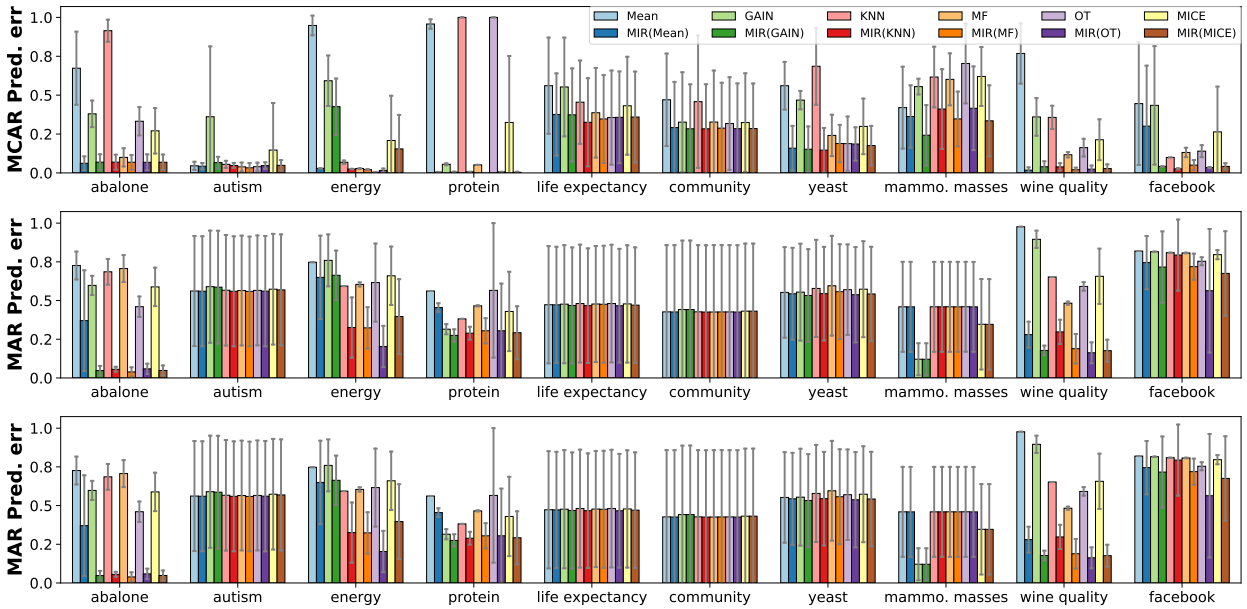


Figure 3.11: MIRACLE on real datasets in terms of predictive error. MIRACLE improves over all baselines across all types of missingness: MCAR **(top)**, MAR **(middle)**, and MNAR **(bottom)**.

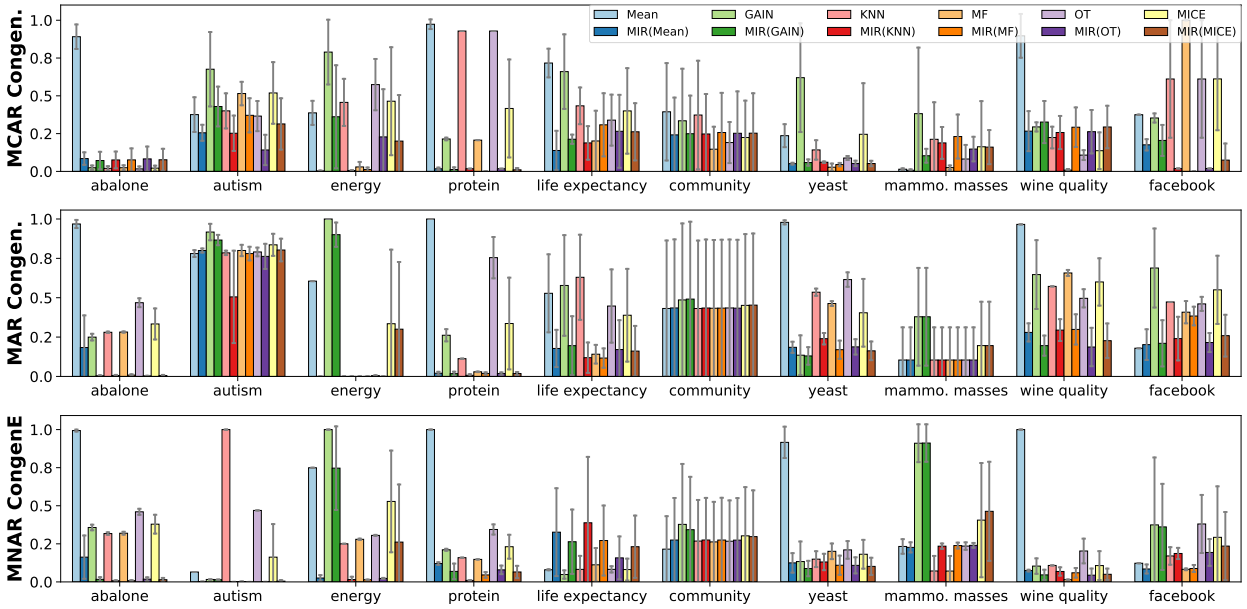


Figure 3.12: MIRACLE on real datasets in terms of congeniality. MIRACLE improves over all baselines across all types of missingness: MCAR **(top)**, MAR **(middle)**, and MNAR **(bottom)**.

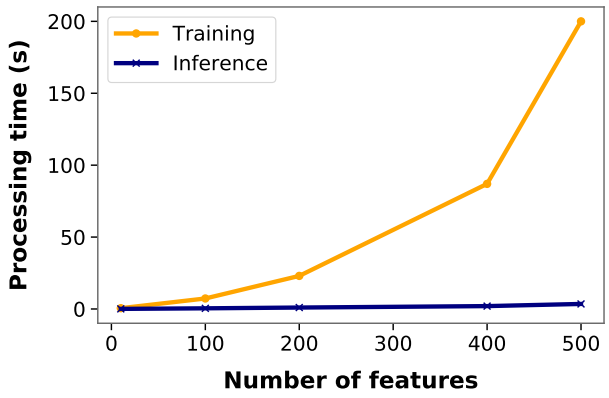


Figure 3.13: MIRACLE scalability analysis

Moreover, the two right-most columns of Table 3.1 give the average estimated functional dependence of X_5 (our target for prediction) and its parents and non-parents. We see that MIRACLE recovers true parents consistently.

2. We see that using causal parents (Pa(X_5) and MIRACLE) for missingness caused by itself, X_5‡, and a noise variable, X_9†, leads to the least amount of improvement.

3. We see that MIRACLE has the most gain when the missingness is caused by a causal parent (X_2 or X_3).

4. Interestingly, for this example, we observe comparable performance when using the Markov blanket features versus all features in our baseline algorithm (MissForest). This suggests that the Markov blanket features are likely used for imputation by the baseline method.

3.4.11 Computational Complexity

Pseudocode for MIRACLE is provided in Algorithm 2. We perform an analysis of the MIRACLE scalability. Using our synthetic data generation, we created datasets of 1000 samples. Using our the synthetic experimental setup presented in the main paper, we present the computational timing results for MIRACLE as we increase the number of input features

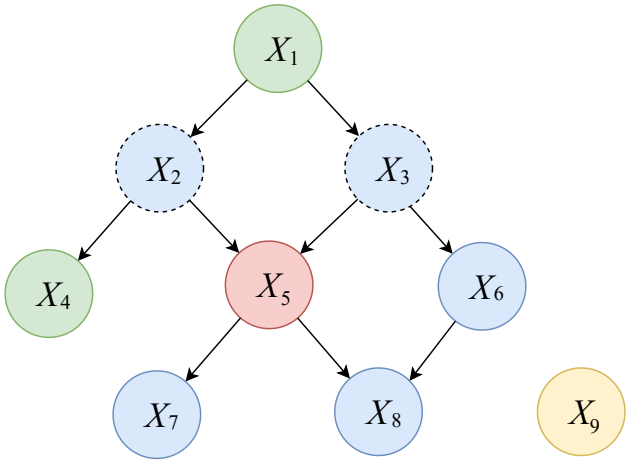


Figure 3.14: A sample DAG. X_5 is the incomplete variable in red. The Markov Blanket $\texttt{MB}(X_5)$ is shown in blue, and the causal parents $\text{Pa}(X_5)$ are shown with dashed borders. X_9 represents a variable that is not causally linked to anything.

Algorithm 2 Train MIRACLE

Input: An incomplete dataset $\mathbf{X}$ with missing values, a missing indicator matrix $\mathbf{R}$ (with 1 indicating observed), an imputed matrix $\tilde{\mathbf{X}}^{(0)}$ by some baseline method,

Output: Imputed dataset $\tilde{\mathbf{X}}^*$ with no missing values.

Initialization: Imputation network f, $\mathcal{G} = \varnothing$ with maximum size $M_{\mathcal{G}}$ for saving imputed matrices over epochs

repeat

Train f for one epoch by optimizing the objective function $\mathcal{L} = \mathcal{L}_1 + \beta_1\mathcal{R}_1 + \beta_2\mathcal{R}_2$ with $\tilde{\mathbf{X}}^{(0)}$ as input.

if $\mathcal{G}$ is full **then**

Remove the first element from $\mathcal{G}$

end if

$\tilde{\mathbf{X}} \leftarrow f(\tilde{\mathbf{X}}^{(0)})$, $\mathcal{G} \leftarrow \mathcal{G} \cup \{\tilde{\mathbf{X}}\}$

$\tilde{\mathbf{X}}^{(0)} \leftarrow$ average all the elements of $\mathcal{G}$.

until MIRACLE converges (i.e., change of $\mathbf{X}^{(0)}$ is small)

return $\tilde{\mathbf{X}}^* \leftarrow \tilde{\mathbf{X}}^{(0)}$

Table 3.1: Understanding the location of missingness. We predict X_5 when its missingness is caused by each variable in the DAG. ‡ and † represent MNAR and MCAR, respectively. All other causes are MAR. The two right-most columns show the learned edge weights into X_5 for the parental and non-parental variables.

	X_5 imputed error (RMSE)				X_5 edge weights (no threshold)	
Cause	Pa(X_5)	MB(X_5)	Baseline	MIRACLE	Pa	non-Pa
X_1	0.11 ± .06	0.15 ±.03	0.27 ± .05	**0.12 ± .07**	0.44 ± 0.14	0.02 ± 0.01
X_2	0.98 ± .08	1.34 ± .05	1.31 ± .06	**0.49 ± .06**	0.64 ± 0.09	0.01 ± 0.01
X_3	1.20 ± .04	1.49 ±.04	1.45 ± .09	**0.50 ± .06**	0.62 ± 0.13	0.02 ± 0.01
X_4	**0.69 ± .05**	1.20 ±.07	1.23 ± .05	1.04 ± .05	0.29 ± 0.11	0.13 ± 0.05
X_5‡	**1.51 ± .03**	1.75 ±.08	1.76 ± .06	1.59 ± .07	0.37 ± 0.18	0.03 ± 0.02
X_6	**0.13 ± .08**	0.17 ±.04	0.18 ± .07	0.14 ± .05	0.34 ± 0.15	0.05 ± 0.02
X_7	1.04 ± .05	1.47 ±.04	1.47 ± .06	**1.01 ± .06**	0.39 ± 0.05	0.04 ± 0.01
X_8	0.21 ± .04	0.28 ±.05	0.23 ± .03	**0.20 ± .03**	0.46 ± 0.10	0.02 ± 0.01
X_9†	0.15 ± .03	0.18 ±.04	0.17 ± .07	**0.14 ± .05**	0.31 ± 0.15	0.02 ± 0.01

on inference and training time in Figure 3.13. Computational time scales linearly with increasing the number of input samples. As expected, we observe that the time to train 1000 samples grows exponentially with the feature size; however, the inference time remains linear. Inference time on 1000 samples with 400 features takes approximately 1.1 seconds, while training time takes nearly 85 seconds. Experiments were conducted on an Ubuntu 18.04 OS using 6 Intel i7-6850K CPUs.

3.4.12 Ablation study

We provide an ablation study on our MIRACLE loss function in Eq. 3.5 to understand the sources of gain of MIRACLE. Here we execute this experiment on our real datasets using the same experimental details highlighted in the main manuscript. We show the results of our ablation on MIRACLE using MissForest as baseline imputation with MAR missingness to highlight our sources of gain in Table 3.2. Here, we observe that MIRACLE (rightmost column) has the most gain over all datasets. Additionally, we observe that $\mathcal{L}_1 + \mathcal{R}_1 + \mathcal{R}_2$ has the most gain when MIRACLE has the most performance improvement over the baseline (see Fig. 3.15 in the manuscript).

3.4.13 Causal discovery and imputation performance

In our experiments, we observe a positive correlation between the quality of learned DAGs (and causal parents) with imputation performance. Consider the left-most plot in Figure 3.15 using OT as a baseline imputer under MAR on our real data sets. Here, we do not have oracle knowledge but assume that the sparseness of the learned DAG implies a coherent DAG. We observe that MIRACLE has the most performance gain when fewer causal parents are identified for the missing variable in the learned DAGs. When MIRACLE is less able to isolate causal parents for prediction, the learned DAGs contains many spurious edges, and MIRACLE only has marginal improvements over the baseline imputer. We note that the gain of MIRACLE is not reproducible via feature selection methods, which are still prone to the spurious correlations in the observed data, as discussed in Section 3.2.1.

Table 3.2: Ablation study of MIRACLE on real datasets to highlight sources of gain.

Dataset	$\mathcal{R}_1 + \mathcal{R}_2$	$\mathcal{L}_1 + \mathcal{R}_2$	$\mathcal{L}_1 + \mathcal{R}_1$	$\mathcal{L}_1 + \mathcal{R}_1 + \mathcal{R}_2$
abalone	0.321 ± 0.108	0.521 ± 0.199	0.312 ± 0.082	0.222 ± 0.062
autism	0.093 ± 0.005	0.094 ± 0.004	0.091 ± 0.004	0.073 ± 0.004
energy	0.106 ± 0.011	0.147 ± 0.077	0.132 ± 0.050	0.065 ± 0.061
protein	0.134 ± 0.016	0.129 ± 0.008	0.119 ± 0.010	0.080 ± 0.008
life expectancy	0.239 ± 0.007	0.223 ± 0.019	0.216 ± 0.014	0.208 ± 0.015
community	0.490 ± 0.015	0.516 ± 0.020	0.479 ± 0.023	0.463 ± 0.010
yeast	0.984 ± 0.013	0.984 ± 0.006	0.988 ± 0.004	0.950 ± 0.014
mammo masses	1.105 ± 0.010	1.150 ± 0.009	1.103 ± 0.013	1.040 ± 0.013
wine quality	0.797 ± 0.004	0.745 ± 0.013	0.724 ± 0.008	0.722 ± 0.003
facebook	1.056 ± 0.005	1.032 ± 0.044	1.034 ± 0.056	0.983 ± 0.002

3.4.14 MIRACLE Convergence

In this subsection, we investigate two dimensions of MIRACLE refinement: (1) baseline imputation quality and (2) sample or instance-wise refinement. Regarding baseline imputation quality, we are interested in understanding the impact of MIRACLE refinement on various baseline imputers that may have disparate performances. In the middle plot of Figure 3.15, we show MIRACLE applied to various baseline imputers on the Abalone dataset. Similar plots for other datasets can be found in Figure 3.10. We observe that even though mean imputation starts off with the worst error, after refinement by MIRACLE, we see that all methods converge to similar RMSEs. For the second experiment, we investigate the sample-wise improvement of MIRACLE on the abalone dataset using MissForest as a baseline imputer. On the right-most plot of Figure 3.15, we observe that a vast majority of the samples are improved by MIRACLE. Note that every point below the diagonal is considered an improvement on an instance over the baseline imputation method. We can see MIRACLE

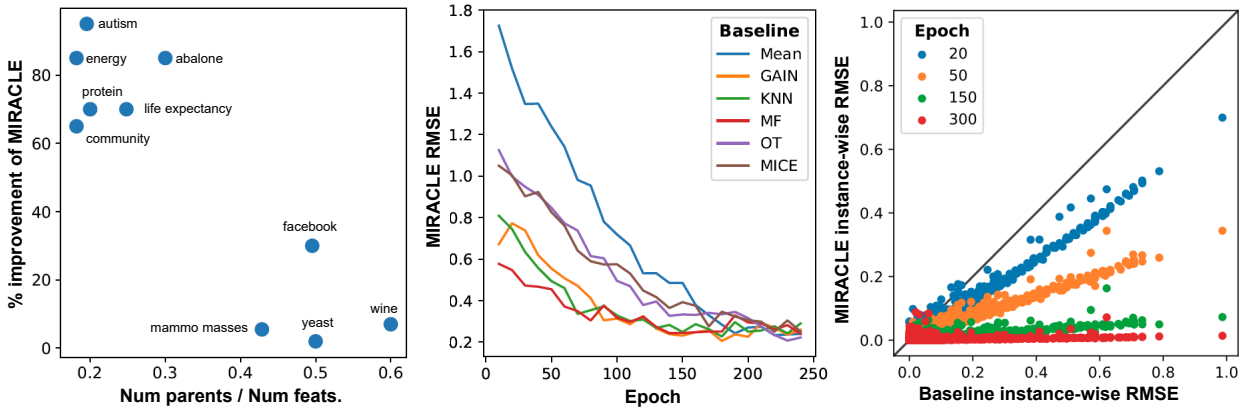


Figure 3.15: **(Left)** Analysis of MIRACLE w.r.t. causal parents on real data. MIRACLE has the most gain when we have identified a sparse set of causal parents. When many features are identified as causal parents, imputation performance degrades. **(Mid)** Convergence of MIRACLE across various baseline imputers. On the abalone dataset, we show that MIRACLE converges to consistent RMSE regardless of baseline imputation. **(Right)** Sample-wise RMSE for MIRACLE across various epochs. MIRACLE is applied to refine MissForest imputations, demonstrating that error is reduced in a sample-wise basis. Note: anything below the diagonal, is an improvement over the baseline imputations.

improves the imputation almost universally except for the instances with small errors in the baseline imputation; on these instances, MIRACLE does not inflate their errors by a large margin. Furthermore, we observe that MIRACLE iteratively improves imputation as training progresses (over each epoch) by the observation that the slope of each line decreases with each epoch.

3.5 Discussion

In conclusion, motivated by the minimax optimization problem (3.1) arising from interventions on missingness indicators in the m-graph that encode the conditional independencies in the data distribution, we proposed MIRACLE, an iterative framework to refine any baseline missing data imputation to use the causal parents embodied in the estimated m-graphs.

MIRACLE learns the causal m-graph as an adjacency matrix embedded in its input layers. We proposed a two-part regularizer based on the causal graph and a moment regularizer based on the missing variable indicators. We demonstrated that MIRACLE significantly improved the imputations of six baseline imputation methods over a variety of synthetic and real datasets. MIRACLE never hurts performance in the worst-case, and we envision MIRACLE becoming a de facto standard in refining missing data imputation.

There are several limitations we would like to identify as paths for future work. First, any violation of the assumptions in Section 3.2 may adversely impact the performance of MIRACLE in practice. Second, causal discovery under missing data is an ongoing research area, and therefore MIRACLE may be discovering DAGs with bias introduced from the baseline methods. However, in experiments, MIRACLE still performs well even if it starts with mean imputation. We expect MIRACLE to improve as causal discovery methods under missingness improve. Third, in its current form, MIRACLE is not extensible to scenarios where causality may not be applicable, such as computer vision. Fourth, because of the causal discovery regularizer and network architecture, MIRACLE may have difficulty scaling to very high dimensional data. Lastly, a more general and detailed discussion is needed between our work and the merits of causality and robustness.

CHAPTER 4

DECAF: Generating Fair Synthetic Data Using Causally-Aware Generative Networks

4.1 Introduction

Generative models are optimized to approximate the original data distribution as closely as possible. Most research focuses on three objectives [9]: fidelity, diversity, and privacy. The first and second are concerned with how closely synthetic samples resemble real data and how much of the real data's distribution is covered by the new distribution, respectively. The third objective aims to avoid simply reproducing samples from the original data, which is important if the data contains privacy-sensitive information [158, 161]. We explore a much-less studied concept: synthetic data fairness.

4.1.1 Motivation.

Deployed machine learning models have been shown to reflect the bias of the data on which they are trained [142, 30, 80, 35, 65]. This has not only unfairly damaged the discriminated individuals but also society's trust in machine learning as a whole. A large body of work has explored ways of detecting bias and creating fair predictors [66, 37, 181, 55, 69, 71, 171], while other authors propose debiasing the data itself [66, 37, 181, 26]. This work's aim is related to the work of [160]: to generate fair synthetic data based on unfair data. Being able to generate fair data is important because end-users creating models based on publicly available data might be unaware they are inadvertently including bias or insufficiently knowledgeable to remove it from their model. Furthermore, by debiasing the data prior to public release,

one can guarantee *any* downstream model satisfies desired fairness requirements by assigning the responsibility of debiasing to the data generating entities.

Goal. From a biased dataset $\mathcal{X}$, we are interested in learning a model G, that is able to generate an equivalent *synthetic* unbiased dataset $\mathcal{X}'$ with minimal loss of data utility. Furthermore, a downstream model trained on the synthetic data needs to make not only unbiased predictions on the synthetic data, but also on real-life datasets (as formalized in Section 4.4.2).

4.1.2 Solution.

We approach fairness from a causal standpoint because it provides an intuitive perspective on different definitions of fairness and discrimination [181, 69, 71, 100, 171]. We introduce DEbiasing CAusal Fairness (DECAF), a generative adversarial network (GAN) that leverages causal structure for synthesizing data. Specifically, DECAF is comprised of d generators (one for each variable) that learn the causal conditionals observed in the data. At inference-time, variables are synthesized topologically starting from the root nodes in the causal graph then synthesized sequentially, terminating at the leave nodes. Because of this, DECAF can remove bias at inference-time through targeted (biased) edge removal. As a result, various datasets can be created for desired (or evolving) definitions of fairness.

4.1.3 Contributions.

We propose a framework of using causal knowledge for fair synthetic data generation. We make three main contributions: i) DECAF, a causal GAN-based model for generating synthetic data, ii) a flexible causal approach for modifying this model such that it can generate fair data, and iii) guarantees that downstream models trained on the synthetic data will also give fair predictions in other settings. Experimentally, we show how DECAF is compatible with several fairness/discrimination definitions used in literature while still maintaining high downstream utility of generated data.

Table 4.1: Overview of related work for synthetic data. We organize related work according to our key areas of interest: (1) Allows post-hoc distribution changes, (2) provides fairness, (3) supports causal notion of fairness, (4) allows inference-time fairness, (5) requires minimal assumptions. We highlight the key contribution, and identify non-neural approaches with "†".

Model	Reference	(1)	(2)	(3)	(4)	(5)	Goal
VAE	[75]	✗	✗	✗	✗	✓	Realistic synth. data.
GANs	[46, 40, 161, 158]	✗	✗	✗	✗	✓	Realistic synth. data.
PSE-DD/DR†	[181]	✓	✓	✓	✗	✗	Discover/Remove bias.
OPPDP†	[26]	✗	✓	✗	✗	✗	Remove bias.
DI†	[37]	✗	✓	✗	✗	✗	Discover/Remove bias.
LFR	[180]	✗	✓	✗	✗	✓	Learn fair representation.
FairGAN	[160]	✗	✓	✗	✗	✓	Realistic and fair synth. data.
CFGAN	[159]	✗	✓	✓	✗	✓	Realistic and fair synth. data.
DECAF	(ours)	✓	✓	✓	✓	✓	Realistic and fair synth.-data.

4.2 Related Works

Here we focus on the related work concerned with data generation, in contrast to fairness definitions for which we provide a detailed overview in Section 4.4 and Section 4.7. As an overview of how data generation methods relate to one another, we refer to Table 4.1 which presents all relevant related methods.

Non-parametric generative modeling. The standard models for synthetic data generation are either based on VAEs [75] or GANs [46, 40, 161, 158]. While these models are well known for their highly realistic synthetic data, they are unable to alter the synthetic data distribution to encourage fairness (except for [160], discussed below). Furthermore, these methods have no causal notion, which prohibits targeted interventions for synthesizing fair data (Section 4.4). We explicitly leave out CausalGAN [72] and CausalVAE [164], which

appear similar by incorporating causality-derived ideas but are different in both method and aim (i.e., image generation).

Fair data generation. In the bottom section of Table 2.1, we present methods that, in some way, alter the training data of classifiers to adhere to a notion of fairness [181, 26, 160, 159, 37, 180]. While these methods have proven successful, they lack some important features. For example, none of the related methods allow for post-hoc changes of the synthetic data distribution. This is an important feature, as each situation requires a different perspective on fairness and thus requires a flexible framework for selecting protected variables. Additionally, only [181, 160] allow a causal perspective on fairness, despite causal notions underlying multiple interpretations of what should be considered fair [69]. Furthermore, only [160, 159, 180] offer a flexible framework, while the others are limited to binary [181, 37] or discrete [26] settings. Most importantly, to the best of our knowledge, we are the only method that regards fairness in the context of downstream model fairness—i.e. a model trained on our data should be fair when rolled out in practice. In essence, from Table 2.1 we learn that DECAF is the only method that combines all key areas of interest. At last, we would like to mention [25], who aim to generate data that resembles a small unbiased reference dataset, by leveraging a large but biased dataset. This is very different to our aim, as we are interested in the downstream model's fairness and explicit notions of fairness.

4.3 Preliminaries

Let $X \in \mathcal{X} \subseteq \mathbb{R}^d$ denote a random variable with distribution $P_X(X)$, with protected attributes $A \in \mathcal{A} \subset \mathcal{X}$ and target variable $Y \in \mathcal{Y} \subset \mathcal{X}$, let $\hat{Y}$ denote a prediction of Y. Let the data be given by $\mathcal{D} = \{\mathbf{x}^{(k)}\}_{k=1}^{N}$, where each $\mathbf{x}^{(k)} \in \mathcal{D}$ is a realization of X. We assume the data generating process can be represented by a directed acyclic graph (DAG)—such that the generation of features can be written as a structural equation model (SEM) [104]—and that this DAG is causally sufficient. Let X_i denote the i^{th} feature in X with causal parents

$\text{Pa}(X_i) \subset \{X_j : j \neq i\}$, the SEM is given by:

$$X_i = f_i(\text{Pa}(X_i), Z_i), \forall i \tag{4.1}$$

where $\{Z_i\}_{i=1}^d$ are independent random noise variables, that is $\text{Pa}(Z_i) = \emptyset$, $\forall i$. Note that each f_i is a deterministic function that places all randomness of the conditional $P(X_i|\,\text{Pa}(X_i))$ in the respective noise variable, Z_i.

4.4 Fairness of Synthetic Data

Algorithmic fairness is a popular topic (e.g., see [22, 69]), but *fair synthetic data* has been much less explored. This section highlights how the underlying graphs of the synthetic and downstream data determine whether a model trained on the synthetic data will be fair in practice. We start with the two most popular definitions of fairness, relating to the legal concepts of *direct* and *indirect* discrimination. We also explore *conditional fairness* [76], which is a generalization of the two. In Section 4.7 we discuss how the ideas in this section transfer to other independence-based definitions [17]. Throughout this section, we separate Y from X by defining $\bar{X} = X \backslash Y$, and we will write $X \leftarrow \bar{X}$ for ease of notation.

4.4.1 Algorithmic fairness

The first definition is called *Fairness Through Unawareness* (e.g. [44]).

Definition 2. *(Fairness Through Unawareness (FTU): algorithm). A predictor $f : X \mapsto \hat{Y}$ is fair iff protected attributes A are not explicitly used by f to predict $\hat{Y}$.*

This definition prohibits *disparate treatment* [22, 179], and is related to the legal concept of *direct discrimination*, i.e., two equally qualified people deserve the same job opportunity independent of their race, gender, beliefs, among others.

Though FTU fairness is commonly used, it might result in *indirect discrimination*: covariates that influence the prediction $\hat{Y}$ might not be identically distributed across different

groups a, a', which means an algorithm might have *disparate impact* on a protected group [37]. The second definition of fairness, *demographic parity* [179], does not allow this:

Definition 3. *(Demographic Parity (DP): algorithm) A predictor $\hat{Y}$ is fair iff $A \perp\!\!\!\perp \hat{Y}$, i.e. $\forall a, a' : P(\hat{Y}|A = a) = P(\hat{Y}|A = a')$.*

Evidently, DP puts stringent constraints on the algorithm, whereas FTU might be too lenient. The third definition we include is based on the work of [76], related to *unresolved discrimination* [71]. The idea is that we do not allow indirect discrimination unless it runs through *explanatory factors* $R \subset X$. For example, in Simpson's paradox [132] there seems to be a bias between gender and college admissions, but this is only due to women applying to more competitive courses. In this case, one would want to regard fairness conditioned on the choice of study [71]. Let us define this as *conditional fairness*:

Definition 4. *(Conditional Fairness (CF): algorithm) A predictor $\hat{Y}$ is fair iff $A \perp\!\!\!\perp \hat{Y}|R$, i.e. $\forall r, a, a' : P(\hat{Y}|R = r, A = a) = P(\hat{Y}|R = r, A = a')$.*

CF generalizes FTU and DP Note that conditional fairness is a generalization of FTU and DP, by setting $R = X \backslash A$ and $R = \emptyset$, respectively. In Section 4.7 we elaborate on the connection between these, and more, definitions.

4.4.2 Synthetic data fairness

Algorithmic definitions can be extended to distributional fairness for synthetic data. Let $P(X), P'(X)$ be probability distributions with protected attributes $A \subset X$ and labels $Y \subset X$. Let $\mathcal{I}(A, Y)$ be a definition of algorithmic fairness (e.g., FTU). Note, that under CF, $\mathcal{I}(A, Y)$ is a function of R as well. We propose $(\mathcal{I}(A, Y), P)$-fairness of distribution $P'(X)$:

Definition 5. *(Distributional fairness) A probability distribution $P'(X)$ is $(\mathcal{I}(A, Y), P)$-fair, iff the optimal predictor $\hat{Y} = f^*(X)$ of Y trained on $P'(X)$ satisfies $\mathcal{I}(A, Y)$ when evaluated on $P(X)$.*

In other words, when we train a predictor on $(\mathcal{I}(A, Y), P)$-fair distribution $P'(X)$, we can only reach maximum performance if our model is fair. Note the explicit reference to $P(X)$,

the distribution on which fairness is evaluated, which does not need to coincide with $P'(X)$. This is a small but relevant detail. For example, when training a model on data $\mathcal{D}' \sim P'(X)$ it could seem like the model is fair when we evaluate it on a hold-out set of the data (e.g., if we simply remove the protected attribute from the data). However, when we use the model for real-world predictions of data $\mathcal{D} \sim P(X)$, disparate impact is possibly observed due to a distributional shift.

By extension, we define synthetic data as $(\mathcal{I}(A, Y), P)$-fair, iff it is sampled from an $(\mathcal{I}(A, Y), P)$-fair distribution. Defining synthetic data as fair w.r.t. an optimal predictor is especially useful when we want to publish a dataset and do not trust end-users to consider anything but performance.[1]

Choosing $\mathbf{P}(\mathbf{X})$. The setting $P(X) = P'(X)$ corresponds to data being fair with respect to itself. For synthetic data generation, this setting is uninteresting as any dataset can be made fair by randomly sampling or removing A; if A is random, the prediction should not directly or indirectly depend on it. This ignores, however, that a downstream user might use the trained model on a real-world dataset in which other variables B are correlated with A, and thus their model (which is trained to use B for predicting Y) will be biased. Of specific interest is the setting where $P(X)$ corresponds to the original data distribution $P_X(X)$ that contains unfairness. In this scenario, we construct $P'(X)$ by learning $P_X(X)$ and removing the unfair characteristics. The data from $P'(X)$ can be published online, and models trained on this data can be deployed fairly in real-life scenarios where data follows $P_X(X)$. Unless otherwise stated, henceforth, we assume $P(X) = P_X(X)$.

4.4.3 Graphical perspective

As reflected in the widely accepted terms direct versus indirect discrimination, it is natural to define distributional fairness from a causal standpoint. Let $\mathcal{G}'$ and $\mathcal{G}$ respectively denote the graphs underlying $P'(X)$ (the synthetic data distribution which we can control) and $P(X)$

[1]Finding the optimal predictor is possible if we assume the downstream user employs any universal function approximator (e.g., MLP) and the amount of synthetic data is sufficiently large.

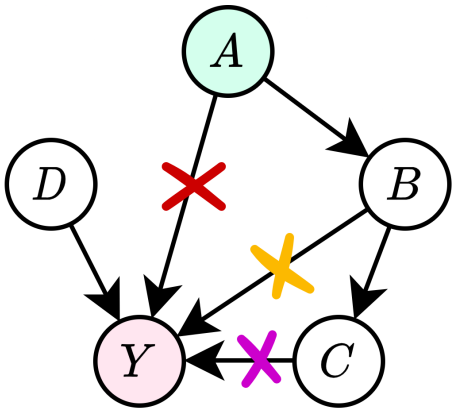


Figure 4.1: Edge removal for fairness. FTU: ✗; DP: ✗✗✗; CF when $R = C$: ✗✗; CF when $B \in R$: ✗ .

(the evaluation distribution that we cannot control). Let $\partial_{\mathcal{G}} Y$ denote the Markov boundary of Y in graph $\mathcal{G}$. We focus on the conditional fairness definition because it subsumes the definition of DP and FTU (Section 4.4.1). Let $R \subset X$ be the set of explanatory features.

Proposition 1. *(CF: graphical condition) If for all $B \in \partial_{\mathcal{G}'} Y$, $A \perp\!\!\!\perp_{\mathcal{G}} B|R$,[2] then distribution $P'(X)$ is CF fair w.r.t $P(X)$ given explanatory factors R.*

Proof. Without loss of generality, let us assume the label is binary.[3] The optimal predictor $f^*(X) = P(Y|X) = P(Y|\partial_{\mathcal{G}'} Y)$. Thus, if $\partial_{\mathcal{G}'} Y$ is d-separated from A in $\mathcal{G}$ given R, prediction $\hat{Y} = f^*(X)$ is independent of A given R and CF holds. □

Corollary 1. *(CF debiasing) Any distribution $P'(X)$ with graph $\mathcal{G}'$ can be made CF fair w.r.t. $P(X)$ and explanatory features R by removing from $\mathcal{G}'$ edges $\tilde{E} = \{(B \to Y)$ and $(Y \to B) : \forall B \in \partial_{\mathcal{G}'} Y$ with $B \not\perp\!\!\!\perp_{\mathcal{G}} A|R\}$.*

Proof. First note $\tilde{E}$ is the necessary and sufficient set of edges to remove for $(\forall B \in \partial_{\mathcal{G}'} Y$, $A \perp\!\!\!\perp_{\mathcal{G}} B|R)$ to be true, subsequently the result follows from Proposition 1. □

For FTU (i.e. $R = X \backslash A$) and DP (i.e. $R = \emptyset$), this corollary simplifies to:

[2] *Where $\perp\!\!\!\perp_{\mathcal{G}}$ denotes d-separation in $\mathcal{G}$. Here we define $A \perp\!\!\!\perp_{\mathcal{G}} B|R$ to be true for all $B \in R$.*

[3] If Y is continuous the same result holds, though the "optimal" predictor will depend on the statistic of interest, e.g. mode, mean, median or the entire distribution $f(X, Y) \approx P(Y|X)$.

Corollary 2. *(FTU debiasing) Any distribution $P'(X)$ with graph $\mathcal{G}'$ can be made FTU fair w.r.t. any distribution $P(X)$ by removing, if present, i) the edge between A and Y and ii) the edge $A \to C$ or $Y \to C$ for all shared children C.*

Corollary 3. *(DP debiasing) Any distribution $P'(X)$ with graph $\mathcal{G}'$ can be made DP fair w.r.t. $P(X)$ by removing, if present, the edge between B and Y for any $B \in \partial_{\mathcal{G}'} Y$ with $B \not\perp\!\!\!\perp_{\mathcal{G}} A$.*

Figure 4.1 shows how the different fairness definitions lead to different sets of edges to be removed.

Faithfulness. Usually one assumes distributions are faithful w.r.t. their respective graphs, in which case the if-statement in Proposition 1 become equivalence statements: fairness is *only* possible when the graphical conditions hold.

Theorem 2. *If $P(X)$ and $P'(X)$ are faithful with respect to their respective graphs $\mathcal{G}$ and $\mathcal{G}'$, then Proposition 1 becomes an equivalence statement and Corollaries 1, 2 and 3 describe the necessary and sufficient sets of edges to remove for achieving CF, FTU and DP fairness, respectively.*

Proof. Faithfulness implies $A \perp\!\!\!\perp_{P(X)} B|R \implies A \perp\!\!\!\perp_{\mathcal{G}} B|R$, e.g. [106]. Thus, if $\exists B \in \partial_{\mathcal{G}'} Y$ for which $A \not\perp\!\!\!\perp_{\mathcal{G}} B|R$, then $A \not\perp\!\!\!\perp B|R$. Because $B \in \partial_{\mathcal{G}'} Y$ and $P'(X)$ is faithful to $\mathcal{G}'$, $\hat{Y} = f^*(X)$ depends on B, and thus $\hat{Y} \not\perp\!\!\!\perp A|R$: CF does not hold. □

Other definitions. Some authors define similar fairness measures in terms of directed paths (cf. d-separation) [181, 71, 100], which is a milder requirement as it allows correlation via non-causal paths. In Section 4.7 we highlight the graphical conditions for these definitions.

4.5 Method: DECAF

The primary design goal of DECAF is to generate fair synthetic data from unfair data. We separate DECAF into two stages. The training stage learns the causal conditionals that are

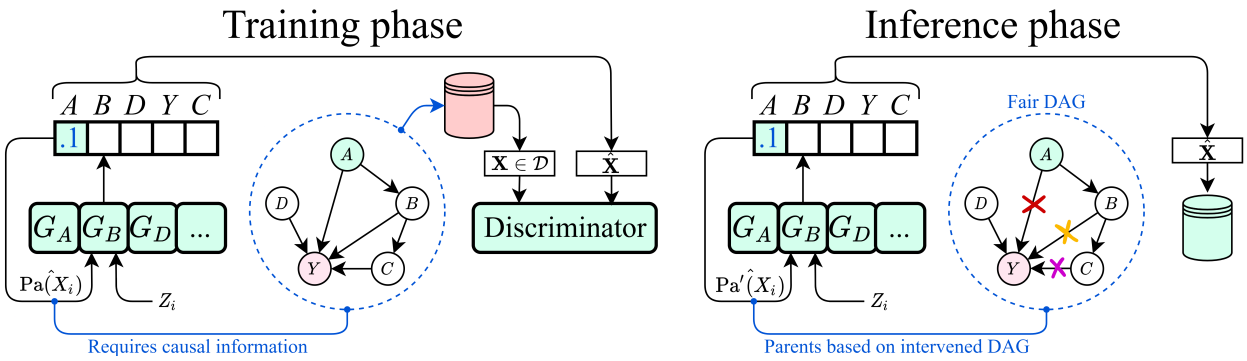


Figure 4.2: Architecture of DECAF. *Training phase*— Each component in $\hat{\mathbf{X}}$ is generated sequentially as a function (where the function is that component's generator G_i) of the component's parents. Parental knowledge is provided by the DAG governing the data. *Inference phase*— As the component-wise generation of the generator network is independent of the DAG governing the data, we can easily replace (or intervene on) the DAG governing parental information. The resulting synthetic data (right) will be governed by the intervened DAG. FTU is achieved by removing edges marked: ✗; DP: ✗✗✗; e.g. CF when $R = C$: ✗✗.

observed in the data through a causally-informed GAN. At the generation (inference) stage, we intervene on the learned conditionals via Corollaries 1-3, in such a way that the generator creates fair data. We assume the underlying DGP's graph $\mathcal{G}$ is known; otherwise, $\mathcal{G}$ needs to be approximated first using any causal discovery method, see Section 4.8.

4.5.1 Training

Overview. This stage strives to learn the causal mechanisms $\{f_i(\text{Pa}(X_i), Z_i)\}$. Each structural equation f_i (Eq. 4.1) is modelled by a separate generator $G_i : \mathbb{R}^{|Pa(X_i)|+1} \to \mathbb{R}$. We achieve this by employing a conditional GAN framework with a causal generator. This process is illustrated in Figure 4.2 and detailed below.

Features are generated sequentially following the topological ordering of the underlying causal DAG: first root nodes are generated, then their children (from generated causal

parents), etc. Variable $\hat{X}_i$ is modelled by the associated generator G_i:

$$\hat{X}_i = G_i(\hat{\text{Pa}}(X_i), Z_i) \quad \forall i, \tag{4.2}$$

where $\hat{\text{Pa}}(X_i)$ denotes the generated causal parents of X_i (for root nodes the empty set), and each Z_i is independently sampled from $P(Z)$ (e.g. standard Gaussian). We denote the full sequential generator by $G(Z) = [G_1(Z_1), ..., G_d(\cdot, Z_d)]$.

Subsequently, the synthetic sample $\hat{\mathbf{x}}$ is passed to a discriminator $D : \mathbb{R}^d \to \mathbb{R}$, which is trained to distinguish the generated samples from original samples. A typical minimax objective is employed for creating generated samples that confuse the discriminator most:

$$\max_{\{G_i\}_{i=1}^d} \min_D \mathbb{E}[\log D(G(Z)) + \log(1 - D(X)], \tag{4.3}$$

with X sampled from the original data. We optimize the discriminator and generator iteratively and add a regularization loss to both networks. Network parameters are updated using gradient descent.

If we assume $P_X(X)$ is compatible with graph $\mathcal{G}$, we can show that the sequential generator has the same theoretical convergence guarantees as standard GANs [46]:

Theorem 3. *(Convergence guarantee) Assuming the following three conditions hold:*

(i) data generating distribution P_X is Markov compatible with a known DAG $\mathcal{G}$;

(ii) generator G and discriminator D have enough capacity; and

(iii) in every training step the discriminator is trained to optimality given fixed G, and G is subsequently updated as to maximize the discriminator loss (Eq. 4.3);

then generator distribution P_G converges to true data distribution P_X

Proof. See Section 4.6 □

Condition (i), compatibility with $\mathcal{G}$, is a weaker assumption than assuming perfect causal knowledge. For example, suppose the Markov equivalence class of the true underlying DAG

has been determined through causal discovery. In that case, any graph $\mathcal{G}$ in the equivalence class is compatible with the data and can thus be used for synthetic data generation. However, we note that debiasing can require the correct directionality for some definitions of fairness, see Discussion.

Remark. The causal GAN we propose, DECAF, is simple and extendable to other generative methods, e.g., VAEs. Furthermore, from the post-processing theorem [34] it follows that DECAF can be directly used for generating *private* synthetic data by replacing the standard discriminator by a differentially private discriminator [158, 64].

4.5.2 Inference-time Debiasing

The training phase yields conditional generators $\{G_i\}_{i=1}^d$, which can be sequentially applied to generate data with the same output distribution as the original data (proof in Section 4.6). The causal model allows us to go one step further: when the original data has characteristics that we do not want to propagate to the synthetic data (e.g., gender bias), individual generators can be modified to remove these characteristics. Given the generator's graph $\mathcal{G} = (X, E)$, fairness is achieved by removing edges such that the fairness criteria are met, see Section 4.4. Let $\tilde{E} \in E$ be the set of edges to remove for satisfying the required fairness definition. For CF, FTU and DP,[4] the sets $\tilde{E}$ are given by Corollaries 1, 2 and 3, respectively.

Removing an edge constitutes to what we call a "surrogate" *do*-operation [104] on the conditional distribution. For example, suppose we only want to remove $(i \to j)$. For a given sample, X_i is generated normally (Eq. 4.2), but X_j is generated using the modified:

$$\hat{X}_j^{do(X_i)=\tilde{x}_{ij}} = G_j(..., X_i = \tilde{x}_{ij}), \tag{4.4}$$

where $X_i = \tilde{x}_{ij}$ is the surrogate parent assignment. Value $\hat{X}_j^{do(X_i)}$ can be interpreted as the counterfactual value of $\hat{X}_j$, had X_i been equal to $\tilde{x}_{ij}$ (see also [171]).

[4]Just like in Corollaries 1 and 3, we assume the downstream evaluation distribution is the same as the biased training data distribution: a predictor trained on the synthetic debiased data, is required to give fair predictions in real-life settings with distribution $P_X(X)$.

Choosing the value of surrogate variable $\tilde{x}_{ij}$ requires background knowledge of the task and bias at hand. For example, surrogate variable $\tilde{x}_{ij}$ can be sampled independently from a distribution for each synthetic sample (e.g., the marginal $P(X_i)$), be set to a fixed value for all samples in the synthetic data (e.g., if X_i: gender, always set $\tilde{x}_{ij} = male$ when generating feature X_j: job opportunity) or be chosen as to maximize/minimize some feature (e.g. $\tilde{x}_{ij} = \arg\max_x \hat{X}_j^{do(X_i)=x}$). We emphasize that we do not set $X_i = \tilde{x}_{ij}$ in the synthetic sample; $X_i = \tilde{x}_{ij}$ is only used for substitution of the removed dependence. We provide more details in Section 4.10.

More generally, we create surrogate variables for all edges we remove, $\{\tilde{x}_{ij} : (i \to j) \in \tilde{E}\}$. Each sample is sequentially generated by Eq. 4.4, with a surrogate variable for each removed incoming edge.

Remark. Multiple datasets can be created based on different definitions of fairness and/or different downstream prediction targets. Because debiasing happens at inference-time, this does not require retraining the model.

4.6 Convergence guarantees DECAF GAN

Assuming the correct underlying data generating DAG is known, well-known theoretical results for GANs transfer to DECAF. We highlight the main results. The typical GAN minimax objective (Eq. 3 paper) is optimized by iteratively updating the discriminator and generator, with respective losses:

$$\mathcal{L}_D(\hat{X}, X) = \log D(\hat{X}) + \log(1 - D(X)) \tag{4.5}$$

$$\mathcal{L}_G(\hat{X}) = -\log D(\hat{X}) \tag{4.6}$$

First, we reiterate the following theorem from [46]. Let P_G and P_X denote generator and original data distributions, respectively.

Theorem 4. *Given fixed optimal discriminator D^*, the global minimum of the generator loss (Eq. 4.6) is achieved if and only if $P_G = P_X$.*

Proof. Noting that we have made no changes to the GAN discriminator, we refer to Theorem 1 of [46]. □

Theorem 5. *(Convergence guarantee) Assuming the following three conditions hold:*

(i) data generating distribution P_X is Markov compatible with a known DAG $\mathcal{G} = (V, E)$;

(ii) generator G and discriminator D have enough capacity; and

(iii) in every training step the discriminator is trained to optimality given fixed G, and G is subsequently updated as to maximise the discriminator loss (Eq. 3 paper);

then generator distribution P_G converges to true data distribution P_X

Proof. This is the direct result of the construction of generator G and follows a similar argument as Proposition 2 of [46]. Note that by the definition of compatibility of P_X and $\mathcal{G} = (V, E)$, we can write:

$$P_X(X) = \prod_{X_i \in V} P(X_i | \{X_j : (X_j \to X_i) \in E\})$$

Given each G_i (see Eq. 2 paper) has enough capacity, G can thus express the full distribution $P_X(X)$. By convexity of the loss functions and the existence of a unique global optimum (Theorem 4), gradient descent is theoretically guaranteed to converge, $P_G \to P_X$ [46]. □

Note that for condition (i) of Theorem 5 to be valid, we do not require that graph $\mathcal{G}$ equals the true underlying DAG of the data generating distribution P_X; P_G is only required to disentangle into the causal factors implied by $\mathcal{G}$. This is highly beneficial, as it enables generation of perfect synthetic data without perfect causal knowledge. For example, if the Markov equivalence class of the true underlying DAG has been determined through causal discovery, any graph $\mathcal{G}$ in the equivalence class satisfies condition (i) of Theorem 5.

Remarks The convergence guarantees do not necessarily hold in practice. First, finite data means there there is no guarantee the algorithm converges to the true underlying data distribution instead of, for example, the observed empirical data distribution. Second, in

practice each generator G_i will have limited capacity and $P(X_i|\,\mathrm{Pa}(X_i))$ might not lie in its support. On a more positive note, these limitations are not specific for DECAF and generally GANs have done well in the past. Additionally, our method is directly extendable to the more stable WGAN-GP [40] and other generative models.

4.7 Compatibility different fairness definitions

Related definitions In the paper we have discussed FTU, DP and CF, which are independence-based definitions and do not take directionality explicitly into account when defining fairness. Some authors use similar definitions, but instead of looking at (conditional) independencies of A and Y, they consider (blocked) directed paths from protected attribute A to Y. These definitions are compatible with DECAF, but mean less edges need to be removed. See Table 4.2 and Figure 4.3. [181] define direct and indirect discrimination, as the "directed path" equivalents of FTU and DP;[5] respectively, there is no edge $A \to Y$ and there is no directed path A to Y. [171] disentangle the total effect of A on Y into direct, indirect and spurious relations. This leads to the same definition for direct discrimination as [181], but a different definition of indirect discrimination as it *does* allow for direct influence of A on Y. A very similar definition, coined counterfactual fairness, is proposed by [69]. [71] introduce *unresolved discrimination* (UD) as the path-equivalent version of conditional fairness. They define *proxy* discrimination as well, which can be considered the dual of UD [71].

Incompatible definitions Some definitions are not compatible with fair synthetic data generation because they rely on the final prediction, e.g. equality of opportunity [55] and calibration (e.g. see [17]). As a consequence, DECAF cannot be used for these. Furthermore, we note that all our fairness definitions are binary: a distribution is fair or unfair. In practice some level of unfairness might be tolerated. For example, the US Supreme Court's 80% rule [8] essentially states that a prediction has disparate impact if for disadvantaged group $A = 1$ and positive outcome $\hat{Y} = 1$, $\frac{P(\hat{Y}=1|A=1)}{P(\hat{Y}=1|A=0)} < 0.8$ [37]. Some authors (e.g. [37]) have explored

[5]Note: the legal definitions of direct and indirect discrimination are in fact defined as FTU and DP.

this continuous definition, but because it requires quantification of path-specific effects work is limited by a linearity assumption. Extending this to nonlinear path-specific effects is an interesting direction for future work, with great relevance for real-life applications.

Table 4.2: Different definitions of fairness that are compatible with DECAF and which edges need removal when evaluation distribution $P(X) = P_X(X)$. The first three definitions are non-causal, the others only prohibit causal paths. A, Y, P, R denote respectively the protected attribute, label, proxy variables and explanatory variables. Let $\pi_{A \to Y}$ denote a directed path from A to Y that ends with $B \to Y$ for some B.

Definition	Edges to remove
Demographic Parity (DP) [179]	$B \leftrightarrow Y : \forall B \in Bl_{G'}(Y)$ with $A \not\perp\!\!\!\perp B$
Conditional Fairness (CF)	$B \leftrightarrow Y : \forall B \in Bl_{G'}(Y)$ with $A \not\perp\!\!\!\perp B \mid R$
Fairness through Unawareness (FTU)	$A \leftrightarrow Y$ and ($A \to C$ or $Y \to C : \forall C$ with $A \to C \leftarrow Y$)
No Indirect Discrim. ($\neg$ ID) [181]	$B \to Y$ if there exists $\pi_{A \to Y}$
No Proxy Discrim. ($\neg$PD) [71]	$B \to Y$ if there exists $\pi_{A \to Y}$ that is blocked by P
No Unresolved Discrim. ($\neg$UD) [71]	$B \to Y$ if there exists $\pi_{A \to Y}$ that is not blocked by R
No Direct Discrim. ($\neg$ DD) [181, 171]	$A \to Y$

4.8 Experiments

In this section, we validate the performance of DECAF for synthesizing bias-free data based on two datasets: i) real data with existing bias and ii) real data with synthetically injected bias. The aim of the former is to show that we can remove real, existing bias. The latter experiment provides a ground-truth unbiased target distribution, which means we can evaluate the quality of the synthetic dataset with respect to this ground truth. For example, when historically biased data is first debiased, a model trained on the synthetic data will likely create better predictions in contemporary, unbiased/less-biased settings than benchmarks that do not debias first.

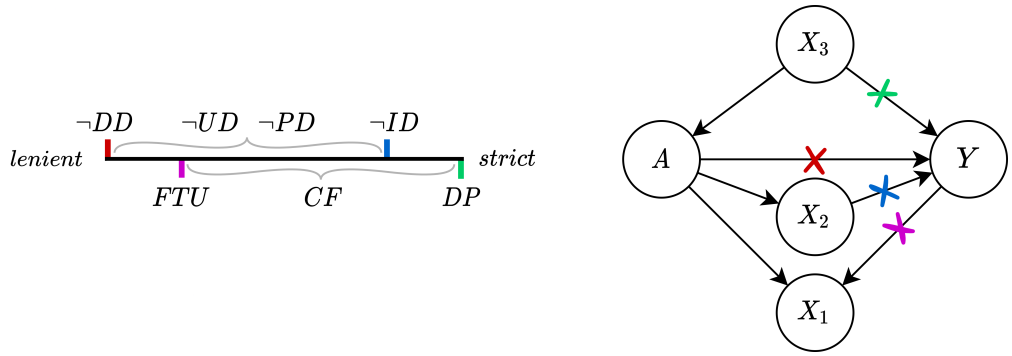


Figure 4.3: (Left) Typical strictness of different definitions. Note that the strictness of CF, ¬ UD and $\neg PD$ depends on the choice of explanatory variables/proxies. (Right) Example showing different definitions and required edge removals. ¬ DD: ✗; FTU: ✗✗; ¬ ID: ✗✗; DP: ✗ ✗ ✗ ✗ . Note that for FTU, $A \to X_1$ could have been removed instead of $Y \to X_1$.

In both experiments, the ground-truth DAG is unknown. We use causal discovery to uncover the underlying DAG and show empirically that the performance is still good.

Benchmarks. We compare DECAF against the following benchmark generative methods: a GAN, a Wasserstein GAN with gradient penalty (WGAN-GP) [40] and FairGAN [160]. FairGAN is the only benchmark designed to generate synthetic fair data,[6] whereas GAN and WGAN-GP only aim to match the original data's distribution, regardless of inherent underlying bias. For these benchmarks, fair data can be generated by naively removing the protected variable – we refer to these methods with the PR (protected removal) suffix and provide more experimental results and insight into PR in Section 4.9. We benchmark DECAF debiasing in four ways: i) with *no inference-time debiasing* (DECAF-ND), ii) under FTU (DECAF-FTU), iii) under CF (DECAF-CF) and iv) under DP fairness (DECAF-DP). We provide DECAF[7].

Evaluation criteria. We evaluate DECAF using the following metrics:

- **Data quality** is assessed using metrics of precision and recall [123, 68, 38]. Additionally,

[6]The works of [181, 26] are not applicable here, as these methods are constrained to discrete data.

[7]`PyTorch Lightning` source code at https://github.com/vanderschaarlab/DECAF.

we evaluate all methods in terms of AUROC of predicting the target variable using a downstream classifier (MLP in these experiments) trained on synthetic data.

- **FTU** is measured by calculating the difference between the predictions of a downstream classifier for setting A to 1 and 0, respectively, such that $|P_{A=0}(\hat{Y}|X) - P_{A=1}(\hat{Y}|X)|$, while keeping all other features the same. This difference measures the direct influence of A on the prediction.

- **DP** is measured in terms of the *Total Variation* [171]: the difference between the predictions of a downstream classifier in terms of positive to negative ratio between the different classes of protected variable A, i.e., $|P(\hat{Y}|A = 0) - P(\hat{Y}|A = 1)|$.

4.8.1 Implementation details.

We instantiate the generator of DECAF with d sub-networks with shared hidden layers. Both the generator and discriminator are constructed having 2 hidden layers with $2d$ neurons and initialized with random uniform weights. Each benchmark is initialized with the same random weights and published hyperparameters. For preprocessing, all continuous variables are standardized. We use the Adam optimizer with a learning rate of 0.001 for up to 50 epochs. We update the generator once for every 10 discriminator updates. We implement DECAF using `PyTorch Lightning`[8].

Computational hardware. All models were trained on an Ubuntu 18.04 OS with 64GB of RAM (Intel Core i7-6850K CPU @ 3.60GHz) and 2 NVidia 1080 Ti GPUs.

Scalability Due to the sequential feature generation, DECAF's run time scales linearly with the number of variables. In practice—for the larger Communities and Crime dataset—this comes down to an average training time of just about 35s per epoch when run on a machine with hexacore Intel i7-6850K CPU. Practical improvements can be made to speed this up further: when the graph is sparse one can parallelize calculations and often one can

[8]Source code is available at https://github.com/vanderschaarlab/DECAF

cluster (some) variables and model clusters together using a single generator network.

Generating discrete variables In both datasets the only non-binary discrete variable is the protected attribute, which for simplicity we have binarised (discriminated vs non-discriminated). All variables are generated in the same way, but binary variables are rounded off after generation.

Table 4.3: Overview datasets

	Credit	Census	Communities
Number of features	15	10	128
- Continuous	3	4	120
- Discrete	12	6	8
Target type	Binary	Binary	Binary
Number of samples	379	32,561	1994
Number of discovered edges	40	22	1288

4.8.2 Debiasing Census Data

In this experiment, we are given a biased dataset $\mathcal{D} \sim P(X)$ and wish to create a synthetic (and debiased) dataset $\mathcal{D}'$, with which a downstream classifier can be trained and subsequently be rolled out in a setting with distribution $P(X)$. We experiment on the Adult dataset [31], with known bias between `gender` and `income` [37, 181]. The Adult dataset contains over 65,000 samples and has 11 attributes, such as `age`, `education`, `gender`, `income`, among others. Following [181], we treat `gender` as the protected variable and use `income` as the binary target variable representing whether a person earns over $50K or not. For DAG $\mathcal{G}$, we use the graph discovered and presented by [181].

DECAF supports both FTU and DP debiasing, i.e. respectively direct and indirect discrimination removal. We use the DAG from [37, 181] as shown in Figure 4.4. FTU is

Table 4.4: Bias removal experiment on the Adult dataset [31].

	Data Quality			Fairness	
Method	Precision↑	Recall↑	AUROC↑	FTU↓	DP↓
Original data $\mathcal{D}$	0.920 ± 0.006	0.936 ± 0.008	0.807 ± 0.004	0.116 ± 0.028	0.180 ± 0.010
GAN	0.607 ± 0.080	0.439 ± 0.037	0.567 ± 0.132	0.023 ± 0.010	0.089 ± 0.008
WGAN-GP	0.683 ± 0.015	0.914 ± 0.005	0.798 ± 0.009	0.120 ± 0.014	0.189 ± 0.024
FairGAN	0.681 ± 0.023	0.814 ± 0.079	0.766 ± 0.029	0.009 ± 0.002	0.097 ± 0.018
DECAF-ND	0.780 ± 0.023	0.920 ± 0.045	0.781 ± 0.007	0.152 ± 0.013	0.198 ± 0.013
DECAF-FTU	0.763 ± 0.033	0.925 ± 0.040	0.765 ± 0.010	0.004 ± 0.004	0.054 ± 0.005
DECAF-CF	0.743 ± 0.022	0.875 ± 0.038	0.769 ± 0.004	0.003 ± 0.006	0.039 ± 0.011
DECAF-DP	0.781 ± 0.018	0.881 ± 0.050	0.672 ± 0.014	0.001 ± 0.002	0.001 ± 0.001

achieved by removing the directed edge between between `sex` and `income` (see Corollary 3), DP is achieved by removing[9] all incoming edges into the target variable that have the protected variable as an ancestor (Corollary 2)- these include edges between the target variable `income` and each of `occupation`, `hours_per_week`, `occupation`, `workclass`, `education`, `relationship`, `marital_status`, and `sex`. DP fairness is overly strict, so to satisfy CF fairness, we allow the variables `occupation`, `hours_per_week`, `workclass`, and `education` while removing the edges from `sex`, `marital_status`, and `relationship`.

We generate synthetic data from the ground truth dataset using each benchmark generator. We randomly hold out a sample of 2000 samples as a test set. We train an MLP using default `scikit-learn` hyperparameters on the generated dataset to use as our downstream classifier. We use a hidden layer with 100 neurons and ReLU activation functions. For the output layer we use a softmax activation and binary cross entropy loss. We use Adam as the optimizer with a learning rate of 0.001.

We repeat this experiment 10 times for each benchmark method and report the average in Table 4.4. As shown, DECAF-ND (no debiasing) performs amongst the best methods in

[9]Specifically, we focus on the scenario of $P(X)$ being the original biased data distribution; we want a model trained on synthetic data $\mathcal{D} \sim P'(X)$ to be DP-fair when evaluated on $P(X)$, see remark Section 4.2.

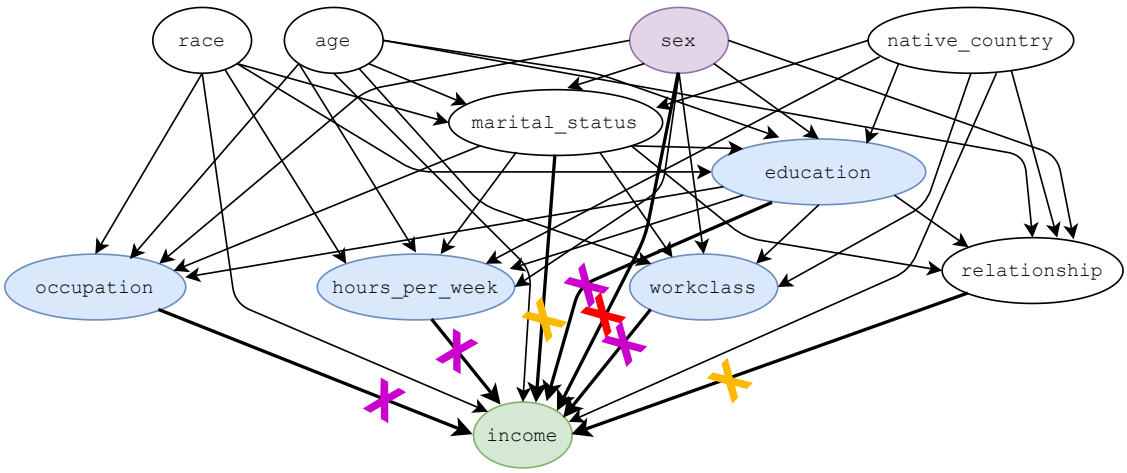


Figure 4.4: Adult dataset DAG from [37, 181]. The target variable is in green, the protected attribute in purple, and the allowed CF variables in blue. *FTU is achieved by removing:* ✗*; DP:* ✗ ✗ ✗ *; CF:* ✗ ✗ . In this particular instance, we follow [160], and remove gender discrimination. However, our method generalizes to removing the highly problematic variable `race` to `income`.

terms of data utility. Because the data utility in this experiment is measured with respect to the original (biased) dataset, we see that the methods DECAF-FTU, DECAF-CF, and DECAF-DP score lower than DECAF-ND because these methods distort the distribution – with DECAF-DP distorting the label's conditional distribution most and thus scoring worst in terms of AUROC. Note also that a downstream user who is only focused on performance would choose the synthetic data from WGAN-GP or DECAF-ND, which are also the most biased methods. Thus, we see that there is a trade-off between fairness and data utility when the evaluation distribution $P(X)$ is the original biased data.

4.8.3 Fair Credit Approval

In this experiment, direct bias, which was not previously present, is synthetically injected into a dataset $\mathcal{D}$ resulting in a biased dataset $\tilde{\mathcal{D}}$. We show how DECAF can remove the injected bias, resulting in dataset $\mathcal{D}'$ that can be used to train a downstream classifier. This is a relevant scenario if the training data $\tilde{D}$ does not follow real-world distribution $P(X)$,

but instead a biased distribution $\tilde{P}(X)$ (due to, e.g., historical bias). In this case, we want downstream models trained on synthetic data $\mathcal{D}'$ to perform well on the real-world data $\mathcal{D}$ instead of $\tilde{\mathcal{D}}$. We show that DECAF is successful at removing the bias and how this results in higher data utility than benchmarks methods trained on $\tilde{D}$.

We use the Credit Approval dataset from [31], with graph $\mathcal{G}$ as discovered by the causal discovery algorithm FGES [112] using `Tetrad` [48]. We inject direct bias by decreasing the probability that a sample will have their credit approved based on the chosen A.[10] The `credit_approval` for this population was synthetically denied (set to 0) with some bias probability β, adding a directed edge between label and protected attribute.

In Figure 4.6, we show the results of running our experiment 10 times over various bias probabilities β. We benchmark against FairGAN, as it is the only benchmark designed for synthetic debiased data. Note that in this case, the causal DAG has only one indirect biased edge between the protected variable (see Figure 4.5), and thus DECAF-DP and DECAF-CF remove the same edges and are the same for this experiment. The plots show that DECAF-FTU and DECAF-DP have similar performance to FairGAN in terms of debiasing; however, all of the DECAF-* methods have significantly better data quality metrics: precision, recall, and AUROC. DECAF-DP is one of the best performers across all 5 of the evaluation metrics and has better DP performance under higher bias. As expected, DECAF-ND (no debiasing) has the same data quality performance in terms of precision and recall as DECAF-FTU and DECAF-DP and has diminishing performance in terms of downstream AUROC, FTU, and DP as bias strength increases.

In Table 4.5, we show the results of running this experiment 10 times over our biased dataset. Note that our method was able to generate synthetic examples that had the highest AUROC (demonstrating FTU fairness). Table 4.5 shows that our method can perform debiasing without performance hits to the synthetic data metrics – i.e., there are no significant difference (outside of a standard deviation) between the top methods.

[10] We let A equal (anonymized) `ethnicity` [2, 3, 33, 81], with randomly chosen $A = 4$ as the disadvantaged population.

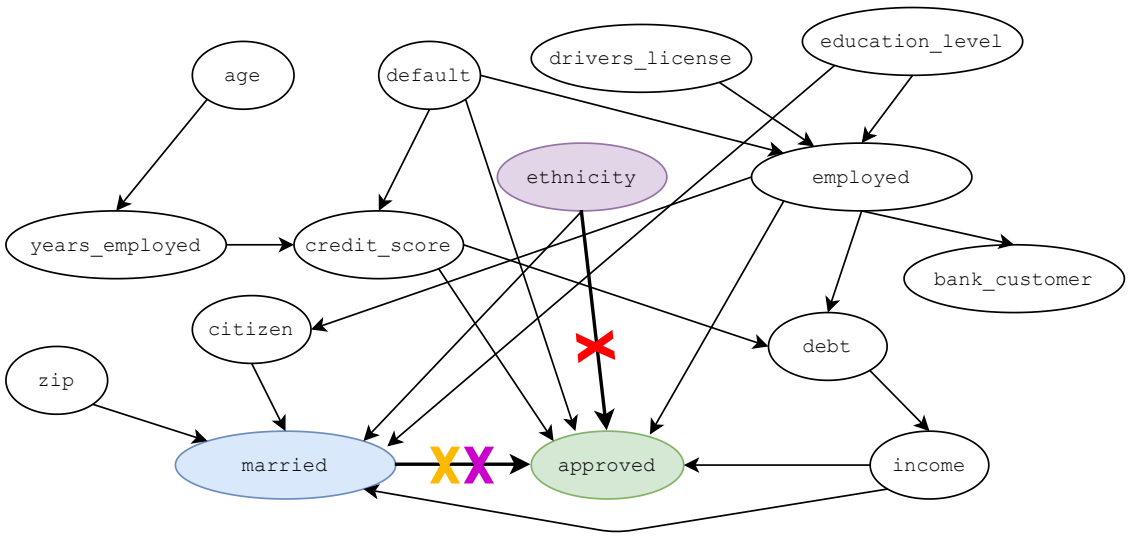


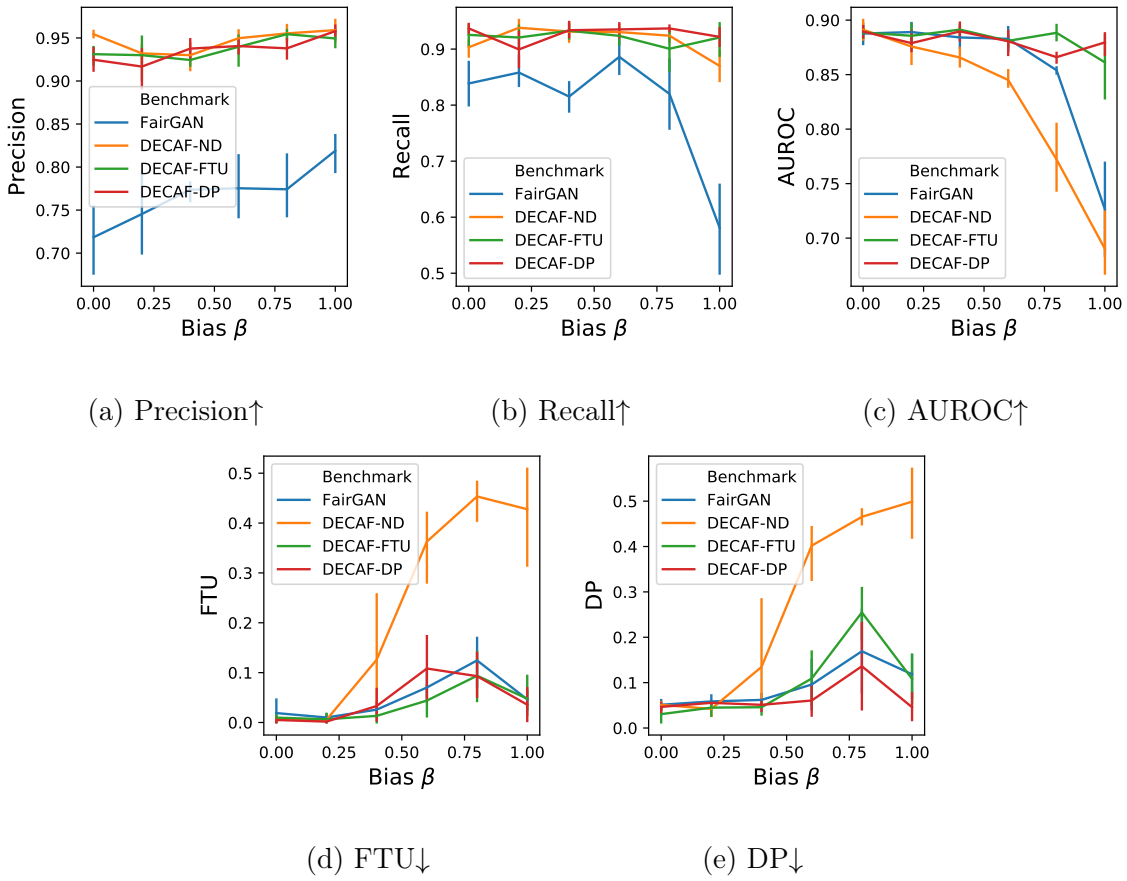
Figure 4.5: Credit Approval DAG discovered using FGES [112] and `Tetrad` [48]. The target variable is in green, the protected attribute in purple, and the allowed CF variables in blue. *FTU is achieved by removing:* ✗*; DP:* ✗✗✗*; CF:* ✗✗. Also, note that in this case CF fairness and DP fairness are the same.

4.9 Protected variable removal

A trivial method for satisfying FTU fairness, is to remove the protected attribute from downstream learners. We first provide a motivating example explaining why this is sub-optimal. We then follow this with an experiment on the Adult dataset.

4.9.1 Example

Defining fairness is task and data dependent. For example, let us assume two datasets are generated by the graphical models in Figure 4.7. Data generated by the top graph is considered fair: *Education* affects past experience (*Resume*), which together affect future job prospects (*Job*). The bottom graph is a historical example of unfairness: even if there would be no bias between *Loan* and *Race*, *redlining* (i.e. the practice of refusing a loan to people living in certain areas) would discriminate indirectly based on race [2, 3, 33, 81]. Human knowledge is thus essential for defining fairness correctly, and making sure (e.g.,



(a) Precision↑ (b) Recall↑ (c) AUROC↑

(d) FTU↓ (e) DP↓

Figure 4.6: Plot of precision **(a)**, recall **(b)**, AUROC **(c)**, FTU **(d)**, and DP **(e)** over bias strength β. FairGAN performs similarly in terms of DP and FTU, but DECAF-FTU and DECAF-DP have significantly better data quality as well as down stream prediction capability (AUROC).

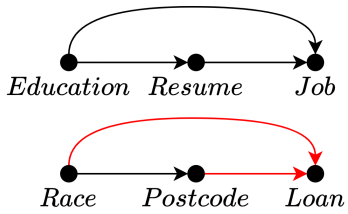


Figure 4.7: Human knowledge is essential for defining fairness.

Table 4.5: Bias removal experiment on Credit Approval dataset. Here we train an MLP on the listed dataset, and report the testing AUROC for credit approval prediction on the ground truth (GT) dataset for the biased population. Methods denoted *-PR represent modifications to the dataset by dropping the protected variable (PR). Note that there the FTU is zero for *-PR methods since the protected variable, P, has been removed.

	Data Quality			**Fairness**	
Method	Precision↑	Recall↑	AUROC↑	DP↓	FTU↓
GAN	0.921 ± 0.036	0.335 ± 0.029	0.743 ± 0.047	0.405 ± 0.077	0.194 ± 0.058
WGAN	0.970 ± 0.007	0.804 ± 0.057	0.698 ± 0.009	0.520 ± 0.036	0.461 ± 0.029
ADSGAN	0.963 ± 0.009	0.841 ± 0.052	0.708 ± 0.009	0.506 ± 0.013	0.429 ± 0.059
GAN-PR	0.794 ± 0.117	0.368 ± 0.080	0.727 ± 0.047	0.203 ± 0.196	0.0 ± 0.0
WGAN-PR	0.941 ± 0.004	0.880 ± 0.017	0.814 ± 0.019	0.406 ± 0.022	0.0 ± 0.0
ADSGAN-PR	0.945 ± 0.008	0.880 ± 0.019	0.827 ± 0.008	0.413 ± 0.029	0.0 ± 0.0
FairGAN	0.951 ± 0.012	0.663 ± 0.046	0.680 ± 0.008	0.510 ± 0.075	0.474 ± 0.054
DECAF	0.954 ± 0.012	0.601 ± 0.015	0.713 ± 0.045	0.511 ± 0.130	0.432 ± 0.127
DECAF-FTU	0.936 ± 0.017	0.901 ± 0.034	0.877 ± 0.009	0.099 ± 0.065	0.014 ± 0.012
DECAF-DP	0.940 ± 0.007	0.922 ± 0.024	0.875 ± 0.010	0.011 ± 0.029	0.015 ± 0.017

historical) bias is not propagated by the models we deploy. This example also shows why simply removing or not measuring a sensitive attribute does not suffice: not only does this ignore indirect bias, but hiding the protected attribute leads to an (additional) correlation between *Postcode* and *Loan* due to confounding. A smart debiasing method is required that can distinguish fair from unfair relations.

4.9.2 Experiment

As explained in the previous example, simply removing the protected attribute is a naive and sub-optimal solution to FTU fairness due to confounding. Let us test this experimentally. We use the same experimental setup described in Section 6 for the Adult dataset, but we include

additional metrics for protected attribute removal. We denote protected attribute removal by the *-PR suffix. In Table 4.6, we observe that naively removing the protected attribute only ensures FTU fairness, as shown by: GAN-PR, WGAN-GP-PR, and DECAF-PR. Furthermore, we observe that synthetic data quality diminishes as well for WGAN-GP-PR and DECAF-PR across precision, recall, and AUROC. For GAN-PR we see a slight improvement in data quality over GAN, however this improvement is very minimal in comparison to DECAF.

Table 4.6: Full table of bias removal experiment on Adult dataset [31] including protected removal (PR) metrics. For methods *-PR, we remove the protected attribute from the dataset before synthesizing data. ‡Note that the FTU values for the *-PR values will be zero since they are removed from the data generation method.

	Data Quality			**Fairness**	
Method	Precision↑	Recall↑	AUROC↑	FTU↓	DP↓
Original data $\mathcal{D}$	0.920 ± 0.006	0.936 ± 0.008	0.807 ± 0.004	0.116 ± 0.028	0.180 ± 0.010
GAN	0.607 ± 0.080	0.439 ± 0.037	0.567 ± 0.132	0.023 ± 0.010	0.089 ± 0.008
WGAN-GP	0.683 ± 0.015	0.914 ± 0.005	0.798 ± 0.009	0.120 ± 0.014	0.189 ± 0.024
FairGAN	0.681 ± 0.023	0.814 ± 0.079	0.766 ± 0.029	0.009 ± 0.002	0.097 ± 0.018
GAN-PR	0.632 ± 0.077	0.509 ± 0.110	0.612 ± 0.106	‡0.0 ± 0.0	0.120 ± 0.012
WGAN-GP-PR	0.640 ± 0.019	0.848 ± 0.028	0.739 ± 0.034	‡0.0 ± 0.0	0.078 ± 0.014
DECAF-PR	0.717 ± 0.021	0.839 ± 0.033	0.769 ± 0.020	‡0.0 ± 0.0	0.044 ± 0.013
DECAF-ND	0.780 ± 0.023	0.920 ± 0.045	0.781 ± 0.007	0.152 ± 0.013	0.198 ± 0.013
DECAF-FTU	0.763 ± 0.033	0.925 ± 0.040	0.765 ± 0.010	0.004 ± 0.004	0.054 ± 0.005
DECAF-CF	0.743 ± 0.022	0.875 ± 0.038	0.769 ± 0.004	0.003 ± 0.006	0.039 ± 0.011
DECAF-DP	0.781 ± 0.018	0.881 ± 0.050	0.672 ± 0.014	0.001 ± 0.002	0.001 ± 0.001

4.10 Surrogate variables

Debiasing in DECAF relies on removing edges from a trained model. As highlighted in Section 5.2, we need surrogate variables with which to replace the removed edges (Eq. 4

paper). In this section, we compare two surrogate variable mechanisms. The aim is show i) that debiasing is successful independent of the choice of surrogate variables, and ii) how prior knowledge helps in choosing surrogate variable mechanism, which leads to better data quality.

Mechanisms Let $\tilde{X}_{ij}$ denote the surrogate variable used for the removed edge $(i \to j)$, i.e. the surrogate variable that replaces the influence of X_i on X_j. Here, we compare two surrogate mechanisms for this setting:

1. $\tilde{X}_{ij} \sim P(X_i)$, i.e. we sample from the parent's marginal distribution,
2. $\tilde{X}_{ij} = \tilde{x}_{ij}$, where $\tilde{x}_{ij}$ is a fixed value.

Mechanism 1 is straightforward and most applicable when one does not know anything about the bias of a particular edge. By sampling from the marginal, each sample might use a different value of $\tilde{X}_{ij}$ when generating feature X_j, which means the diversity of the generated X_j is retained better compared to mechanism 2. Mechanism 1 for all experiments in Section 6.

On the other hand, mechanism 2 is more suitable when we know explicitly that there is bias for some values of X_i, e.g. if X_i is the protected attribute we might know there is a group $A = 0$ that is being discriminated. In this case, sampling $\tilde{X}_{ij}$ from the marginal of A is not desired: even though this means we remove direct bias from A to Y, it still means we disadvantage some individuals randomly, i.e. every time we sample $\tilde{x}_{ij} = 0$. We can employ the second mechanism instead, i.e. set $\tilde{x}_{ij} = 1$ for all individuals. This corresponds to treating everyone like they are from the advantaged group.

We repeat the experiment from Section 6.2, in which we insert direct bias from A to Y by denying loans for a disadvantaged group $A = 0$ with probability β. Our aim is to remove the direct bias from A to Y and we evaluate the synthetic data quality and bias with respect to the original, unbiased dataset. As we will see, in this setting mechanism 2 is more appropriate: we want to treat everyone from group $A = 0$ like they are from group $A = 1$, thereby removing the bias we inserted. Meanwhile, we do not want to change the way

we generate data for the advantaged group. More specifically, even though it would not be considered discrimination against a protected group, randomly denying loans to individuals of any group should still be considered unfair.

In Figure 4.8 we plot the quality metrics and FTU for three generation methods: DECAF-ND (no debiasing), DECAF-FTU1 (DECAF-FTU with surrogate mechanism 1) and DECAF-FTU2 (DECAF-FTU with mechanism 2). We plot three columns; on the left we plot the metrics for all generated data, in the middle we plot the metrics as computed on the discriminated group and on the right for the non-discriminated group.

As we can see in the FTU plots (bottom), both debiasing mechanisms are equally valid for removing the injected bias from A to Y. However, the precision metric tells a different story. Mechanism 1 disadvantages individuals randomly whenever it samples $\tilde{x}_{ij} = 0$, but this is not in line with what we want the data to be like (no disadvantage like this at all). As a result, we see that the quality of both the discriminated group goes down. The same result can be observed in the recall and AUROC plot, though the overlapping error bars prohibit strong conclusions.

In a nutshell, these results indicate that for different mechanisms for surrogate variables, data fairness is guaranteed. However, knowledge about the origins of the bias can help increase the data utility.

4.11 DAG Sensitivity

In this section, we investigate DECAF under imperfect knowledge. Here, we are curious to understand what happens when our causal knowledge has: 1) has missing edges, 2) has spurious edges, i.e., edges that we assumed falsely, and 3) edges that are reversed in directionality.

We perform this experiment on the credit approval dataset [31], with the known DAG used in the manuscript. Using an identical experimental setup as described in Section 6.2 and a bias of $\beta = 0.8$, we run our experiment 10 times each under random DAG perturbations. Starting

with the baseline DAG used in our credit approval experiment, we perform a sensitivity analysis to the following DAG perturbations:

- **Edge removal** is done by randomly edges from the baseline DAG.
- **Edge addition** is done by randomly adding edges that are constrained by the following two criteria: 1) it does not create any cycles, and 2) it does not create any new indirect bias measures. For the second condition, we ensure this by asserting that an edge is not added between the protected attribute `ethnicity` and an ancestor of `approved`. We do this to ensure that the indirect bias is held consistent across each DAG instantiation and experimental run.
- **Edge reversal** is done by randomly reversing edges in the baseline DAG while preserving acyclicity.

Results for this experiment are shown in Figure 4.9. As expected, we see that edge removal degrades synthetic data quality (precision, recall, and AUROC) as the number of edges removed increases; this is not the case for adding and reversing edges – where stable synthetic data quality is preserved. In terms of debiasing, we see that DECAF-FTU and DECAF-ND is still able to debias consistently across all DAG perturbations.

4.12 Discussion

We have proposed DECAF, a causally-aware GAN that generates fair synthetic data. DECAF's sequential generation provides a natural way of removing these edges, with the advantage that the conditional generation of other features is left unaltered. We demonstrated on real datasets that the DECAF framework is both versatile and compatible with several popular definitions of fairness. Lastly, we provided theoretical guarantees on the generator's convergence and fairness of downstream models. We next discuss limitations as well as applications and opportunities for future work.

Definitions. DECAF achieves fairness by removing edges between features, as we have

shown for the popular FTU and DP definitions. Other independence-based [17] fairness definitions can be achieved by DECAF too, as we show in Section 4.7. Just like related debiasing works [37, 26, 181, 160], DECAF is not compatible with fairness definitions based on separation or sufficiency [17], as these definitions depend on the downstream model more explicitly (e.g. Equality of Opportunity [55]).

Incorrect DAG specification. Our method relies on the provision of causal structure in the form of a DAG for i) deciding the sequential order of feature generation and ii) deciding which edges to remove to achieve fairness. This graph need not be known a priori and can be discovered instead. If discovered, the DAG needs not equal the true DAG for many definitions of fairness, including FTU and DP, but only some (in)dependence statements are required to be correct (see Proposition 1). This is shown in the Experiments, where the DAG was discovered with the PC algorithm [129] and TETRAD [48]. Furthermore, in Section 4.6 we prove that the causal generator converges to the right distribution for any graph that is Markov compatible with the data. We reiterate, however, that knowing (part of) the true graph is still helpful because i) it often leads to simpler functions $\{f_i\}_{i=1}^{d}$ to approximate,[11] and ii) some causal fairness definitions do require correct directionality—see Section 4.7.

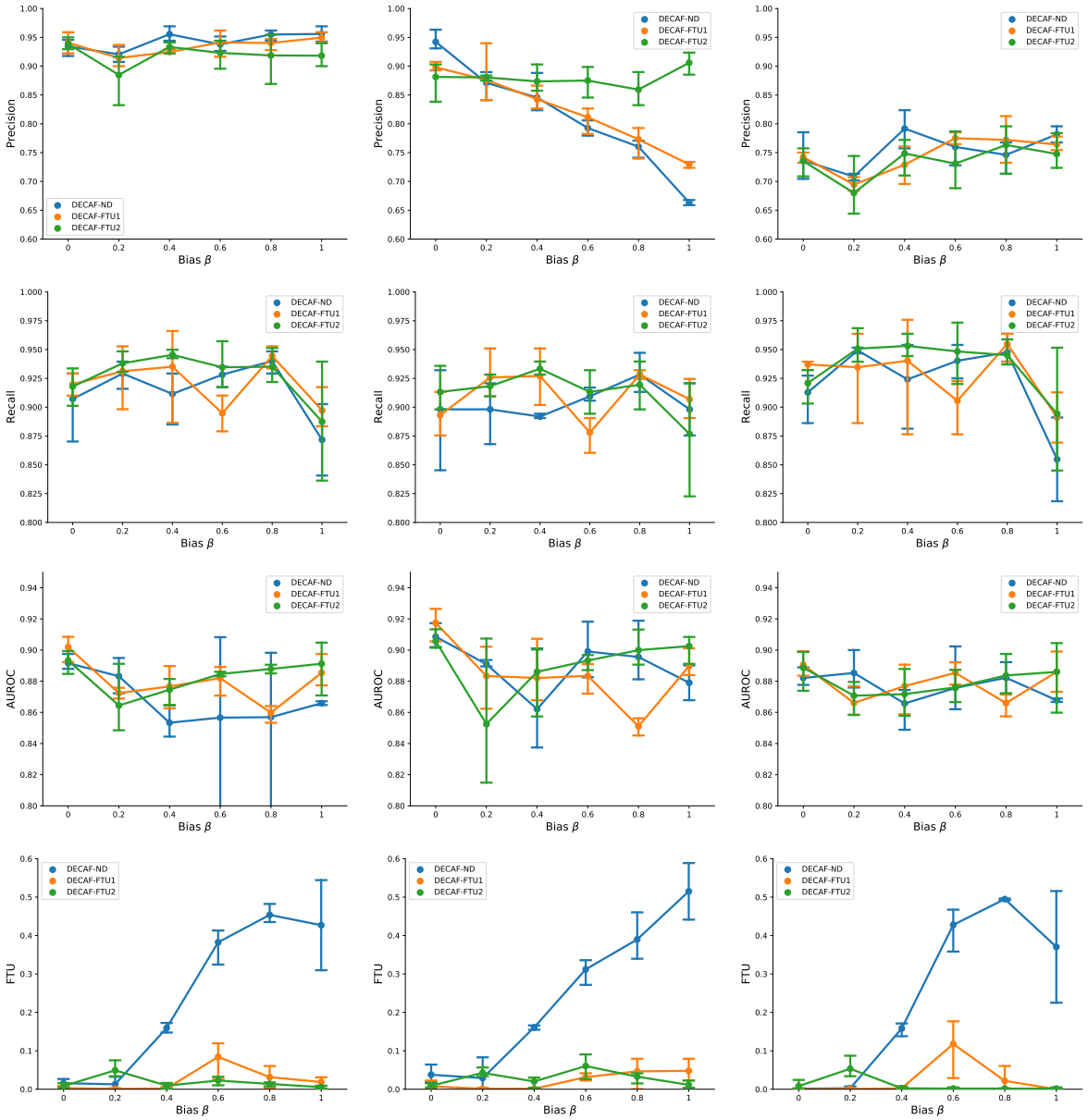
Causal sufficiency. We have focused on just one type of graph: causally-sufficient directed graphs. Extending this to undirected or mixed graphs is possible as long as the generation order reflects a valid factorization of the observed distribution. This includes settings with hidden confounders. We note that for some definitions of bias, e.g., counterfactual bias, directionality is essential and hidden confounders would need to be corrected for (which is not generally possible).

Time-series. We have focused on the tabular domain. The method can be extended to other domains with causal interaction between features, e.g., time-series. Application to image data is non-trivial, partly because, in this instance, the protected attribute (e.g., skin color) does not correspond to a single observed feature. DECAF might be extended to

[11] Specifically, this is the case if modeling the causal direction is simpler than modeling the anti-causal direction. For many classes of models this is true when algorithmic independence holds, see [106].

this setting in the future by first constructing a graph in a disentangled latent space (e.g., [72, 164]).

Social implications. Fairness is task and context-dependent, requiring careful public debate. With that being said, DECAF empowers data issuers to take responsibility for downstream model fairness. We hope that this progresses the ubiquity of fairness in machine learning.



(a) All (b) Discriminated (c) Non-discriminated

Figure 4.8: Plot of precision, recall, AUROC, and FTU over various bias strengths for **(a)** both populations (discriminated and non-discriminated), **(b)** discriminated population, and **(c)** non-discriminated population.

(a) Edge removal (b) Edge addition (c) Edge reversal

Figure 4.9: Plot of precision, recall, AUROC, FTU, and DP over **(a)** edge removal, **(b)** edge addition, and **(c)** edge reversal on the credit approval dataset.

CHAPTER 5

Exploiting Causal Structure for Robust Model Selection in Unsupervised Domain Adaptation

5.1 Introduction

There are a growing number of healthcare practitioners seeking means of leveraging machine learning in practice. However, a critical impediment to this arises when deploying a model on a testing domain with covariate distributions that differ from the training distribution, i.e., there is *covariate shift* [130]. Prior works have shown predicting under *covariate shift* may lead to unexpected behavior in the target domain [102]. This is further exacerbated by the fact that it is often the case that labels do not exist in the target domain, and transfer learning methods, such as unsupervised domain adaptation (UDA), must be performed. For example, in our experimental section, we demonstrate how during a global pandemic (COVID-19), we can transfer models from regions already afflicted by the outbreak to benefit other regions that are in the early stages of spread and still have time to respond appropriately.

There exist a number of approaches for UDA that include deep neural networks [141], generative models [122], adversarial learning [49, 144], distribution matching [86] and similarity learning [105] among others. Despite a large number of available models or approaches, there exist very few methods for UDA model selection, where the goal is to select the models that generalize best to target domains. These methods include [79] and [169], which base their model selection on domain risk estimates that leverage weighted discrepancies between the density ratios calculated from the input covariates in the target and source domains. Regardless, these methods base their estimates of target domain risk on only *model predictions*

in the source domain, assuming that the extrapolated behavior for each model will be identical in the target domain.

In contrast to these methods, in this paper, we introduce a method that calculates a score *based on model predictions on the target domain without labels*. To enable this, we exploit the notion that causality is a property of the physical world, and therefore the true causal graph representing the underlying data generating process (DGP) is invariant across domains. This assumption of strong generalizability for domain adaptation has been exploited by many, including [20, 178, 103, 87, 111] to name a few. Our method exploits graphical structure that can either be discovered from observational data via causal discovery algorithms [52] or may be known ahead of time through other more traditional means, such as experimentation and randomized trials [129, 104].

■ **Contributions.** Our primary contribution is to provide a new selection criterion that leverages causal knowledge, in the form of a causal graph, to improve model selection for UDA. The main idea is to select models whose predictions from a set of variables least violate the known causal relationships captured in the structure of the causal graph representing the underlying DGP. In doing so, our method diverges from existing UDA approaches and is uniquely able to leverage predictions on the target domain in the absence of labels. We propose a proof-of-concept implementation of our approach and show that our method can identify and select the models that better generalize to test domains where covariate distributions differ. We provide a thorough analysis of our data on oracle causal structure using synthetic data. We demonstrate on several real-world healthcare domain transfer problems, including COVID-19, that our selection method outperforms the state-of-the-art.

5.2 Related Works

5.2.1 UDA Model Selection

There exists a plethora of research addressing domain adaptation, a sub-task in the field of transfer learning. For a general overview, we refer to [110]. Our work focuses only on

Table 5.1: Related UDA selection methods. Target Domain is checked if method expoloits model predictions in the target domain.

Method	Domain adaptation	Unsup. Selection	Leverages Causality	Target Domain
Source Risk	✗	✓	✗	✗
TrCV [175]	✓	✗	✗	✗
IWCV [79]	✓	✓	✗	✗
DEV [169]	✓	✓	✗	✗
Proposed	✓	✓	✓	✓

Table 5.2: Overview of related causal domain transfer methods. General ML is checked if the method applies to general machine learning (rather than just SCMs). Partial DAGs is checked if the method applies to methods with partial graphs (incomplete causal DAGs). Intervention agnostic is checked when the method is agnostic to the intervention/perturbation location in the DAG. Non-linear is checked if the method does not make any assumptions on linearity of underlying functional connections. Model selection is checked when the method can be used for model selection.

Method	General ML	Partial DAGs	Interv. Agnostic	Non-linear	Model Selection
[20]	✗	✗	✗	✓	✗
[176]	✓	✗	✗	✓	✗
[111]	✓	✓	✗	✗	✗
[87]	✓	✓	✓	✗	✗
CAM	✓	✓	✓	✓	✓

UDA methods. [49] used a measure of source (training) risk; however, source risk is a poor estimator of target risk when target domains differ significantly in terms of *covariate shift* [169]. [175] proposed Transfer Cross-Validation (TrCV) by considering both marginal and conditional distributions in target domains but is a supervised method that requires labeled data in the test domains. [79] used an algorithm called Importance-Weighted Cross-Validation (IWCV), first proposed by [136], to select hyperparameters and models for domain adaptation under *covariate shift*. Deep Embedded Validation (DEV) was later proposed by [169] as a model selection method for UDA that provided an unbiased estimation of the target risk with bounded variance. DEV built on IWCV by learning the target distribution density ratio using neural networks. These methods do not leverage the predictions of the candidate model on the unlabeled target domain. Leveraging just a few model predictions in the target domain can significantly improve transfer performance, as shown by semi-supervised works, such as [137] and [146]. We provide a summary of related works in Table 5.1. We propose a metric that calculates a score based on model predictions on a target domain in the absence of labels by exploiting the domain invariance of causality.

5.2.2 Causality for Domain Adaptation

Structural causal models (SCM) have been applied to domain invariance or adaptation by leveraging the invariance of the causal graph that describes the underlying DGP and has origins as early as [129] or perhaps even earlier. [20] provides a theory for identifiability under transportability, assuming that a causal graph and the intervention targets are known. [178, 176] assume perfect interventions with known targets and expand the methods to more than SCMs. [87] and [111] attempt to identify some subset of covariates that will lead to the most domain transferable predictions. However, [87] points out that such an invariant set may not exist, or their algorithm may not converge on such a set. In this work, we are not interested in feature selection, building causal models, or causal discovery; instead, we propose a method for selecting the model that will result in the lowest error in the test domain. Table 5.2 provides an overview of related causal domain transfer methods that attempt to

leverage causality for domain transfer and strong generalizability. This table highlights the differences in approaches and the beneficial coverage of our proposed method. To the best of our knowledge, our method is the first causal-based method for model selection.

5.3 Causal Preliminaries

We base our notation on the framework of [104]. A causal structure of a set of variables V is a directed acyclic graph (DAG) in which each vertex corresponds to a distinct element in V, and each link represents direct functional relationships between the corresponding variables. An SCM is a pair $\langle G, \Theta_G \rangle$ consisting of a causal structure, G, and a set of parameters Θ_G compatible with G. The parameters, Θ_G, assign a function $v_i = \pi_i(pa_i, u_i)$ to each $v_i \in V$, where pa_i represents the parents (direct causes) of v_i in G and where each u_i is some i.i.d disturbance according to $P(u_i)$.

Our primary assumption, which we will refer to as *causal invariance*, is:

Assumption 7 (Causal invariance)**.** *Let G be a causal DAG representing variables V, E be a set of environments or domains, $P(V, e)$ be the corresponding distribution on V in e, and $I(P(V, e))$ denote the set of all conditional independence relationships embodied in $P(V, e)$ for a domain $e \in E$, then $\forall e_i, e_j \in E, I(P(V, e_i)) = I(P(V, e_j))$.*

Assumption 7 states that the the conditional independence relationships between variables and therefore DAG structure are invariant across domains. Similar assumptions have been made in other works [87, 111, 103, 120, 39]. We assume that the causal model M and the graph G satisfy the Markov and faithfulness conditions [104], meaning that any conditional independencies in the joint distribution of $P(V)$ are indicated by d-separation in G and vice-versa. In this work, we do not assume that the domain shift is attributed to interventional mechanisms but rather due to deviations in noise terms.

5.4 Exploiting Causality for Model Selection

In this section, we formalize our problem of model selection for UDA, present limitations of existing UDA methods, introduce definitions and theory for causal model selection, and detail our proposed methodology.

5.4.1 UDA Model Selection

In this section, we formalize the UDA model selection problem. Let E be a set of environments or domains, where $e \in E$ is a binary variable that has the value of 0 and 1 when from the source or target domain, respectively. We are particularly interested in predicting a target variable Y (with realization y) from a random variable of input features $\mathbf{X}$ (with realization $\mathbf{x}$), which take their values in a label space $\mathcal{Y}$ and feature space $\mathcal{X}$, respectively. The source training domain may differ from the target test domain, such that $P(\mathbf{X}, e=0) \neq P(\mathbf{X}, e=1)$, but we assume that *covariate shift* holds, such that $P(Y \mid \mathbf{X}, e=0) = P(Y \mid \mathbf{X}, e=1)$ [14]. Given a finite set of candidate machine learning models $\mathcal{M} : \mathcal{X} \to \mathcal{Y}$ that use input features $\mathbf{X}$ to predict a target label Y, the goal in UDA model selection is to find the model $\widetilde{m}$ having the smallest expected test error in the target test domain ($e=1$) given by

$$\widetilde{m} = \arg\min_{m \in \mathcal{M}} \mathbb{E}_{\mathbf{x} \sim P(\mathbf{x}, e=1)} \ell(m(\mathbf{x}), y), \tag{5.1}$$

where $\ell(\cdot, \cdot)$ is the desired testing loss function. Note that our goal here is not to down-select features as done in [87, 111], and we train our models using all features $\mathbf{X}$.

In UDA access to the target labels, Y are not available in the target domain, and only the features $\mathbf{X}$ are known ahead of time. Therefore, training each $m \in \mathcal{M}$ using supervised learning leverages only labeled samples from the source training domain. Specifically, the source training dataset is defined as $\mathcal{D}_{src} = \{(\mathbf{x}_i^{src}, y_i^{src})\}_{i=1}^{n^{src}}$, where n^{src} is the number of source samples. $\mathcal{D}_{src}$ can be further partitioned into a training and validation dataset $\mathcal{D}_{train} = \{(\mathbf{x}_i^{train}, y_i^{train})\}_{i=1}^{n^{train}}$ and $\mathcal{D}_{val} = \{(\mathbf{x}_i^{val}, y_i^{val})\}_{i=1}^{n^{val}}$, respectively, where $\mathcal{D}_{train} \cap \mathcal{D}_{val} = \emptyset$. The target testing set is $\mathcal{D}_{test} = \{\mathbf{x}_i^{test}\}_{i=1}^{n^{test}}$, where n^{test} is the number of test samples. For clarity, throughout this paper we denote $\widehat{Y}$ as $m(\mathbf{X})$, and our target domain is always $\mathcal{D}_{test}$.

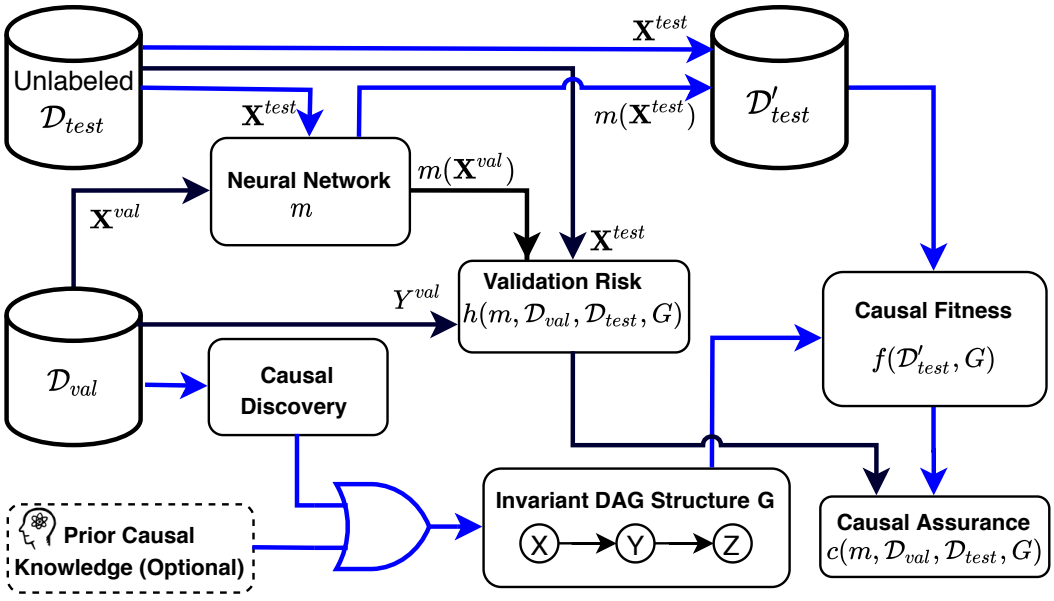


Figure 5.1: Schematic for calculating CAM c. We compare the fitness of $\mathcal{D}'_{test}$ to the invariant causal DAG structure. $\mathcal{D}'_{test}$ is generated by augmenting $\mathcal{D}_{test}$ by using $m(\mathbf{X}^{test})$ in place of Y (which does not exist in the target domain). Black arrows show the existing pathways for estimating target risk [169, 79, 136], which are unable to leverage target domain predictions. We use the causal graph to restrict our model selection. Blue arrows denote pathways unique to CAM.

5.4.2 Leveraging Predictions on the Target Domain

In this section, consider the proposed schematic in Fig. 5.1. We present the limitations of existing UDA methods (IWCV and DEV), which do not factor in the predictions of $m(\mathbf{X})$ on the target domain $\mathcal{D}_{test}$. IWCV and DEV are similar methods for approximating the target risk of m, and use the same underlying density function ρ. The density ratio or importance weighting is described by [169] as $\rho_m(\mathbf{X}) = \frac{p(e=1|\mathbf{X})}{p(e=0|\mathbf{X})} \frac{n^{src}}{n^{test}}$, where the first term $\frac{p(e=1|\mathbf{X})}{p(e=0|\mathbf{X})}$ can be estimated by a discriminative model trained to determine the domain (source or target) of a sample. Both methods for model selection are variants based on the same underlying metric function $\psi = \frac{1}{n^{src}} \sum_{i=1}^{n^{src}} \rho_m(\mathbf{x}_i^{src}) \ell(m(\mathbf{x}_i^{src}), y_i^{src})$, which does not factor in any predictions of m in the target domain. If we had access to labeled samples in $\mathcal{D}_{test}$, we could accurately

calculate the target risk as $\frac{1}{n^{test}} \sum_{i=1}^{n^{test}} \ell(m(\mathbf{x}_i^{test}), y_i^{test})$. However, this is not the case in UDA, and we use the predictions of $m(\mathbf{X})$ as a proxy for variable Y when comparing fitness to graphical structure in the target domain.

5.4.3 Causal Assurance Metric

In this section, we discuss our metric for UDA model selection. Based on our assumption of *causal invariance*, we provide a theorem of *causal preservation* as follows:

Theorem 6 (Causal preservation)**.** *Let $P(\mathbf{X}, Y, e = 0)$ be a source distribution with causal graph structure G. If $m \in \mathcal{M}$ is a perfect discriminative model, such that $Y = m(\mathbf{X})$, then*

$$I_G(G) = I(P(\mathbf{X}, m(\mathbf{X}), e = 1)), \tag{5.2}$$

where $I_G(G)$ and $I(P(\mathbf{X}, m(\mathbf{X}), e = 1))$ returns all the conditional independence relationships in G and $P(\mathbf{X}, m(\mathbf{X}), e = 1)$, respectively.

Proof. By the Markov and faithfullness assumptions, the conditional independencies in G are the same in P, such that

$$I_G(G) = I(P(\mathbf{X}, Y, e = 0)). \tag{5.3}$$

Since the goal of a perfect discriminative model m is to model a conditional distribution $P(Y|\mathbf{X})$, so that $Y = m(\mathbf{X})$, it follows that

$$I(P(\mathbf{X}, Y, e = 0)) = I(P(\mathbf{X}, m(\mathbf{X}), e = 0)). \tag{5.4}$$

By our assumption of *causal invariance* from Assumption 1, we have

$$I(P(\mathbf{X}, m(\mathbf{X}), e = 0)) = I(P(\mathbf{X}, m(\mathbf{X}), e = 1)), \tag{5.5}$$

such that we obtain

$$I_G(G) = I(P(\mathbf{X}, m(\mathbf{X}), e = 1)). \tag{5.6}$$

□

Theorem 6 allows us to replace Y by $m(\mathbf{X})$ in the target domain. The intuition behind Theorem 1 is based on the simple notion that a machine learning model should ideally issue predictions in any domain that preserve the conditional independencies of the true underlying causal structure.

Theorem 6 provides an equality that allows bridging the domain gap, such that the conditional independencies in the source domain, $I_G(G)$, equal the conditional independencies of $\mathbf{X}$ and $m(\mathbf{X})$ in the target domain, *without the need for any labels* Y. In other words, Theorem 6 implies that we desire selecting models that allow us to replace Y with $m(\mathbf{X})$ in the target domain and the conditional independencies $I_G(G)$ remain unchanged. We will later show experimentally that our method also works using subgraphs.

We can constrain our formalization in Eq. 5.1 with Theorem 6 to present an improved causal-based model selection objective in the following definition of *causal assurance*:

Definition 6 (Causal assurance)**.** *Given an invariant DAG structure G for the variables $\mathbf{X}$ and Y, a finite set of models $\mathcal{M}$, and data from a source domain ($e = 0$) and a target domain ($e = 1$). We say that a machine learning model $\widehat{m} \in \mathcal{M}$ is causally assured if and only if:*

$$\begin{aligned} \widehat{m} = &\arg\min_{m \in \mathcal{M}} \mathbb{E}_{\mathbf{x} \sim P(\mathbf{x}, e=1)} \ell(m(\mathbf{x}), y) \\ s.t.\ I_G(G) = &I(P(\mathbf{X}, m(\mathbf{X}), e = 1)). \end{aligned} \tag{5.7}$$

By the Markov and faithfulness assumptions, the constraint presented in Eq. 5.7 can be rewritten as $I_G(G) = I_G(P(\mathbf{X}, m(\mathbf{X}), e = 1))$. Since $I_G(G)$ is held constant, by Assumption 9, the only term that changes is $I_G(P(\mathbf{X}, m(\mathbf{X}), e = 1))$. Therefore, our constraint can be approximated by a metric of likelihood (DAG fitness) of G to the dataset $\mathcal{D}'_{test} = \{(\mathbf{x}_i^{test}, m(\mathbf{x}_i^{test}))\}_{i=1}^{n^{test}}$, which embodies the graphical conditional independence relationships of $I_G(P(\mathbf{X}, m(\mathbf{X}), e = 1))$. We denote this DAG fitness function as $f(\mathcal{D}'_{test}, G)$. Note that we construct $\mathcal{D}'_{test}$ from $\mathcal{D}_{test}$ simply by using $m(\mathbf{x}_i^{test})$ as a proxy for Y, which does not exist in $\mathcal{D}_{test}$. We calculate f using a graphical fitness metric used in score-based causal discovery, which we detail in the next subsection.

By the Lagrangian method, we can rewrite Eq. 5.7 as a loss function as follows:

$$\mathcal{L} = \mathbb{E}_{\mathbf{x}\sim P(\mathbf{x},e=1)}\ell(m(\mathbf{x}), y) + \lambda f(\mathcal{D}'_{test}, G). \tag{5.8}$$

The first term in the loss $\mathbb{E}_{\mathbf{x}\sim P(\mathbf{x},e=1)}\ell(m(\mathbf{x}), y)$ is the target risk and is approximated by a validation measure, such as source risk or target risk estimates, which we denote as the function $h(m, \mathcal{D}_{val}, \mathcal{D}_{test})$.

From Eq. 5.8 we provide a metric of causal assurance c in the following definition:

Definition 7 (Causal assurance metric (CAM))**.** *Let m be a trained model, $\mathcal{D}_{val}$ be a labeled validation set from the source domain, $\mathcal{D}_{test}$ be an unlabeled test dataset, and G be an invariant causal DAG for the variables $\mathbf{X}$ and Y. We define CAM as a function c, which is defined as:*

$$c(m, \mathcal{D}_{val}, \mathcal{D}_{test}, G) = h(m, \mathcal{D}_{val}, \mathcal{D}_{test}) + \lambda f(\mathcal{D}'_{test}, G), \tag{5.9}$$

where f is a scoring function that measures the fitness of G to the dataset $\mathcal{D}'_{test}$, and h is a function that returns the validation risk.

Term h can be either source risk, such as MSE or accuracy (if doing classification), or can be an approximation of target risk from methods such as IWCV or DEV, hence the inclusion of $\mathcal{D}_{test}$. For metrics that we wish to maximize, such as accuracy, subtract h instead. We present a schematic for calculating our CAM in Fig. 5.1. Note that the blue arrows denote connections that are unique to our approach and highlight that CAM is the only method to use the predictions on the test dataset, $m(\mathbf{x}_i^{test})$, in calculating its score. We provide pseudocode summarizing our methodology in Algorithm 3.

5.4.4 Appraising Causal Knowledge

The tuning factor λ controls the weighting between our *casual assurance* term and machine learning model performance. In this work, we define λ in terms of the uncertainty or probability that we know the true DAG G. Intuitively, we have defined $\lambda = \frac{|\alpha|}{|\alpha|+|\beta|}$, where α is the directed set of edges accepted to be true (by either randomized trials or causal discovery),

and β is the set of undirected edges whose causal directionality cannot be determined (by either causal discovery or experimentation). For notation, $|\alpha|$ is referring to the cardinality of α. λ will have a value of 1 when we know the orientations of all edges in the graph ($\beta = \emptyset$) and will converge to 0 as the number of edges in β increases. Therefore if we do not know any edges in our graph ($\alpha = \emptyset$) our CAM in Eq. 5.9 will be equal to the score determined by h only. When calculating c, we min-max normalize f and h between 0 and 1 over all candidate models.

5.4.5 Model Scoring and Selection

In score-based causal discovery, the Bayesian Information Criterion (BIC) is a common score that is used to discover the completed partially directed acyclic graph (CPDAG), representing all DAGs in the Markov equivalence class (MEC), from observational data. Under the Markov and faithfullness assumptions, every conditional independence in the MEC of G is also in $\mathcal{D}$. The BIC is defined as:

$$BIC(G \mid \mathcal{D}) = -LL(G \mid \mathcal{D}) + \left(\frac{\log_2 n}{2}\right) ||G||, \tag{5.10}$$

where n is the data set size, $LL(G|\mathcal{D})$ is the log-likelihood of G given $\mathcal{D}$, and $||G||$ is the dimensionality of G. For our function f in Eq. 5.9, we use the BIC score. However, since n and $||G||$ are held constant in our proposed method our function $f = -LL(G|\mathcal{D})$. To find the $LL(G|\mathcal{D})$ we use the following decomposition: $LL(G \mid \mathcal{D}) = -n \sum_{v_i pa_i} H_{\mathcal{D}}(v_i \mid pa_i)$, where pa_i are the parent nodes of v_i in G, and H is the conditional entropy function which is given by [29] for discrete variables and by [115] for continuous or mixed variables. Note that when using causal discovery in our method and their are multiple candidate DAGs in the MEC, any can be chosen for use in our metric. That is because we are holding the DAG constant and changing the data (via $m(\mathbf{X})$ in place of Y), such that all DAGs in the MEC will have the same statistical score by definition [29, 104].

The causal DAG of the variables in $\mathcal{D}$ is often never wholly known ahead of time. In most practical cases, only a few of the causal relationships may be known a priori. If it were the

Algorithm 3 Select model with lowest CAM

Input: A labeled source dataset $\mathcal{D}_{src}$ with set of input features $\mathbf{X}$ and target label Y, a set of untrained models $\mathcal{M}$, an unlabeled target dataset $\mathcal{D}_{test}$ containing only features $\mathbf{X}$, and optional prior causal DAG G_p.
Output: The most *causally assured* model $\widehat{m} \in \mathcal{M}$.
Function: `SelectModel`$(M, \mathcal{D}_{src}, \mathcal{D}_{test}, [G_p])$
Divide $\mathcal{D}_{src}$ into two disjoint sets $\mathcal{D}_{train}$ and $\mathcal{D}_{val}$ for model training and validation, respectively.
Train each model $m \in \mathcal{M}$ on $\mathcal{D}_{train}$ until performance of m on $\mathcal{D}_{val}$ converges (stops improving).
if G_p is completely known **then**
    $G \leftarrow G_p$.
else if G_p is partially known **then**
    $G \leftarrow$ causal discovery on D_{src} constrained by G_p.
else
    $G \leftarrow$ causal discovery on D_{src}.
end if
for $m \in \mathcal{M}$ **do**
    Generate dataset $\mathcal{D}'_{test}$ from $\mathcal{D}_{test}$ by concatenating with $m(X)$.
    Calculate the causal assurance term $f(\mathcal{D}'_{test}, G)$ by a graphical fitness metric, such as BIC.
    Calculate the validation error $h(m, \mathcal{D}_{val}, \mathcal{D}_{test}, G)$.
end for
return $\widehat{m} \in \mathcal{M}$ with lowest $c(m, \mathcal{D}_{val}, \mathcal{D}_{test}, G)$

case that all causal edges were known ahead of time (with certainty), then causal methods, such as a causal Bayesian network [104], could be used. However, we emphasize that the goal of this work is model selection and not model development.

Algorithm 3 scales linearly with the number of models in $\mathcal{M}$. Because our CAM selection

algorithm requires one external loop over $\mathcal{M}$, our method scales linearly with the number of models in $\mathcal{M}$. The overall computational complexity of calculating CAM is $\mathcal{O}(|\mathcal{M}| \times T(G, \mathcal{D}))$, where $|\mathcal{M}|$ is the number of candidate models and $T(G, \mathcal{D})$ is the computational complexity of calculating dataset $\mathcal{D}$ to DAG G fitness. Since we use the LL as our graph fitness score, this requires calculating the conditional entropy of each node given its predecessors. This has a worst-case computational complexity of $\mathcal{O}(|G|^2)$, since the asymptotic maximum number of connections in a graph is $\frac{|G|(|G|-1)}{2}$, where $|G|$ is the number of nodes in G.

5.5 Experiments

Experiments were performed on both synthetic and real-world datasets. For the synthetic data experiments, the true causal graph was first established and used as the underlying DGP to generate each dataset. Conversely, the correct causal graph was not fully known for the real-world datasets, and causal discovery was used to recover the causal graph. We implemented our method using `Tensorflow`[1].

5.5.1 Evaluation Details

The following describes our experimental settings used for both synthetic and real-world data.

5.5.1.1 Benchmark methods

We compare our CAM to three benchmark UDA selection methods: validation source risk (validation MSE), IWCV [79], and DEV [169]. For each method, we use their published hyperparameters. For IWCV, since the target density ratio may not be known ahead of time, we use a discriminative neural network to learn the density ratio as part of the DEV method provided by [169]. Although our method applies to machine learning models in general, our focus is on neural networks.

[1]Source code will be made available upon acceptance.

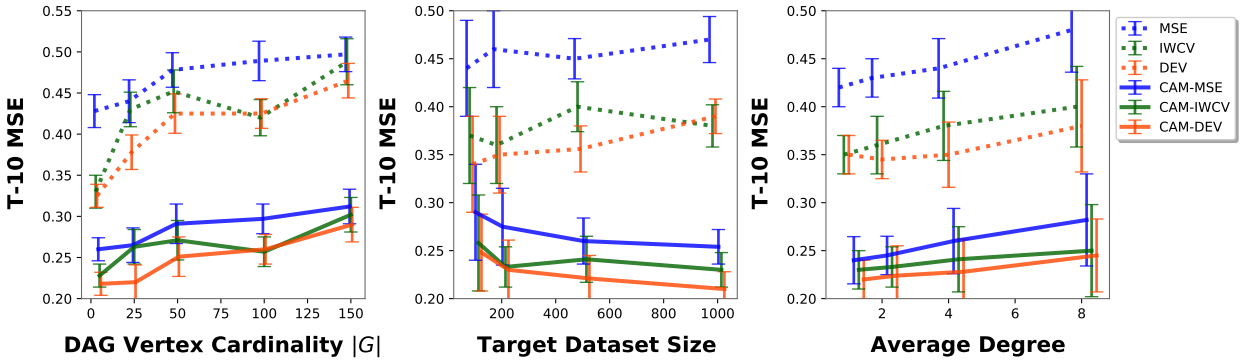


Figure 5.2: T-10 MSE (± standard error) of UDA methods over various DAG cardinalities (**left**), target dataset sizes (**middle**) and average degree of G (**right**). CAM-* denotes CAM used with each * as function h in Eq. 5.9 (e.g. CAM-DEV uses DEV for h). Using CAM significantly improves over each benchmark as shown in the difference between the solid and dashed lines.

5.5.1.2 Evaluation metrics

We evaluate our models based on the following performance metrics. For each of our experiments, we evaluated and ranked models according to their performance on our validation set by each method. The first metric we propose examines the top 10% of models selected by candidate methods on the validation set to show the generalization improvement in terms of test error. We refer to the average test error of the top 10% of selected models by each method as T-10 error. The second metric we propose examines the ability to rank the entire set of trained models (rather than selecting the top 10%) by candidate methods in terms of test performance. Here, we use a list inversion count (IC), which measures the number of element-wise inversions required for sorting a list. IC can be thought of as a metric of how sorted or ranked a list is, where a perfectly sorted list has an IC of 0. The asymptotic maximum number of inversions for a list of n elements is $n(n-1)/2$, which we use to normalize our reported IC metric. Note that there are differences in the two metrics, where T-10 highlights the top-end "best" selection performance and IC examines the entire list of candidate models.

5.5.1.3 Network architecture and training pool

To generate a diverse set of candidate models to choose from, we draw networks from the following pool of MLPs. Each MLP can have anywhere between 2 and 4 layers, with each layer having between d and $2d$ hidden neurons, where d is the number of input features. Furthermore, each layer uses ReLU activation and is followed by a dropout layer with a droprate randomly selected from $\{0\%, 20\%, 40\%, 60\%\}$. Training batch sizes are also varied and randomly chosen from $\{32, 64, 128\}$. Each network is trained with the Adam optimizer using a learning rate of 0.001 until validation error stops improving.

5.5.1.4 Discovered causal graphs

For each of the real datasets we made no prior assumptions on the underlying causal graph and used causal discovery to identify the causal graph. Specifically, we used the fast greedy equivalence search (FGES) [112] algorithm on the entire dataset using the `Tetrad` software package. Only the directed edges that were output in the CPDAG by FGES were considered as known edges in the causal graphs, which are shown in Fig. 2.

`Tetrad` allows prior knowledge to be specified in terms of required edges that must exist, forbidden edges that will never exist, and temporal restrictions (variables that must precede other variables). For an example output, see Fig. 5.5, where the green vertices represent the prediction target. Solid bordered and dashed bordered vertices represent continuous and discrete variables, respectively. Solid edges represent the known causal connections, and dashed edges represent discovered edges. The `Tetrad` software package automatically handles mixed connections, i.e., edges between discrete and continuous variables.

5.5.2 Synthetic Experiments

Using synthetic data, we can evaluate our method under oracle conditions when the complete causal graph is known. For each of the simulations, we generated a random Erdos-Renyi random DAG, G, with $|G|$ vertices and between $|G|$ and $\frac{|G|(|G|-1)}{2}$ edges (the maximum

number of edges in a DAG) between them. Using the structure of G we synthesized two datasets $\mathcal{D}_{src}$ and $\mathcal{D}_{test}$ with functional relationships between variables with directed edges between them in G and applied Gaussian noise to each. $\mathcal{D}_{src}$ was generated by sampling 5000 data points with a Gaussian noise having a mean of 0 and variance of 1 and was randomly partitioned into a model training and validation sets of 80% and 20%, respectively. The validation set was used to terminate model training. During model selection and calculation of our CAM, we used only the unlabeled input $\mathbf{x}$ values from $\mathcal{D}_{test}$ to select models, and only used the y values for evaluation of model generalization performance. The input features of $\mathcal{D}_{src}$ were min-max normalized between 0 and 1, and the min and max values for each feature were saved for scaling $\mathcal{D}_{test}$ accordingly.

[111] shows that for predictive models, when the location of the interventional perturbation is not known, the invariant set of predictors is the causal parents of the target variable. This stems from the fact that an intervention of a child node will not propagate in the anti-causal direction to its parents [120]. Because we make no assumption to the location of any perturbation in G, throughout the remainder of this manuscript, assume that the outgoing edges of the target V_i have been removed in G (rendering V_i conditionally independent of any child) when calculating our *causal assurance* metric.

We enumerated all nodes in G randomly. To prevent the magnitude of the leaf nodes from becoming overwhelmingly large relative to the root nodes, each node was instantiated as a function of its parents' values that were either added or subtracted for even or odd enumerated parents, respectively. We provide pseudocode for our synthetic DGP in Algorithm 4. We also specify in Algorithm 4 how perturbations can be applied to nodes p with mean μ_p and standard deviation σ_p. Note that perturbations are used for shifting noise variables at test time.

For each synthesized DAG G, we randomly selected a target variable from G connected to at least one other variable in G. We then trained 50 random deep neural networks from our pool of candidate networks on our training set to predict our target variable. We evaluated and ranked each of the 50 models using each method and repeated this for 50 different DAG

Algorithm 4 Synthetic Data Generation (DGP)

Input: An Erdos-Renyi graphical structure G, a mean μ, standard deviation σ, edge weights w, a dataset size n, a list of perturbation nodes p, a perturbation mean μ_p and a perturbation standard deviation σ_p.

Output: A dataset according to G with n samples and perturbation applied at nodes p.

Function: `gen_data`$(G, \mu, \sigma, w, n, \mu_p, \sigma_p)$**:**

$e \leftarrow$ edges of G

$G_{sorted} \leftarrow$ `topological_graph_sort`(G)

$ret \leftarrow$ empty list

for $node \in G$ **do**

  if $node \in p$ **then**

    Append to $ret[node]$ a list of Gaussian (μ_p and σ_p) randomly sampled list of size n.

  else

    Append to $ret[node]$ a list of Gaussian (μ and σ) randomly sampled list of size n.

  end if

end for

for $node \in G_{sorted}$ **do**

  for $par \in \{parents(node)\}$ **do**

    if `is_even`(par) **then**

      $ret[node] \mathrel{+}= ret[par] * w(par, node)$, where $w(par, node)$ is the edge weight from par to $node$.

    else

      $ret[node] \mathrel{-}= ret[par] * w(par, node)$, where $w(par, node)$ is the edge weight from par to $node$.

    end if

  end for

end for

return ret.

instantiations each.

Because our prior knowledge about the underlying causal graph is perfect, we set $\lambda = 1$ for calculating our CAM in Eq. 5.9. Unless specified, we performed experiments on default test sets of 2000 samples and DAGs having nodes between 10 and 100 vertices. We created our test datasets $\mathcal{D}_{test}$ with at least one of the variables in G randomly perturbed with a mean of 1 and a variance of 2 (rather than mean of 0 and variance of 1 as used in $\mathcal{D}_{src}$ for training, validation, and selection). The noise terms in all of the remaining variables are unchanged in $\mathcal{D}_{src}$.

We perform three experiments to investigate various sensitivities of CAM. Furthermore, we use each benchmark (MSE, IWCV, or DEV) as our function h when calculating our CAM to demonstrate how CAM can be used in conjunction with these target risk estimators. Fig. 5.2 shows that IWCV and DEV provide a noticeable improvement over MSE. However, CAM provides a larger improvement in terms of T-10 error over each benchmark across all vertex cardinalities, target dataset sizes, and DAG degrees. This is shown by the improvement of the solid lines over the dashed lines in Fig. 5.2. Also, we observe that as the target data size increases, CAM provides more consistent selection results.

In Fig. 5.3, we show a sensitivity analysis of our method on synthetic data. We use the same experimental setup used in the synthetic experiments from the main paper on a random DAG instantiation. Here we see that there is a linear relationship between λ and performance in terms of T-10 error.

5.5.3 Erroneous or Incomplete DAGs

In this experiment, we investigate the performance trade-off under two practical conditions. In the first hypothetical scenario, we investigate the sensitivity of CAM to incomplete knowledge, where we know only a subgraph perfectly. In the second hypothetical situation, we investigate the sensitivity of our method to "imposter" DAGs, where we know a portion of the causal graph correctly, but there is some number of edges that are spurious or reversed. We used our same synthetic experimental setup, except we mutilate our DAGs to form either subgraphs

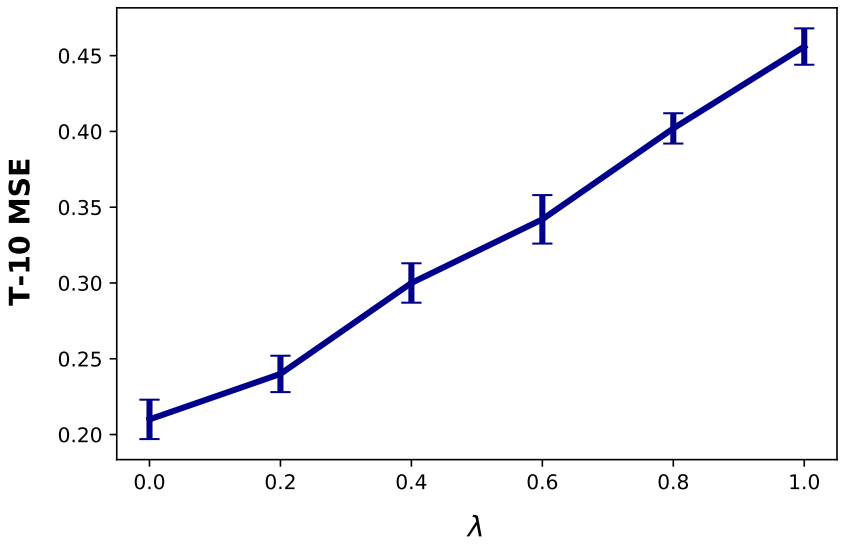


Figure 5.3: Sensitivity analysis of λ on synthetic data.

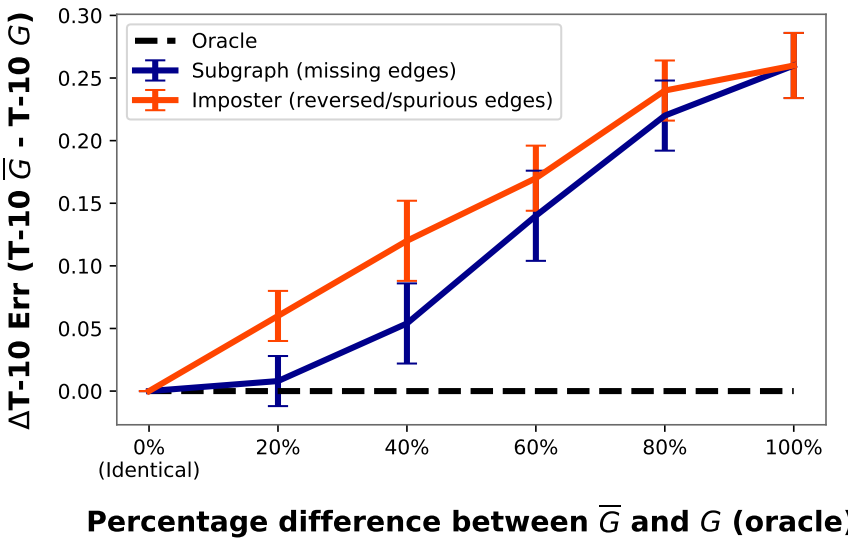
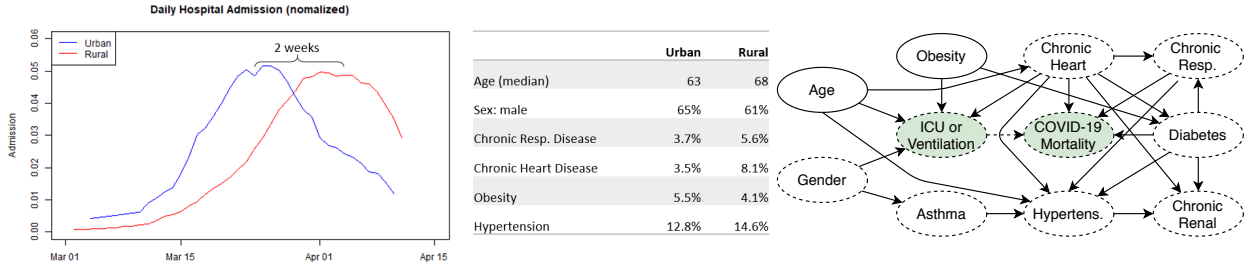


Figure 5.4: Performance of CAM on subgraph and imposter experiment: ΔT-10 error is the difference of the T-10 error of $\overline{G}$ and G using our CAM metric versus the percentage of graphical distance (in terms of total edges). Note that G is the oracle causal graph and is held static across the x-axis.

or "imposters". We set λ to 1 since we are assuming the graph is truth (even though it is incorrect). We use DEV as our validation risk metric and show our results in Fig. 5.4, which

shows the ΔT-10 error, i.e., the difference in T-10 error of the subgraph or imposter graph $\overline{G}$ and G, versus the percentage graph difference (between G and $\overline{G}$). The graphical difference is calculated in terms of the percentage of edges that are mutated or removed. Fig. 5.4 shows the correlation between the correctness of the causal graph and the relative model selection improvement. This correlation testifies to the validity of our approach. Furthermore, Fig. 5.4 suggests that we should be conservative in adding edges to our causal knowledge and favor using a subgraph over a false "imposter" DAG that may have erroneously added or flipped edges.



	Urban	Rural
Age (median)	63	68
Sex: male	65%	61%
Chronic Resp. Disease	3.7%	5.6%
Chronic Heart Disease	3.5%	8.1%
Obesity	5.5%	4.1%
Hypertension	12.8%	14.6%

Figure 5.5: **Left:** In the UK, COVID-19 pandemic hit urban areas before spreading to rural areas, which motivates us to transfer a model learned from the urban to the rural population. **Middle:** Feature subset showing there exists a significant *covariate shift* between urban and rural populations with the urban population younger and with fewer preexisting conditions. **Right:** Discovered COVID-19 DAG for all covariates. Dashed and solid lines represent discrete or continuous variables respectively.

5.5.4 Responding to COVID-19

In this section, we demonstrate improved model selection for *covariate shift* using our CAM on several real-world healthcare prediction problems. The primary problem we focus on is centered around COVID-19 where we predict two types of ventilation, patient mortality, and the number of ICU beds using a real-world COVID-19 dataset.

The COVID-19 pandemic has resulted in significant mortality and has challenged healthcare systems worldwide. At the peak of the COVID-19 outbreak, many countries, unfortu-

Table 5.3: Top 10% model selection test performance (in terms of T-10 AUROC with standard error) on real data experiments. CAM represents our algorithm used with each DEV as our function h in Eq. 5.9. Bold denotes best-performing methods. Note that all of our proposed CAM results in higher testing AUROC on target domains.

Dataset	h:AUROC	IWCV	DEV	CAM
COVID-19 Invasive Vent. (**UK Urban** → **UK Rural**)	0.629 ± 0.011	0.633 ± 0.015	0.641 ± 0.012	$\mathbf{0.662 \pm 0.014}$
COVID-19 Non-Invasive Vent. (**UK Urban** → **UK Rural**)	0.791 ± 0.008	0.795 ± 0.006	0.798 ± 0.010	$\mathbf{0.811 \pm 0.007}$
COVID-19 Mortality (**UK Urban** → **UK Rural**)	0.582 ± 0.023	0.588 ± 0.029	0.589 ± 0.032	$\mathbf{0.602 \pm 0.021}$
COVID-19 ICU Beds (**UK Urban** → **UK Rural**)	0.718 ± 0.010	0.724 ± 0.013	0.725 ± 0.012	$\mathbf{0.743 \pm 0.012}$
Prostate Cancer Mortality (**UK Biobank** → **US SEER**)	0.612 ± 0.029	0.627 ± 0.013	0.627 ± 0.025	$\mathbf{0.638 \pm 0.018}$
MAGGIC Chronic heart failure (**Europe** → *)	0.720 ± 0.005	0.732 ± 0.010	0.733 ± 0.011	$\mathbf{0.748 \pm 0.009}$

nately, experienced a shortage of life-saving equipment such as ventilators and ICU beds. Considering data from the UK outbreak, we observed that the pandemic hit the urban area first before spreading to the rural areas (Fig. 5.5). This implies that we could potentially transfer models trained on the urban population to benefit the rural areas immediately if we reacted promptly. However, there is a significant domain shift as the rural population are older and have more comorbidities. Furthermore, there may be no labeled samples available at the time of model deployment in rural areas.

5.5.4.1 COVID-19 patient statistics across geographical locations

We acquired de-identified COVID-19 Hospitalization in England Surveillance System (CHESS) data from Public Health England (PHE). The data contains 7,714 hospital admission, including 3,092 ICU admissions from 94 NHS trusts across England. The dataset features comprehensive information on patients' general health condition, COVID-19 specific risk factors (e.g., comorbidities), basic demographic information (age, sex, etc.), whether they were admitted to the ICU, what treatment (e.g., ventilation) they received, and their outcome by April 20th, 2020 (609 deaths and 384 discharges). We split the data set into a source dataset containing 2,552 patients from urban areas (mostly Greater London area) and a target

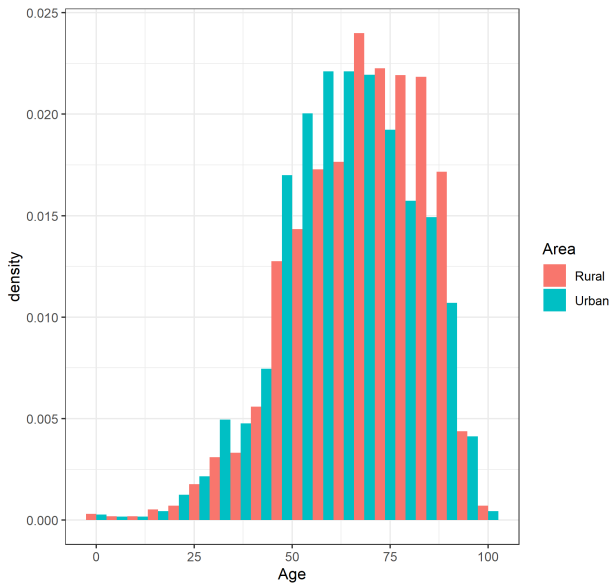


Figure 5.6: Age distribution for urban and rural patients. The median age of rural patients is five years older than urban.

dataset of the remaining 5,162 rural patients. The characteristics of the two populations are summarized in Fig. 5.5.

Figure 5.6 shows the histogram of the age distribution for urban and rural patients. It is clear from the plot that the rural population is older, and therefore at higher risk of COVID-19. Table 5.4 presents statistics about the prevalence of preexisting medical conditions, the treatments received, and the final outcomes for patients in urban and rural areas. We can see that rural patients tend to have more preexisting conditions such as chronic heart disease and hypertension. The higher prevalence of comorbid conditions complicates the treatment for this population.

5.5.4.2 COVID-19 UK (urban) $\rightarrow$ UK (rural)

We first performed causal discovery using the FGES algorithm on the patients from the urban area. The discovered graph (Fig. 5.5) agrees well with the literature [156, 99]. We selected 50 random models from our pool of candidate models and used the same training regime as in the synthetic experiments along with the discovered COVID-19 causal DAG shown in

Table 5.4: Comparison of key features of urban and rural COVID-19 patients in the data set.

	Urban		Rural	
	Perc.	**Count**	**Perc.**	**Count**
Sex at Birth	65%	1446	62%	3388
Chronic Respiratory	4%	81	6%	310
Obesity	5%	121	4%	225
Chronic Heart	4%	80	8%	444
Hypertension	13%	285	15%	798
Asthma	4%	92	6%	326
Diabetes	9%	197	11%	589
Chronic Renal	2%	45	3%	175
Noninvasive Ventilation	7%	160	6%	342
Invasive Ventilation	21%	456	16%	879
Death	18%	402	19%	1014
Discharge	12%	276	21%	1164

Fig. 5.5 as our invariant DAG. We evaluated the best model selected by each model selection method based on the T-10 AUROC on each task in Table 5.3. We see that CAM identified the models that resulted in higher AUROC in the UK's rural areas without access to labeled data.

5.5.5 Results on Real Healthcare Data

We provide additional illustrative results for predicting prostate cancer in the US Surveillance, Epidemiology, and End Results (SEER) [28] dataset for models trained on UK Biobank [127] data. We also apply our method to predict chronic heart failure across a collection of 30 independent studies using the Meta-Analysis Global Group in Chronic Heart Failure (MAGGIC) [154] datasets.

5.5.5.1 Prostate cancer

In this case, we are interested in deploying a machine learning model for prostate cancer but have access to only labeled data in the UK Biobank [127] dataset, which has approximately 10,000 patients. We would like to deploy our models in the United States, where we have access to many samples of patient features. For this target domain, we use the SEER [28] dataset, which has over 100,000 samples. Our objective is to predict patient mortality, given the patient features.

5.5.5.2 MAGGIC dataset

This MAGGIC dataset [154] is a collection of 30 studies; it is comprised of 46,817 patients collected across various locations to study the survival time after heart failure. Each patient may have one or more comorbidities such as myocardial infarction or angina, which are documented in this dataset along with patient attributes such as gender or age. We assumed as prior knowledge that the patient attributes such as gender, age, and ethnicity, could not be a descendant (effect) of any of the other observed variables. We also assumed for prior

knowledge that the last observation was the survival time, which could not be an ancestor (cause) of any of the other observed variables.

We use the same experimental set-up detailed for the prior experiments. Table 5.3 shows that our CAM can reliably select the models that result in improved performance in terms of T-10 AUROC for both prediction problems. For the MAGGIC experiment, we trained on the European population and report the average testing error applied to the remaining 29 other studies.

Table 5.5: Top 10% model selection test performance (in terms of T-10 MSE (**top**) and inversion count (**bottom**) with standard error) on real data experiments. CAM-* represents our algorithm used with each * as our function h in Eq. 5 (e.g. CAM-DEV uses DEV as our algorithm for calculating function h). Bold denotes best performing models. Note that all of our methods CAM-* have lower testing MSE on target domains and inversion counts than * methods across all datasets (shown on RHS).

Dataset	MSE	IWCV	DEV	CAM-MSE	CAM-IWCV	CAM-DEV
Pima Diab.	0.577 ± 0.084	0.350 ± 0.041	0.344 ± 0.053	0.499 ± 0.064	0.301 ± 0.034	**0.293 ± 0.032**
Stud. Exams	0.424 ± 0.054	0.333 ± 0.032	0.319 ± 0.031	0.350 ± 0.042	0.248 ± 0.039	**0.246 ± 0.013**
Powerlift	0.325 ± 0.056	0.301 ± 0.080	0.300 ± 0.050	0.205 ± 0.021	0.182 ± 0.025	**0.176 ± 0.053**
Bike Share	0.105 ± 0.003	0.081 ± 0.016	0.080 ± 0.086	0.021 ± 0.004	0.018 ± 0.016	**0.012 ± 0.004**
PIMA	0.614 ± 0.044	0.472 ± 0.036	0.482 ± 0.025	0.504 ± 0.044	0.432 ± 0.034	**0.414 ± 0.030**
Student Exams	0.523 ± 0.034	0.491 ± 0.038	0.501 ± 0.033	0.424 ± 0.032	0.402 ± 0.041	**0.392 ± 0.032**
Power Lifting	0.284 ± 0.063	0.398 ± 0.073	0.437 ± 0.080	0.202 ± 0.062	0.219 ± 0.044	**0.193 ± 0.073**
Bike Sharing	0.347 ± 0.019	0.249 ± 0.023	0.201 ± 0.009	0.301 ± 0.035	0.242 ± 0.021	**0.134 ± 0.017**

5.5.6 Results on Public Datasets

In practice, often times the complete underlying causal DAG is unknown, and we must rely on DAG recovery via causal discovery. In this experiment, we explore using incomplete prior knowledge on four publicly available datasets and a medical dataset. The publicly available

datasets include the Pima Indian Diabetes Database [31], Student Performance in Exams [67], Open Powerlifting [109], and Bike Sharing in Washington D.C. [36] datasets. The Bike Sharing dataset was used in prior causal works by [114].

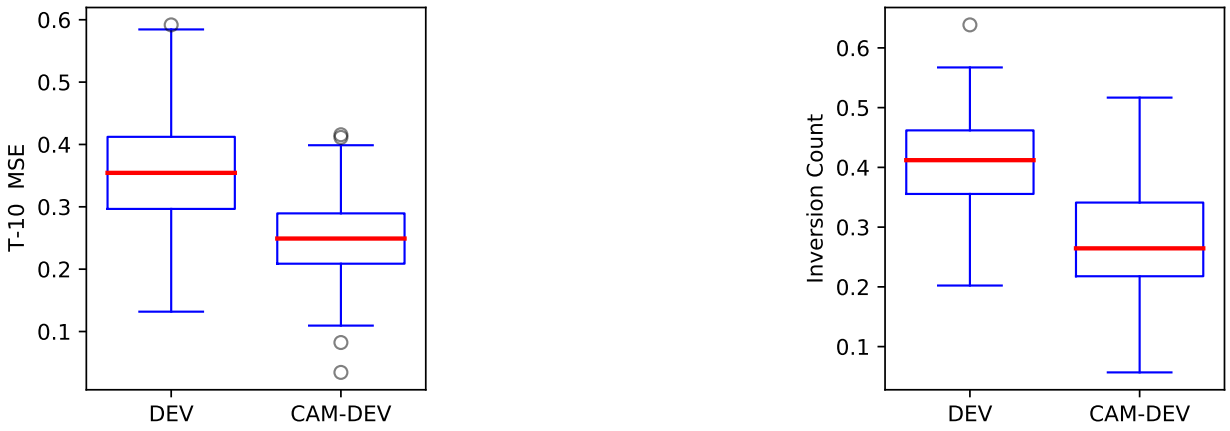
For each dataset, we used either prior knowledge or a causal discovery algorithm to determine the full causal graph. If the complete causal graph was not known ahead of time, we discovered the remaining causal connections from the data using the Fast Greedy Equivalence Search (FGES) algorithm by [112] on the entire source dataset using the `Tetrad` software package [48].

For each of the datasets, we create target test sets in either of two ways: 1) randomly choosing a continuous variable and randomly holding-out either 20% of the lowest or greatest samples for that variable such that these end-point values were never seen during any phase other than testing, or 2) by randomly choosing a discrete variable and holding out one of the labels for the test set (e.g., training on only females and testing on males).

We randomly split the source training set into an 80% training and 20% validation split, the latter of which is used for calculating our CAM score. For each dataset, we identically trained 50 random deep learning models from our pool of candidate architectures on our training set. We then evaluated and ranked each of the 50 models by their performance on our selection set, by each candidate method, and repeated this 100 times for each dataset. Results in terms of T-10 MSE and inversion count are presented in Table 5.5, showing that using CAM, we were able to select the most performant models across each dataset.

5.5.7 Going Beyond Existing Feature Selection Algorithms

In this subsection, we demonstrate how CAM can be used on a subraph to improve state-of-the-art causal domain adaptation algorithms. We use the invariant feature selection methods presented in [87], which is a more generalized approach than [111]. We used the same experimental setup mentioned in the previous subsection, but instead of using all of the input features, we use the input features identified by [87] that will result in the most domain transferable predictions. We will refer to this feature selection approach as CIFS (for causally



(a) T-10 Error

(b) Inversion Count

Figure 5.7: Performance improvement of CIFS (causal invariant feature selection [87]) using CAM. (a) T-10 error. (b) Inversion count.

invariant feature selection). Again, we used our synthetic experimental setup, except we apply there CIFS method to select the most invariant causal features to use as input features. We apply our selection method to 100 trained CIFS models. We use DEV as our baseline risk estimate on the reduced feature space (by CIFS). Fig. 5.7 shows that our method is able to improve model selection in terms of both inversion count and T-10 error model selection.

5.6 Conclusion

We have presented a model selection method that considers a candidate model's predictions on an unlabeled test domain by leveraging the invariance of causal graphs to improve UDA. To the best of our knowledge, this is the first such method for UDA to explicitly leverage model predictions in the target domain. We have demonstrated improved performance over the state-of-the-art on synthetic data with oracle knowledge and real data using causal discovery. For future work, we would like to integrate our methodology into a differentiable loss that can be used during model training. Although we frame our method in a healthcare setting, we envision our algorithm being leveraged by any party interested in deploying machine learning models across domains.

CHAPTER 6

Selecting Treatment Effects Models for Domain Adaptation Using Causal Knowledge

6.1 Introduction

Causal inference models for estimating individualized treatment effects (ITE) are designed to provide actionable intelligence as part of decision support systems and, when deployed on mission-critical domains, such as healthcare, require safety and robustness above all [134, 10]. In healthcare, it is often the case that the observational data used to train an ITE model may come from a setting where the distribution of patient features is different from the one in the deployment (target) environment, for example, when transferring models across hospitals or countries. Because of this, it is imperative to select ITE models that are robust to these covariate shifts across disparate patient populations. In this paper, we address the problem of *ITE model selection in the unsupervised domain adaptation (UDA)* setting where we have access to the response to treatments for patients on a source domain, and we desire to select ITE models that can reliably estimate treatment effects on a target domain containing only unlabeled data, i.e., patient features.

UDA has been successfully studied in the predictive setting to transfer knowledge from existing labeled data in the source domain to unlabeled target data [49, 144]. In this context, several model selection scores have been proposed to select predictive models that are most robust to the covariate shifts between domains [136, 169]. These methods approximate the performance of a model on the target domain (*target risk*) by weighting the performance on the validation set (*source risk*) with known (or estimated) density ratios.

However, ITE model selection for UDA differs significantly in comparison to selecting predictive models for UDA [126]. Notably, we can only approximate the estimated counterfactual error [12], since we only observe the factual outcome for the received treatment and cannot observe the counterfactual outcomes under other treatment options [129]. Consequently, existing methods for selecting predictive models for UDA that compute a weighted sum of the validation error as a proxy of the target risk [169] is suboptimal for selecting ITE models, as their validation error in itself is only an approximation of the model's ability to estimate counterfactual outcomes on the source domain.

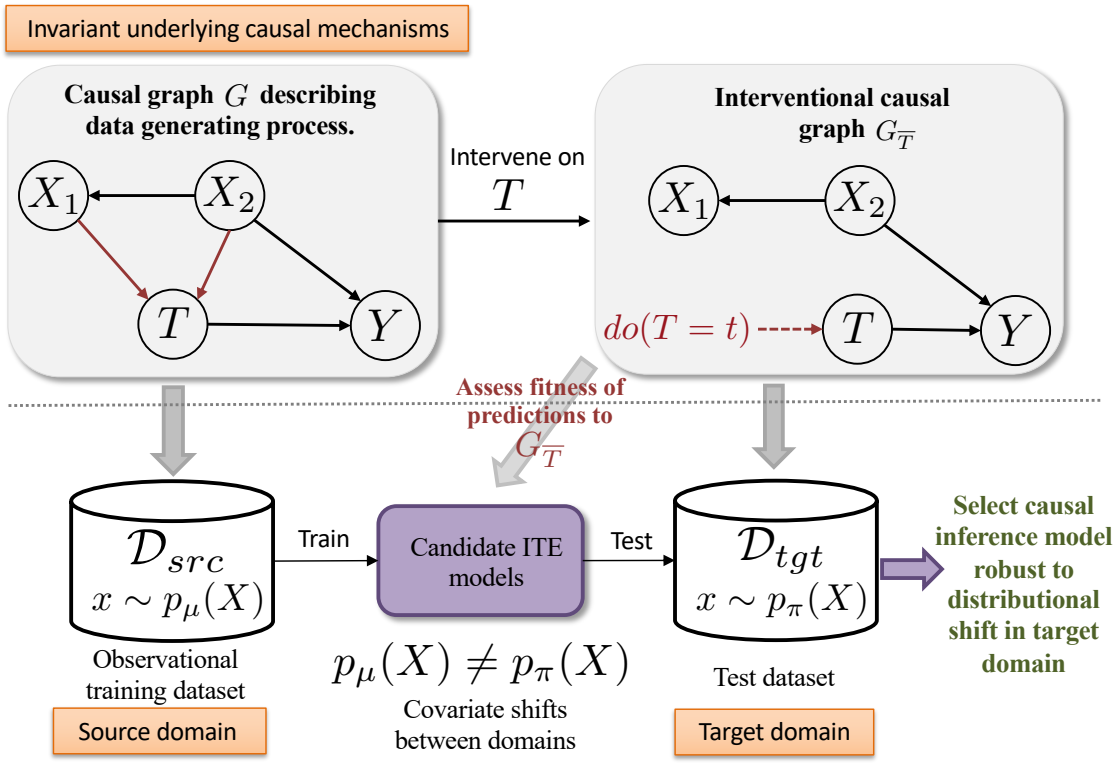


Figure 6.1: Method overview. We propose selecting ITE model whose predictions of the treatment effects on the target domain satisfy the causal relationships in the interventional causal graph $G_{\overline{T}}$.

To better approximate target risk, we propose to leverage the invariance of causal graphs across domains and select ITE models whose predictions of the treatment effects also satisfy known or discovered causal relationships. It is well-known that causality is a property of the

physical world, and therefore the physical (functional) relationships between variables remain invariant across domains [120, 21, 111, 87]. As shown in Figure 6.1, we assume the existence of an underlying causal graph that describes the generating process of the observational data. We represent the selection bias present in the source observational datasets by arrows between the features $\{X_1, X_2\}$, and treatment T. In the target domain, we only have access to the patient features, and we want to estimate the patient outcome (Y) under different settings of the treatment (intervention). When performing such interventions, the causal structure remains unchanged except for the arrows into the treatment node, which are removed.

6.1.1 Contributions

To the best of our knowledge, we present the first UDA selection method specifically tailored for machine learning models that estimate ITE. Our ITE model selection score uniquely leverages the estimated patient outcomes under different treatment settings on the target domain by incorporating a measurement of how well these outcomes satisfy the causal relationships in the interventional causal graph $G_{\overline{T}}$. This measure, which we refer to as causal risk, is computed using a log-likelihood function quantifying the model predictions' fitness to the underlying causal graph. We provide a theoretical justification for using the causal risk, and we show that our proposed ITE model selection metric for UDA prefers models whose predictions satisfy the conditional independence relationships in $G_{\overline{T}}$ and are thus more robust to changes in the distribution of the patient features. We also show experimentally that adding the causal risk to existing state-of-the-art model selection scores for UDA results in selecting ITE models with improved performance on the target domain. We provide an illustrative example of model selection for several real-world datasets for UDA, including ventilator assignment for COVID-19.

6.2 Related Works

6.2.1 ITE models.

Recently, a large number of machine learning methods for estimating heterogeneous ITE from observational data have been developed, leveraging ideas from representation learning [63, 134, 165], adversarial training, [163], causal random forests [155] and Gaussian processes [10, 11]. Nevertheless, no single model will achieve the best performance on all types of observational data [32] and even for the same model, different hyperparameter settings or training iterations will yield different performance.

6.2.2 ITE model selection.

Evaluating ITE models' performance is challenging since counterfactual data is unavailable, and consequently, the true causal effects cannot be computed. Several heuristics for estimating model performance have been used in practice [124, 151]. Factual model selection only computes the error of the ITE model in estimating the factual patient outcomes. Alternatively, inverse propensity weighted (IPTW) selection uses the estimated propensity score to weigh each sample's factual error and thus obtain an unbiased estimate [151]. [10] propose using influence functions to approximate ITE models' error in predicting both factual and counterfactual outcomes. Influence function (IF) based validation currently represents the state-of-the-art method in selecting ITE models. However, existing ITE selection methods are not designed to select models robust to distributional changes in the patient populations, i.e., for domain adaptation.

6.2.3 UDA model selection.

UDA is a special case of domain adaptation, where we have access to unlabeled samples from the test or target domain. Several methods for selecting predictive models for UDA have been proposed [110]. Here we focus on the ones that can be adapted for the ITE setting. The first

unsupervised model selection method was proposed by [79], who used Importance-Weighted Cross-Validation (IWCV) [136] to select hyperparameters and models for covariate shift. IWCV requires that the importance weights (or density ratio) be provided or known ahead of time, which is not always feasible in practice. Later, Deep Embedded Validation (DEV), proposed by [169], was built on IWCV by using a discriminative neural network to learn the target distribution density ratio to provide an unbiased estimation of the target risk with bounded variance. However, these proposed methods do not consider model predictions on the target domain and are agnostic of causal structure.

6.2.4 Causal structure for domain adaptation.

Recently, [74] proposed Causal Assurance (CA) as a domain adaptation selection method for predictive models that leverages prior knowledge in the form of a causal graph. Because their work is centered around predictive models, it is suboptimal for ITE models, where the edges into the treatment (or intervention) will capture the selection bias of the observational data. Furthermore, their method does not allow for examining the target domain predictions, which is a key novelty of this work. We leverage *do*-calculus [104] to manipulate the underlying directed acyclical graph (DAG) into an interventional DAG that more appropriately fits the ITE regime. More recently, researchers have focused on leveraging the causal structure for predictive models by identifying subsets of variables that serve as invariant conditionals [111, 87].

6.3 Preliminaries

6.3.1 Individualized treatment effects and model selection for UDA

Consider a training dataset $\mathcal{D}_{src} = \{(x_i^{src}, t_i^{src}, y_i^{src})\}_{i=1}^{N_{src}}$ consisting of N_{src} independent realizations, one for each individual i, of the random variables (X, T, Y) drawn from the source joint distribution $p_\mu(X, T, Y)$. Let $p_\mu(X)$ be the marginal distribution of X. Assume that we also have access to a test dataset $\mathcal{D}_{tgt} = \{x_i^{tgt}\}_{i=1}^{N_{tgt}}$ from the target domain, consisting

of N_{tgt} independent realizations of X drawn from the target distribution $p_\pi(X)$, where $p_\mu(X) \neq p_\pi(X)$. Let the random variable $X \in \mathcal{X}$ represent the context (e.g. patient features) and let $T \in \mathcal{T}$ describe the intervention (treatment) assigned to the patient. Without loss of generality, consider the case when the treatment is binary, such that $\mathcal{T} = \{0, 1\}$. However, note that our model selection method is also applicable for any number of treatments. We use the potential outcomes framework [119] to describe the result of performing an intervention $t \in \mathcal{T}$ as the potential outcome $Y(t) \in \mathcal{Y}$. Let $Y(1)$ represent the potential outcome under treatment and $Y(0)$ the potential outcome under control. Note that for each individual, we can only observe one of potential outcomes $Y(0)$ or $Y(1)$. We assume that the potential outcomes have a stationary distribution $p_\mu(Y(t) \mid X) = p_\pi(Y(t) \mid X)$ given the context X; this represents the *covariate shift* assumption in domain adaptation [130].

Observational data can be used to estimate $\mathbb{E}[Y \mid X = x, T = t]$ through regression. Assumption 1 describes the causal identification conditions [116], such that the potential outcomes are the same as the conditional expectation: $\mathbb{E}[Y(t) \mid X = x] = \mathbb{E}[Y \mid X = x, T = t]$.

Assumption 8 (Consistency, Ignorability and Overlap)**.** *For any individual (unit) i, receiving treatment t_i, we observe $Y_i = Y(t_i)$. Moreover, $\{Y(0), Y(1)\}$ and the data generating process $p(X, T, Y)$ satisfy strong ignorability $Y(0), Y(1) \perp\!\!\!\perp T \mid X$ and overlap $\forall x : P(T \mid X = x) > 0$.*

The ignorability assumption, also known as the no hidden confounders (unconfoundness) assumptions, means that we observe all variables X that causally affect the assignment of the intervention and the outcome. Under unconfoundness, X blocks all backdoor paths between Y and A [104].

Under Assumption 1, the conditional expectation of the potential outcomes can also be written as the interventional distribution obtained by applying the $do-$operator under the causal framework of [104]: $\mathbb{E}[Y(t) \mid X = x] = \mathbb{E}[Y \mid X = x, do(T = t)]$. This equivalence will enable us to reason about causal graphs and interventions on causal graphs in the context of selecting ITE methods for estimating potential outcomes.

6.3.1.1 Evaluating ITE models.

Methods for estimating ITE learn predictors $f : \mathcal{X} \times \mathcal{T} \to \mathcal{Y}$ such that $f(x,t)$ approximates $\mathbb{E}[Y \mid X = x, T = t] = \mathbb{E}[Y(t) \mid X = x] = \mathbb{E}[Y \mid X = x, do(T = t)]$. The goal is to estimate the ITE, also known as the conditional average treatment effect (CATE):

$$\tau(x) = \mathbb{E}[Y(1) \mid X = x] - \mathbb{E}[Y(0) \mid X = x] \tag{6.1}$$

$$= \mathbb{E}[Y \mid X = x, do(T = 1)] - \mathbb{E}[Y \mid X = x, do(T = 0)]. \tag{6.2}$$

The CATE is essential for individualized decision making as it guides treatment assignment policies. A trained ITE predictor $f(x,t)$ approximates CATE as: $\hat{\tau}(x) = f(x,1) - f(x,0)$. Commonly used to assess ITE models is the precision of estimating heterogeneous effects (PEHE) [53]:

$$PEHE = \mathbb{E}_{x \sim p(x)}[(\tau(x) - \hat{\tau}(x))^2], \tag{6.3}$$

which quantifies a model's estimate of the heterogeneous treatment effects for patients in a population.

6.3.1.2 UDA model selection.

Given a set $\mathcal{F} = \{f_1, \dots f_m\}$ of candidate ITE models trained on the source domain $\mathcal{D}_{src}$, our aim is to select the model that achieves the lowest target risk, that is the lowest PEHE on the target domain $\mathcal{D}_{tgt}$. Thus, ITE model selection for UDA involves finding:

$$\hat{f} = \arg\min_{f \in F} \mathbb{E}_{x \sim p_\pi(x)}[(\tau(x) - \hat{\tau}(x))^2] = \arg\min_{f \in F} \mathbb{E}_{x \sim p_\pi(x)}[(\tau(x) - (f(x,1) - f(x,0)))^2]. \tag{6.4}$$

For this purpose, we propose using the invariance of causal graphs across domains to select ITE predictors that are robust to distributional shifts in the marginal distribution of X.

6.3.2 Causal graphs framework

In this work, we use the semantic framework of causal graphs [104] to reason about causality in the context of model selection. We assume that the unknown data generating process in

the source domain can be described by the causal directed acyclic graph (DAG) G, which contains the relationships between the variables $V = (X, T, Y)$ consisting of the patient features X, treatment T, and outcome Y. We operate under the Markov and faithfulness conditions [113, 104], meaning that any conditional independencies in the joint distribution of $p_\mu(X, T, Y)$ are indicated by d-separation in G and vice-versa.

In this framework, an intervention on the treatment variable $T \in V$ is denoted through the do-operation $do(T = t)$ and induces the interventional DAG $G_{\overline{T}}$, where the edges into T are removed. The interventional DAG $G_{\overline{T}}$ corresponds to the interventional distribution $p_\mu(X, Y \mid do(T = t))$ [104]. The only node on which we perform interventions in the target domain is the treatment node. Consequently, this node will have the edges into it removed, while the remainder of the DAG is unchanged. We assume that the causal graph is invariant across domains [120, 39, 87] which we formalize for interventions as follows:

Assumption 9 (Causal invariance)**.** *Let $V = (X, T, Y)$ be a set of variables consisting of patient features X, treatment T, and outcome Y. Let Δ be a set of domains, $p_\delta(X, Y \mid do(T = t))$ be the corresponding interventional distribution on V in domain $\delta \in \Delta$, and $I(p_\delta(V))$ denote the set of all conditional independence relationships embodied in $p_\delta(V)$, then*

$$\forall \delta_i, \delta_j \in \Delta, I(p_{\delta_i}(X, Y \mid do(T = t))) = I(p_{\delta_j}(X, Y \mid do(T = t))). \tag{6.5}$$

6.4 ITE Model Selection for UDA

Let $\mathcal{F} = \{f_1, f_2, \dots f_m\}$ be a set of candidate ITE models trained on the data from the source domain $\mathcal{D}_{src}$. Our aim is to select the model $f \in \mathcal{F}$ that achieves the lowest PEHE on the target domain $\mathcal{D}_{tgt}$, as described in Equation 6.4. Let G be a causal graph, either known or discovered, that describes the causal relationships between the variables in X, the treatment T and the outcome Y. Let $G_{\overline{T}}$ be the interventional causal graph of G that has edges removed into the treatment variable T.

6.4.1 Prior causal knowledge and graph discovery.

The invariant graph G can be arrived at in two primary ways. The first would be through experimental means, such as randomized trials, which does not scale to a large number of covariates due to financial or ethical impediments. The second would be through the causal discovery of DAG structure from observational data (for a listing of current algorithms we refer to [52]), which is more feasible in practice. Under the assumption of no hidden confounding variables, score-based causal discovery algorithms output a completed partially directed acyclical graph (CPDAG) representing the Markov equivalence class (MEC) of graphs, i.e., those graphs which are statistically indistinguishable given the observational data and therefore share the same conditional independencies. Provided a CPDAG, it is up to an expert (or further experiments) to orient any undirected edges of the CPDAG to convert it into the DAG [104]. This step is the most error-prone, and we show in our real data experiments how a subgraph (using only the known edges) can still improve model selection performance.

6.4.2 Improving target risk estimation.

For the trained ITE model f, let $\hat{y}(0) = f(x, 0)$ and let $\hat{y}(1) = f(x, 1)$ be the predicted potential outcomes for $x \sim p_\pi(x)$. We develop a selection method that prefers models whose predictions on the target domain preserve the conditional independence relationships between X, T and Y in the interventional DAG $G_{\overline{T}}$ with edges removed into the treatment variable T. We first propose a Theorem, which we later exploit for model selection.

Theorem 7. *Let $p_\mu(X, T, Y)$ be a source distribution with corresponding DAG G. If $Y = f(X, T)$, i.e., f is an optimal ITE model, then*

$$I_G(G_{\overline{T}}) = I(p_\pi(X, f(X, t) \mid do(T = t))), \tag{6.6}$$

where $p_\pi(X, f(X, t) \mid do(T = t))$ is the interventional distribution for the target domain and $I_G(G_{\overline{T}})$ and $I(p_\pi(X, f(X, t) \mid do(T = t)))$ returns all the conditional independence relationships in $G_{\overline{T}}$ and $p_\pi(X, f(X, t) \mid do(T = t))$, respectively.

Proof. In the source domain, by the Markov and faithfullness assumptions the conditional independencies in G are the same in $p_\mu(X, T, Y)$, such that

$$I_G(G) = I(p_\mu(X, T, Y)). \tag{6.7}$$

To estimate the potential outcomes $Y(t)$, we apply the *do*-operator to obtain the interventional DAG $G_{\overline{T}}$ and interventional distribution $p_\mu(X, Y \mid do(T = t))$, such that:

$$I_G(G_{\overline{T}}) = I(p_\mu(X, Y \mid do(T = t))). \tag{6.8}$$

Since we assume $Y = f(X, T)$ we obtain:

$$I_G(G_{\overline{T}}) = I(p_\mu(X, f(X, t) \mid do(T = t))). \tag{6.9}$$

By Assumption 7, we know that the conditional independence relationships in the interventional distribution are the same in any environment, so that

$$I(p_\mu(X, f(X, t) \mid do(T = t))) = I(p_\pi(X, f(X, t) \mid do(T = t))), \tag{6.10}$$

such that we obtain:

$$I_G(G_{\overline{T}}) = I(p_\pi(X, f(X, t) \mid do(T = t))). \tag{6.11}$$

□

Theorem 7 provides an equality relating the predictions of f in the target domain to the interventional DAG $G_{\overline{T}}$. Therefore we desire the set of independence relationships in $G_{\overline{T}}$ to equal $I(p_\pi(X, f(X, t) \mid do(T = t)))$. In our case, we do not have access to the true interventional distribution $p_\pi(X, f(X, t) \mid do(T = t))$, but we can approximate it from the dataset obtained by augmenting the unlabeled target dataset $\mathcal{D}_{tgt}$ with the model's predictions of the potential outcomes: $\hat{\mathcal{D}}_{tgt} = \{(x_i^{tgt}, 0, \hat{y}_i^{tgt}(0)), (x_i^{tgt}, 1, \hat{y}_i^{tgt}(1))\}_{i=1}^{N_{tgt}}$, where $\hat{y}_i^{tgt}(t) = f(x_i^{tgt}, t)$, for $x_i^{tgt} \in \mathcal{D}_{tgt}$. We propose to improve the formalization in Eq. 6.4 by adding a constraint on preserving the conditional independencies of $G_{\overline{T}}$ as follows:

$$\arg\min_{f \in F} \mathcal{R}_T(f) \text{ s.t. } \mathbb{E}[NCI(G_{\overline{T}}, \hat{\mathcal{D}}_{tgt})] = 0, \tag{6.12}$$

where $\mathcal{R}_T(f)$ is a function that approximates the target risk for a model f, $NCI(G_{\overline{T}}, \hat{\mathcal{D}}_{tgt})$ is the number of conditional independence relationships in the graph $G_{\overline{T}}$ that are not satisfied by the test dataset augmented with the model's predictions of the potential outcomes $\hat{\mathcal{D}}_{tgt}$.

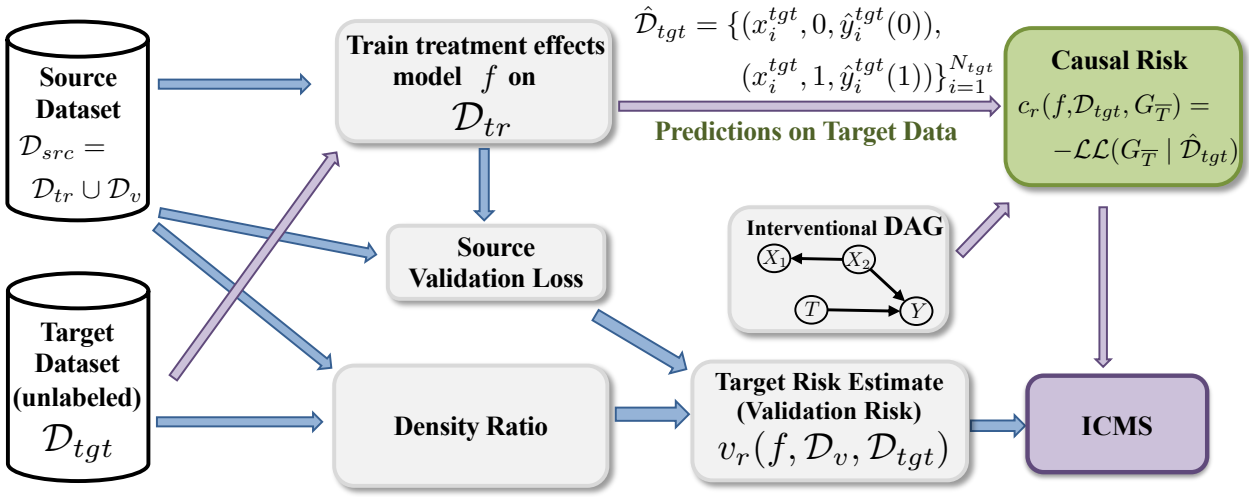


Figure 6.2: ICMS is unique in that it calculates a causal risk (green) using predictions on target data. Purple arrows denote pathways unique to ICMS.

6.4.3 Interventional causal model selection.

Consider the schematic in Figure 6.2. We propose an interventional causal model selection (ICMS) score that takes into account the model's risk on the source domain, but also the fitness to the interventional causal graph $G_{\overline{T}}$ on the target domain according to Eq. 6.4. A score that satisfies this is provided by the Lagrangian method:

$$\mathcal{L} = \mathcal{R}_T(f) + \lambda \mathbb{E}[NCI(G_{\overline{T}}, \hat{\mathcal{D}}_{tgt})]. \tag{6.13}$$

The first term $\mathcal{R}_T(f)$ is equivalent to the expected test PEHE which at selection time can be approximated by the validation risk (either source or target risk), which we represent as $v_r(f, \mathcal{D}_v, \mathcal{D}_{tgt})$. In some cases the second term $\mathbb{E}[NCI(G_{\overline{T}}, \hat{\mathcal{D}}_{tgt})]$ may never equal 0, because of this we approximate it by using a causal fitness score that measures the likelihood of a DAG given some data on the test dataset, which we rewrite as $c_r(f, \mathcal{D}_{tgt}, G_{\overline{T}})$. Consider partitioning the source dataset $\mathcal{D}_{src} = \{(x_i^{src}, t_i^{src}, y_i^{src})\}_{i=1}^{N_{src}}$ into a training dataset $\mathcal{D}_{tr}$ and a

validation dataset $\mathcal{D}_v$ such that $\mathcal{D}_{src} = \mathcal{D}_{tr} \cup \mathcal{D}_v$. From Eq. 6.13 we define our ICMS score r as follows:

Definition 8 (ICMS score)**.** *Let f be an ITE predictor trained on $\mathcal{D}_{tr}$. Let $\mathcal{D}_{tgt} = \{(x_i^{tgt})\}_{i=1}^{N_{tgt}}$ be test dataset and let $G_{\overline{T}}$ be the interventional causal graph. We define the following selection score:*

$$r(f, \mathcal{D}_v, \mathcal{D}_{tgt}, G_{\overline{T}}) = v_r(f, \mathcal{D}_v, \mathcal{D}_{tgt}) + \lambda c_r(f, \mathcal{D}_{tgt}, G_{\overline{T}}) \tag{6.14}$$

where v_r measures the validation risk on the validation set $\mathcal{D}_v$ and c_r is a scoring function, which we call causal risk, that measures the fitness of the interventional causal graph $G_{\overline{T}}$ to the dataset $\hat{\mathcal{D}}_{tgt} = \{(x_i^{tgt}, 0, \hat{y}_i^{tgt}(0)), (x_i^{tgt}, 1, \hat{y}_i^{tgt}(1))\}_{i=1}^{N_{tgt}}$, where $\hat{y}_i^{tgt}(t) = f(x_i^{tgt}, t)$, for $x_i^{tgt} \in \mathcal{D}_{tgt}$.

The validation risk $v_r(f, \mathcal{D}_v, \mathcal{D}_{tgt})$ can either be (1) source risk where we use existing model selection scores for ITE [12, 151], or (2) an approximation of target risk using the preexisting methods of IWCV or DEV [136, 169]. We describe in the following section how to compute the causal risk $c_r(f, \mathcal{D}_{tgt}, G_{\overline{T}})$. λ is a tuning factor between our causal risk term and validation risk v_r. We currently set $\lambda = 1$ for our experiments, but ideally, λ would be proportional to our certainty in our causal graph.

6.4.4 Assessing causal graph fitness.

The causal risk term $c_r(f, \mathcal{D}_{tgt}, G_{\overline{T}})$ as part of our ICMS score requires assessing the fitness of the dataset $\hat{\mathcal{D}}_{tgt}$ to the invariant causal knowledge in $G_{\overline{T}}$. Some options include noteworthy maximum-likelihood algorithms such as the Akaike Information Criterion (AIC) [6] and Bayesian Information Criterion (BIC) [125]. Both the BIC and AIC are penalized versions of the log-likelihood function of a DAG given data, e.g., $\mathcal{LL}(G_{\overline{T}} \mid \hat{\mathcal{D}}_{tgt})$. In score based causal discovery, the DAG that best fits the data will maximize the $\mathcal{LL}(G_{\overline{T}} \mid \hat{\mathcal{D}}_{tgt})$ subject to some model complexity penalty constraints. In this work, we are not searching between candidate causal graphs and only care about maximizing our DAG to dataset fitness. Thus, we use the negative log-likelihood of G given $\hat{\mathcal{D}}_{tgt}$, i.e. $-\mathcal{LL}(G_{\overline{T}} \mid \hat{\mathcal{D}}_{tgt})$, for our *causal risk* term c_r. The $-\mathcal{LL}(G_{\overline{T}} \mid \hat{\mathcal{D}}_{tgt})$ has a smaller value when G is closer to modeling the probability

distribution in $\hat{\mathcal{D}}_{tgt}$, i.e the predicted potential outcomes satisfy the conditional independence relationships in G.

6.4.5 Pseudocode for ICMS

To clarify our methodology further we have provided pseudocode in Algorithms 5 and 6. Algorithm 5 calculates the ICMS score (from Eq. 6.14) from a given model. The values for c_r and v_r are min-max normalized between 0 and 1 across all models. Algorithm 6 returns a ranked list of models by ICMS score from a set of ITE models $\mathcal{F}$. It takes optional prior knowledge in the form of a causal graph or known connections.

Algorithm 5 Calculate ICMS

Input: ITE model f; source validation dataset $\mathcal{D}_v$; unlabeled target test set $\mathcal{D}_{tgt} = \{x_i^{tgt}\}_{i=1}^{N_{tgt}}$; interventional DAG $G_{\overline{T}}$; scale factor λ.

Output: ICMS score: $r(f, \mathcal{D}_v, \mathcal{D}_{tgt}, G_{\overline{T}})$

Function: $\texttt{ICMS}(f, \mathcal{D}_v, \mathcal{D}_{tgt}, G_{\overline{T}}, \lambda)$**:**

$\hat{y}_i^{tgt}(t) \leftarrow f(x_i^{tgt}, t)$, for $x_i^{tgt} \in \mathcal{D}_{tgt}$

$\hat{\mathcal{D}}_{tgt} \leftarrow \{(x_i^{tgt}, 0, \hat{y}_i^{tgt}(0)), (x_i^{tgt}, 1, \hat{y}_i^{tgt}(1))\}_{i=1}^{N_{tgt}}$

$c_r \leftarrow$ Measure of $\hat{\mathcal{D}}_{tgt}$ to DAG $G_{\overline{T}}$ fitness.

$v_r \leftarrow$ Validation risk of f on $\mathcal{D}_v$ and $\mathcal{D}_{tgt}$.

return $c_r + \lambda v_r$ (from Eq. 6.14).

6.4.6 Limitations of UDA selection methods for predictive models

In the ideal scenario, we would be able to leverage labeled samples in the target domain to estimate the target risk of a machine learning model. We can express the target risk $\mathcal{R}_{tgt}$ in terms of the testing loss as follows:

$$\mathcal{R}_{tgt} = \frac{1}{N_{tgt}} \sum ((Y^{tgt}(1) - Y^{tgt}(0)) - (f(x^{tgt}, 1) - f(x^{tgt}, 0))^2 \tag{6.15}$$

However, in general, we do not have access to the treatment responses for patients in the target set and, even if we did, we can only observe the factual outcome. Moreover, existing

Algorithm 6 ICMS Selection

Input: Source dataset $\mathcal{D}_{src} = \{(x_i^{src}, t_i^{src}, y_i^{src})\}_{i=1}^{N_{src}}$ split into a training set $\mathcal{D}_{tr}$ and validation set $\mathcal{D}_v$; set of ITE models $\mathcal{F}$ trained $\mathcal{D}_{tr}$; unlabeled test set $\mathcal{D}_{tgt}$; optional prior knowledge in the form of a DAG G_π, scale factor λ.

Output: A list $\mathcal{F}'$ of models in $\mathcal{F}$ ranked by ICMS score.

Function: `ICMS_sel`$(\mathcal{F}, \mathcal{D}_{tr}, \mathcal{D}_v, \mathcal{D}_{tgt}, \lambda, G_\pi = \emptyset)$**:**

$G_d \leftarrow$ causal discovery on $\mathcal{D}_{tr}$

$G \leftarrow$ assumed invariant DAG from G_π or G_d

$G_{\overline{T}} \leftarrow$ interventional DAG of G (remove edges into T)

$\mathcal{F}' \leftarrow$ Sort $\mathcal{F}$ by `ICMS`$(f, \mathcal{D}_v, \mathcal{D}_{tgt}, G_{\overline{T}}, \lambda)$ ascending

return $\mathcal{F}'$.

model selection methods for UDA for predictive models only consider predictions on the source domain and do not take into account the predictions of the candidate model in the target domain. Specifically, DEV and IWCV calculate a density ratio or importance weight between the source and target domain as follows:

$$w_f(x) = \frac{p(d=1|x)}{p(d=0|x)} \frac{N^{tgt}}{N^{src}}, \tag{6.16}$$

where d designates dataset domain (source is 0, target is 1), and $\frac{p(d=1|x)}{p(d=0|x)}$ can be estimated by a discriminative model to distinguish source from target samples [169]. Both calculate their score as a function of Δ as follows:

$$\Delta = \frac{1}{N_v} \sum_{i=1}^{N_v} w_f(x_i^v) l(y_i^v, f(x_i^v, 0), f(x_i^v, 1)) \tag{6.17}$$

where $l(\cdot, \cdot, \cdot)$ is a validation loss, such as influence-function based validation [12]. Note that the functions l and w are only defined in terms of validation features x_i^v from the source dataset. Such selection scores can be used to compute the validation score $v_r(f, \mathcal{D}_v, \mathcal{D}_{tgt})$ part of the ICMS score.

However, our ICMS score also computes the likelihood of the interventional causal graph given the predictions of the model in the target domain as a proxy for the risk in the target

domain. By adding the causal risk, we the improve the estimation of target risk. Additionally, we specifically make use of the estimated potential outcomes on the test set $f(x^{tgt}, 0)$ and $f(x^{tgt}, 1)$ to calculate our selection score as shown in Eq. 6.14. Fig. 6.2 depicts how we use the predictions of the target data to calculate our ICMS score.

6.5 Experiments

We evaluate methods by the test performance in terms of PEHE of the top 10% of models in the list return by the model selection benchmarks. We will refer to this as the PEHE-10 test error.

6.5.1 Benchmark ITE models.

We show how the ICMS score improves model selection for state-of-the-art ITE methods based on neural networks: GANITE [163], CFRNet [61], TARNet [61], SITE [165] and Gaussian processes: CMGP [10] and NSGP [11]. These ITE methods use different techniques for estimating ITE and currently achieve the best performance on standard benchmark observational datasets [12]. We iterate over each model multiple times and compare against various DAGs and held-out test sets. Having various DAG structures results in varying magnitudes of test error. Therefore, without changing the ranking of the models, we min-max normalize our test error between 0 and 1 for each DAG, such that equal weight is given to each experimental run, and a relative comparison across benchmark ITE models can be made.

6.5.2 Benchmark methods.

We benchmark our proposed ITE model selection score ICMS against each of the following UDA selection methods developed for predictive models: IWCV [79] and DEV [169]. To approximate the source risk, i.e., the error of ITE methods in predicting potential outcomes on the source domain (validation set $\mathcal{D}_v$), we use the following standard ITE scores: MSE on the factual outcomes, inverse propensity weighted factual error (IPTW) [151] and influence

functions (IF) [12]. Note that each score (MSE, IPTW, etc.) can be used to estimate the target risk in the UDA selection methods: IWCV, DEV, or ICMS. Specifically, we benchmark our method in conjunction with each combination of ITE model errors {MSE, IPTW, IF} with validation risk {$\emptyset$, IWCV, DEV}. We include experiments with $\emptyset$, to demonstrate using source risk as an estimation of validation risk.

6.5.3 Synthetic UDA model selection

6.5.3.1 Data generation.

In this section, we evaluate our method in comparison to related selection methods on synthetic data. For each of the simulations, we generated a random DAG, G, with n vertices and up to $n(n-1)/2$ edges (the asymptotic maximum number of edges in a DAG) between them. We construct our datasets with functional relationships between variables with directed edges between them in G and applied Gaussian noise (0 mean and 1 variance) to each. Using the structure of G, we synthesized 2000 samples for our observational source dataset $\mathcal{D}_{src}$. We randomly split $\mathcal{D}_{src}$ into a training set $\mathcal{D}_{tr}$ and validation set $\mathcal{D}_v$ with 80% and 20% of the samples, respectively. To generate the testing dataset $\mathcal{D}_{tgt}$, we use G to generate 1000 samples where half of the dataset receives treatment, and the other half does not. For $\mathcal{D}_{tgt}$, we randomly shift the mean between 1 and 10 of an ancestor of Y in G, whereas in $\mathcal{D}_{src}$ a mean of 0 is used. It is important to note that the actual outcome or response is never seen when selecting our models. Furthermore, the training dataset $\mathcal{D}_{src}$ is observational and contains selection bias into the treatment node, whereas the synthetic test set $\mathcal{D}_{tgt}$ does not, since it was generated by intervention at the treatment node. Our algorithm has only access to the covariates X in $\mathcal{D}_{tgt}$.

6.5.3.2 Improved selection for all ITE models.

Table 6.1 shows results of ICMS on synthetic data over the benchmark ITE models. Here, we evaluate three different types of selection baseline methods: MSE, IPTW, and IF. We

then compare each baseline selection method with UDA methods: IWCV, DEV, and ICMS (proposed). We repeated the experiment over 50 different DAGs with 30 candidate models for each model architecture. Each of the candidate algorithms was trained using their published settings and hyperparameters. In Table 6.1, we see that our proposed method (ICMS) improves on each baseline selection method by having a lower testing error in terms of PEHE-10 over all treatment models.

6.5.4 Application to the COVID-19 Response

ICMS facilitates and improves model transfer across domains with disparate distributions, i.e., time, geographical location, etc., which we will demonstrate in this section for COVID-19. The COVID-19 pandemic challenged healthcare systems worldwide. At the peak of the outbreak, many countries experienced a shortage of life-saving equipment, such as ventilators and ICU beds. Considering data from the UK outbreak, the pandemic hit the urban population before spreading to the rural areas (Figure 5.5). This implies that if we reacted in a timely manner, we could transfer models trained on the urban population to the rural population. However, there is a significant domain shift as the rural population is older and has more preexisting conditions. Furthermore, at the time of model deployment in rural areas, there may be no labeled samples available.

6.5.4.1 Dataset

We obtained de-identified COVID-19 Hospitalization in England Surveillance System (CHESS) data from Public Health England (PHE) for the period from 8$^{\text{th}}$ February (data collection start) to 14$^{\text{th}}$ April 2020, which contains 7,714 hospital admission, including 3,092 ICU admissions from 94 NHS trusts across England. The data set features comprehensive information on patients' general health condition, COVID-19 specific risk factors (e.g., comorbidities), basic demographic information (age, sex, etc.), and tracks the entire patient treatment journey: hospitalization time, ICU admission, what treatment (e.g., ventilation) they received, and their outcome by April 20th, 2020 (609 deaths and 384 discharges). We split the data set

into a source dataset containing 2,552 patients from urban areas (mostly Greater London area) and a target dataset of the remaining 5,162 rural patients. The characteristics of the two populations are summarized in Figure 5.5.

6.5.4.2 COVID-19 Ventilation UK (urban) → UK (rural)

Using the urban dataset, we performed causal discovery on the relationships between the patient covariates, treatment, and outcome. The discovered graph (Figure 5.5) agree well with the literature [156, 99]. To be able to evaluate the ITE methods on how well they estimate all counterfactual outcomes, we created a semi-synthetic version of the dataset with outcomes simulated according to the causal graph. Our training observational dataset consists of the patient features, ventilator assignment (treatment) for the COVID-19 patients in the urban area, and the synthetic outcome generated based on the causal graph.

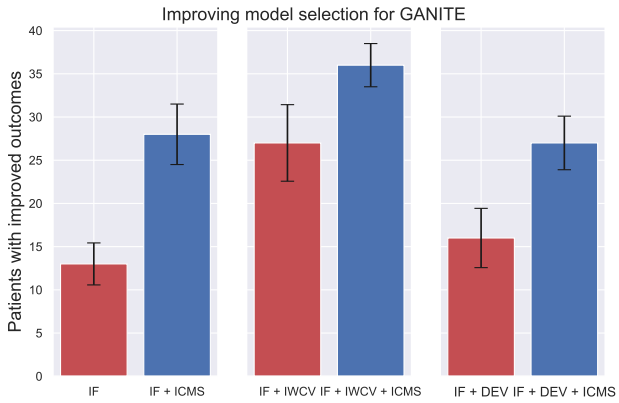


Figure 6.3: Performance of model selection methods in terms of the additional number of patients with improved outcomes compared to selecting models based on the factual error on the source domain.

For each benchmark ITE model, we used 30 different hyperparameter settings and trained the various models to estimate the effect of ventilator use on the patient risk of mortality. We used the same training regime as in the synthetic experiments and the discovered COVID-19 causal DAG (using FGES) shown in Figure 5.5. We evaluated the best ITE model selected by each model selection method in a ventilator assignment task. Using each selected ITE

model, we assigned 2000 ventilators to the rural area patients that would have the highest estimated benefit (individualized treatment effect) from receiving the ventilator. Using the known synthetic outcomes for each patient, we then computed how many patients would have improved outcomes using each selected ITE model for assigning ventilators. By considering selection based on the factual outcome (MSE) on the source dataset as a baseline, in Figure 6.3, we computed the additional number of patients with improved outcomes by using ICMS on top of existing UDA methods when selecting GANITE models with different settings of the hyperparameters. We see that ICMS (in blue) identified the GANITE models that resulted in better patient outcomes in the UK's rural areas without access to labeled data.

6.5.4.3 Additional experiments:

On the TWINS dataset [5], we show how our method improves UDA model selection even with partial knowledge of the causal graph (i.e., using only a known subgraph for computing the ICMS score). Note also that in the Twins dataset, we have access to real patient outcomes.

6.6 Conclusion

We provide a novel ITE model selection method for UDA that uniquely leverages the predictions of candidate models on a target domain by preserving invariant causal relationships. To the best of our knowledge, we have provided the first model selection method for ITE models specifically for UDA. We provide a theoretical justification for using ICMS and have shown on a variety of synthetic, semi-synthetic, and real data that our method can improve on existing state-of-the-art UDA methods.

Table 6.1: PEHE-10 performance (with standard error) using ICMS on top of existing UDA methods. ■ + ICMS means that the ■ was used in conjunction with ICMS. For example, DEV($\star$)+ICMS represents DEV($\star$) selection used as the validation risk v_r in the ICMS. The $\star$ indicates the method used to approximate the validation error on the source dataset. Our method (in bold) improves over each selection method over all models and source risk scores (Src.).

Selection Method	GANITE	CFR	TAR	SITE	CMGP	NSGP
MSE	0.395 (0.051)	0.363 (0.042)	0.391 (0.050)	0.157 (0.035)	0.131 (0.046)	0.282 (0.049)
MSE+ICMS	**0.222 (0.049)**	**0.212 (0.036)**	**0.264 (0.034)**	**0.126 (0.027)**	**0.120 (0.050)**	**0.210 (0.047)**
IWCV(MSE)	0.348 (0.046)	0.393 (0.044)	0.364 (0.052)	0.185 (0.033)	0.201 (0.041)	0.209 (0.040)
IWCV(MSE)+ICMS	**0.212 (0.043)**	**0.220 (0.051)**	**0.256 (0.039)**	**0.149 (0.033)**	**0.183 (0.055)**	**0.172 (0.043)**
DEV(MSE)	0.398 (0.056)	0.414 (0.042)	0.427 (0.049)	0.198 (0.038)	0.239 (0.058)	0.183 (0.048)
DEV(MSE)+ICMS	**0.224 (0.042)**	**0.210 (0.039)**	**0.269 (0.035)**	**0.120 (0.040)**	**0.160 (0.047)**	**0.160 (0.042)**
IPTW	0.381 (0.049)	0.355 (0.046)	0.394 (0.052)	0.357 (0.045)	0.182 (0.046)	0.292 (0.045)
IPTW+ICMS	**0.220 (0.049)**	**0.217 (0.039)**	**0.272 (0.032)**	**0.228 (0.031)**	**0.140 (0.050)**	**0.207 (0.047)**
IWCV(IPTW)	0.269 (0.055)	0.518 (0.049)	0.433 (0.038)	0.416 (0.053)	0.417 (0.043)	0.475 (0.053)
IWCV(IPTW)+ICMS	**0.053 (0.028)**	**0.121 (0.034)**	**0.119 (0.035)**	**0.207 (0.039)**	**0.304 (0.059)**	**0.328 (0.058)**
DEV(IPTW)	0.302 (0.072)	0.472 (0.056)	0.414 (0.049)	0.400 (0.057)	0.441 (0.071)	0.493 (0.086)
DEV(IPTW)+ICMS	**0.087 (0.035)**	**0.194 (0.052)**	**0.120 (0.027)**	**0.220 (0.031)**	**0.282 (0.041)**	**0.355 (0.050)**
IF	0.222 (0.041)	0.255 (0.050)	0.250 (0.046)	0.321 (0.059)	0.392 (0.051)	0.376 (0.057)
IF+ICMS	**0.127 (0.039)**	**0.166 (0.042)**	**0.190 (0.044)**	**0.215 (0.056)**	**0.212 (0.053)**	**0.250 (0.054)**
IWCV(IF)	0.180 (0.059)	0.364 (0.051)	0.286 (0.041)	0.293 (0.043)	0.415 (0.048)	0.437 (0.057)
IWCV(IF)+ICMS	**0.058 (0.018)**	**0.104 (0.025)**	**0.108 (0.033)**	**0.173 (0.028)**	**0.292 (0.062)**	**0.331 (0.051)**
DEV(IF)	0.193 (0.058)	0.415 (0.045)	0.292 (0.046)	0.214 (0.038)	0.490 (0.043)	0.544 (0.053)
DEV(IF)+ICMS	**0.069 (0.026)**	**0.191 (0.048)**	**0.107 (0.029)**	**0.147 (0.025)**	**0.229 (0.054)**	**0.364 (0.056)**

BIBLIOGRAPHY

[1] Bryon Aragam, Arash A Amini, and Qing Zhou. Learning directed acyclic graphs with penalized neighbourhood regression. *arXiv preprint arXiv:1511.08963*, 2015.

[2] Robert B. Avery, Kenneth P. Brevoort, and Glenn Canner. Credit scoring and its effects on the availability and affordability of credit. *Journal of Consumer Affairs*, 43(3):516–537, 2009.

[3] Robert B. Avery, Kenneth P. Brevoort, and Glenn Canner. Does credit scoring produce a disparate impact? *Real Estate Economics*, 40(s1):S65–S114, 2012.

[4] Sanjeev Arora, Aditya Bhaskara, Rong Ge, and Tengyu Ma. Provable bounds for learning some deep representations. In *International conference on machine learning*, pages 584–592. PMLR, 2014.

[5] Douglas Almond, Kenneth Y Chay, and David S Lee. The costs of low birth weight. *The Quarterly Journal of Economics*, 120(3):1031–1083, 2005.

[6] Hirotogu Akaike. *Information Theory and an Extension of the Maximum Likelihood Principle*, pages 199–213. Springer New York, New York, NY, 1998.

[7] A. Allen and W. Li. Generative adversarial denoising autoencoder for face completion,. https://www.cc.gatech.edu/~hays/7476/projects/Avery_Wenchen/, 2016.

[8] Anita M Alessandra. When doctrines collide: Disparate treatment, disparate impact, and watson v. fort worth bank & trust. *U. Pa. L. Rev.*, 137:1755, 1988.

[9] Ahmed M Alaa, Boris van Breugel, Evgeny Saveliev, and Mihaela van der Schaar. How faithful is your synthetic data? sample-level metrics for evaluating and auditing generative models. *arXiv preprint arXiv:2102.08921*, 2021.

[10] Ahmed M Alaa and Mihaela van der Schaar. Bayesian inference of individualized treatment effects using multi-task gaussian processes. In *Advances in Neural Information Processing Systems*, pages 3424–3432, 2017.

[11] Ahmed Alaa and Mihaela van der Schaar. Limits of estimating heterogeneous treatment effects: Guidelines for practical algorithm design. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 129–138, 2018.

[12] Ahmed Alaa and Mihaela van der Schaar. Validating causal inference models via influence functions. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*,

volume 97 of *Proceedings of Machine Learning Research*, pages 191–201, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[13] Mohammad Taha Bahadori, Krzysztof Chalupka, Edward Choi, Robert Chen, Walter F. Stewart, and Jimeng Sun. Causal regularization. *CoRR*, abs/1702.02604, 2017.

[14] John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman. Learning bounds for domain adaptation. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 129–136. Curran Associates, Inc., 2008.

[15] Melanie L Bell, Mallorie Fiero, Nicholas J Horton, and Chiu-Hsieh Hsu. Handling missing data in rcts; a review of the top medical journals. *BMC medical research methodology*, 14(1):1–8, 2014.

[16] Stef Buuren and Catharina Groothuis-Oudshoorn. Mice: Multivariate imputation by chained equations in r. *Journal of Statistical Software*, 45, 12 2011.

[17] Solon Barocas, Moritz Hardt, and Arvind Narayanan. Fairness in machine learning. *Nips tutorial*, 1:2, 2017.

[18] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. In *Proceedings of the 19th International Conference on Neural Information Processing Systems*, NIPS'06, page 153–160, Cambridge, MA, USA, 2006. MIT Press.

[19] Rohit Bhattacharya, Razieh Nabi, Ilya Shpitser, and James Robins. Identification in missing data models represented by directed acyclic graphs. In *Proc. of Conference on Uncertainty in Artificial Intelligence (UAI)*, 06 2019.

[20] Elias Bareinboim and Judea Pearl. Transportability of causal effects: Completeness results. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI'12, pages 698–704. AAAI Press, 2012.

[21] Elias Bareinboim and Judea Pearl. Causal inference and the data-fusion problem. *Proceedings of the National Academy of Sciences*, 113(27):7345–7352, 2016.

[22] Solon Barocas and Andrew D Selbst. Big data's disparate impact. *Calif. L. Rev.*, 104:671, 2016.

[23] Yoshua Bengio, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'13, page 899–907, Red Hook, NY, USA, 2013. Curran Associates Inc.

[24] Krzysztof Chalupka, Frederick Eberhardt, and Pietro Perona. Estimating causal direction and confounding of two discrete variables. *arXiv preprint arXiv:1611.01504*, 2016.

[25] Kristy Choi, Aditya Grover, Trisha Singh, Rui Shu, and Stefano Ermon. Fair generative modeling via weak supervision. In *International Conference on Machine Learning*, pages 1887–1898. PMLR, 2020.

[26] Flavio P. Calmon, Dennis Wei, Karthikeyan Natesan Ramamurthy, and Kush R. Varshney. Optimized data pre-processing for discrimination prevention, 2017.

[27] Ruichu Cai, Feng Xie, Clark Glymour, Zhifeng Hao, and Kun Zhang. Triad constraints for learning causal structure of latent variables. In *Advances in Neural Information Processing Systems 32*, pages 12883–12892. Curran Associates, Inc., 2019.

[28] Máire Duggan, William Anderson, Sean Altekruse, Lynne Penberthy, and Mark Sherman. The surveillance, epidemiology, and end results (seer) program and pathology: Toward strengthening the critical relationship. *The American Journal of Surgical Pathology*, 40:1, 10 2016.

[29] Adnan Darwiche. *Modeling and Reasoning with Bayesian Networks*. Cambridge University Press, New York, NY, USA, 1st edition, 2009.

[30] Jeffrey Dastin. Amazon scraps secret AI recruiting tool that showed bias against women. *Reuters*, 2018.

[31] Dheeru Dua and Casey Graff. UCI machine learning repository, 2020.

[32] Vincent Dorie, Jennifer Hill, Uri Shalit, Marc Scott, Dan Cervone, et al. Automated versus do-it-yourself methods for causal inference: Lessons learned from a data analysis competition. *Statistical Science*, 34(1):43–68, 2019.

[33] Will Dobbie, Andres Liberman, Daniel Paravisini, and Vikram Pathania. Measuring bias in consumer lending. Working Paper 24953, National Bureau of Economic Research, August 2018.

[34] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

[35] Daniel de Vassimon Manela, David Errington, Thomas Fisher, Boris van Breugel, and Pasquale Minervini. Stereotype and skew: Quantifying gender bias in pre-trained and fine-tuned language models. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2232–2242, Online, April 2021. Association for Computational Linguistics.

[36] Hadi Fanaee. Bike sharing in washington d.c., 2019. Retrieved from Kaggle, March 2019.

[37] Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 259–268, 2015.

[38] Peter Flach and Meelis Kull. Precision-recall-gain curves: Pr analysis done right. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[39] AmirEmad Ghassami et al. Learning causal structures using regression invariance. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 3011–3021. Curran Associates, Inc., 2017.

[40] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[41] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[42] Pascal Germain, Francis Bach, Alexandre Lacoste, and Simon Lacoste-Julien. Pac-bayesian theory meets bayesian inference. In *Advances in Neural Information Processing Systems*, pages 1884–1892, 2016.

[43] Yu Gong, Hossein Hajimirsadeghi, Jiawei He, Megha Nawhal, Thibaut Durand, and Greg Mori. Variational selective autoencoder. In Cheng Zhang, Francisco Ruiz, Thang Bui, Adji Bousso Dieng, and Dawen Liang, editors, *Proceedings of The 2nd Symposium on Advances in Approximate Bayesian Inference*, volume 118 of *Proceedings of Machine Learning Research*, pages 1–17. PMLR, 08 Dec 2020.

[44] Nina Grgic-Hlaca, Muhammad Bilal Zafar, Krishna P Gummadi, and Adrian Weller. The case for process fairness in learning: Feature selection for fair decision making. In *NIPS Symposium on Machine Learning and the Law*, volume 1, page 2, 2016.

[45] Tian Gao and Qiang Ji. Local causal discovery of direct causes and effects. pages 2512–2520. Curran Associates, Inc., 2015.

[46] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Y. Bengio. Generative adversarial networks. *Advances in Neural Information Processing Systems*, 3, 06 2014.

[47] Alexander Gain and Ilya Shpitser. Structure learning under missing data. *Proceedings of machine learning research*, 72:121, 2018.

[48] Clark Glymour, Richard Scheines, Peter Spirtes, and Joseph Ramsey. Tetrad, 2019.

[49] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17(1):2096–2030, January 2016.

[50] Lovedeep Gondara and Ke Wang. Multiple imputation using deep denoising autoencoders. *ArXiv*, abs/1705.02737, 2017.

[51] Bin-Bin Gao, Chao Xing, Chen-Wei Xie, Jianxin Wu, and Xin Geng. Deep label distribution learning with label ambiguity. *IEEE Transactions on Image Processing*, 26:2825–2838, 04 2017.

[52] Clark Glymour, Kun Zhang, and Peter Spirtes. Review of causal discovery methods based on graphical models. *Frontiers in Genetics*, 10:524, 2019.

[53] Jennifer L Hill. Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, 20(1):217–240, 2011.

[54] Arthur Hoerl and Robert Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12:55–67, 04 2012.

[55] Moritz Hardt, Eric Price, and Nathan Srebro. Equality of opportunity in supervised learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 3323–3331, Red Hook, NY, USA, 2016. Curran Associates Inc.

[56] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *ArXiv*, abs/1207.0580, 2012.

[57] Tianyu He, Xu Tan, Yingce Xia, Di He, Tao Qin, Zhibo Chen, and Tie-Yan Liu. Layer-wise coordination between encoder and decoder for neural machine translation. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 7944–7954. Curran Associates, Inc., 2018.

[58] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 448–456. JMLR.org, 2015.

[59] Dominik Janzing. Causal regularization. In *Advances in Neural Information Processing Systems 32*, pages 12704–12714. Curran Associates, Inc., 2019.

[60] Janus Christian Jakobsen, Christian Gluud, Jørn Wetterslev, and Per Winkel. When and how should multiple imputation be used for handling missing data in randomised clinical trials–a practical guide with flowcharts. *BMC medical research methodology*, 17(1):1–10, 2017.

[61] Fredrik D Johansson, Nathan Kallus, Uri Shalit, and David Sontag. Learning weighted representations for generalization across designs. *arXiv preprint arXiv:1802.08598*, 2018.

[62] Dominik Janzing and Bernhard Schölkopf. Semi-supervised interpolation in an anticausal learning scenario. *J. Mach. Learn. Res.*, 16(1):1923–1948, January 2015.

[63] Fredrik Johansson, Uri Shalit, and David Sontag. Learning representations for counterfactual inference. In *International conference on machine learning*, pages 3020–3029, 2016.

[64] James Jordon, Jinsung Yoon, and M. Schaar. Pate-gan: Generating synthetic data with differential privacy guarantees. In *ICLR*, 2019.

[65] Achuta Kadambi. Achieving fairness in medical devices. *Science*, 372(6537):30–31, 2021.

[66] Faisal Kamiran and Toon Calders. Classifying without discriminating. In *2009 2nd International Conference on Computer, Control and Communication*, pages 1–6. IEEE, 2009.

[67] Royce Kimmons. Student performance, 2019. Retrieved from Kaggle, March 2019.

[68] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[69] Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. Counterfactual fairness. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[70] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

[71] Niki Kilbertus, Mateo Rojas Carulla, Giambattista Parascandolo, Moritz Hardt, Dominik Janzing, and Bernhard Schölkopf. Avoiding discrimination through causal reasoning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[72] Murat Kocaoglu, Christopher Snyder, Alexandros G Dimakis, and Sriram Vishwanath. Causalgan: Learning causal implicit generative models with adversarial training. *arXiv preprint arXiv:1709.02023*, 2017.

[73] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, page 1097–1105, Red Hook, NY, USA, 2012. Curran Associates Inc.

[74] Trent Kyono and Mihaela van der Schaar. Improving model robustness using causal knowledge. *CoRR*, abs/1911.12441, 2019.

[75] Diederik P. Kingma and M. Welling. Auto-encoding variational bayes. *ICLR*, abs/1312.6114, 2014.

[76] Faisal Kamiran, Indrė Žliobaitė, and Toon Calders. Quantifying explainable discrimination and removing illegal discrimination in automated decision making. *Knowledge and Information Systems*, 1:in press, 06 2012.

[77] Po-Ling Loh and Peter Bühlmann. High-dimensional learning of linear causal networks via inverse covariance estimation. *The Journal of Machine Learning Research*, 15(1):3065–3105, 2014.

[78] Sébastien Lachapelle, Philippe Brouillard, Tristan Deleu, and Simon Lacoste-Julien. Gradient-based neural DAG learning. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, 2020.

[79] Mingsheng Long, ZHANGJIE CAO, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 1640–1650. Curran Associates, Inc., 2018.

[80] Kaiji Lu, Piotr Mardziel, Fangjing Wu, Preetam Amancharla, and Anupam Datta. Gender bias in neural natural language processing. *CoRR*, abs/1807.11714, 2018.

[81] P. K. Lohia, K. Natesan Ramamurthy, M. Bhide, D. Saha, K. R. Varshney, and R. Puri. Bias mitigation post-processing for individual and group fairness. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2847–2851, 2019.

[82] Sebastian Lunz, Ozan Öktem, and Carola-Bibiane Schönlieb. Adversarial regularizers in inverse problems. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 8507–8516. Curran Associates, Inc., 2018.

[83] David Lopez-Paz, Robert Nishihara, Soumith Chintala, Bernhard Scholkopf, and Leon Bottou. Discovering causal signals in images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[84] Lei Le, Andrew Patterson, and Martha White. Supervised autoencoders: Improving generalization performance with unsupervised regularizers. In *Advances in Neural Information Processing Systems*, pages 107–117, 2018.

[85] John Langford and John Shawe-Taylor. Pac-bayes & margins. In *Advances in neural information processing systems*, pages 439–446, 2003.

[86] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. Deep transfer learning with joint adaptation networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70 of *Proceedings of Machine Learning Research*, pages 2208–2217, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[87] Sara Magliacane et al. Domain adaptation by using causal inference to predict invariant conditional distributions. In S. Bengio et al., editors, *Advances in Neural Information Processing Systems 31*, pages 10846–10856. Curran Associates, Inc., 2018.

[88] David A. McAllester. Some pac-bayesian theorems. In *Machine Learning*, pages 230–234. ACM Press, 1998.

[89] Nicolai Meinshausen. Causality from a distributional robustness point of view. In *2018 IEEE Data Science Workshop (DSW)*, pages 6–10. IEEE, 2018.

[90] Pierre-Alexandre Mattei and Jes Frellsen. MIWAE: Deep generative modelling and imputation of incomplete data sets. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4413–4423. PMLR, 09–15 Jun 2019.

[91] Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *Journal of Machine Learning Research*, 11(80):2287–2322, 2010.

[92] Boris Muzellec, Julie Josse, Claire Boyer, and Marco Cuturi. Missing data imputation using optimal transport. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 7130–7140, Virtual, 13–18 Jul 2020. PMLR.

[93] Karthika Mohan and Judea Pearl. On the testability of models with missing data. In *Artificial Intelligence and Statistics*, pages 643–650, 2014.

[94] Andreas Maurer, Massimiliano Pontil, and Bernardino Romera-Paredes. The benefit of multitask representation learning. *The Journal of Machine Learning Research*, 17(1):2853–2884, 2016.

[95] Karthika Mohan, Judea Pearl, and Jin Tian. Graphical models for inference with missing data. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26, pages 1277–1285. Curran Associates, Inc., 2013.

[96] Behnam Neyshabur, Srinadh Bhojanapalli, and Nathan Srebro. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. *arXiv preprint arXiv:1707.09564*, 2017.

[97] Razieh Nabi, Rohit Bhattacharya, and Ilya Shpitser. Full law identification in graphical models of missing data: Completeness results. In *Proc. of International Conference on Machine Learning (ICML)*, 2020.

[98] Vaishnavh Nagarajan and J Zico Kolter. Deterministic pac-bayesian generalization bounds for deep networks via generalizing noise-resilience. *arXiv preprint arXiv:1905.13344*, 2019.

[99] Claire L Niedzwiedz, Catherine A O'Donnell, Bhautesh D Jani, Evangelia Demou, Frederick K Ho, Carlos Celis-Morales, Barbara I Nicholl, Frances Mair, Paul Welsh, Naveed Sattar, Jill Pell, and Srinivasa Vittal Katikireddi. Ethnic and socioeconomic differences in sars-cov-2 infection: prospective cohort study using uk biobank. *medRxiv*, 2020.

[100] Razieh Nabi and Ilya Shpitser. Fair inference on outcomes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[101] Hyeonwoo Noh, Tackgeun You, Jonghwan Mun, and Bohyung Han. Regularizing deep neural networks by noise: Its interpretation and optimization. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 5115–5124, Red Hook, NY, USA, 2017. Curran Associates Inc.

[102] Y. Ovadia, E. Fertig, J. Ren, Zachary Nado, D. Sculley, Sebastian Nowozin, Joshua V. Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *NeurIPS*, 2019.

[103] Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):947–1012, 2016.

[104] J. Pearl. *Causality*. Causality: Models, Reasoning, and Inference. Cambridge Univ. Press, 2009.

[105] Pedro H. O. Pinheiro. Unsupervised domain adaptation with similarity learning. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8004–8013, 2018.

[106] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.

[107] Arnu Pretorius, Steve Kroon, and Herman Kamper. Learning dynamics of linear denoising autoencoders. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4141–4150. PMLR, 10–15 Jul 2018.

[108] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *ICML*, 2012.

[109] Open Powerlifting. Power lifting, 2019. Retrieved from Kaggle, March 2019.

[110] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, Oct 2010.

[111] Mateo Rojas-Carulla, Bernhard Schölkopf, Richard Turner, and Jonas Peters. Invariant models for causal transfer learning. *Journal of Machine Learning Research*, 19(36):1–34, 2018.

[112] Joseph Ramsey, Madelyn Glymour, Ruben Sanchez-Romero, and Clark Glymour. A million variables and more: the fast greedy equivalence search algorithm for learning high-dimensional graphical causal models, with an application to functional magnetic resonance images. *International Journal of Data Science and Analytics*, 3(2):121–129, Mar 2017.

[113] Thomas Richardson. Markov properties for acyclic directed mixed graphs. *Scandinavian Journal of Statistics*, 30(1):145–157, 2003.

[114] Dominik Rothenhausler, Nicolai Meinshausen, Peter Buhlmann, and Jonas Peters. Anchor regression: heterogeneous data meet causality. *CoRR*, abs/1801.06229, 2018.

[115] Brian C. Ross. Mutual information between discrete and continuous data sets. *PLOS ONE*, 9(2):1–5, 02 2014.

[116] Paul R Rosenbaum and Donald B Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.

[117] Marc'Aurelio Ranzato and Martin Szummer. Semi-supervised learning of compact document representations with deep networks. In *Proceedings of the 25th International Conference on Machine Learning*, pages 792–799, 01 2008.

[118] James Robins, Mariela Sued, Quanhong Lei-Gomez, and Andrea Rotnitzky. Comment: Performance of double-robust estimators when" inverse probability" weights are highly variable. *Statistical Science*, 22(4):544–559, 2007.

[119] Donald B Rubin. Causal inference using potential outcomes: Design, modeling, decisions. *Journal of the American Statistical Association*, 100(469):322–331, 2005.

[120] Bernhard Schoelkopf et al. On causal and anticausal learning. *Proceedings of the 29th International Conference on Machine Learning (ICML)*, 2, 06 2012.

[121] Daniel J. Stekhoven and Peter Bühlmann. Missforest - non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28 1:112–8, 2012.

[122] Swami Sankaranarayanan, Yogesh Balaji, Carlos D. Castillo, and Rama Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8503–8512, 2018.

[123] Mehdi S. M. Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[124] Alejandro Schuler, Michael Baiocchi, Robert Tibshirani, and Nigam Shah. A comparison of methods for model selection when estimating individual treatment effects. *arXiv preprint arXiv:1804.05146*, 2018.

[125] Gideon Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978.

[126] Elizabeth A Stuart, Eva DuGoff, Michael Abrams, David Salkever, and Donald Steinwachs. Estimating causal effects in observational studies using electronic health data: challenges and (some) solutions. *Egems*, 1(3), 2013.

[127] Cathie Sudlow, John Gallacher, Naomi Allen, Valerie Beral, Paul Burton, John Danesh, Paul Downey, Paul Elliott, Jane Green, Martin Landray, Bette Liu, Paul Matthews, Giok Ong, Jill Pell, Alan Silman, Alan Young, Tim Sprosen, Tim Peakman, and Rory Collins. Uk biobank: An open access resource for identifying the causes of a wide range of complex diseases of middle and old age. *PLOS Medicine*, 12(3):1–10, 03 2015.

[128] Shaun Seaman, John Galati, Dan Jackson, and John Carlin. What is meant by "missing at random"? *Statistical Science*, pages 257–268, 2013.

[129] Peter Spirtes, Clark N Glymour, Richard Scheines, and David Heckerman. *Causation, prediction, and search*. MIT press, 2000.

[130] Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.

[131] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

[132] Edward H Simpson. The interpretation of interaction in contingency tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, 13(2):238–241, 1951.

[133] Bernhard Schölkopf, Dominik Janzing, Jonas Peters, Eleni Sgouritsa, Kun Zhang, and Joris Mooij. On causal and anticausal learning. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, page 459–466, Madison, WI, USA, 2012. Omnipress.

[134] Uri Shalit, Fredrik D Johansson, and David Sontag. Estimating individual treatment effect: generalization bounds and algorithms. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3076–3085. JMLR. org, 2017.

[135] Patrick Schwab and Walter Karlen. Cxplain: Causal explanations for model interpretation under uncertainty. In *Advances in Neural Information Processing Systems 32*, pages 10220–10230. Curran Associates, Inc., 2019.

[136] Masashi Sugiyama, Matthias Krauledat, and Klaus-Robert Müller. Covariate shift adaptation by importance weighted cross validation. *J. Mach. Learn. Res.*, 8:985–1005, December 2007.

[137] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. *ICCV*, 2019.

[138] Ilya Shpitser, Karthika Mohan, and Judea Pearl. Missing data as a causal and probabilistic problem. In *Proc. of Conference on Uncertainty in Artificial Intelligence (UAI)*, 2015.

[139] John Shawe-Taylor and Robert C Williamson. A pac analysis of a bayesian estimator. In *Proceedings of the tenth annual conference on Computational learning theory*, pages 2–9, 1997.

[140] Jonathan AC Sterne, Ian R White, John B Carlin, Michael Spratt, Patrick Royston, Michael G Kenward, Angela M Wood, and James R Carpenter. Multiple imputation for missing data in epidemiological and clinical research: potential and pitfalls. *Bmj*, 338, 2009.

[141] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3723–3732, 2018.

[142] Jason Tashea. Courts are using ai to sentence criminals. that must stop now. *WIRED*, Apr 2017.

[143] Olga Troyanskaya, Mike Cantor, Gavin Sherlock, Trevor Hastie, Rob Tibshirani, David Botstein, and Russ Altman. Missing value estimation methods for dna microarrays. *Bioinformatics*, 17:520–525, 07 2001.

[144] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2962–2971, 2017.

[145] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[146] Shuhan Tan, Jiening Jiao, and Wei-Shi Zheng. Weakly supervised open-set domain adaptation by dual-domain collaboration. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5389–5398, 2019.

[147] Joel A Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of computational mathematics*, 12(4):389–434, 2012.

[148] Ruibo Tu, Cheng Zhang, Paul Ackermann, Karthika Mohan, Hedvig Kjellström, and Kun Zhang. Causal discovery in the presence of missing data. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1762–1770. PMLR, 2019.

[149] S. van Buuren. *Flexible Imputation of Missing Data*. Chapman & Hall/CRC Interdisciplinary Statistics. CRC Press LLC, 2018.

[150] Sara Van de Geer, Peter Bühlmann, et al. l_0-penalized maximum likelihood for sparse directed acyclic graphs. *The Annals of Statistics*, 41(2):536–567, 2013.

[151] Mark J Van der Laan and James M Robins. *Unified methods for censored longitudinal data and causality*. Springer Science & Business Media, 2003.

[152] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008.

[153] Pascal Vincent, H. Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.*, 11:3371–3408, 2010.

[154] Chih M. Wong et al. Heart failure in younger patients: the Meta-analysis Global Group in Chronic Heart Failure (MAGGIC). *European Heart Journal*, 35(39):2714–2721, 06 2014.

[155] Stefan Wager and Susan Athey. Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, 113(523):1228–1242, 2018.

[156] Elizabeth Williamson, Alex J Walker, Krishnan J Bhaskaran, Seb Bacon, Chris Bates, Caroline E Morton, Helen J Curtis, Amir Mehrkar, David Evans, Peter Inglesby, Jonathan Cockburn, Helen I Mcdonald, Brian MacKenna, Laurie Tomlinson, Ian J Douglas, Christopher T Rentsch, Rohini Mathur, Angel Wong, Richard Grieve, David Harrison, Harriet Forbes, Anna Schultze, Richard T Croker, John Parry, Frank Hester, Sam Harper, Rafael Perera, Stephen Evans, Liam Smeeth, and Ben Goldacre. Factors associated with covid-19-related death using opensafely. *Nature*, 584, 2020.

[157] Stefan Wager, Sida Wang, and Percy Liang. Dropout training as adaptive regularization. *Advances in Neural Information Processing Systems*, 07 2013.

[158] Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. Differentially private generative adversarial network. *CoRR*, abs/1802.06739, 2018.

[159] Depeng Xu, Yongkai Wu, Shuhan Yuan, Lu Zhang, and Xintao Wu. Achieving causal fairness through generative adversarial networks. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, 2019.

[160] Depeng Xu, Shuhan Yuan, Lu Zhang, and Xintao Wu. Fairgan: Fairness-aware generative adversarial networks. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 570–575. IEEE, 2018.

[161] Jinsung Yoon, L. Drumright, and M. van der Schaar. Anonymization through data synthesis using generative adversarial networks (ads-gan). *IEEE Journal of Biomedical and Health Informatics*, 24:2378–2388, 2020.

[162] Jinsung Yoon, James Jordon, and Mihaela van der Schaar. Gain: Missing data imputation using generative adversarial nets. In *ICML*, 2018.

[163] Jinsung Yoon, James Jordon, and Mihaela van der Schaar. Ganite: Estimation of individualized treatment effects using generative adversarial nets. *International Conference on Learning Representations (ICLR)*, 2018.

[164] Mengyue Yang, Furui Liu, Zhitang Chen, Xinwei Shen, Jianye Hao, and Jun Wang. Causalvae: Disentangled representation learning via neural structural causal models, 2021.

[165] Liuyi Yao, Sheng Li, Yaliang Li, Mengdi Huai, Jing Gao, and Aidong Zhang. Representation learning for treatment effect estimation from observational data. In *Advances in Neural Information Processing Systems*, pages 2633–2643, 2018.

[166] Larry S. Yaeger, Richard F. Lyon, and Brandyn J. Webb. Effective training of a neural network character classifier for word recognition. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems 9*, pages 807–816. MIT Press, 1997.

[167] Jiaxuan You, Xiaobai Ma, Daisy Yi Ding, Mykel Kochenderfer, and Jure Leskovec. Handling missing data with graph representation learning. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2020.

[168] Seongwook Yoon and Sanghoon Sull. Gamin: Generative adversarial multiple imputation network for highly missing data. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[169] Kaichao You, Ximei Wang, Mingsheng Long, and Michael Jordan. Towards accurate model selection in deep unsupervised domain adaptation. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7124–7133, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[170] Xun Zheng, Bryon Aragam, Pradeep Ravikumar, and Eric P. Xing. DAGs with NO TEARS: Continuous Optimization for Structure Learning. In *Advances in Neural Information Processing Systems*, 2018.

[171] Junzhe Zhang and Elias Bareinboim. Fairness in decision-making—the causal explanation formula. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[172] Shengyu Zhu and Zhitang Chen. Causal discovery with reinforcement learning. *CoRR*, abs/1906.04477, 2019.

[173] Hongyi Zhang, Moustapha Cisse, Yann Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *Proceedings of the 6th International Conference on Learning Representations (ICLR)*, 2018.

[174] Xun Zheng, Chen Dan, Bryon Aragam, Pradeep Ravikumar, and Eric Xing. Learning sparse nonparametric dags. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 3414–3425. PMLR, 26–28 Aug 2020.

[175] Erheng Zhong, Wei Fan, Qiang Yang, Olivier Verscheure, and Jiangtao Ren. Cross validation framework to choose amongst models and datasets for transfer learning. In José Luis Balcázar, Francesco Bonchi, Aristides Gionis, and Michèle Sebag, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 547–562, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[176] 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, pages 3150–3157. AAAI Press, 2015.

[177] Qingyuan Zhao and Trevor J. Hastie. Causal interpretations of black-box models. In *Journal of Business & Economic Statistics*, 2017.

[178] Kun Zhang, Bernhard Schölkopf, Krikamol Muandet, and Zhikun Wang. Domain adaptation under target and conditional shift. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning (ICML)*, volume 28 of *Proceedings of Machine Learning Research*, pages 819–827, 2013.

[179] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P Gummadi. Fairness constraints: Mechanisms for fair classification. In *Artificial Intelligence and Statistics*, pages 962–970. PMLR, 2017.

[180] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *International conference on machine learning*, pages 325–333. PMLR, 2013.

[181] Lu Zhang, Yongkai Wu, and Xintao Wu. A causal framework for discovering and removing direct and indirect discrimination. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 3929–3935, 2017.