Understanding Hyperparameters in Machine Learning for Causal Estimation from Observational Data

Damian Machlanski
School of Computer Science and Electronic Engineering
University of Essex

A thesis submitted for the degree of
Doctor of Philosophy
September 2024

To Karolina.

Acknowledgements

Throughout this PhD journey, I have been supported by many people in all sorts of ways, towards whom I would like to express my sincere gratitude in the following paragraphs.

First and foremost, to my PhD supervisors **Spyros Samothrakis** and **Paul Clarke**, for introducing me to the fascinating world of causality, for constant support, advice, encouragement, understanding and patience, but also for trusting me and giving space to find my own way into research.

This project would not be possible without the **ESRC Research Centre on Micro-Social Change (MiSoC), ISER** (grant number: ES/S012486/1), who funded my PhD and continued to support me in the last months. I also want to thank the **ESRC Business and Local Government Data Research Centre** (grant number: ES/S007156/1), for supporting my work through computational resources that made my experiments possible. Thank you to **Institute for Analytics and Data Science (IADS)**, for providing me with the working environment, including me in the group that made a big difference in the last year of my PhD, and providing additional support related to participation in conferences.

To my former MSc supervisors, **Ana Matran-Fernandez** and **Luca Citi**, who reignited my interest in research that subsequently led me to this PhD, with special thanks to Ana for her continued support beyond my MSc, providing me with teaching opportunities, and being there for me when I needed it the most.

To **Annalivia** and **JunKyu**, for insightful and inspiring conversations, companionship, feedback, and countless PhD and career-related advice that I cannot thank for enough.

To **Tasos**, for sharing the thesis template that made the preparation of this report a much smoother and more enjoyable job.

To the **University of Essex community, staff and students**, which I have been part of since 2018, for being always so kind and welcoming, but also for helping me find myself throughout my entire journey at Essex. This includes but is not limited to: MiSoC/ISER students **Tommaso**, **Omar** and **Hettie**; my ISER roommates **Laura**, **Selin** and **Jim**; IADS colleagues **David**, **Nina** and **Maria**; **Angus** for our conversations about running; and my gym buddies **Şefki** and **Priyanka**. Thank you all for making Essex the place it is.

To my family, for being the mental pillars that allowed me to withstand all the challenges put in front of me. **Cezary**, **Emanuela** and **Marcelina**, for their friendship and unconditional kindness. My sister **Eliza**, for always believing in her little brother. My mother **Jolanta**, for her love and sacrifice. And lastly, but most importantly, my friend, love and wife, **Karolina**; thank you for everything.

Abstract

Causal analysis is fundamental to science and decision-making. It unravels the structure of the process underlying the data and estimates the effectiveness of interventions. Deriving causal notions from randomised experiments is well understood and relatively simple analytically. However, there is also considerable interest in the analysis of far more widely available non-experimental observational data, which is substantially more challenging. There has been considerable research on causal analysis in the statistics literature, but more recently also from computer scientists since the robustness and performance of existing statistical approaches to causal estimation can be improved by machine learning (ML). However, despite growing evidence that hyperparameters of ML methods are critical for strong predictive performance, this aspect is neglected in causality. To make matters worse, the use of observational data, while convenient due to their availability, creates serious obstacles for model evaluation and tuning. The relationship between hyperparameters and model performance is understudied in ML, let alone in significantly more challenging causal settings. This work fills this gap by investigating the intricate interplay of challenges posed by causal settings, ML methods, hyperparameter selection, and estimation performance, all within the context of two causal estimation tasks: treatment effect estimation and causal structure learning. This unique direction has led this study to several original contributions, such as a novel estimation method, or the first of their kind extensive performance evaluation analyses from the perspective of hyperparameters. The results form the ultimate claim of this thesis, which is that hyperparameters play a pivotal role in causal estimation performance, but crucially, their optimisation is significantly limited by incomplete observations within observational causal data. Consequently, this work calls for more careful treatment of hyperparameters in practice and more fundamental research into causal hyperparameter optimisation to harness the full potential of ML in causal estimation.

Contents

List of Figures

List of Tables

Glossary and Abbreviations

ANM . . . . . *Additive Noise Model*

ATE . . . . . *Average Treatment Effect*

ATT . . . . . *Average Treatment effect on the Treated*

BIC . . . . . *Bayesian Information Criterion*

CAM . . . . . *Causal Additive Model*

CATE . . . . . *Conditional Average Treatment Effect*

CGNN . . . . *Causal Generative Neural Network*

CPDAG . . . . *Completed Partially Directed Acyclic Graph*

CSL . . . . . *Causal Structure Learning*

CV . . . . . *Cross-Validation*

DAG . . . . . *Directed Acyclic Graph*

DeGeT . . . . *Debiasing Generative Tree*

DeGeF . . . . *Debiasing Generative Forest*

DGP . . . . . *Data Generating Process*

DML . . . . . *Double Machine Learning*

FCI . . . . . *Fast Causal Inference*

FGES . . . . . *Fast Greedy Equivalence Search*

FN . . . . . *False Negative*

FP . . . . . *False Positive*

GAN . . . . . *Generative Adversarial Network*

GMM . . . . . *Gaussian Mixture Model*

HP . . . . . *Hyperparameter*

HPO . . . . . *Hyperparameter Optimisation*

IID . . . . . *Independent and Identically Distributed*

IHDP . . . . . *Infant Health Development Program*

IPTW . . . . . *Inverse Probability of Treatment Weighting*

IPW . . . . . *Inverse Propensity Weighting*

ITE . . . . . *Individual Treatment Effect*

kNN . . . . . *k-Nearest Neighbours*

LiNGAM . . . *Linear Non-Gaussian Acyclic Model*

MADE . . . . *Masked Autoencoder for Distribution Estimation*

ML . . . . . *Machine Learning*

MSE . . . . . *Mean Squared Error*

NN . . . . . *Neural Network*

NSWP . . . . *National Supported Work Program*

oMSE . . . . . *observable Mean Squared Error*

OOD . . . . . *Out-Of-Distribution*

pMSE . . . . . *potential Mean Squared Error*

PC . . . . . *Peter-Clark*

PDAG . . . . . *Partially Directed Acyclic Graph*

PEHE . . . . . *Precision in Estimation of Heterogeneous Effect*

PO . . . . . *Potential Outcome*

RC . . . . . *Rank Correlation*

RCT . . . . . *Randomised Controlled Trial*

RMSE . . . . . *Root Mean Squared Error*

RL . . . . . *Reinforcement Learning*

SCM . . . . . *Structural Causal Model*

SEM . . . . . *Structural Equation Model*

SGD . . . . . *Stochastic Gradient Descent*

SHD . . . . . *Structural Hamming Distance*

SotA . . . . . *State-of-the-Art*

SUTVA . . . . *Stable Unit Treatment Value Assumption*

TMLE . . . . . *Targeted Maximum Likelihood Estimation*

TP . . . . . *True Positive*

VAE . . . . . *Variational Auto-Encoder*

1 Introduction

This chapter introduces the thesis in a bottom-up manner. It first briefly presents its components independently, such as causality and machine learning, which after connecting all together form a unique perspective that unravels novel questions and ultimately motivates this work. A summary of contributions and contents ahead concludes the chapter.

Contents

1.1 Causality

Gaining causal knowledge from data is fundamental to science. It allows going beyond ambiguous associations and instead answers questions that are causal in nature, such as "*Does smoking cause cancer?*", as opposed to a limited perspective

of "*Is smoking associated with cancer?*". However, many statisticians would argue that it is not important if smoking (S) causes cancer (C) or whether people with cancer just happen to smoke as long as the joint distribution $P(S, C)$[1] can be modelled accurately. The situation changes when we consider patient data from different hospitals, resulting in different marginals $P(S)$. The model that follows the causal factorisation[2] $P(C|S)P(S)$ (Figure 1.1 left) and which contains the causal component (*causal mechanism* [1]) $P(C|S)$ will generalise across hospitals, but the *anti-causal* model[3] that subsumes factorisation $P(S|C)P(C)$ (Figure 1.1 right) will not show such robustness. The fact that the causal factorisation is faithful to the true data generating process (DGP), where smoking causes cancer, makes it invariant to domain changes. And although these data shifts violate the usual assumption of independent and identically distributed (IID) data, they are prevalent in real-world applications[4].

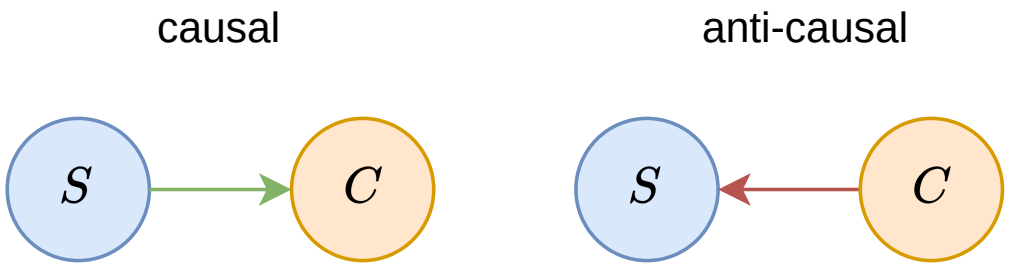


Figure 1.1: Causal graphs depicting the problem with Smoking (S) and Cancer (C) variables.

Indeed, it is causal questions and functions that are usually of interest and which require addressing to make meaningful progress. However, statistics have forbidden researchers to think in causal terms for decades by saying that "*correlation is not causation*", or at best by trivialising causality as merely a special case of

[1] P denotes probability.

[2] Factorisation of the joint distribution into its constituent conditional and marginal distributions that respects the topology of the assumed causal graph is called a causal factorisation.

[3] Anti-causal model involves assuming variable relationships that have the opposite direction to causal ones, hence 'anti-'.

[4] IID assumes the same distributions in training and production, but different hospitals encountered in production may involve distributions different from those seen in training.

correlation (Pearson correlation coefficient of ± 1). This unnatural constraint, in turn, gave rise to obscure situations like *spurious correlations*[5] and many other well-known paradoxes (e.g. Simpson's paradox [2][6]) that could not be explained by standard statistical tools.

First attempts to unlock causal language in science date back to 1920 when Wright introduced his *path analysis* in a guinea pig study [3], further generalised to a discussion on correlation and causation [4]. Such controversial at the time thoughts met strong opposition from statistics and hence did not lead to this approach being more commonly incorporated, despite fair recognition from other fields [5–7] and Wright's defence 60 years later [8].

It was only in the early 2000s when causality was finally "*mathematised*"[7] by Pearl's *do-calculus* [10]. Crucially, this formal mathematical framework allows distinguishing between probabilities that are affected by observing X, as in $P(Y|X)$, and those where X is intervened on, expressed as $P(Y|do(X))$. In this context, by intervention, we mean performing an action, such as applying a medical treatment, which in practical terms translates to setting X to a desired value that represents the action (e.g. $P(Y|do(X = 1))$; see Figure 1.2). Furthermore, such notation can be used to express causal relationships by saying that "X causes Y" if $P(Y|do(X)) > P(Y)$, as opposed to $P(Y|X) > P(Y)$ that only implies observing X. With this approach, it is possible to build complex graphs that capture all assumed causal relationships in the underlying DGP. These *graphical models* represent a causal model and provide a way of explicitly expressing data assumptions made by the analyst.

Equipped with this framework, it is possible to solve problems beyond the reach of standard statistics. As depicted by Pearl in his *ladder of causation* [9], standard tools are limited to associations and involve questions like "*How would Y change if*

[5] Coincidental correlation of otherwise unrelated events.

[6] For example, we may observe a positive trend across the entire population, but negative per-group trends upon creating groups with a new input variable.

[7] Credit to Pearl [9].

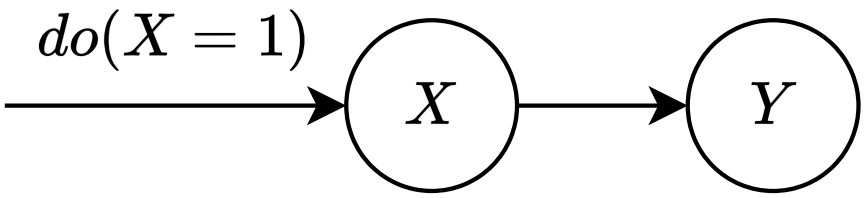


Figure 1.2: Example causal graph with two variables X and Y. The do-notation represents intervention, in this case setting X to 1.

I observe X?". It is only with causal tools that questions involving interventions or retrospective thinking (counterfactuals) can be answered, such as "*Will Y change if I do X?*" or "*What would Y be if I had acted differently?*" respectively. In real-world situations, many sought-after questions are in fact causal. As opposed to asking "*What should be the price of a given house based on its properties?*", what is often of real interest is "*How the price of this house will change if we modify X?*", or even better "*What would be the cheapest investment in the house that would increase its value the most?*".

While the above framework enables the expression of causal questions mathematically, serious obstacles to computing the answers through data still exist. To demonstrate one of the main problems clearly, let us use Rubin's *potential outcomes* framework [11]. The outcomes of units (e.g. health index) that received treatment ($T = 1$; e.g. experimental drug) are denoted with Y_1, and those in the control group ($T = 0$; placebo) have outcomes Y_0. Now, in order to determine if the intervention achieves desirable effects, one must answer the question "*Does T improve Y?*", which requires computing $Y_1 - Y_0$ for each individual in the study. However, it is possible to observe only one outcome for each individual. We do not know what would have happened to patients that received the treatment, had they not received it in the first place (we observe Y_1 but not Y_0), and vice versa. The outcomes that correspond to the alternative reality that did not happen are called *counterfactuals*. The fact that they are missing, but are needed to compute causal effects entails the **fundamental problem of causal inference**, demonstrated also in Table 1.1.

ID	X_1	X_2	X_3	T	Y_0	Y_1
1	1.397	0.996	0	1	**?**	4.771
2	0.269	0.196	1	0	2.956	**?**
3	1.051	1.795	1	1	**?**	4.164
4	0.662	0.196	0	1	**?**	6.172
5	0.856	1.795	1	0	7.834	**?**

Table 1.1: Demonstration of the fundamental problem of causal inference. $X_1 - X_3$ are background features, T intervention/treatment status, and Y_T correspond to observed outcomes. Question marks denote unobserved (missing) counterfactuals (Y_{1-T}). The problem stems from the fact that we always observe only one outcome per data unit that corresponds to the assigned treatment status. The outcome for the same unit but a different treatment status is always unknown. Thus, the effect of the form $Y_1 - Y_0$ cannot be directly computed.

Although causal effects cannot be directly computed, they can be approximated through the appropriate use of statistical methods. This gave rise to many causal estimation methods, which in general can be grouped into two categories based on the type of questions they attempt to answer. Questions of type "*How X causes Y?*" involve the approximation of the function between the two variables and ultimately the quantification of the causal effect. This is mostly known as *causal inference* or *treatment effect estimation* [10]. The other type of question aims to determine whether two (or more) variables are causally connected at all, and if so, whether X causes Y or the other way around. With multiple variables involved, the product is a structure of interconnected causal variables – causal graphs. The task of inferring such models is often referred to as *causal discovery* or *causal structure learning* (CSL) [12]. Both types of causal tasks are further discussed in Chapter 2.

1.1.1 Observational Data

Data derived from Randomised Controlled Trials (RCTs) are the gold standard in science when it comes to causal analysis. Owing to their randomisation, the treatment assignment does not depend on the data unit itself, whereas their controlled nature reduces discrepancies between the treated and control groups. Both characteristics are paramount to causal inference as they ensure unbiased estimates (Section 2.4.1), but conducting such experiments is often expensive or

even infeasible due to ethical reasons (e.g. drinking alcohol or smoking). Utilising much more accessible observational data collected passively without any controlled protocol overcomes these issues. For example, even though it is impossible to prescribe smoking as part of an experiment, information about existing smokers and non-smokers can be collected through surveys. However, because people decide on their own whether to smoke or not, the treatment assignment is no longer random, and it depends on the unit's background features $P(T|X)$. Note, while not considered in this work, T may depend on omitted variables that are not conditionally independent of $Y|X$. This non-random assignment in practice leads to data biases (see Section 2.4.1) where certain data subpopulations are either underrepresented or missing (e.g. no/very few young smokers). As a result, distributions of treated and control units differ in the input space to the extent that it makes accurate causal effect estimation very challenging. This problem is more generally known in the literature as *covariate shift*, where there is variation in the response pattern across response surfaces.

Another potential problem arises due to possible *confounding* of the causal effect, wherein both treatment T and outcome Y depend on the same background features X. This considerably complicates identification and modelling as all the input features need to be carefully handled to *de-confound* the effect of interest, which notably may increase the complexity of the target function. Another common take on this problem is to simply assume no confounding as part of the set of assumptions made a priori.

There is also a parallel to be drawn with *domain adaptation*, wherein generalising to a distribution different from that seen in training is the core problem. The major difference between the areas is that in domain adaptation the distributional discrepancies occur between the training and test sets, whereas in causal inference they usually take place between treated and control units. Both challenges essentially entail performing well on previously unseen (parts of) distributions that can arise due to covariate shift, known as *out-of-distribution* (OOD) generalisation problem.

In terms of observational data, there are two major types important to distinguish between. The distinction is based on the number of times each unit is observed over time. *Longitudinal* (or *panel*) data involve multiple observations per data unit across a longer period of time. The readings are usually taken at regular time intervals (e.g. every month or year) at which point information about all units is collected. This time-series data structure shares similarities with standard forecasting problems, with the crucial difference being that in panel data we deal with multiple entities observed over time as opposed to just one in classic forecasting (e.g. a single stock price). Though multi-channel time-series is quite common in, for example, biomedical engineering. Perhaps a closer resemblance to the DGP behind panel data is Reinforcement Learning (RL), wherein an agent interacts with the environment across multiple actions (creates observations over time) repeated over separate game episodes (creates distinct units). A noteworthy difference in this case, however, is that RL agents are often allowed to perform one of multiple possible actions, whereas in causal inference it is customary to consider only two actions (treated or untreated status). In addition, the number of actions performed by the agent can vary among game episodes, whereas in panel data each unit is expected to be observed an equal amount of times.

The second type of observational data involves only a single observation per data unit; there are no repeated measurements over time. This variant is often referred to as *cross-sectional* data and is notably much more common than longitudinal/panel data. It closely resembles classic tabular data structures known in machine learning (ML) with the only exception of the treatment status variable T which is considered a special input feature on top of the usual background covariates X.

In this work, we focus specifically on cross-sectional observational data and consider all aforementioned challenges. A more detailed discussion about causal data is further continued in Chapter 2.

1.2 Machine Learning

Finding patterns in data has always been strongly rooted in statistics [13], but perhaps what sets ML apart is its focus on the extraction of algorithms from data for machine intelligence [14] that can be used to predict future outcomes (i.e. extrapolation) as opposed to merely explaining existing data (i.e. interpolation) that is often the case with classic statistics.

Today's ML is commonly divided into supervised and unsupervised learning tasks where the data in the former include expected answers (or *labels*) that can be utilised to identify an approximating function that fits given data well enough. The approximators are further grouped into regressors and classifiers depending on the type of the target variable (continuous or discrete respectively). With the lack of labels, unsupervised learning is mostly concerned with finding meaningful structures within the data. RL is a third ML category that involves agents interacting with an environment and which often receive a form of a reward (or penalty) to incentivise desired behaviour.

Regardless of the three categories mentioned above, the (artificial) Neural Network (NN) architecture has always been an important part of ML. Despite its early Perceptron form [15] encountered some setbacks [16] (e.g. inability to solve the XOR problem), it was eventually proved to have universal approximation properties [17] that laid the foundations for deep learning [18] and its subsequent successes over the last three decades. Some notable NN-based deep learning breakthroughs occurred in the areas of computer vision [19–21], text translation [22], RL in games [23] and recently in chatbots [24].

The ML successes showed that it is possible to find accurate approximations in highly complex nonlinear settings, even with very little to no data preprocessing, also known as end-to-end learning. This sparked a lot of interest in using ML methods in other application areas, including causal analysis.

1.2.1 Hyperparameters

The parameters of a model define a space and form of possible solutions, with different forms often referred to as *model or hypothesis classes* $\mathcal{H}$. Through optimisation, an exact solution ($h \in \mathcal{H}$) in the form of specific parameter values can be identified. For example, a regression line of the form $\mathcal{H}_1$: $y = mx + b$ consists of two such parameters, where m is the slope and b is the intercept. Through optimisation (e.g. minimisation of the least squares loss), one possible solution could be h_1: $y = x + 2$ ($m = 1$, $b = 2$). Further, by adding more polynomial terms we can expand the space of solutions to more complex models, such as $\mathcal{H}_2$: $y = \theta_2 x^2 + \theta_1 x + \theta_0$, with $\theta \in \{\theta_0, \theta_1, \theta_2\}$ being the parameters. Or more generally: $\mathcal{H}(n)$: $y = \theta_n x^n + \theta_{n-1} x^{n-1} + \cdots + \theta_1 x + \theta_0$.

The difference between any two hypothesis classes in this case can be captured by the variable n that controls the number of parameters and associated polynomial terms. That is, by changing n it is possible to move from one solution space to another. To differentiate between model parameters θ and special variables like n that change the hypothesis class $\mathcal{H}$, the latter can be named **hyperparameters** (HPs). Many ML methods incorporate them in one way or another, regardless of their parametric or non-parametric nature. They can be also thought of as parameters of the algorithm that do not necessarily correspond to the parameters of the underlying model that generated the data. Some examples of HPs include L1 (sparsity) and L2 (model complexity) regularisation terms in linear regression, maximum depth in Decision Trees, or the number of hidden layers and neurons in NNs.

Hyperparameters can often control how general a hypothesis is, as is the case with n above, where higher n values create more specific (or complex) hypotheses that can fit given data more closely but can also result in overfitting. Thus, hyperparameters clearly play a vital role in the bias-variance trade-off and their optimisation, or *tuning*, can reduce prediction error if done correctly [25].

In the context of this thesis, we are particularly interested in the types of HPs that control model complexity as well as those that specify and control optimisation algorithms. When it comes to selecting between different model classes, this task falls under the name of **model selection**. Furthermore, we define HP selection as a sub-task of model selection. That is, in this work, model selection entails selecting HP values **and** selecting model classes. In other words, we treat all modelling and learning choices that are confined within a single model class as **hyperparameter selection**, whereas the task of **model selection** involves selecting between model classes and HP values.

On a high level, tuning can be broken down into three major sub-problems: exploration of hyperparameter values, performance evaluation of candidate values, and selecting the winning combination. While the principles are straightforward, each sub-problem involves non-trivial challenges. In exploration, larger search spaces increase the probability of finding the right hypothesis class for given data but also raise computational demands. This puts exhaustive grid searches at a specific disadvantage, creating the need for alternative exploration approaches, such as greedy or evolutionary algorithms. Model evaluation requires a metric that will be used to score candidate values, but different metrics may favour certain types of predictors [26]. The decision about whether to evaluate candidates on a single validation set or through Cross-Validation (CV) may also affect the final solution, not to mention further details on data splits such as split percentages or the number of folds in CV. Model assessment in ML also entails an important assumption that training and test sets come from the same distribution, which might be invalidated under covariate shifts, resulting in incorrectly selected $\mathcal{H}$. Picking the right combination may also be more complex as sometimes it might be more desirable to pick not one but multiple top candidates concerning the metric to increase robustness.

With all types of choices above, hyperparameter tuning becomes a challenge that makes it prone to mistakes. At the same time, tuning also grows in importance

together with increased interest in NNs due to their extreme modelling flexibility and sensitivity to HPs. Despite this, HPs are often neglected. Partly due to convenient implementations nowadays, practitioners tend to underappreciate the critical influence HPs have on the methods they use, often deferring to default hyperparameter values suggested by package providers as a consequence, unknowingly ending up with subpar modelling outcomes. Surprisingly, ML researchers also have their fair share of bad practices in this area, with the prime example being the case when the newly introduced method was thoroughly tuned but the baselines that are compared to received little to no attention when it comes to their HPs. Such practice certainly questions the efficacy of new developments and hinders future research.

The widespread use of ML methods on non-standard datasets that pose new challenges (e.g. covariate shift) combined with hyperparameter tuning ignorance certainly calls for more attention, making the investigation of this problem one of the pillars of this work.

We continue our discussion around hyperparameters in Chapter 2.

1.3 Machine Learning for Causal Estimation

The relationship between machine learning and causality is mutually beneficial; one helps address the shortcomings of the other. Some issues within ML are model interpretability and OOD generalisation, just to name a few. The efforts under the flag of *causality for machine learning*, also termed *causal machine learning* or *causal representation learning* and inspired by the formalisation of the causal framework [10], aim at bringing the ideas from causality to solve some pressing issues in ML. That is, make models more transparent by forging intuitive causal notions into them, or incorporate model design insights from causal estimators that are meant to generalise in more challenging data conditions [1, 27].

Causality has made great strides in the past few decades, but its estimation techniques were mostly centred around population-level predictions with strongly linear assumptions. Since an increasing interest in personalised estimates in recent years, this has become a serious limitation. Encouraged by the successes of modern ML methods capable of high prediction accuracy even in nonlinear settings, the efforts of using ML in causal problems have begun, encompassed under the name of *machine learning for causality* [28, 29], or here as *machine learning for causal estimation* to make our interest in estimation tasks explicit.

Causal estimation problems are commonly divided into *causal inference* and *causal discovery*. The former is about inferring causal parameters, such as causal (or treatment) effects from observed data, in which ML methods have already made promising advancements via neural networks [30], decision trees [31], or more general meta-learners [32, 33]. A noteworthy part of causal inference is also the problem of recommending the best intervention among the set of possible ones, more commonly known in ML as *policy optimisation*, which differs from treatment effect estimation significantly enough [34][8] to have its own family of estimators [35] (i.e. policy learners).

In causal discovery, the goal is to estimate (or recover) causal graphs that explain causal relationships among input features. This task differs significantly from causal inference as it is unsupervised in nature due to the graph structures being always unknown a priori. Despite the immense challenges of CSL, ML methods have already proved their usefulness via standard NNs [36], adversarial approaches [37] and variational auto-encoders [38].

In this work, our focus is specifically on machine learning for causality, with treatment effect estimation and CSL being the main estimation tasks of interest. However, we intend to go beyond the aspect of simply using ML methods and

[8]Policy learning focuses more on the sign of the causal effect (positive/negative) rather than the exact quantification of the causal effect itself.

rather investigate the challenges that arise from the use of ML in causal settings that naturally differ from standard ML problems.

1.4 Motivation

This thesis brings together previously discussed topics into a unique perspective that unveils knowledge gaps and new challenges to be considered. As previously pointed out, there is considerable interest in using ML methods in causal settings due to their past advancements in ML as well as already successful applications in causal inference and CSL. However, even though ML procedures received widespread attention mostly because of their high performances, a critical ingredient that drives those achievements has been severely overlooked – hyperparameters. To make matters worse, causal data are often more challenging than classic ML problems due to almost universal covariate shifts but also because the ground truth is either partly (counterfactuals) or completely (graphs) hidden. As a result, many standard ML procedures, such as hyperparameter tuning and model evaluation, are not readily available for causal applications out of the box.

While there has been some work on bringing ML methods into causal problems (see Section 1.3), hyperparameters have been heavily neglected by both researchers and practitioners. Given the acknowledged importance of HPs in ML, we deem investigating their role in causality crucial for meaningful future progress in causal estimation. Thus, the main goal of this thesis is to investigate the influence of HPs in ML methods used for treatment effect estimation and CSL for the analysis of cross-sectional observational data. This brings us to the core questions and contributions of this work.

1.5 Research Questions

The identified literature gaps gave rise to the following questions:

Q1. What are the challenges of causal estimation from observational data?

Q2. How do identified challenges affect performance evaluation and hyperparameter tuning?

Q3. How does hyperparameter selection affect the performance of causal estimation methods?

Q4. How does the choice of hyperparameter values influence the selection of causal estimation methods in ensemble settings?

1.6 Main Findings

Based on conducted studies, this work claims the following findings:

F1. Incomplete observations in the form of missing counterfactuals are the root cause of challenges in all causal estimation tasks that use observational data.

F2. In causal inference, F1 leads to data biases and covariate shifts that severely impact the estimation of individualised causal effects and hyperparameter tuning.

F3. In causal discovery, F1 emerges as unobserved interventions that limit causal graph identification, but it is the lack of ground truth and subsequent limited performance evaluation that makes the selection of hyperparameters and algorithms almost impossible outside simulations.

F4. Hyperparameter tuning is critical for reliable causal effect estimation and is more important than model class selection, but its reliability depends on used evaluation metrics.

F5. The choice of a method is still important in causal discovery, but the choice of hyperparameter values does impact the final performance when selecting among multiple approaches.

1.7 Contributions

This work offers the following contributions:

C1. A novel data augmentation method that improves the estimation of individualised effects through reduced bias.

C2. Identification of the major challenges in causal estimation from observational data and their impact on hyperparameter tuning and performance evaluation.

C3. Empirical evaluation and comparison of causal estimation methods across a variety of metrics and datasets.

C4. Empirical analysis of the impact of hyperparameter selection on the performance of individual causal estimators.

C5. Empirical analysis and comparison of the influence of choosing different hyperparameter values on the estimation performance in ensemble settings.

1.8 Publications

This thesis includes the following three papers.

- *Undersmoothing Causal Estimators With Generative Trees.* Available on arXiv and GitHub. **Published** in *IEEE Access*. Part of **Chapter 3**.

- *Hyperparameter Tuning and Model Evaluation in Causal Effect Estimation.* Available on arXiv and GitHub. Submitted to *Machine Learning*. Decision: **revise and resubmit**. Part of **Chapter 4**.

- *Robustness of Algorithms for Causal Structure Learning to Hyperparameter Choice.* Available on arXiv and GitHub. **Presented** at the *3rd Conference on Causal Learning and Reasoning*. **Published** in the *Proceedings of Machine Learning Research*. Part of **Chapter 5**.

1.9 Overview of Chapters

This project evolved over time as new knowledge and evidence were gathered. While some of the main chapters (3-5) may seem disconnected, they reflect the journey behind the project which upon some additional explanation should come together as a meaningful whole.

The initial research idea was to combine both causal inference and discovery into a singular framework for complete causal analysis capable of recovering and processing causal graphs for superior estimation of personalised causal effects of any desired interventions. This approach was mainly motivated by recent breakthroughs in CSL algorithms that seemed to be ready for real applications, or so the published research claimed. Upon further investigation, this endeavour failed as we found the algorithms extremely volatile and ineffective in environments outside of simulations. Thus, a practical decision was made to abandon the development of the CSL module in the envisioned framework and focus solely on effect estimation methodology in individualised settings where a fixed causal graph is usually assumed. This is how we arrived at our new method presented in Chapter 3.

Although we developed the new method successfully, we struggled with showcasing the method's full performance capabilities due to newly discovered difficulties with hyperparameter tuning. It was at this moment that we realised that model

evaluation and tuning are substantially more challenging in causal inference than in standard ML, but we were not aware yet of the consequences. Surprisingly, our experiences were barely (if at all) mentioned in the literature, let alone thoroughly studied. Thus, we investigated the issue of model selection and tuning systematically and arrived at astonishingly counterintuitive conclusions, presented in Chapter 4.

Equipped with the new knowledge of the importance of hyperparameters in effect estimation and still remembering the difficulties of using causal discovery methods due to their instabilities, we started wondering about the role of hyperparameters in CSL. The motivation to study the issue more closely grew even stronger upon realising that HPs are heavily neglected, almost completely ignored, in the causal discovery literature; a state of things much worse compared to already quite neglected treatment effect estimation methodology. As a consequence, the entire project circled back to the previously abandoned topic of CSL but now within a fresh context of HPs and robustness, included in Chapter 5.

While this work covers multiple areas, the (unspoken) importance of the findings related to hyperparameters (Chapter 4 and 5) made this aspect the core topic of this thesis. Even though the work in Chapter 3 does not explicitly talk about hyperparameter-related problems, the issues discovered there through practice were the catalyst for further research into hyperparameters that ultimately set the tone for this thesis.

The content of this thesis is organised in the following manner:

Chapter 2 introduces the reader to the background knowledge necessary to appreciate more technical chapters of this work. It gives a summary of the two major application areas this work is concerned about: causal inference and causal discovery. The discussion is further supplemented by a more detailed presentation of types of causal data and associated with them challenges. The topic of hyperparameters, and their methods of search and evaluation, are also included.

Chapter 3 presents our novel idea for improved treatment effect estimation in individualised settings under covariate shifts. It proposes a new data augmentation method based on recently developed generative trees. Based on collected evidence, we show that it increases data complexity and subsequently improves the accuracy of estimated heterogeneous treatment effects of causal estimators trained on the data augmented by our method, as compared to standard causal estimators trained on original data.

Chapter 4 investigates hyperparameters in the causal effect estimation setting motivated by tuning difficulties discovered in the previous chapter. This work performs an extensive set of experiments across multiple datasets, evaluation metrics, causal estimators and their hyperparameters. One of the most surprising outcomes is that many commonly used estimators are able to reach or beat State-of-the-Art (SotA) performance levels if their hyperparameters are well specified.

Chapter 5 builds upon findings on hyperparameters in the previous chapter but here applies learnt lessons into the causal discovery area that was initially explored at the start of the project. The experimental setup consists of many data scenarios, learning algorithms and hyperparameter values, but having in mind previously discovered issues with algorithms' instabilities, this work focuses strongly on robustness and performances under misspecified hyperparameters. The main outcome of this chapter is that algorithms respond differently to incorrectly specified hyperparameters in terms of prediction performance, which is arguably an important property in CSL where evaluation, and hence tuning, is borderline impossible due to missing ground truth.

Chapter 6 consolidates the previous three chapters by revisiting the main discussion points that appeared throughout. The core questions and findings, as well as the original contributions this work offers, are also discussed in more

detail. The overall goal of this chapter is to help readers appreciate each chapter's contributions and how they collectively form this thesis.

Chapter 7 concludes the thesis by summarising the core findings and contributions. A look-ahead into possible future research, divided into opportunities either directly raised by this work or identified in the literature, finalise this report.

2 Background

This chapter covers knowledge foundations that should aid appreciation of technical aspects of this thesis. It spans the main causal estimation tasks of interest, such as causal discovery and inference, but also describes the types of causal data used in forthcoming chapters. The topic of hyperparameters is also discussed due to its central role throughout the totality of this work.

Contents

2.1 Causal Discovery

The gold standard of causal analysis is still considered to be RCTs. However, they are not always feasible due to ethical or financial reasons. Given the vast amounts of data being collected nowadays, there is a lot of effort in the field to make use of such non-experimental data to get useful causal insights. Apart from the inference part, it is often of great interest to automatically extract, or *discover*, the true underlying causal graph from given data. The reasons could be to simply better understand the data at hand, or to incorporate the discovered causal structure in further inference tasks. Causal discovery is a relatively young research area, though highly vibrant in recent years. For a gentle introduction and recent review of the area, the reader is referred to [39] and [40]. More complete approaches to the topic can be found in [41] and [42].

2.1.1 Graphs, Models and Definitions

We start with a brief overview of the most important aspects of causality deemed relevant to the task of inferring causal graphs from data. For a more extensive review, we refer the reader to classic positions on causal analysis [41, 43, 44].

Here, we focus on two major approaches to causal graphs: graphical and functional. With respect to the former, we are specifically interested in Directed Acyclic Graphs (DAGs). Functional models, on the other hand, are often referred to as Structural Causal Models (SCMs). The two approaches are demonstrated in Figure 2.1, which also shows it is possible to capture the same graph with either of the two approaches.

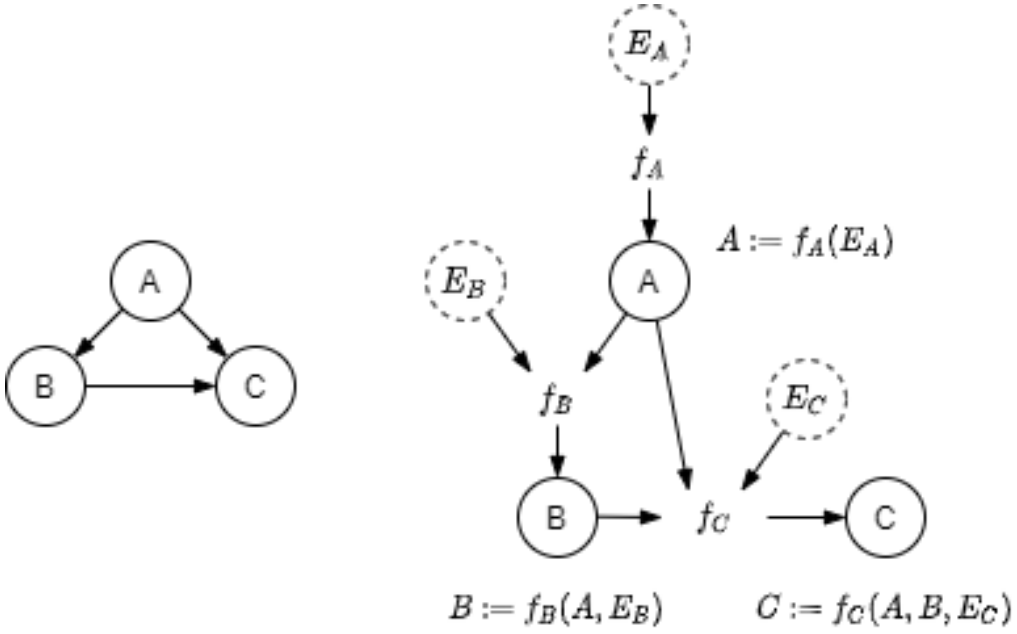


Figure 2.1: An example DAG (left) and its corresponding SCM (right).

Investigating the effects of interventions is an important part of causal analysis. An efficient way of formally expressing such perturbations on a system is Pearl's *do-calculus* [43] (see also Section 1.1), allowing us to write $do(X = x)$ to denote an intervention on variable X. It is also a convenient way of distinguishing between interventions and statistical conditioning. For instance, continuing with the example from Figure 2.1, we can define the distribution for C as $\mathcal{L}_C = \mathcal{L}(C|A, B)$. However, if we intervene on node B, that is, $do(B = b)$, we get $\mathcal{L}_C^{do(B=b)} = \mathcal{L}(C|A, do(B = b))$. This operation is reflected in DAGs by removing all incoming edges to the node we intervene on (no $A \rightarrow B$ link in the example), and by assigning a constant to a variable instead of using its corresponding function to compute it in SCMs ($B := \textit{constant}$ instead of $B := f_B()$). As a consequence, it is clear that $\mathcal{L}_C \neq \mathcal{L}_C^{do(B=b)}$, and by extension $\mathcal{L}_{ABC} \neq \mathcal{L}_{ABC}^{do(B=b)}$. This shows an important difference between observational and interventional distributions and why they must be handled differently in modelling and inference tasks.

D-Separation

A crucial concept in DAGs (and common assumptions defined below) is *d-separation* [43]. For completion, let us define it here as well.

Definition 1 (d-separation)**.** Two graph nodes X and Y are **d-separated** if all **paths** between them are **blocked**.

A **path** is a sequence of graph nodes traversed through the connecting edges but disregarding their direction. For example, (B, A, C) from Figure 2.1 is a path.

A path is **blocked** if two nodes 'meet' in a third node while respecting the direction of the edges. This third node is also called a **collider**. For example, assuming connection $A \to B$ does not exist in Figure 2.1, then C is a collider and all paths between A and B would be blocked, making A and B d-separated.

In addition, a path can be also **blocked** by a conditioning set Z. That is, a path is blocked by Z if it contains a member of Z. For example, assuming connection $B \to C$ does not exist in Figure 2.1, then B and C are d-separated if conditioned on A.

Finally, a path is **not** blocked if the conditioning set includes a collider or its descendants.

2.1.2 Common Assumptions

Causal discovery from observations is an extremely difficult task and without certain prior biases, it is, in fact, an ill-posed problem [41]. Over the years of CSL advances, a set of commonly used assumptions have been established to make the search more feasible. Here we provide the most important ones.

Assumption 1 (Sufficiency)**.** There are no hidden confounders.

Assumption 2 (Markov condition)**.** Two variables are independent in $\mathcal{L}(\mathbf{X})$ if they are *d-separated* in $\mathcal{G}$.

Assumption 3 (Faithfulness)**.** Two variables are *d-separated* in $\mathcal{G}$ if they are independent in $\mathcal{L}(\mathbf{X})$.

Assumption 4 (Acyclicity)**.** Any vertex can be visited only once when traversing the graph through its directed paths. This assumption applies by default when considering DAGs.

It is common to assume that *d-separation* links to conditional independence through *Markov* and *faithfulness* assumptions defined above. This has an important practical implication as it means that all variables are independent of their ancestors when conditioned on their parents. As a result, causal mechanisms are effectively independent of each other as well, subject to the error terms being independent of each other. This specific observation has been formally formulated as the Independent Causal Mechanisms principle [41]. Given this, the entire joint distribution $\mathcal{L}(\mathbf{X})$ can be written as a product of distributions of all variables conditioned on their parents $X_{\mathbf{PA}_j^{\mathcal{G}}}$. In other words:

$$\mathcal{L}(\mathbf{X}) = \prod_j \mathcal{L}(X_j | X_{\mathbf{PA}_j^{\mathcal{G}}}) \tag{2.1}$$

2.1.3 Causal Structure Learning Methods

The goal of CSL is to infer (or identify) graph $\mathcal{G}$ given the distribution $\mathcal{L}(\mathbf{X})$. If it is possible to do this, we say $\mathcal{G}$ is **identifiable** from $\mathcal{L}$.

The four assumptions defined above gave rise to highly effective discovery algorithms, often categorised as constraint-based. For instance, the Peter-Clark (PC) method [12] exploits all four aforementioned assumptions and relies particularly on conditional independence tests. As the tests are part of an exhaustive search, the algorithm does not scale well to high-dimensional data sets, though there are attempts to speed it up through parallel computing [45]. Other ideas involve greedy search in the DAG space [46], which, perhaps unsurprisingly, suffer from the local optimum issue and hence provide suboptimal solutions. As having more assumptions often translates to poorer applicability of a given method to practical problems, some methods try to relax specific constraints. The classic examples

include Fast Causal Inference (FCI) [12] that drops sufficiency or Cyclic Causal Discovery [47] allowing for cycles.

A major limitation of the aforementioned constraint-based methods is the fact they are not able to identify a unique DAG solution. Instead, they provide multiple possible solutions up to a Markov equivalence class. That is, some of the edges of the resulting graph might be undirected. In fact, it has been proved that only a Markov equivalence class can be identified in the case of multinomial distributions [48] or linear SCMs with Gaussian noise [49]. This in turn raises questions about the graph's identifiability if one is willing to diverge from either Gaussianity or linearity. As it turns out, it is possible to converge to a single solution under the assumption that causal mechanisms are linear and the noise terms adhere to a non-Gaussian distribution [50]. This approach, named Linear Non-Gaussian Acyclic Model (LiNGAM), can be formally written as:

$$X_j = f_j(X_{\mathbf{PA}_j^{\mathcal{G}}}) + \epsilon_j, \ \ for\ j = 1, \ldots, d \tag{2.2}$$

Where f_j is a linear function and ϵ_j is a non-Gaussian noise independent of X_j. Similar findings have been reported for nonlinear cases but without the non-Gaussianity constraint [51, 52], and for linear models with Gaussian noise terms of equal variances [53]. Equation (2.2) still applies in the nonlinear case, but f_j is now a nonlinear function and the error term ϵ_j strictly enters additively. Note in this setting we do not assume any specific noise distribution. This Additive Noise Models (ANMs) approach was further extended to post-nonlinear models [54] to allow for more flexibility in terms of how the noise enters the model. Thus, Equation (2.2) transforms into:

$$X_j = g_j(f_j(X_{\mathbf{PA}_j^{\mathcal{G}}}) + \epsilon_j), \ \ for\ j = 1, \ldots, d \tag{2.3}$$

Another interesting perspective on the problem is provided via additive SCMs [55], also termed Causal Additive Models (CAM), which are formally defined as:

$$X_j = \sum_{k \in \mathbf{PA}_j^{\mathcal{G}}} f(X_k) + \epsilon_j \tag{2.4}$$

A recent breakthrough in causal discovery research has created yet another type of algorithms called gradient-based. It has started with [56] and proposed NOTEARS procedure, where the combinatorial search space of DAGs has been reformulated as a continuous optimisation problem, which in turn can be solved via standard optimisers, such as Stochastic Gradient Descent (SGD). The method learns the causal graph in the form of a weighted adjacency matrix, where the weights correspond to coefficients in the linear SCM. This approach was further generalised in [36] to handle nonlinear settings by modelling causal mechanisms separately through function approximators and extracting the adjacency matrix from the models through partial derivatives.

This line of work inspired alternative ways of handling nonlinear cases through continuous optimisation of the weighted adjacency matrix that embeds causal relationships, including graph autoencoders [57, 58] or normalising flow architectures [59]. One of the nonlinear generalisations, called GraNDAG [60], proposes an alternative way of extracting the weighted adjacency matrix from the models, which involves calculating the so-called neural paths between variables that effectively are products of consecutive neural network weights. Furthermore, the work in [61], again building on top of NOTEARS, recognised the fact that in nonlinear settings we are no longer interested in weight values of the adjacency matrices, but rather in the presence of a connection between each pair of variables. Thus, they incorporated the Gumbel-Softmax trick [62] to compute a binary adjacency matrix, where positive entries denote existing connections. This is an important idea as such a binary matrix can be used as a mask in neural networks to control their

input or, in fact, any weights. The masking idea was also explored in [37], but the neural networks were trained in an adversarial manner.

2.1.4 Generative Approach to Causality

The usual approach to causal inference involves fitting a model to carefully selected features based on a given causal graph. The choice of variables is of great importance in order to avoid selection bias and inaccurate causal effect estimations as a consequence. However, even with a perfect set of features, the entire inference process is still limited by the approximators modelling the data. For instance, some model families require vast amounts of data to learn something useful and generalise reasonably. But perhaps the biggest downside of modern learning algorithms is their limited ability to generalise out of the distribution provided in training. As a result, this entire approach to causal inference has rather limited capabilities of handling interventional distributions well enough, a central part of answering causal and counterfactual questions.

There is perhaps an alternative approach to causal inference, offering solutions to some of those issues. A general idea is to model causal relations, that is, mechanisms, among all variables with respect to the underlying causal graph and then generate new data samples from interventional distributions. The choice of modelling each individual causal mechanism instead of fitting the entire joint distribution at once is expected to not only capture complex interactions between variables but also provide the ability to produce new outputs in the form of new data points. Thus, on a high level, this can be seen as a data simulator that respects the underlying causal graph and is able to respond to any interventions. Ultimately, in an ideal case, this framework reconstructs the true DGP with access to its inner workings. When compared to the classic causal inference approach, this framework not only has the ability to produce infinite amounts of data but also allows to reshape distributions in order to investigate various interventions and their effects.

This simulation-based causal inference is a relatively new idea, practically uncharted territory waiting to be explored. For this reason, there is very little existing work on the topic. This direction of thought possibly starts in [63] where the authors mention the generative capabilities of their Causal Generative Neural Network (CGNN) method. According to the design description of the proposed method, it is indeed possible to generate new data by sampling noise terms from assumed distributions and letting the data propagate through the modelled functions with respect to the causal graph. In addition, by substituting a selected causal mechanism with a constant value, it is possible to simulate interventions. Unfortunately, the authors provide no experimental results when it comes to causal inference apart from claiming the method could be used in that way. This is because the primary goal of CGNN is causal discovery. Similar observations can be drawn from [37], which again focuses on CSL, but due to the way its SAM method was designed also allows to control the error terms and ultimately generate new data as a consequence. Unfortunately, the algorithm was not tested against causal inference tasks.

2.1.5 Causal Representation Learning

Most of the methods of causal analysis assume that provided datasets consist of meaningful causal features, similar to data produced by randomised controlled trials. However, it is not difficult to think about counterexamples, where it would be rather meaningless to attempt to learn causal relations or structures directly on the supplied variables, let alone investigate responses to interventions applied directly to them. An intuitive example is images, where we would expect to perform causal analysis not directly on individual pixels, but on high-level concepts or objects visible on the picture instead. In those settings, there is a clear need to learn meaningful causal representations first. In fact, this resembles the problem of manual feature engineering and how deep learning helped to automate the process and created end-to-end approaches capable of learning useful representations on their own [18]. Given those recent successes of representation learning, the idea of combining it

with causal discovery methods to automatically extract high-level causal features is indeed an interesting direction, often referred to as causal representation learning [1].

Despite the early stages of this research avenue, the first solutions have already been proposed. One of the first rigorous investigations dates back to 2013 where Hoel, Albantakis, and Tononi showed that causal analysis based on macro-level features can outperform the one incorporating micro ones only [64]. Further works focused on learning causal mechanisms in the latent space, though without learning the causal structure explicitly [65]. In fact, the method requires prior knowledge about the macro features, known there as targets. Thus, effectively, it learns a mapping between provided high-dimensional data and manually defined target high-level aspects, making it a strongly supervised technique. The work was further extended in [66] to capture multiple levels of causal features, not just micro and macro ones.

More recent efforts have finally started bringing the best of both worlds, that is, causal discovery and representation learning, by utilising the latest CSL optimisation tricks [56], masked layers [61] and generative neural networks in the form of Variational AutoEncoders (VAEs) [67, 68] or Generative Adversarial Networks (GANs) [69]. For instance, CausalVAE [70], learns the weighted adjacency matrix in the latent space, assuming linear relationships among the macro features. It first maps the inputs into their latent representations and then performs CSL, both happening simultaneously during training. A substantial limiting factor of this approach, however, is that it requires prior knowledge about the final causal variables, rendering it a fully supervised method. In fact, almost the same idea was already explored in [65], but without VAEs. Other works tried alternative generators in the form of GANs [71], wherein the proposed method also relies on linear SCMs, but more importantly, again assumes strong supervised signals, this time about the underlying causal graph among high-level features. Lastly, the work in [72] models nonlinear SCMs through multiple neural networks (one model per each latent feature). The reported results are promising as, contrary to previous works, they are not tightly coupled to images, though the experiments involve simulated data only.

2.1.6 Masked Neural Networks

The standard architecture of neural networks assumes that consecutive layers are fully connected, that is, each neuron is connected with all other neurons in neighbouring layers. Apart from the dropout technique where such connections can be disabled randomly during training as a form of regularisation [73], there also exists a type of networks where selected weights can be turned off permanently. This idea was originally introduced in [74] and coined as Masked Autoencoder for Distribution Estimation (MADE), where the connections can be controlled by predefined binary masks. Moreover, the masks in MADE are defined in a very specific way, so the outputs reconstruct the inputs using only allowed variables. As a result, MADE decomposes the joint distribution into individual ones, a product of which again produces the joint one. Note that in the causal context if the masks adhere to the underlying causal graph, so variables depend only on their parents in the SCM sense, the resulting distributions are actually causal mechanisms. However, this idea has not been explored with MADE so far. Figure 2.2 shows an example of MADE and its corresponding decomposed joint distribution in Equation (2.5). They assume an input vector $\mathbf{X} = [X_1, X_2, X_3]$ and approximations of input variables denoted as $\hat{X}_1$, $\hat{X}_2$, and $\hat{X}_3$.

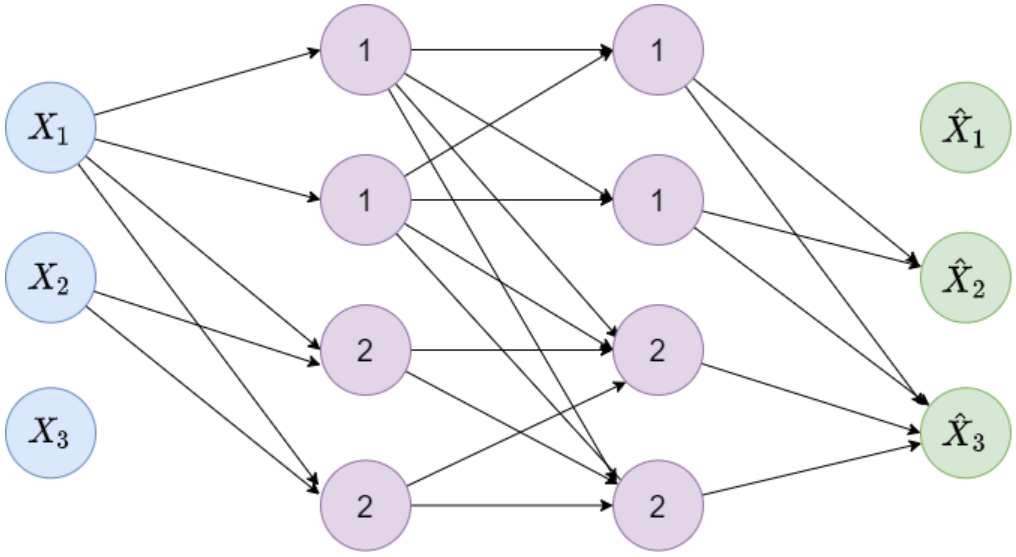


Figure 2.2: An example of MADE architecture with three variables. Note the hidden neurons of group 1 depend on X_1 only, whereas group 2 on both X_1 and X_2. Feature X_3 does not contribute to any other variable. X_1 does not depend on other features.

$$\mathcal{L}(\mathbf{X}) = \mathcal{L}(\hat{X}_1)\mathcal{L}(\hat{X}_2|\hat{X}_1)\mathcal{L}(\hat{X}_3|\hat{X}_1, \hat{X}_2) \tag{2.5}$$

Although MADE has not been fully explored in the causal context yet, the idea of masking certain connections, specifically in the first layer to control the contribution of inputs, has appeared in recent causal discovery methods [37, 61, 72].

2.1.7 Discussion

The last two decades of research have been quite promising for causal discovery, delivering a multitude of interesting methods and deepening our understanding of the problem. Despite tremendous progress, the task remains extremely difficult. Even a few data features pose enough challenges, not to mention real-life projects that involve hundreds of them. Current SotA procedures excel at artificially generated problems but clearly do not transfer those capabilities to real-world applications. Although this is acceptable to some extent due to the lack of maturity of the field, recent research suggests this may be also due to flawed experimental designs that most methodologists follow. For instance, the latest methods do not necessarily find causal relations among variables as it is hoped, but pick up noise patterns instead [75]. Other research shows that simulated DAGs, which are used as testing benchmarks for new methods, can be easily abused in certain ways [76], seriously undermining the validity of many discovery methods proposed.

Our personal experiences with the latest methods confirm those issues. At the current stage, most causal discovery algorithms are very difficult to use with real data as they lack prediction stability and trustworthy tools of detecting prediction mistakes. Another possible reason could be the discrepancy between simulated datasets and real ones. CSL methods proved to work well in simulated environments, which are built from specific assumptions about the DGP that are also incorporated in CSL algorithms. This proves the algorithms solve their intended problems. However, this algorithmic development approach assumes those simulated datasets accurately reflect the properties of real data, which in the end might be a very

strong assumption. If the simulations are indeed very different from real problems, it would explain the difficulties of using those methods in practice. And the fact that we cannot detect which of our assumptions actually hold when dealing with real data only adds complexity to the issue.

CSL models are also often highly inaccurate when it comes to causal effect prediction, again making the CSL part highly questionable as the two subtasks (causal graph and effect learning) should be in line at least to some degree if they truly are to be causal. There is also an ongoing discussion that using observational data alone makes complete causal graph recovery inherently impossible, with suggestions to explore mixtures of experimental and non-experimental data settings or even fully interactive ones that are possible in reinforcement learning setups.

2.2 Causal Inference

In the majority of practical use cases, causal analysis boils down to quantification of the causal (or treatment) effect, leaving the explanatory part in the form of causal graph discovery as an entirely optional, or even redundant, effort. Thus, the methods approximating such effects and relevant literature are often focused exclusively on that single task. Here, we discuss the essential background as well as the most important algorithms concerned with causal effect estimation. Recent surveys provide a more extensive view of the field [77, 78].

2.2.1 The Two Frameworks

A common way of calculating causal effects is the Potential Outcome (PO) framework [11]. More formally, a potential outcome $\mathcal{Y}_t^{(i)}$ is the observed outcome when individual i receives treatment t. In many ways, this approach is equivalent to SCMs, though certainly not the same. In SCMs, instead of focusing on concrete individuals and treatments, we write a more general equation $Y = f_Y(X_{PA_Y}, \epsilon_Y)$, where Y represent the node Y, f_Y its function, and ϵ_Y its noise term. Then, assuming

that treatment variable $T \in PA_Y$ (i.e. T is a parent of Y), we can calculate the outcome for a specific individual and treatment. This example also shows important differences between the frameworks. The PO one focuses specifically on treatment and outcome variables, removing other covariates from the picture, hence not requiring the knowledge of the full causal graph. This approach is particularly convenient in problems concerned with outcome estimation, where developing estimators and formulating the estimation task is of major importance. SCMs, on the other hand, aim to capture the entire graph and the mechanisms within, providing a way of investigating the effects of interventions on any variable. This method is specifically advantageous when modelling the entire DGP is of priority. Thus, both frameworks clearly have their valid use cases, though there has been some scepticism in the literature about the practicality of SCMs and DAGs in the context of estimating causal effects [79].

2.2.2 Common Assumptions

When it comes to causal effect estimation, there are three common assumptions about the DGP that many estimators build upon. As we incorporate them in this work as well, we include their brief description for the convenience of the reader.

Assumption 5 (Stable Unit Treatment Value Assumption (SUTVA))**.** The potential outcomes of any unit (individual) do not affect the treatment assignment of any other unit. Furthermore, there are no different levels or forms of the same treatment. In short: a) no inter-unit interaction, and b) no hidden treatment variations.

Assumption 6 (Ignorability)**.** Given background covariates X, potential outcomes $\mathcal{Y}$ are independent from observed treatment T. That is, $\mathcal{Y}_1, \mathcal{Y}_0 \perp\!\!\!\perp T|X$. In practice, it means that for individuals with the same X their treatment assignment T can be perceived as random because there are no unmeasured hidden variables (confounders), which is why this assumption is often referred to as *unconfoundedness*.

Assumption 7 (Positivity)**.** Treatment assignment T is not deterministic for all individuals X. That is, $P(T = t|X = x) > 0$ for all t and x. Informally, it requires

the existence of potential outcomes for all treatments and individuals (and their combinations). Ignorability and positivity together form *strong ignorability*.

2.2.3 Treatment Effect Estimation

Using Rubin's potential outcome framework [11] (see also Section 2.2.1), which is particularly convenient in outcome estimation without knowing the full causal graph, the problem of estimating causal (treatment) effects can be defined as follows. First, the main three variables of interest are background covariates X, treatment assignment T, and outcome $\mathcal{Y}$. Second, a potential outcome $\mathcal{Y}_t^{(i)}$ is the observed outcome when individual i receives treatment $t \in \{0, 1\}$.

Given this, the Individual Treatment Effect (ITE) is formulated as the difference between the outcomes under treatment ($\mathcal{Y}_1^{(i)}$) and no treatment ($\mathcal{Y}_0^{(i)}$), as in Equation (2.6).

$$\mathrm{ITE_i} = \mathcal{Y}_1^{(\mathrm{i})} - \mathcal{Y}_0^{(\mathrm{i})}. \tag{2.6}$$

Thus, to compute such a value for individual i, we need access to both potential outcomes, $\mathcal{Y}_1^{(i)}$ and $\mathcal{Y}_0^{(i)}$, but only one, called the *factual*, is observed: the other potential outcome, called the *counterfactual*, cannot be observed. The fact that we only observe factuals but also need the counterfactuals to properly compute causal effects is known as the fundamental problem of causal inference: ITEs are not identified by the observed data.

However, parameters such as the Conditional Average Treatment Effect (CATE) and Average Treatment Effect (ATE) are identified, defined in Equations (2.7) and (2.8) respectively.

$$\mathrm{CATE(x)} = \tau(\mathrm{x}) = \mathbb{E}\left[\mathcal{Y}_1 \mid \mathrm{X} = \mathrm{x}\right] - \mathbb{E}\left[\mathcal{Y}_0 \mid \mathrm{X} = \mathrm{x}\right] \tag{2.7}$$

$$\mathrm{ATE} = \mathbb{E}\left[\tau(\mathrm{X})\right] = \mathbb{E}\left[\mathcal{Y}_1 - \mathcal{Y}_0\right], \tag{2.8}$$

where $\mathbb{E}[.]$ denotes (statistical) expectation over the target population. The ATE in Equation (2.8) is essentially the average ITE for the entire population (hence

the expectation operator and no index i); the CATE in Equation (2.7) is the average ITE for everyone in the subpopulation characterised by $X = x$. The ATE is not meaningful if there is substantial heterogeneity of the ITEs between subpopulations. In such circumstances, CATE is more informative about ITEs as it allows the effect to be conditioned on the subpopulation of interest. The ITE can be thought of as a special case of CATE where individual i is the only member of the subpopulation. While ITE_i cannot be identified, CATE for the subpopulation $X = x$ which includes individual i will be a better estimate of it than ATE (under the reasonable assumption that between-subpopulation variation in ITEs is greater than that within subpopulations).

2.2.4 Traditional ATE Estimators

Many estimators are available with which to estimate ATE under the above assumptions. Arguably, the most intuitive approach is regression adjustment involving a single regression learner $\mu(t, x) = \mathbb{E}[\mathcal{Y} \mid T = t, X = x]$. Recognising the need for different modelling complexity per treatment arm, it is natural to extend the approach to two separate learners $\mu_t(x) = \mathbb{E}[\mathcal{Y} \mid T = t, X = x]$ for $t = 0$ and $t = 1$. Single- and two-learner regression adjustment approaches have been formalised as S- and T-Learners respectively (e.g. [32]).

Another method of adjustment is done as part of so-called reweighting methods that seek to transform the observed support of input covariates for the treated and control groups via *propensity scores*. These are simply defined as the unit's probability of receiving the treatment, that is, $e(x_i) = P(t_i|x_i)$. The propensity scores can be further used to create so-called Inverse Propensity Weights (IPW), in which the weight w_i for sample i depends on treatment status t_i and propensity $e(x_i)$, as in Equation (2.9).

$$w_i = \frac{t_i}{e(x_i)} + \frac{1 - t_i}{1 - e(x_i)}. \tag{2.9}$$

The weight in Equation (2.9) can be also perceived as sample importance. The higher the weight, the more impact that sample should have on the estimator. Its

inverse nature refers to the weight and probability of treatment being inversely proportional. Thus, assuming treated units are less numerous as compared to the untreated ones, this approach assigns higher weights to treated samples hence providing a balancing effect.

The inverse weights can be used as sample importance weights in conjunction with the S-Learner, resulting in weighted regression. In traditional causal estimation of average effects however, the weights are more commonly used to create the IPW estimator [80], also known as Inverse Probability of Treatment Weighting (IPTW) estimator, defined in Equation (2.10) to estimate ATEs.

$$\widehat{\text{ATE}}_{IPW} = \frac{1}{n}\sum_{i}^{n}\left[\frac{y_i t_i}{\hat{e}(x_i)} - \frac{y_i(1-t_i)}{1-\hat{e}(x_i)}\right] \tag{2.10}$$

where y_i and t_i are quantities observed in the data.

In practice, both $\mu(x)$ and $e(x)$ can be subject to model misspecification. This led to the development of the Doubly Robust estimators [81] which allow consistent estimation of target parameters (large sample theory where bias tends to zero as the sample size increases) even if one (but not both) of nuisance models $\mu(x)$ and $e(x)$ is misspecified. The Doubly Robust estimator, which builds on IPW estimation from (2.10), is defined as in Equation (2.11).

$$\widehat{\text{ATE}}_{DR} = \frac{1}{n}\sum_{i}^{n}\left[\frac{y_i t_i - (t_i - \hat{e}(x_i))\hat{\mu}(x_i, t_i)}{\hat{e}(x_i)} - \frac{y_i(1-t_i) - (t_i - \hat{e}(x_i))\hat{\mu}(x_i, t_i)}{1-\hat{e}(x_i)}\right] \tag{2.11}$$

where the outcome nuisance model $\mu(x, t)$ depends on background covariates x and treatment t. The Targeted Maximum Likelihood Estimation (TMLE) approach builds on this idea to incorporate ML for the nuisance models $\mu(x)$ and $e(x)$, but notably uses ensemble learning (also called *super-learning*) to estimate them [82].

An alternative improvement is to use propensity scores to balance not only samples but covariates as well [83]. IPW can also result in extremely small scores, making the entire estimation very unstable. One possible solution is trimming, that

is, to eliminate samples with propensity scores lower than a specified threshold [84]. Another way of improving and stabilising propensity score models is post-calibration [85].

2.2.5 CATE Estimators

The literature on CATE estimation is more recent. Doubly Robust and TMLE estimators can be extended to CATE but require users to specify a family of parametric models. Double Machine Learning (DML) [33] introduces a general framework for parametric CATEs, implementation of which can assume linear effects [86] or be applied to manually specified non-linear cases. DML additionally proposes the use of Neyman-orthogonal estimating equations for CATEs. These have the property of limiting the impact of bias from the estimation of the nuisance parameters ($\mu(x)$ and $e(x)$) which can be minimised when combined with cross-fitting[1]. Hence, DML proposes a two-stage estimation. When CATE is linear in X, the first stage involves estimating nuisance functions $\mu(x)$ and $e(x)$; then, the residuals $\mathcal{Y} - \mu(x)$ and $T - e(x)$ are used to minimise the residual on the CATE square loss.

The DML approach above notably assumes parametric form of CATEs. However, more recent work, known as the R-Learner [87], uses the same two-stage procedure, but shows that CATEs can be estimated correctly without assuming any parametric form. This allows to perform the second stage modelling non-parametrically and hence is more general than DML.

All of these approaches (DML and R-Learner) fall into the class of Orthogonal Learning methods [88] due to their dependence on the aforementioned Neyman-orthogonal equations.

A different approach but also involving multi-stage modelling is X-Learner [32]. In its first stage, it behaves as T-Learner and models $\mu_0(x)$ and $\mu_1(x)$ (see Section 2.2.4 above). Second stage involves computing *imputed effects* $\mathcal{D}_0^{(i)} = \hat{\mu}_1(x_0^{(i)}) - y_0^{(i)}$ and

[1] Analogous to cross-validation but merges per-fold predictions to obtain predictions for the entire dataset.

$\mathcal{D}_1^{(i)} = y_1^{(i)} - \hat{\mu}_0(x_1^{(i)})$, followed by their modelling, defined as $\tau_0(x) = \mathbb{E}[\mathcal{D}_0|X = x]$ and $\tau_1(x) = \mathbb{E}[\mathcal{D}_1|X = x]$. The third and final stage is about CATE estimation that is notably weighted by the propensity scores, defined as $\hat{\tau}(x) = \hat{e}(x)\hat{\tau}_0(x) + (1 - \hat{e}(x))\hat{\tau}_1(x)$. This approach is specifically useful for problems where one treatment arm (usually the untreated/control) dominates the other in terms of sample size. X-Learner has also been shown to be a special case of the R-Learner (defined above).

CATE estimators outside Orthogonal Learning have also been explored. In the realm of tree-like algorithms, there is Causal Forest [31] that generalises Decision Trees and Random Forests to CATE estimation. Unlike standard trees which splits are based on, for instance, Mean Squared Error (MSE) or entropy, causal trees are based on CATE heterogeneity, resulting in the tree leaves containing both treated and control samples but assigned to the same heterogeneous group. The within-group difference then enables CATE estimation. Using ensembles of trees, one gets Causal Forests, which offer a unique approach to CATE estimation, though their non-standard design makes hyperparameter tuning more challenging as it renders standard performance metrics (e.g. MSE) unusable[2]. Trees have also been used as a data augmentation tool to address data imbalances via Generative Trees for improved CATE estimation [89].

Neural networks have also been used in CATE estimation, many of which established new SotA performance levels. Some of the more important ideas involve penalising imbalanced representations (between treatment groups) using standard feedforward networks [90], further extended to two-headed structures (one head/output per treatment group) and more sophisticated distribution discrepancy metrics [91] (Maximum Mean Discrepancy and Wasserstein distance), as well as solutions that encourage the preservation of local similarities on top of balanced representations [92]. In particular, the two-headed structure without penalisation, termed TARNet [91], inspired many other approaches, combining it with TMLE [30] or ensemble

[2]Causal Forests do not expose predicted potential outcomes, only causal effects.

learning [93]. Other methods incorporate generative NNs in the form of VAEs [94] and GANs [95].

2.2.6 Modern Estimators

The latest developments show even greater diversity in proposed methodologies and are a testament to continued research interest in the causal estimation problem. Neural architectures are still being explored in this line of work, most recently in the form of *normalising flows* that target CATE distribution modelling instead of expected values, providing uncertainty quantification as a consequence [96]. On another research front, there are continued developments concerned with metalearners. One important example includes the addition of the conformal prediction framework on top of metalearning that enables predictive intervals for ITEs as opposed to CATE point estimates [97]. Conformal learning indeed has been gathering increased interest in causal estimation as it found its way into offline off-policy prediction problems as well [98]. B-Learner constitutes another proposed metalearner that tackles the hidden confounding problem and provides bounds on predicted CATEs [99]. Some other notable work involves forecasting treatment outcomes over time [100] and a comparative study on performance differences between parametric and nonparametric causal effect modelling [101]. Lastly, the DML framework has been notably also extended to panel data [102].

2.2.7 Discussion

Apart from the classic effect estimators, there is growing interest in utilising NNs as well, which indeed have been shown to be capable of handling causal notions [103]. In addition, there is increasing evidence that simple and relatively small NNs can deliver astonishing performance and generalise extremely well [104]. Both of the points make NNs a promising future direction for causality.

As generative NNs are also a part of recent machine learning successes, they are being explored in the treatment estimation setting as well. These architectures are, for example, used to simulate new data sets that resemble real-world distributions

but provide access counterfactuals for benchmarking purposes [105, 106]. Another use case is to use generative networks to generate counterfactuals [107] as such models are known to perform well at filling gaps in data.

As much as NNs can deliver great results, it is important to consider their drawbacks as well, such as explainability or stability of training. These issues may make the alternatives more desirable, such as much more transparent generative trees [108]. To counteract stability issues with NNs, one may consider ensemble learning [109].

When it comes to model capabilities of being truly causal (i.e. following the causal factorisation) and reaching ultimate generalisation, recent research suggests it is all about the balance between under- and over-parametrisation [110]. However, standard cross-validation and hyperparameter search techniques are useless in causal settings (discussed in Chapter 4), so finding such balance is non-trivial. One proposition is to introduce a special kind of regularisation that depends on the strength of confounding in the data [111].

The majority of causal inference literature have been so far focused on measuring effects and their predictions. However, the latest research suggests this is not always in line with the actual application targets when it comes to real world projects [34]. More precisely, the outcome the decision makers are actually after is individual treatment recommendation, not necessarily the effect itself. Although accurate treatment suggestions can be derived from predicted effects, this leads to training instabilities and suboptimal policy learnt in the finite sample regime. One of the latest works in this line of research combines value and policy learning under well-known double robustness, capable of handling multiple treatments as well as continuous ones [35].

2.2.8 Evaluation Metrics

The main focus of utilised performance evaluation metrics is on the quantification of the errors made by provided predictions. Thus, the metrics are often denoted as ϵ_X, where the error ϵ is measured with respect to prediction type X (lower is better).

In terms of treatment outcomes, $\mathcal{Y}_t^{(i)}$ and $\hat{y}_t^{(i)}$ denote true and predicted outcomes respectively for treatment t and individual i. Note that one of the true outcomes (i.e. counterfactual) is never observed thus the metrics below that use both of the true outcomes can be used only in simulated environments where the two outcomes are accessible, which is often the case with causal inference benchmark datasets.

Following the definition of ITE in Equation (2.6), the difference $\mathcal{Y}_1^{(i)} - \mathcal{Y}_0^{(i)}$ gives a true effect, whereas $\hat{y}_1^{(i)} - \hat{y}_0^{(i)}$ a predicted one. Following this, we can define Precision in Estimation of Heterogeneous Effect (PEHE), which is the root mean squared error between predicted and true effects, as given in Equation (2.12).

$$\text{PEHE} = \sqrt{\frac{1}{\text{n}}\sum_{\text{i=1}}^{\text{n}}(\hat{\text{y}}_1^{(\text{i})} - \hat{\text{y}}_0^{(\text{i})} - (\mathcal{Y}_1^{(\text{i})} - \mathcal{Y}_0^{(\text{i})}))^2}. \tag{2.12}$$

Following the definition of ATE in Equation (2.8), we measure the error on predicted ATE as the absolute difference between predicted and true average effects, formally written as in Equation (2.13). Note, instead of expected values used in Equation (2.8), here we switch to sample averages as the data sets used in experiments are of finite size, denoted in Equation (2.13) with n.

$$\epsilon_{ATE} = \left| \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_1^{(i)} - \hat{y}_0^{(i)}) - \frac{1}{n}\sum_{i=1}^{n}(\mathcal{Y}_1^{(i)} - \mathcal{Y}_0^{(i)}) \right|. \tag{2.13}$$

Given a set of treated subjects T that are part of sample E[3] coming from an experimental study, and a set of control group C, we can define the true Average Treatment effect on the Treated (ATT) as per Equation (2.14). It is a difference between the average outcome of the treated units and the average outcome of control units that come from experimental data. Note that $|A|$ denotes the cardinality of a set A and $A \cap B$ is the intersection of sets A and B.

$$\text{ATT} = \frac{1}{|\text{T}|}\sum_{\text{i}\in\text{T}}\mathcal{Y}^{(\text{i})} - \frac{1}{|\text{C}\cap\text{E}|}\sum_{\text{i}\in\text{C}\cap\text{E}}\mathcal{Y}^{(\text{i})}. \tag{2.14}$$

[3]Sample E contains treated and control units, but not all control units come from E. Thus, $C \cap E$ denotes controls specifically from E.

The error on predicted ATT is then defined as the absolute difference between the true and predicted ATT as in Equation (2.15).

$$\epsilon_{ATT} = \left| \text{ATT} - \frac{1}{|\text{T}|} \sum_{\text{i} \in \text{T}} (\hat{\text{y}}_1^{(\text{i})} - \hat{\text{y}}_0^{(\text{i})}) \right|. \tag{2.15}$$

To measure the risk (or regret) of a policy (or treatment assignment) recommendation, let us define policy π that depends on background features x such that $\pi(x) = 1$ if $\hat{y}_1 - \hat{y}_0 > 0$; $\pi(x) = 0$ otherwise. Following such a policy means recommending the treatment to any individual who will benefit (positive effect) from it based on predicted outcomes. The risk of applying such policy, as opposed to following the optimal policy, is defined as policy risk $\mathcal{R}_{pol}$ in Equation (2.16).

$$\mathcal{R}_{pol} = 1 - (\mathbb{E}\left[\mathcal{Y}_1 | \pi(x) = 1\right] \mathcal{P}(\pi(x) = 1) + \mathbb{E}\left[\mathcal{Y}_0 | \pi(x) = 0\right] \mathcal{P}(\pi(x) = 0)), \tag{2.16}$$

with mathematical expectation $\mathbb{E}[.]$ being switched to sample averages on data sets of finite size.

2.2.9 Benchmark Datasets

There is a set of well-established benchmark datasets commonly used in the causal inference literature (see e.g. [90–92, 94]) to evaluate and compare estimation performances. Since they are extensively used throughout this work, we provide brief descriptions for each of the datasets here; see respective references for additional details. These are also summarised in Table 2.1 and openly accessible online [112]. Note we use the same metrics and data splits as in the literature to make our results comparable.

It is worth noting that real-world causal inference observational data sets are naturally "broken" as they inherently suffer from selection biases and covariate shifts, often in the form of distributional differences between treated and untreated units. Thus, in order to test causal estimators in conditions similar to real-world situations, a common practice is to purposefully "break" existing data sets by introducing biases and shifts. This thesis incorporates such datasets as overcoming

said data challenges, and measuring the degree of success via appropriate evaluation metrics, is the goal of this work.

data set	# samples (t/c)	# features	outcome
IHDP	747 (139/608)	25	continuous
JOBS	3,212 (297/2,915)	17	binary
NEWS	5,000 (2,289/2,711)	3,477	continuous
TWINS	11,984 (5,992/5,992)	194	binary

Table 2.1: Summary of incorporated data sets. Letters t/c denote the amount of treated and control samples respectively.

IHDP. Introduced by [113], based on Infant Health Development Program (IHDP) clinical trial [114]. The experiment measured various aspects of premature infants and their mothers, and how receiving specialised childcare affected the cognitive test scores of the infants later on. We use a semi-synthetic version of this data set, where the outcomes are simulated through the NPCI package[4] (setting 'A') based on real pre-treatment covariates. Moreover, the treatment groups are made imbalanced by removing a subset of the treated individuals. We calculate PEHE and ϵ_{ATE} errors (Equations (2.12) and (2.13) respectively) for this dataset. The data consists of 1,000 realisations with recommended training/test splits of 90/10.

JOBS. This data set, proposed by [115], is a combination of the experiment done by [116] as part of the National Supported Work Program (NSWP) and observational data from the Panel Study of Income Dynamics [117]. Overall, the data captures people's basic characteristics, whether they received job training from NSWP (treatment), and their employment status (outcome). We use ϵ_{ATT} (see Equation (2.15)) and $\mathcal{R}_{pol}$ (see Equation (2.16)) metrics for this dataset, averaged over 10 runs with 80/20 training/test ratio splits.

NEWS. Introduced by [90], which consists of news articles in the form of word counts with respect to a predefined vocabulary. The treatment is represented as the device type (mobile or desktop) used to view the article, whereas the simulated

[4]https://github.com/vdorie/npci

outcome is defined as the user's experience. Similarly to IHDP, we calculate PEHE (see Equation (2.12)) and ϵ_{ATE} (see Equation (2.13)) metrics for this dataset. There are 50 realisations within this dataset, with 90/10 training/test ratio splits.

TWINS. The data set comes from official records of twin births in the US in the years 1989-1991 [118]. The data are preprocessed to include only individuals of the same sex and where each of them weighs less than 2,000 grams. The treatment is represented as whether the individual is the heavier one of the twins, whereas the outcome is the mortality within the first year of life. As both factual and counterfactual outcomes are known from the official records, that is, the mortality of both twins, one of the twins is intentionally hidden to simulate an observational setting. Here, we incorporate the approach taken by [94], where new binary features are created and flipped at random (0.33 probability) in order to hide confounding information. We calculate ϵ_{ATE} (as in Equation (2.13)) and PEHE (as in Equation (2.12)) metrics for this dataset, averaged over 10 iterations with 80/20 training/test ratio splits.

2.3 Joint Discovery and Inference

Notably, there has been very little work that would combine both discovery and effect inference parts under a single framework, which is partly understandable due to the immaturity of the causal graph learning methods. One of the first promising attempts involves autoregressive models [119], though they can handle relatively small graphs only. A more scalable and complete approach builds on the latest tricks in causal graph discovery (Gumbel-softmax, sparsity penalty and differentiable acyclicity) and NNs (attention, graph networks) that ultimately delivers quite a powerful method [120]. Good results, however, come here at the cost of the method's high complexity and advanced training techniques that may reduce potential applicability by practitioners.

2.4 Causal Data

Causal data take many forms depending on the application area. In most cases, they are categorised based on their source (experimental/observational) and whether they involve the time component (cross-sectional/longitudinal). In this work, we are specifically interested in **observational cross-sectional** data, mostly due to their prevalence in ML. In the following sections, we discuss the main characteristics as well as challenges that entail such data.

2.4.1 Observational

When it comes to the source of data, experimental ones, formally known as RCTs, are the gold standard for causal effect inference, with the now widely used Neyman's potential outcome framework (see Sections 2.2.1 and 2.2.3) originally developed for effect inference from such randomised experiments [121] (original date 1923, later translated in 1990).

RCTs are considered the gold standard because causal effects are identifiable by design in those cases; data alone is sufficient to obtain the effects. This can be attributed to the treatment assignment being strictly random, which leads to *exchangeability* [122, Chapter 2]. That is, causal effects would have transferred between the treatment groups (treated and control) had we changed treatment assignment to opposite values (i.e. controls become treated and vice versa).

This type of data is presented in Figure 2.3, wherein said randomisation can be noticed as node T not having any node parents, meaning T does not causally depend on any other variable. Less formally, it is often said that randomisation "breaks" any variable dependence T might otherwise have.

Other noteworthy design requirements in this setup, in which the time component plays an important role, though only implicitly, are: a) all background covariates X must be recorded before applying the treatment T, b) no background covariates X after the outcome Y.

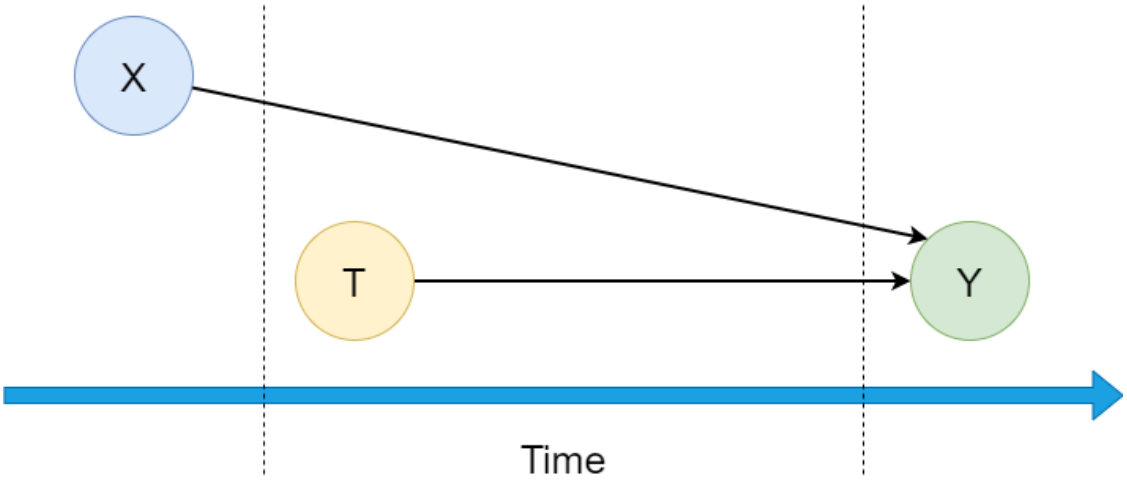


Figure 2.3: A diagram representing experimental data via a causal graph. Nodes: X - background covariates, T - treatment assignment, Y - outcome. Directed arrows denote causal relationships (cause and effect). Note how the variables take specific order in time.

Randomised experiments are favourable due to their design and identifiability of causal effects. However, they are often difficult to obtain due to multiple possible reasons, such as high costs of experiments or ethical aspects (e.g. administering smoking or drinking alcohol as the treatment). Not to mention some experiments are just plain impossible due to physical barriers (e.g. what if London was located 1,000 metres above the sea level?).

These issues are alleviated by much more accessible observational data. The costs are only those relating to data collection, for example via surveys. Ethical concerns are also not as limiting as we can collect data about people who decide themselves whether to smoke or drink and compare them to controls. As for London, moving it vertically is still impossible, but with observational data, we can compare cities across different vertical locations.

Accessibility has certainly been one of the main drivers behind the popularity of observational data, and their widespread use has seen the potential outcome framework being eventually extended to this type of data as well, resulting in what is known today as Neyman-Rubin causal model [11] (see Section 2.2.3 for more details).

However, this accessibility comes at a cost, mostly due to non-randomised treatment

assignment. That is, treatment is no longer assigned to individuals by an external force according to a coin flip. In non-experimental data, subjects make their own decisions about the actions they take (e.g. drink coffee, take aspirin, etc.), which means the treatment assignment T now depends on subject-specific covariates X. This is reflected in Figure 2.4 by the added causal link from X to T.

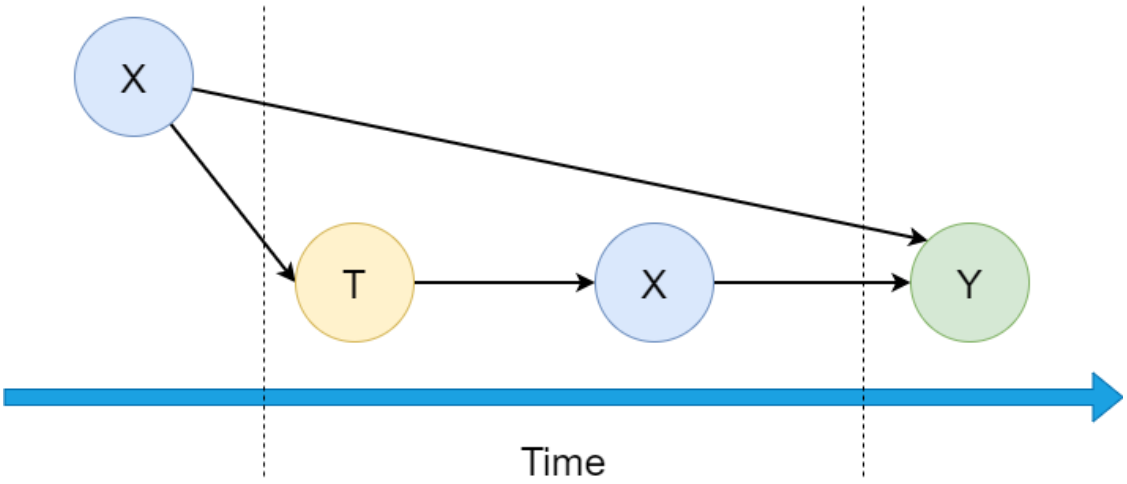


Figure 2.4: A diagram representing observational (non-experimental) data via a causal graph. The layout and notation is similar to that found in Figure 2.3. Note two major discrepancies between the two figures: a) T now depends on X due to non-random selection, b) lack of experimental control may introduce mediators (the X between T and Y).

The added $X \to T$ connection breaks the idealistic design of RCTs. As a result, exchangeability, by default, is no longer valid; causal effects are not identifiable. To make the effects identifiable again, additional assumptions about the data are necessary, which may, or may not, hold in practice (see Section 2.2.2 for common assumptions). Thus, data alone are not enough for effect estimation as it was the case with RCTs; it is data combined with additional assumptions that enable causal inference from observational data [122, Chapter 2].

Bias

Further, lack of T randomisation leads to all sorts of **biases**. Overall, **statistical bias** 'refers to any type of error or distortion that is found with the use of statistical analyses' [123]. Further, there are multiple types of statistical biases to consider, but in this work we are mostly interested in **selection** and **estimator** biases.

As stated by Bareinboim, Tian, and Pearl (2022), 'selection bias is induced by preferential selection of units for data analysis, usually governed by unknown factors including treatment, outcome, and their consequences, and represents a major obstacle to valid causal and statistical inferences' [124].

Whereas estimator bias occurs when the expected value of an estimator differs from the underlying parameter that we want to estimate [125]. In addition, we call an estimator **unbiased** if its estimation bias is zero; and **biased** otherwise.

In the context of causal estimation from observational data, selection bias takes the form of data gaps or underrepresented data regions in one of the treatment groups (e.g. very few or no young smokers). This leads to covariate shifts that in turn results in biased causal estimates. These problems generally entail most of the challenges in effect estimation from observational data, especially with individualised effects as they are sensitive to covariate shifts.

Mixed Data

Interestingly, in methodological research, observational and experimental data both serve their purpose. For instance, there are cases where data from clinical trials [114] are purposefully biased and "broken" to create their observational equivalents [113] in order to test the estimation performance of causal models under conditions that closely resemble real-world problems. Other cases involve mixing observational and RCT data into a single dataset, where the experimental units can be used as quasi-ground truth, which is useful in the absence of the proper ground truth.

2.4.2 Cross-Sectional

It is also useful to group data based on whether they involve the time variable or not, that is, whether each unit is observed more than once over time, possibly at regular time intervals. This is important because the two data types require different inference approaches.

Among the two data types we discuss here, cross-sectional ones involve observing the units only once, as demonstrated in Table 2.2. This characteristic can be either beneficial or limiting, depending on a perspective. On one hand, this form of data is more simplistic and hence easier to deal with as the only possible source of variability is the one across units. In addition, due to this data structure being closely similar to that found in standard supervised ML, a plethora of ML methods is readily available to use in these settings out of the box as a consequence. On the other hand, observing each unit's outcome only for one treatment leaves many alternative scenarios possibly rich in information not explored and hence unavailable for inference, inevitably limiting potential prediction accuracy due to incomplete information provided.

ID	X_1	X_2	X_3	Y
1	1.39	0.99	0	4.77
2	0.26	0.19	1	2.95
3	1.05	1.79	1	4.16
4	0.66	0.19	0	6.17
5	0.85	1.79	1	7.83
⋮	⋮	⋮	⋮	⋮

Table 2.2: Example cross-sectional data. Each unit is observed once (ID) and entails background variables X and outcome Y.

The inclusion of the time component in longitudinal, or panel, data partly alleviates the limitations of cross-sectional data mentioned above, with an example of panel data being provided in Table 2.3. More observations over time per unit translates to possibly more information available within the data that can be exploited for inference and more accurate estimates. However, this benefit comes at a cost since now there are two possible sources of variability within the data: across and within units. As a result, such data requires different, possibly non-standard, inference methods tailored towards time series datasets. This is partly because, while units are mutually independent, the multiple observations per individual unit are not, breaking the IID assumption so fundamental to many ML methods.

ID	**time**	X_1	X_2	X_3	Y
1	1	1.39	0.99	0	4.77
1	2	1.31	0.92	0	4.72
1	3	1.45	1.02	0	4.84
2	1	0.26	0.19	1	2.95
2	2	0.24	0.18	1	2.99
2	3	0.21	0.11	1	2.15
⋮	⋮	⋮	⋮	⋮	⋮

Table 2.3: Example longitudinal (panel) data. The notation is similar to that used in Table 2.2. Note that here each unit (ID) is observed three times (time column). Also, notice how the background variables can change over time within the same unit.

As mentioned above, cross-sectional data closely resembles those found in supervised ML. This similarity made the adoption of ML methods into causal estimation much easier, with now many ML-based approaches setting current SotA in causal estimation. A possibly similar but underexplored relationship is the one between longitudinal data and data seen in offline RL, as they both involve multiple observations over time. Though multiple treatments per unit are not necessarily explored within panel data, unlike *crossover* data [122, Chapter 2]. This direction can possibly unlock a new family of ML-based methods applicable to this causal data type, notably widely used in social and economic sciences.

2.5 Hyperparameters

Hyperparameters have always been part of learning algorithms in one way or another, as they provide a means of adjusting learners to specific data at hand. For instance, HPs can enforce feature sparsity in high-dimensional cases, or encourage simpler hypotheses to avoid overfitting when necessary. In general, they provide a mechanism through which the analyst can find the right balance between bias and variance that is required for model generalisation.

There are three major methodological components the topic of HPs can be divided into: the **search** among available HPs and values, performance **evaluation** of

candidate solutions, and overall **optimisation** process formed by combining the two. These are further discussed in Sections 2.5.2 - 2.5.4. However, before we discuss those details, we first specify the types of HPs we are focusing on in this thesis and draw a line between HP and model selection (Section 2.5.1).

2.5.1 Hyperparameters and Models

HPs can be found across many learning tasks (e.g. supervised, unsupervised, RL) and frameworks (e.g. parametric, semi-parametric, non-parametric). But because they can also be specific to certain ML algorithms, they can take different forms, such as controlling the model's complexity (e.g. *L1, L2, max depth*), or affecting the optimisation process itself via *learning rates* or different variants of *optimisation algorithms* (i.e. versions of gradient descent).

In the context of this thesis, we view HPs as options defined **within the scope of individual ML algorithms**, and the choices that involve those options as HP selection or tuning. Any other modelling decisions that involve selecting among different ML (or causal) algorithms are classified in this work as a more broadly defined **model selection**. More broadly because this work views model selection as more general than HP selection, and which also involves model class choices on top of HP tuning.

To clarify further, we classify as HPs those algorithmic options that:

- Specify model complexity
  - e.g. number of hidden layers/neurons, L1/L2 regularisation, maximum tree depth
- Specify the optimisation algorithm
  - e.g. Stochastic Gradient Descent, Adam, Adagrad
- Control an optimisation algorithm

  - e.g. learning rate, momentum, decay

And the problem of selecting values across those options is termed here as HP selection (or tuning).

Whereas the following are the algorithmic decisions that can be classified here as model selection:

- Selection between different model classes
  - (ML base learners) e.g. Decision Tree, Random Forest, Neural Network
  - (causal estimators) e.g. X-Learner, Causal Forest, Double ML
- Hyperparameter selection (as defined above)

The reason we draw a line between HP and model selection is because in this work we are often interested in analysing performances of individual causal and ML learners as they usually employ different assumptions and showcase different properties. In those cases, we take into account different HPs and values, but defined per individual causal/ML learner. But whenever our interest shifts towards performances across the learning methods (while still tuning HPs; think ensemble learning), then we use the more general term that is model selection.

To complete the picture of what a (candidate) **model** entails in this work, we define it here as a candidate solution (or hypothesis) that consists of a model class and HP values. Following this logic, changing either the model class or HPs to different values results in a different model. For this reason, whenever we evaluate performances of candidate models, we also use the term **model evaluation**.

2.5.2 Search

There are several distinct ways to approach the hyperparameter search. First and foremost, it is possible to disregard the search process entirely by using the default

values recommended by the authors of the learning method. These are often derived as the best setting averaged across different datasets and hence might be suboptimal for the data at hand, which motivates performing the search in the first place. Manually trying different options might be convenient initially due to its simplicity, but is cumbersome and time-consuming in the long term owing to its sequential nature, not to mention one needs to also keep track of already tested combinations that further increase the burden of this approach.

More principled search procedures take away those negative aspects as they automate the search process in many ways. Perhaps the most intuitive approach is the one that performs the search exhaustively according to specified HP candidate values, also known as *grid search*. It specifically suits discrete search spaces as one needs to define all the candidate values that ought to be explored but may leave continuous HPs underexplored. Random search methods [126] notably allow to set the distribution to be explored instead, but the experiments using those might be more difficult to replicate due to their randomness.

The above search methods solve many problems found in manual exploration, but they are not without challenges. The main issue is computational demands that increase with every explored HP candidate while at the same time exploring the HP space as much as possible to increase the chances of identifying optimal HP values. While parallelisation can alleviate this classic exploration-exploitation dilemma [127][5] to some extent, it ultimately motivates more sophisticated search methods. Some of them focus on directly reducing computational requirements, such as "halving" which works as a tournament-like competition of candidate HP values [128]. Other approaches focus on specific estimator types, like neural networks in the case of Neural Architecture Search [129], with some of them building upon evolutionary algorithms [130].

[5] A decision between exploitation of current knowledge and exploration that will potentially enrich said knowledge, with both taking up computational resources.

2.5.3 Evaluation

Performance evaluation is an integral part of HP optimisation since all candidate settings must be tested and compared to each other to identify the best solution. A core part of the evaluation is scoring metrics that allow ranking candidate HPs from best to worst. The choice of metrics may depend on the type of the task at hand. For example, one can consider *mean squared error* for regression, but *accuracy* in classification tasks. Further, multiple metrics are usually available within the same task, such as *precision* and *recall* which are alternatives to accuracy in classification problems. All these choices have a considerable impact on the final HP selection and model behaviour as different metrics can enforce certain desirable outcome criteria by preferring those candidates that perform better in that regard, such as the reduction of false positives or false negatives. As such, there are *no free lunches* in metric selection as they highly depend on the data at hand and desirable modelling criteria.

Accurate evaluation can notably be more challenging in certain situations. Data shifts, for instance, can break the relationship between metric scores seen in training and deployment, resulting in wrongly selected HPs for production and poor generalisation. In other cases like unsupervised learning tasks, hardly any metrics are available for tuning purposes, or in the rare cases where metrics can be used, the ones available for HP tuning do not necessarily reflect the actual target (e.g. goodness of data fit vs. accuracy of predicted causal graphs).

Another aspect affecting evaluation is data splitting and the use of separate validation sets of the data. The basic approach is to split the data into training, validation and test sets, wherein a single validation part is used for evaluation across all candidates. The simplicity of this approach is certainly advantageous, but it can arguably be sensitive to the choice of the validation set that may not fully reflect the characteristics of the entire dataset. Cross-validation [131] addresses this undesirable factor by covering the whole dataset across its folds, though at the cost of increased compute demands due to repeated model training and evaluation per each CV fold.

Performance evaluation is also a connecting factor between HP tuning and model selection (see also 2.5.1) as both involve metrics and evaluation of candidate hypotheses. This shared characteristic enables the exchange of methodologies between the two, though only to a certain extent as not all learners support the same validation metrics, which is precisely what makes model selection arguably even more challenging than HP optimisation.

2.5.4 Optimisation

Hyperparameter optimisation (HPO), also known as *tuning*, brings together previously discussed search and evaluation components, but also introduces new problems, such as the strategy of selecting the final solution. In its simplest and most common form, it entails picking the winning candidate concerning a given evaluation metric. An alternative approach is to use the metric to rank the candidates and then select not one but multiple top performers from which an ensemble estimator can be created to achieve better robustness [109].

The dilemma between exploration and exploitation found in HP optimisation is also well recognised in RL wherein the agent can either exploit the currently learnt policy or search for better ones. As such, HP tuning can be framed as a type of RL, particularly a *multi-armed bandit* problem [128][6]. A similar problem was also investigated in the general ML literature in the context of hyperparameter *tunability* [132], where certain HPs were shown to have a greater impact on prediction performance.

The literature on HPO consistently shows its importance [25] and is still an active area of research [133]. In addition, recent efforts under the name of automated ML (AutoML), among other aspects, attempt to automate the selection of models and HPs, so the users are not required to have expert knowledge of the learning methods to effectively perform HPO [134].

[6] Each bandit arm refers to a different hyperparameter.

Summary

- **Causal discovery**, also known as causal structure learning, is about **recovering the causal graph structure** that underlies the data generating process using only observed data. This task is unsupervised, as the true causal graph is unknown a priori.
- **Causal inference** entails **estimating causal effects** from observed data despite that the effects are not identifiable on the individual level due to fundamentally missing counterfactuals.
- **Causal data** that are of interest in this work can be categorised as **observational** and **cross-sectional**, both of which pose challenges to causal estimation in the form of data shifts and incomplete information within.
- **Hyperparameters** can control many aspects of learning algorithms that adjust the bias-variance trade-off and hence are important in achieving accurate predictions. Their **optimisation** involves two major components: **search** and **performance evaluation**.

3

Undersmoothing Causal Estimators With Generative Trees

Having learnt that current causal discovery algorithms are not ready for applications, this project adjusted its course toward causal inference. This chapter presents the first steps in this direction. The result is a novel data augmentation method that improves the estimation of individualised causal effects under covariate shifts. It is also the first utilisation of generative trees in this type of problem.

Contents

3.1 Introduction

In the absence of data from randomised experiments, analysts must use observational data (see Section 2.4.1) to make inferences about the causal effects of interventions or treatments, that is, what would happen if they intervened to change the treatment status of individual units in a population. The estimation of average causal effects — the average effect of the treatment aggregated across every unit in a population — has been studied in considerable depth (see Section 2.2.4). However, there is now growing interest in estimating heterogeneous treatment effects for individuals characterized by a possibly large number of input variables or covariates (see Section 2.2.5). If there is substantial heterogeneity across units, such systems can unlock the analysis of targeted interventions, for instance, in the form of personalised healthcare based on covariates that describe patients' symptoms and health histories.

The use of observational data creates challenges for the estimation of heterogeneous causal effects. First, the analyst must make assumptions, for example, that treatment selection is strongly ignorable given the available covariates (see Section 2.2.2). We take ignorability to hold throughout, and focus on the second problem, namely, that non-random treatment selection can lead to observed data in which the distributions of covariates among the treated and untreated units are very different. In practice, this can make it difficult for conventional causal estimators to learn the true relationship between the treatment effect and covariates across the entire support of the covariates, and so result in poor performance when tested on other datasets.

More generally, this issue is known as 'covariate shift', which in this setting means the learning target $P(Y|X)$ remains unchanged, while the marginal distributions of the covariate inputs $P(X)$ for treated and untreated can be very different. Most existing methods attempt to transform the observational distribution by sample reweighting schemes usually based on propensity scores [31–33, 80, 81] (but not exclusively, see e.g. domain adaptation methods; see also Section 2.2.4 for propensity scores). However, reweighting seeks to standardise the observed

support of X for the treated and untreated groups, and so generally performs well for estimating treatment effects averaged across the common support of X, but less so for estimating conditional average treatment effects at points outside the observed support; in other words, as pointed out by [135], reweighting does not address the problem of model misspecification which can be detrimental when it comes to estimating individualised treatment effects [136].

A promising alternative to these classical approaches is undersmoothing, where the model is allowed to fit the data very closely to capture $P(X)$ in the two treatment groups, and in doing so potentially produce more accurate individualised predictions. Encouraged by suggestions elsewhere - [137, footnote 3] and [138, 139] - in this work, we develop a novel approach to causal effect estimation that improves accuracy by undersmoothing the observed data.

Specifically, we propose to undersmooth using fast and straightforward generative trees [108] to augment the existing data, and in doing so facilitate more robust learning of downstream estimators of key causal parameters. The trees are used to 'discretise' the input space into subpopulations of similar units (subclassification); the distributions of these groups are then modelled separately via mixtures of Gaussians, from which we sample equally to reduce data biases (see Section 2.4.1).

The concept of model misspecification comes from the world of finite-dimensional parametric models [136] when the analyst uses a parametric model family for prediction that does not include the true prediction rule implied by the DGP. In our context, where no such family is specified, the data augmentation algorithm leads to individualised predictions which can be viewed as coming from an infinite-dimensional model family for statistical functionals that, while not nonparametric, is richer than those induced by existing alternative algorithms. We argue that the implied model family is more likely to include the DGP because data augmentation oversampling the underrepresented data regions is effectively performing targeted undersmoothing and so reduces estimation bias [138] (see also Section 2.4.1). The

practical upshot of this should be that, even with covariate shift, learners trained on data augmented by our method offer more accurate predictions.

Data augmentation is a widely recognised data preprocessing method of improving overall data quality through synthetic sample generation for better prediction performance [140]. It has proven very effective in computer vision [141, 142] which greatly influenced the wider popularisation of data augmentation techniques. Data augmentation constitutes an important part of dealing with imbalanced tabular data [143], specifically by oversampling minority classes in imbalanced classification problems [144–147]. Interestingly, causal notions and ability to simulate interventions has been attributed to successes of data augmentation [148]. This links to a specific type of augmentation that focuses on generating counterfactuals (unobserved outcomes), called counterfactual data augmentation, which found its use in classification problems [149, 150] and reinforcement learning [151]. Other methods focus on text classification [152], or mitigating the effects of confounding [153]. In our case, the method we propose could be seen as oversampling underrepresented data regions instead of just classes or specific outcomes like counterfactuals, making our approach much more general.

Generative models have also been investigated in causal inference literature, mostly in two major strands of work. In one, generative models are used for benchmarking purposes to create new synthetic data sets that closely resemble real data but with access to true, though synthetically generated, effects [105, 106]. The other branch of research is concerned with generating causal effects [94], with more recent works applied to bounding confounded average effects [154], continuous treatments under confounding [155], and longitudinal data [156]. In this work, unlike the two strands above, we use generative modelling for targeted data augmentation.

Arguably the closest work to ours that combines data augmentation and generative models within the causal inference setting is [107]. Despite a similar approach on a high level, that is, training downstream causal estimators on augmented data, we

believe our frameworks differ substantially upon further examination. More precisely, [107] incorporates neural network-based generative models to specifically generate counterfactuals, and focuses on conditions where the treatment is continuous. In this work, our proposed method: a) is based on simple and widely-used decision trees, b) does not specifically generate counterfactuals, but oversamples heterogeneous data regions (more general), and c) works with classic discrete treatments.

In terms of this work's contributions, we show empirically that the choice of model class can have a substantial effect on the estimator's final performance, and that standard reweighting methods can struggle with individual treatment effect estimation. Given our experiments, we also provide evidence that our proposed method increases data complexity[1], reduces estimation bias by training on augmented data (targeted undersmoothing), and leads to statistically significant improvements in individual treatment effect estimation, while keeping the average effect predictions competitive. Our experimental setup incorporates a wide breadth of non-neural standard causal inference methods and data sets. We specifically focus on non-neural solutions as they are more commonly used by practitioners.

The rest of this chapter is structured as follows. First, we revisit fundamental concepts that should aid understanding of the technical part of this work. Next, we formally discuss the problem of model misspecification, followed by a thorough description of our proposed method. We then present our experimental setup and obtained results. The next section provides further discussion of the results and their implications. The final section concludes the chapter, including the considered limitations of the method.

[1]Modelling complexity required to fit the data accurately.

3.2 Preliminaries

Most of the background knowledge necessary for this chapter, including the problem of causal effect estimation and relevant inference methods, can be found in Section 2.2 of Chapter 2. A brief discussion on the topic of covariate shift, which is specifically relevant to this chapter, is provided in the following section.

3.2.1 Covariate Shift

The covariate-shift problem generally occurs when there are distributional discrepancies in input variables X among certain groups of data samples. These differences lead to 'gaps', that is, regions of the common support of X where the observed data are non- or weakly informative about the target parameter. As a result, point estimates of the target (e.g. ITEs or CATEs) in these gaps are inaccurate. Note this issue is not as severe for population-level estimates (e.g. ATEs) because these are averages across the entire support of X and so more robust to gaps. The problem varies depending on the characterisation of the groups of data among which the covariate shift occurs. In ML, this problem is often realised when there are differences between training and test (target) distributions. These can happen due to, for instance, different circumstances between data collection and model deployment. In causal inference, on the other hand, said distributional discrepancies exist between treated and control units. For example, in a study of smoking effects on health, the observational data at hand may include very little to no information about young smokers. Methods of dealing with covariate shift include data adaptation [157], importance sample reweighting (see Section 2.2.4), or causal effect estimation in general (see Section 2.2.5).

Covariate shift falls under a broader category of distribution shifts, in which a second major shift problem occurs due to changes in the conditional distribution $Y|X$ between the target and input covariates. Specific reasons behind this may involve confounding, but also a change in subject behaviour over time [158]. Regardless of the nature of data shifts, they have been shown to have a clear impact on

prediction performance [158, 159], and the fact that real-world data sets often suffer from not one but a mixture of types of shifts only adds importance to this issue. Robustness to data shifts is also discussed from the perspective of out-of-distribution generalisation [160], a topic researched with renewed interest in ML recently partly due to still unresolved issues with model inaccuracies after deployment.

3.3 Model Misspecification

The choice of model class (see Section 2.5.1) occurs at some point in any learning task. Such a decision is made based on available data, usually the training part of it, while the environment of the actual application can be different, a scenario often mimicked via a separate test set. The occurring discrepancies between those two data sets are known as the covariate shift problem. Within causal inference, this manifests as differences between observational and interventional distributions, ultimately making effect estimation extremely difficult. More formally, given input covariates x, treatment t, and outcome y, the conditional distribution $P(y|x,t)$ remains unchanged across the entire data set, whereas marginal distributions $P(x,t)$ differ between observational and interventional data. This is where model misspecification occurs as the model class is selected based on available observations only, which does not generalise well to later predicted interventions.

Let us consider a simple example as presented in Figure 3.1. It consists of a single input feature x, output variable y (both continuous), and binary treatment t. For convenience, let us denote this data set as $\mathcal{D}$. Note the effect is clearly heterogeneous as it differs in $\mathcal{D}(x < 0.5)$ and $\mathcal{D}(x > 0.5)$. Furthermore, the two data regions closer to the top of the figure, that is, $\mathcal{D}(x < 0.5, t = 1)$ and $\mathcal{D}(x > 0.5, t = 0)$, are in minority with respect to the rest of the data. These scarce data points will likely be treated as outliers by many learners, resulting in lower variance than needed to provide accurate estimates. Thus, naively fitting the data will lead to biased estimates, an example of which is depicted on the figure as *Biased T* and

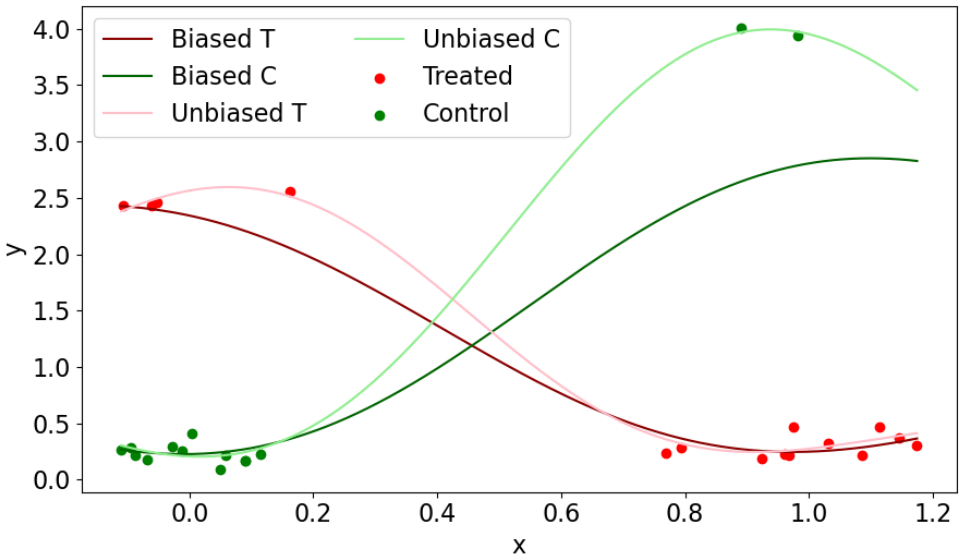


Figure 3.1: An example highlighting model misspecification issue. T and C denote Treated and Control respectively. The difference in ITE error is almost twice as in ATE.

Biased C. However, what we aim for is an unbiased estimator that captures the data closely while still generalising well, a scenario showcased by *Unbiased T* and *Unbiased C* on the figure (see Section 2.4.1 for bias).

For ITE estimation, fitting the data closely is especially important. Although in the case of average effect estimation the difference between biased and unbiased estimators can be negligible, the individualised case usually exacerbates the issue. For instance, in the presented example, the difference in ATE error is 0.44, but it grows to 0.77 in ITE error.

In this work, instead of altering the sample importance, as many existing methods do, we aim to augment provided data in a way that underrepresented data regions are no longer dominated by the rest of the samples, leading to estimators no longer treating those data points as outliers and fitting them more closely, ultimately resulting in less biased solutions, decreased misspecification, and more accurate ITE estimates. The following section describes our proposed method in detail.

3.4 Debiasing Generative Trees

As described in the previous section, model misspecification can be caused by underrepresented or missing data regions. Reweighting partially addresses this problem, but struggles with ITE estimation, not to mention propensity score approximators are subject to misspecification too. To avoid these pitfalls, we tackle the misspecification through undersmoothness by augmenting the original data with new data points that carry useful information and help the final estimators achieve better ITE predictions. As the injected samples are expected to be informative to the learners, the overall data complexity increases as a consequence. Moreover, because this is a data augmentation procedure, it is estimator agnostic, that is, it can be used by any existing estimation method. It is also worth pointing out that simply modelling and oversampling the entire joint distribution would not work as the learnt joint would include any existing data imbalances. In other words, underrepresented data regions would remain in the minority, not addressing the problem at hand.

This observation led us to a conclusion that there is a need to identify smaller data regions, or clusters, and model their distributions in separation instead, giving us control over which areas to sample from and with what ratios. To achieve this, we incorporate recently proposed Generative Trees [108], which retain all the benefits of standard decision trees, such as simplicity, speed and transparency. They can also be easily extended to ensembles of trees, often improving the performance significantly. In practice, a standard decision tree regressor is used to learn the data. Once the tree is constructed, the samples can be assigned to tree leaves according to the learnt decision paths, forming distinct subpopulations that we are after. The distributions of these clusters are then separately modelled through Gaussian Mixture Models (GMMs). Similarly to decision trees, we again prioritise simplicity and ease of use here, which is certainly the case with GMMs. The next step is to sample equally from the modelled distributions, that is, to draw the same amount of new samples per each GMM. In this way, we reduce data imbalances. A merge of new and original data is then provided to a downstream estimator, resulting in

Algorithm 1 Debiasing Generative Trees

Input: D - data set (columns $[X, T, Y]$), E - causal estimator
Parameter: N - number of generated samples
Output: E_D - debiased estimator

1: Let $D_G = \varnothing$.
2: Split D into treated and control units using column T (D_T and D_C).
3: Train a Decision Tree regressor on D_T (e.g. `.fit(X, Y)`).
4: Identify tree leaf index each D_T unit is predicted as (e.g. `.apply(X)`).
5: Common tree leaf indexes form subpopulations S.
6: Let $N_G = N/(2 \times len(S))$.
7: **for** S_i in S **do**
8: Model S_i with Gaussian Mixture Models (e.g. `.fit([X, T, Y])`).
9: Obtain Gaussian Mixture Model G_i.
10: Draw N_G samples from G_i (e.g. `.sample(`N_G`)`). Store them in D_G.
11: **end for**
12: Repeat steps 3-11 for D_C.
13: Merge D and D_G into a single data set D_M.
14: Train estimator E on D_M (e.g. `.fit(X, T, Y)`). Get debiased estimator E_D.
15: **return** debiased estimator E_D

a less biased final estimator. Through experimentation, we find that splitting the original data at the beginning of the process into treated and control units and learning two separate trees for each group helps achieve a better overall effect. A step-by-step description of the proposed procedure is presented in Algorithm 1.

As ensembles of trees almost always improve over simple ones, we incorporate Extremely Randomised Trees for an additional performance gain. The procedure remains the same on a high level, differing only in randomly selecting inner trees at the time of sampling. Overall, we call this approach Debiasing Generative Trees (DeGeTs) as a general framework, with DeGe Decision Trees (DeGeDTs) and DeGe Forests (DeGeFs) for realisations with Decision Trees and Extremely Randomised Trees respectively.

There are a few important hyperparameters (recall Section 2.5.1) to take care of when using the method. Firstly, the depth of trees controls the granularity of identified subpopulations. Smaller clusters may translate to less accurately modelled distributions (too few samples), whereas too shallow trees will bring the modelling

closer to the entire joint distribution which may result in not solving the problem of interest at all. The other tunable knob is the number of new data samples to generate, where more data usually equates to a stronger effect, but also higher noise levels, which must be controlled to avoid destroying meaningful information in the original data. Finally, the number of components in GMMs is worth considering, where more complex distributions may require higher numbers of components.

As our method encourages higher modelling complexity, it is important to consider overfitting, which can be taken care of through the standard practice of tuning the hyperparameters mentioned above and cross-validation. This can be done by using a downstream estimator's performance as a feedback signal as to which parameters work the best, which can also be tailored to a specific estimator of choice. The number of GMM components can be alternatively optimised through Bayesian Information Criterion (BIC) score. In order to make this method as general and easy to use as possible, we instead provide a set of reasonable defaults that we find work well across different data sets and settings. Default parameters: $max\_depth = \lceil \log_2 N_f \rceil - 1$, where N_f denotes the number of input features, $n\_samples = 0.5 \times size(training\_data)$, $n\_components \in [1, 5]$ — pick the one with the lowest BIC score.

In addition, we observe the fact that DeGeTs framework goes beyond applied Generative Trees and GMMs. This is because the data splitting part can, in fact, be performed by other methods, such as clustering. Consequently, GMMs can be substituted by any other generative models.

3.5 Experiments

There are a few aspects we aim to investigate. Firstly, how the established reweighting methods perform in ITE estimation (see Section 2.2.3). Secondly, how the choice of model class impacts estimation accuracy (misspecification). Thirdly,

how our proposed method affects the performance of the base learners, and how it compares to other methods. Finally, we also study how our method influences the number of rules in pruned decision trees as an indirect measure of data complexity.

Although we do perform hyperparameter search (recall Section 2.5 for HPs) to some extent in order to get reasonable results, it is not our goal to achieve the best results possible, hence the parameters used here are likely not optimal and can be improved upon more extensive search. The main reason is the setups presented as part of this work are intended to be as general as possible. This is why in our analysis we specifically focus on the relative difference in performance between settings rather than comparing them to absolute SotA results.

The source code that allows for a full replication of the presented experiments is available online[2] and is based on the *CATE benchmark*[3].

3.5.1 Data and Evaluation Metrics

We follow recent literature (e.g. [90–92]) in terms of incorporated datasets and evaluation metrics. As such, our experimental setting involves the following datasets: IHDP, Jobs, Twins, and News. The metrics used may differ across the datasets. PEHE and ϵ_{ATE} are used with IHDP, Twins and News, whereas ϵ_{ATT} and $\mathcal{R}_{pol}$ find their use with Jobs. See Sections 2.2.8 and 2.2.9 of Chapter 2 for more details about metrics and datasets involved in this set of experiments.

3.5.2 Setup

We incorporate the following estimators.

Base Learners. Linear methods: Lasso (l1) and Ridge (l2). Simple Trees: pruned Decision Trees, Extremely Randomised Trees (ET) [161]. Gradient Boosted Trees: CatBoost [162], LightGBM [163]. Kernel Ridge regression with nonlinearities. Dummy regressor returning the mean as a reference only.

[2] https://github.com/misoc-mml/undersmoothing-data-augmentation

[3] https://github.com/misoc-mml/cate-benchmark

Reweighting Methods. Causal Forest [31], Double Machine Learning (DML) [33], and Meta-Learners [32] in the form of T and X variations. See Sections 2.2.4 and 2.2.5 for a refresher on methods.

Debiasing Generative Trees. Our proposed method (Section 3.4). We include the stronger performing DeGeF variation that involves Extremely Randomised Trees.

A general approach throughout all conducted experiments was to train a method on the training set and evaluate it against appropriate metrics on the test set. 5 base learners were trained and evaluated in that way: l1, l2, Simple Trees, Boosted Trees and Kernel Ridge. DML and Meta-Learners were combined with different base learners as they needed them to solve intermediate regression and classification tasks internally. This resulted in $3 \times 5 = 15$ combinations of distinct estimators. Similarly, DeGeF was combined with the same 5 base learners to investigate how they react to our data augmentation method. Causal Forest and dummy regressor were treated as standalone methods. Overall, we obtained 27 distinct estimators per each data set. In terms of Simple and Boosted Trees, we defaulted to ETs and CatBoost respectively. For NEWS, due to its high dimensionality, we switched to computationally less expensive Decision Trees and LightGBM instead.

As our DeGeF method is a data augmentation approach, it affects only the training set that is later used by base learners. It does not change the test set in any way as the test portion is used specifically for evaluation purposes to test how methods generalise to unseen data examples. More specifically, DeGeF injects new data samples into the existing training set, and that augmented training set is then provided to base learners.

Hyperparameter search was also performed wherever applicable, though not too extensive to keep our study as general and accessible as possible. The following is a list of base learners and their hyperparameters we explored. ETs: $max\_leaf\_nodes \in \{10, 20, 30, None\}$, $max\_depth \in \{5, 10, 20\}$. Kernel Ridge: $alpha \in \{0, 1e-1, 1e{-}2, 1e{-}3\}$, $gamma \in \{1e{-}2, 1e{-}1, 0, 1e{+}1, 1e{+}2\}$, $kernel \in \{rbf, poly\}$, $degree$

$\in \{2, 3, 4\}$. CatBoost: *depth* $\in \{6, 8, 10\}$, *l2_leaf_reg* $\in \{1, 3, 10, 100\}$. LightGBM: *max_depth* $\in \{5, 7, 10\}$, *reg_lambda* $\in \{0, 0.1, 1, 5, 10\}$. Causal Forest: *max_depth* $\in \{5, 10, 20\}$. For ETs, CatBoost, LightGBM and Causal Forest we set the number of inner estimators to 1000. To find the best set of hyperparameters, we performed 5-fold cross-validation. When it comes to DeGeF, we set the number of estimators to 10. The other parameters, like the number of new samples, tree depth and GMM components, were set to defaults as recommended in the description of the framework. All randomisation seeds were set to a fixed number (1) throughout all experiments.

Most of our experimental runs were performed on a Linux-based machine with 12 CPUs and 60 GB of RAM. More demanding settings, such as NEWS combined with tree-based methods, were delegated to one with 96 CPUs and 500 GBs of RAM, though such a powerful machine is not required to complete those runs.

3.5.3 Results

We incorporate the following estimator names throughout the presented tables: **l1** - Lasso, **l2** - Ridge, **kr** - Kernel Ridge, **dt** - Decision Tree, **et** - Extremely Randomised Trees, **cb** - CatBoost, **lgbm** - LightGBM, **cf** - Causal Forest, **dml** - Double Machine Learning, **xl** - X-Learner, **tl** - T-Learner, **degef** - our DeGeF method. Causal estimators can use different base learners to estimate, for instance, the nuisance functions (see Section 2.2.5). This creates multiple possible combinations of causal estimators and base learners. We denote different such combinations with a hyphen, for instance, 'dml-l1'. All presented numbers (excluding relative percentages explained below) denote means and 95% confidence intervals (CIs).

Estimation performance. Tables 3.1 - 3.4 present the main results, where we specifically focus on: a) relevant to a given data set metrics, and b) changes in performance relative to a particular base learner. The latter is calculated as $((r_a - r_b)/r_b) \times 100\%$, where r_a and r_b denote results of advanced methods and base learners respectively. The reason for analysing these relative changes rather than absolute values is because in this study we are specifically interested in how more

complex approaches (including ours) affect the performance of the base learners, even if not reaching SotA results. For example, if a relative change for *xl-et* reads '−20', it means this estimator decreased the error by 20% when compared to plain *et* learner for that particular metric. Changes greater than zero denote an increase in errors (lower is better).

Data complexity. Table 3.5 shows the number of rules obtained from a pruned Decision Tree while trained on original data and augmented by *degef*. The purpose of this experiment is to explain the mechanism through which our method affects the estimation performance of downstream learners. Here, we interpret the number of induced tree rules as a model complexity level required to fit the data accurately. Since our goal is bias reduction and undersmoothing, an increase in model complexity after data augmentation would be desirable. Thus, by measuring model complexity this way we can inspect the existence and strength of such desirable properties. Note that sensitivity to noise and overfitting are not the subjects of interest in this particular experiment.

3.6 Discussion

In terms of IHDP data set (Table 3.1), the classic methods (*dml*, *tl*, and *xl*) strongly improve in ATE, but can also be unstable[4] as it is the case with *dml*, specifically *dml-cb* and *dml-lgbm*. Against PEHE, the situation is much worse as those methods significantly decrease in performance when compared to the base learners, not to mention catastrophic setbacks in the worst cases (deltas above 200%). Note that not a single traditional method improves in PEHE (all deltas positive). Our *degef*, on the other hand, often improves in both ATE and PEHE (see negative deltas). Even in the worst cases with *l1* and *l2*, *degef* is still very stable and does

[4]Unstable in the sense that certain combinations with base learners result in disproportionately weak performances. We regard a causal estimator as 'stable' if all considered base learners combinations for the estimator perform similarly.

name	ϵ_{ATE}	$\Delta\%$	PEHE	$\Delta\%^{a}$
dummy	$4.408 \pm .103$	-	$7.898 \pm .473$	-
cf	$0.397 \pm .045$	-	$3.387 \pm .318$	-
l1	$0.981 \pm .106$	-	$5.790 \pm .514$	-
dml-l1	$0.387 \pm .043$	-60.52	$7.782 \pm .691$	34.42
tl-l1	$0.273 \pm .033$	-72.19	$7.858 \pm .678$	35.73
xl-l1	$0.282 \pm .034$	-71.27	$7.660 \pm .678$	32.31
degef-l1	$1.051 \pm .107$	7.15	$5.809 \pm .514$	0.33
l2	$0.974 \pm .104$	-	$5.786 \pm .514$	-
dml-l2	$0.381 \pm .040$	-60.91	$7.859 \pm .691$	35.82
tl-l2	$0.273 \pm .034$	-72.02	$7.810 \pm .679$	34.99
xl-l2	$0.287 \pm .034$	-70.53	$7.723 \pm .678$	33.47
degef-l2	$1.093 \pm .107$	12.16	$5.820 \pm .514$	0.58
dt	$0.636 \pm .084$	-	$4.025 \pm .402$	-
dml-dt	$1.262 \pm .116$	98.50	$6.679 \pm .570$	65.95
tl-dt	$0.406 \pm .044$	-36.22	$8.012 \pm .698$	99.07
xl-dt	$0.529 \pm .065$	-16.81	$7.317 \pm .653$	81.79
degef-dt	$0.542 \pm .075$	-14.83	$3.882 \pm .384$	-3.55
kr	$0.356 \pm .031$	-	$2.276 \pm .170$	-
dml-kr	$0.616 \pm .059$	73.06	$8.174 \pm .728$	259.16
tl-kr	$0.167 \pm .010$	-53.02	$8.024 \pm .706$	252.60
xl-kr	$0.247 \pm .023$	-30.65	$7.847 \pm .698$	244.82
degef-kr	$0.316 \pm .031$	-11.18	$2.149 \pm .181$	-5.58
et	$0.519 \pm .074$	-	$3.093 \pm .322$	-
dml-et	$0.869 \pm .082$	67.61	$6.532 \pm .563$	111.23
tl-et	$0.306 \pm .042$	-41.01	$7.445 \pm .643$	140.75
xl-et	$0.453 \pm .053$	-12.63	$6.875 \pm .597$	122.32
degef-et	$0.394 \pm .052$	-24.03	$2.818 \pm .273$	-8.89
cb	$0.404 \pm .038$	-	$2.179 \pm .210$	-
dml-cb	$1.123 \pm .052$	177.88	$6.976 \pm .580$	220.18
tl-cb	$0.224 \pm .027$	-44.48	$7.715 \pm .664$	254.10
xl-cb	$0.388 \pm .044$	-3.97	$6.894 \pm .604$	216.42
degef-cb	$0.328 \pm .032$	-18.73	$2.013 \pm .190$	-7.63
lgbm	$0.412 \pm .052$	-	$2.866 \pm .273$	-
dml-lgbm	$1.516 \pm .142$	268.30	$7.544 \pm .632$	163.25
tl-lgbm	$0.255 \pm .028$	-38.10	$8.002 \pm .678$	179.25
xl-lgbm	$0.435 \pm .046$	5.53	$7.602 \pm .650$	165.29
degef-lgbm	$0.397 \pm .051$	-3.54	$2.691 \pm .250$	-6.09

Table 3.1: Results for IHDP data set. Metrics are *mean* $\pm$ 95%CI (lower is better). [a]$\Delta\%$ = change over the baseline (negative means improvement).

name	ϵ_{ATT}	$\Delta\%$	$\mathcal{R}_{pol}$	$\Delta\%^{a}$
dummy	$0.029 \pm .000$	-	$0.326 \pm .000$	-
cf	$0.025 \pm .000$	-	$0.294 \pm .000$	-
l1	$0.005 \pm .000$	-	$0.296 \pm .000$	-
dml-l1	$0.012 \pm .000$	146.75	$0.366 \pm .000$	23.43
tl-l1	$0.012 \pm .000$	140.15	$0.374 \pm .000$	26.10
xl-l1	$0.022 \pm .000$	361.49	$0.356 \pm .000$	20.16
degef-l1	$0.054 \pm .012$	1010.26	$0.296 \pm .000$	0.00
l2	$0.034 \pm .000$	-	$0.296 \pm .000$	-
dml-l2	$0.008 \pm .000$	-77.14	$0.374 \pm .000$	26.20
tl-l2	$0.007 \pm .000$	-79.25	$0.370 \pm .000$	24.75
xl-l2	$0.011 \pm .000$	-67.37	$0.361 \pm .000$	21.91
degef-l2	$0.056 \pm .009$	62.76	$0.296 \pm .000$	0.00
dt	$0.029 \pm .000$	-	$0.365 \pm .000$	-
dml-dt	$0.149 \pm .000$	408.57	$0.336 \pm .000$	-7.80
tl-dt	$0.035 \pm .000$	21.07	$0.351 \pm .000$	-3.68
xl-dt	$0.037 \pm .000$	27.98	$0.296 \pm .000$	-18.75
degef-dt	$0.048 \pm .014$	64.01	$0.335 \pm .015$	-8.12
kr	$0.017 \pm .000$	-	$0.400 \pm .000$	-
dml-kr	$0.007 \pm .000$	-61.39	$0.374 \pm .000$	-6.52
tl-kr	$0.005 \pm .000$	-70.54	$0.305 \pm .000$	-23.81
xl-kr	$0.003 \pm .000$	-80.58	$0.279 \pm .000$	-30.32
degef-kr	$0.019 \pm .012$	11.82	$0.299 \pm .013$	-25.16
et	$0.006 \pm .000$	-	$0.276 \pm .000$	-
dml-et	$0.099 \pm .000$	1686.11	$0.353 \pm .000$	27.66
tl-et	$0.010 \pm .000$	86.50	$0.295 \pm .000$	6.81
xl-et	$0.004 \pm .000$	-36.17	$0.235 \pm .000$	-14.87
degef-et	$0.015 \pm .009$	167.98	$0.270 \pm .014$	-2.24
cb	$0.026 \pm .000$	-	$0.308 \pm .000$	-
dml-cb	$0.010 \pm .000$	-60.23	$0.368 \pm .000$	19.42
tl-cb	$0.026 \pm .000$	-0.50	$0.250 \pm .000$	-18.86
xl-cb	$0.045 \pm .000$	72.93	$0.239 \pm .000$	-22.56
degef-cb	$0.019 \pm .007$	-26.61	$0.257 \pm .030$	-16.51
lgbm	$0.029 \pm .000$	-	$0.247 \pm .000$	-
dml-lgbm	$0.191 \pm .000$	555.20	$0.387 \pm .000$	56.81
tl-lgbm	$0.004 \pm .000$	-86.33	$0.305 \pm .000$	23.62
xl-lgbm	$0.021 \pm .000$	-29.31	$0.297 \pm .000$	20.20
degef-lgbm	$0.021 \pm .007$	-27.62	$0.283 \pm .024$	14.83

Table 3.2: Results for JOBS data set. Metrics are *mean* $\pm$ 95%CI (lower is better). [a]$\Delta\%$ = change over the baseline (negative means improvement).

name	ϵ_{ATE}	$\Delta\%$	PEHE	$\Delta\%^{a}$
dummy	$0.033 \pm .002$	-	$0.318 \pm .004$	-
cf	$0.064 \pm .001$	-	$0.323 \pm .005$	-
l1	$0.042 \pm .000$	-	$0.319 \pm .004$	-
dml-l1	$0.028 \pm .003$	-33.55	$0.318 \pm .004$	-0.29
tl-l1	$0.052 \pm .001$	23.80	$0.324 \pm .005$	1.59
xl-l1	$0.053 \pm .001$	25.46	$0.322 \pm .004$	0.71
degef-l1	$0.064 \pm .004$	53.10	$0.323 \pm .004$	1.18
l2	$0.047 \pm .002$	-	$0.320 \pm .004$	-
dml-l2	$0.042 \pm .001$	-11.32	$0.334 \pm .009$	4.25
tl-l2	$0.042 \pm .000$	-10.47	$0.337 \pm .011$	5.19
xl-l2	$0.042 \pm .001$	-10.95	$0.335 \pm .010$	4.83
degef-l2	$0.067 \pm .004$	41.28	$0.324 \pm .004$	1.10
dt	$0.004 \pm .005$	-	$0.319 \pm .004$	-
dml-dt	$0.070 \pm .011$	1859.14	$0.327 \pm .002$	2.53
tl-dt	$0.062 \pm .000$	1631.81	$0.334 \pm .004$	4.67
xl-dt	$0.059 \pm .000$	1549.54	$0.323 \pm .004$	1.20
degef-dt	$0.064 \pm .013$	1697.62	$0.349 \pm .005$	9.37
kr	$0.045 \pm .001$	-	$0.320 \pm .004$	-
dml-kr	$0.055 \pm .028$	20.87	$0.323 \pm .012$	0.99
tl-kr	$0.050 \pm .000$	9.18	$0.334 \pm .006$	4.45
xl-kr	$0.043 \pm .002$	-4.96	$0.325 \pm .007$	1.73
degef-kr	$0.033 \pm .004$	-27.17	$0.320 \pm .004$	0.15
et	$0.027 \pm .006$	-	$0.322 \pm .003$	-
dml-et	$0.047 \pm .002$	74.36	$0.320 \pm .005$	-0.32
tl-et	$0.051 \pm .000$	87.25	$0.327 \pm .006$	1.76
xl-et	$0.050 \pm .001$	85.14	$0.323 \pm .006$	0.53
degef-et	$0.054 \pm .007$	96.91	$0.335 \pm .002$	4.23
cb	$0.039 \pm .000$	-	$0.319 \pm .004$	-
dml-cb	$0.078 \pm .011$	99.66	$0.328 \pm .002$	2.65
tl-cb	$0.051 \pm .000$	31.77	$0.331 \pm .008$	3.65
xl-cb	$0.048 \pm .002$	22.63	$0.323 \pm .006$	1.04
degef-cb	$0.051 \pm .003$	31.48	$0.326 \pm .004$	2.06
lgbm	$0.038 \pm .000$	-	$0.327 \pm .005$	-
dml-lgbm	$0.034 \pm .007$	-10.56	$0.362 \pm .008$	10.90
tl-lgbm	$0.042 \pm .002$	9.79	$0.393 \pm .009$	20.34
xl-lgbm	$0.039 \pm .002$	2.02	$0.366 \pm .009$	12.18
degef-lgbm	$0.042 \pm .002$	8.61	$0.328 \pm .006$	0.56

Table 3.3: Results for TWINS data set. Metrics are *mean* $\pm$ 95%CI (lower is better). [a]$\Delta\%$ = change over the baseline (negative means improvement).

name	ϵ_{ATE}	$\Delta\%$	PEHE	$\Delta\%^{a}$
dummy	$2.714 \pm .212$	-	$4.381 \pm .361$	-
cf	$0.544 \pm .089$	-	$3.907 \pm .481$	-
l1	$0.244 \pm .068$	-	$3.370 \pm .365$	-
dml-l1	$0.233 \pm .062$	-4.50	$2.469 \pm .269$	-26.73
tl-l1	$0.298 \pm .052$	22.13	$2.166 \pm .201$	-35.74
xl-l1	$0.220 \pm .045$	-9.75	$2.152 \pm .186$	-36.14
degef-l1	$0.225 \pm .048$	-7.86	$3.370 \pm .361$	0.00
l2	$0.260 \pm .068$	-	$3.371 \pm .366$	-
dml-l2	$0.236 \pm .080$	-9.08	$5.108 \pm .394$	51.52
tl-l2	$0.173 \pm .030$	-33.33	$4.182 \pm .343$	24.06
xl-l2	$0.174 \pm .036$	-33.09	$4.162 \pm .345$	23.45
degef-l2	$0.178 \pm .041$	-31.64	$3.366 \pm .362$	-0.16
dt	$0.344 \pm .076$	-	$2.717 \pm .277$	-
dml-dt	$4.523 \pm .783$	1216.23	$5.875 \pm .676$	116.18
tl-dt	$0.329 \pm .062$	-4.12	$2.638 \pm .222$	-2.92
xl-dt	$0.290 \pm .060$	-15.47	$2.639 \pm .263$	-2.87
degef-dt	$0.355 \pm .080$	3.22	$2.727 \pm .266$	0.35
kr	$0.715 \pm .133$	-	$3.316 \pm .367$	-
dml-kr	$2.544 \pm .256$	255.79	$4.186 \pm .399$	26.25
tl-kr	$0.198 \pm .150$	-72.27	$2.677 \pm .290$	-19.26
xl-kr	$0.229 \pm .112$	-68.00	$2.695 \pm .297$	-18.72
degef-kr	$0.582 \pm .102$	-18.61	$3.256 \pm .349$	-1.80
et	$0.276 \pm .051$	-	$2.063 \pm .200$	-
dml-et	x	-	x	-
tl-et	x	-	x	-
xl-et	x	-	x	-
degef-et	$0.290 \pm .052$	5.13	$2.013 \pm .167$	-2.40
cb	$0.127 \pm .029$	-	$1.880 \pm .179$	-
dml-cb	x	-	x	-
tl-cb	x	-	x	-
xl-cb	x	-	x	-
degef-cb	x	-	x	-
lgbm	$0.162 \pm .045$	-	$2.074 \pm .241$	-
dml-lgbm	$1.461 \pm .181$	799.12	$3.240 \pm .386$	56.27
tl-lgbm	$0.161 \pm .033$	-0.81	$1.861 \pm .138$	-10.25
xl-lgbm	$0.131 \pm .042$	-19.41	$2.005 \pm .228$	-3.31
degef-lgbm	$0.151 \pm .042$	-6.78	$2.038 \pm .228$	-1.71

Table 3.4: Results for NEWS data set. Metrics are *mean* $\pm$ 95%CI (lower is better). [a]$\Delta\%$ = change over the baseline (negative means improvement). Estimators marked with 'x' – no results due to unreasonably excessive training time.

not destroy the predictions as it happened with the other approaches. Thus, our method clearly offers the best improvements in PEHE and competitive predictions in ATE while providing a good amount of stability.

In the JOBS data set (Table 3.2), classic methods again achieve strong improvements in average effect estimation (see ATT) in best cases, though they can be substantially worse as well (e.g. *dml-et*). In policy predictions, an equivalent of ITE, traditional techniques are even less likely to provide improvements, except the *X-Learner*. With respect to *degef*, it can also worsen the quality of predictions in ATT, as shown with *degef-l1*, though it does not get as bad as with *dml-et*. However, even in that worst example, policy predictions are not destroyed. The best cases in *degef*, on the other hand, achieve strong improvements in policy. Similarly to IHDP, here *degef* provided solid improvements in ITE predictions (policy) while staying on par with traditional methods in ATT, obtaining reasonable improvements and keeping the worst cases still better than the worst ones in the other methods, proving again its stability.

TWINS data set (Table 3.3), proved to be very difficult for all considered methods when it comes to PEHE, though they did not worsen the predictions as well. Some good improvements in ATE can be observed, but also noticeable decreases in performance in the worst cases (combinations with *dt*). Our method behaves similarly to the classic ones, offering occasional gains and keeping the decreases within reasonable bounds. The stability of *degef* is especially noticeable in PEHE as the worst decrease (*degef-dt*) is still better than in other methods.

The last data set, NEWS (Table 3.4), showed the traditional approaches can provide some improvements in PEHE as well, at least in their best efforts, though performance decreases are also noticeable in the worst ones. They also offer quite stable improvements in ATE, except extremely poor *dml-dt*. The *X-Learner* performs particularly well across both metrics (most deltas negative). Our proposed method offers reasonable gains in ATE as well while keeping performance decreases at bay even in the worst efforts. Even though *degef* provides little improvement

in PEHE, it does not destroy individualised predictions either. Overall, this data set showcases superior stability properties of *degef* particularly well, making it a preferable choice if small but safe performance gains are desirable over potentially higher but riskier improvements.

In general terms, the results show that performance can vary substantially depending on the model class, even within the same advanced method (*dml, xl, degef*). For instance, *DML* proved to work particularly well with *L1* and *L2* as base learners, whereas *X-Learner* often outperforms *T-Learner*, adding more stability to the results as well. Our proposed technique usually offers significant improvements in ITE predictions in the best cases, often better than traditional methods, while keeping the predictions stable even in the worst examples. Classic methods are clearly strong in ATE estimates but can struggle in individualised predictions. Overall, these methods (*dml, xl*) proved to be less stable than ours, where the worst cases can perform quite poorly, especially *dml*. This makes *degef* a safer choice on average when considering various estimators, even more so when achieving the best possible performance is not considered a priority.

The observation that the choice of model class can significantly impact estimation performance opens up a more general question about possible reasons behind said performance differences. We investigated this question closely from the perspective of hyperparameters in another study [164], which offers rather surprising lessons. We have found that hyperparameter selection plays a significant role in this in the sense that, if done optimally (with access to a tuning oracle), the performance differences among individual estimators become small, rendering model selection secondary and suggesting that *free lunches* are possible under the right conditions. A wider implication of this is that causal estimators are generally comparable with respect to *potential* performance (only potential performance because optimal tuning is impossible in practice) and that some of the performance differences we found can be attributed to imperfect model evaluation, which in turn suffers from the same challenges as causal estimation itself – missing counterfactuals and covariate

shifts. As a result, while theoretically, many estimators are capable of similar performance levels, the best one can do in practice is to perform hyperparameter tuning as thoroughly as possible (which we do in our experiments here) to reduce the influence of model evaluation imperfections.

We also investigate the number of rules in pruned Decision Trees as a proxy for data complexity and required model complexity to accurately fit the data. As presented in Table 3.5, *degef* significantly increases the number of rules across all data sets, translating to an increase in data complexity. This proves that augmented data encourages richer model families that are more likely to include the true DGP, subsequently leading to reduced bias, undersmoothing, and decreased misspecification. In addition, we observe that modest data complexity increases in IHDP and JOBS correlate with strong *degef* gains in ITE estimation in those two data sets, whereas a much bigger difference in TWINS (from 9.6 to 59.1) correlated with considerably lower prediction performance gains (Table 3.3). This suggests there is a practical limit to increased data complexity beyond which performance benefits decrease.

After combining all the results, we can observe that *degef*: a) improves effect predictions (Tables 3.1 - 3.4), and b) increases data complexity (Table 3.5). Both points essentially demonstrate the positive effects our method has on prediction performance (point (a)) and specific mechanisms enabling such benefits (point (b)). More specifically, *degef* encourages higher modelling complexity (undersmoothing) through increased complexity of the augmented data (point (b)). This reduces bias and misspecification, which in practical terms improves prediction performance, even under covariate shift (as per point (a)). In terms of theoretical guarantees, we rely on [136] and [138], which provide a thorough formal analysis of the problem of model misspecification and undersmoothing respectively.

data set	original data	augmented data	$\Delta\%^{a}$
IHDP	33.6 ± 2.0	53.3 ± 2.6	58.63
JOBS	6.0 ± 0.0	11.3 ± 5.3	88.33
TWINS	9.6 ± 0.9	59.1 ± 11.9	515.63
NEWS	19.4 ± 2.5	32.0 ± 4.7	64.95

Table 3.5: Number of rules in a pruned Decision Tree with and without *degef* augmentation. Numbers are *mean* $\pm$ 95%CI. $^{a}\Delta\%$ = relative change in number of rules. We interpret the number of tree rules as a proxy for model complexity required to fit the data (more rules, more complex model), which is an indirect proof for increased data complexity as a result of *degef* data augmentation ($\Delta\%$ column), confirming the desired undersmoothing effect and reduced model misspecification has been achieved successfully.

3.7 Conclusion

In this work, we proposed Debiasing Generative Trees, a novel data augmentation method based on generative trees for improved estimation of heterogeneous causal effects. Data augmented by *DeGeTs* through oversampling underrepresented data regions reduces bias and undersmooths causal estimators trained on the data. Higher modelling complexity of downstream learners achieved this way enriches the model family that is more likely to include the true DGP and hence reduces model misspecification. In practice, this results in more accurate predictions, even with covariate shift, especially in individualised estimation where the consequences of misspecification are exacerbated.

Our key finding is that our proposed approach offers significantly better performance improvements in individual effect estimation, as compared to traditional reweighting procedures while staying competitive on average effect tasks. Our method also exhibits better stability in terms of provided gains than other approaches, rendering it a safer option overall. Furthermore, we show through our experiments that the choice of model class can significantly affect achieved performance, and that reweighting methods can struggle on individualised estimation tasks. This links with our recent research on hyperparameters suggesting that tuning alone can be a source of major differences between performances [164]. Note, however, that

hyperparameter optimisation is highly non-trivial in causal settings as it suffers from the same challenges as causal estimation itself.

3.7.1 Limitations

In terms of possible limitations of our method, we assume the data sets we work with have relatively low noise levels. This is because in noisy environments, the inner GMMs would likely pick up a lot of noise and thus sampling from them would result in even more noisy data samples. The result would be the opposite of what we aim for, that is, to increase data complexity and bring new informative samples, not to introduce bias in the form of noise. Thus, our method would likely worsen base learners' performance in such environments. Furthermore, we expect extremely high-dimensional data sets may cause computational issues due to the increasing depth of the inner trees. This is partly why setting a reasonable depth limit is important. Our proposed method is also subject to the standard set of assumptions (SUTVA and strong ignorability; see Section 2.2.2). Thus, scenarios that violate those are outside the applicability of the method.

3.7.2 Future Work

In terms of possible future directions, it might be interesting to investigate the feasibility of replacing generative trees with neural networks to handle extremely high-dimensional problems. Another direction would be to instantiate *DeGeTs* framework with alternative methods, such as standard clustering and generative neural networks. Furthermore, extending our approach to data sets with a high degree of noise could increase its applicability to a wider set of real-world tasks. In addition, an in-depth theoretical analysis of specific mechanisms behind the increased estimation robustness of the proposed method may further explain its effectiveness.

Summary

- A novel data augmentation method based on Generative Trees proved to successfully achieve the undersmoothing effect, as a result improving the accuracy of downstream causal effect estimators.
- The proposed data augmentation approach is competitive with standard methods on average treatment effects while performing significantly better on individualised treatment effects.
- The choice of model class can significantly affect the final effect estimation performance.
- Standard reweighting methods can struggle in individualised effect estimation.

4

Hyperparameter Tuning and Model Evaluation in Causal Effect Estimation

Facing surprisingly non-trivial hyperparameter optimisation challenges as part of the previous chapter motivated a thorough investigation of the issue. This is the central topic of this upcoming chapter, which studies the complex interplay of model class selection, tuning and metrics, and their impact on causal effect estimation performance.

Contents

4.1 Introduction

Modern ML methods are remarkably flexible and hence perfectly suited to causal effect estimation, either as subcomponents of an overall causal inference procedure (e.g. [32, 33]) or as an "algorithmic substrate" upon which further developments take place (e.g. [31, 91]). However, the hyperparameters of ML learners must be highly tuned to the dataset at hand to avoid large estimation errors [25]. In practice, hyperparameter tuning heavily relies on accurate model evaluation (recall Section 2.5.1), but this task is more challenging for causal than non-causal ML for the fundamental reason that causal parameters (e.g. ATE, CATE) depend on unobservable counterfactuals.

Specifically, a key practice in ML is to follow a model selection procedure that reliably identifies the best solutions across a rich space of candidate models and hyperparameters, each of which is evaluated using performance-validation metrics and cross-validation. This task is especially difficult in causal effect estimation because causal parameters are functions of the difference in 'potential outcomes' $\mathcal{Y}_1 - \mathcal{Y}_0$, where $\mathcal{Y}_1$ represents individuals' outcomes under the treatment and $\mathcal{Y}_0$ represents those outcomes for the same individuals had they instead not received the treatment: hence, one potential outcome is always 'counterfactual' and unobservable [165] (see also Section 2.2.3).

For model evaluation, this means it is only possible to measure prediction error compared to the observed outcomes $Y = (1 - T)\mathcal{Y}_0 + T\mathcal{Y}_1$, where $T = 1$ for individuals who are treated and $T = 0$ for individuals who are not. The prediction error is then assessed using e.g. the *observable* Mean Squared Error (oMSE) based on Y. Ideally, we would measure prediction error directly on $\mathcal{Y}_1 - \mathcal{Y}_0$, but the corresponding *potential* Mean Squared Error (pMSE) is inaccessible (see Section 2.2.8 for such metrics). Were the observed data balanced between treated and untreated units across the entire support of our pre-treatment covariates X, as in

the case for data from RCTs, we could expect oMSE to be an accurate proxy for pMSE. Or, to put it differently, this idealised setup would make it reasonable to assume that the models with the best observed data fit (lowest oMSE) also have the best potential data fit (lowest pMSE). We demonstrate the problem between oMSE and pMSE with examples in Section 4.3.

Observational data, however, does not come from RCTs because treatment selection is non-random and assumed to depend on X. This leads to a domain adaptation, or covariate-shift, problem, as the distribution of X among treated and untreated subjects can be very different [89] (see also Section 2.4.1). As a result, oMSE can be a poor approximation of pMSE, as illustrated in Section 4.3. Because hyperparameter tuning is so crucial for ML methods, bad oMSE can lead to badly tuned ML models and, ultimately, to causal estimators with poor pMSE [166, 167].

To the best of our knowledge, this is the first study to investigate rigorously the influence of hyperparameter tuning on causal estimator performance. In order to achieve this, we look at a) the impact of the quality of hyperparameters on causal estimation performance, and b) the effectiveness of observable metrics (oMSE) at approximating ideal but inaccessible target metrics (pMSE) in the task of model selection and tuning. The role of hyperparameters in causal estimation is understudied. Past research investigated the impact of broader model selection on causal performance, that is, broader in the sense that both models and hyperparameters are part of a single selection search space, making it impossible to analyse the impact of only hyperparameters on performance (e.g. [168, 169]). At the same time, many papers proposing new causal estimators employ different hyperparameter tuning strategies, which makes it difficult to compare the performance of different estimators in different studies and makes it difficult to assess the influence of tuning on the final performance.

In terms of the effectiveness of observable metrics, other studies have already acknowledged that oMSE can be a poor proxy for pMSE. Some propose new

metrics to alleviate the issue [87, 166, 170–172], whereas others systematically review currently available oMSE metrics [168, 169]. However, none of these studies explores the potential performance achievable using pMSE metrics, which is crucial to understanding the impact on practice, because counterfactuals are by definition unobserved. We overcome this problem by leveraging causal inference benchmark datasets, namely, IHDP, Jobs, Twins and News (Section 2.2.9).

These benchmark datasets are designed specifically to test the performance of causal estimators and hence include the ground truth in the form of either counterfactuals or causal effects, making metrics like pMSE available to us. However, note that the training data sets used herein comprise only observational data (i.e., no counterfactuals or causal effects) such that the ground truth and pMSE metrics are used only for performance assessment purposes and not for model selection/tuning.

Our interest in model selection and evaluation metrics combined with the empirical nature of this study, create a similar context to that found in two recent papers. [168] compare the effectiveness of multiple observable metrics in model selection using simulated data. [169] perform a similar analysis but attempt to make their datasets more realistic through generative modelling [106], a concept previously proposed by [173]. Both are useful for identifying the best model selection metrics and for observing how model selection affects estimation performance. Our work differs in two important aspects. First, in some of our experiments we limit our model selection search spaces to hyperparameters only in order to study their exclusive impact on individual causal estimators and their estimation performance (see a) above). Second, we include ideal pMSE metrics in model selection to obtain potential performances. This allows us to better understand the magnitude of the consequences of oMSE inaccurately approximating pMSE (important for hyperparameter tuning), and explore potential performances of causal estimators given the right hyperparameters (see b) above).

The rest of this chapter is structured as follows. Section 4.2 gives a brief background overview on the topic of model selection and evaluation, followed by Section 4.3 discussing the challenges of model evaluation in the causal effect estimation setting. The proposed methodology is described in Section 4.4. Obtained results are presented and discussed in Section 4.5, with supplementary results also presented in Appendix A.1. Section 4.6 concludes the chapter.

4.2 Preliminaries

There are two major topics fundamental to this work – the problems of causal effect estimation and causal model selection. A thorough discussion on the former can be found in Section 2.2 of Chapter 2 due to its wider use in this thesis. Model selection, evaluation, and the specific challenges of the two posed by causal estimation, are described further in this section. For a more comprehensive treatment of the subject of model selection in causal settings, please refer to [168, 170].

4.2.1 Model Selection

No single model is universally better than others in all situations (*no free lunch* theorem). Thus, a model most suitable to the task at hand must be selected every time a new task is encountered, either manually by employing prior knowledge about the problem or in a data-driven manner, the latter being the focus of this work. More formally, the task is to select a model $\mathcal{M}_{m^*}$ from a collection of candidate models $\mathcal{M} = \{\mathcal{M}_1, ..., \mathcal{M}_M\}$ that minimises incurred loss $\mathcal{L}$ on validation data $\mathcal{D}_{val}$ after being trained on training data $\mathcal{D}_{tr}$. To simplify the notation, let us also introduce a model $\mathcal{M}_m$ trained on training data $\mathcal{D}_{tr}$ as a predictive model $\mathcal{P}_m$. We use $\mathcal{P}_m$ to connect $\mathcal{M}_m$ specifically with $\mathcal{D}_{tr}$. Putting both equations together, we have

$$\mathcal{P}_m = \mathcal{P}(\mathcal{M}_m, \mathcal{D}_{tr}) \tag{4.1}$$

$$m^* = \underset{m=1,\dots,M}{\operatorname{argmin}} \mathcal{L}(\mathcal{P}_m, \mathcal{D}_{val}) \tag{4.2}$$

To reduce the clutter, let us refer to winning models $\mathcal{M}_{m^*}$ and $\mathcal{P}_{m^*}$ as simply $\mathcal{M}^*$ and $\mathcal{P}^*$ respectively. The goal could also be adjusted to maximise a scoring criterion, instead of minimising a loss function. Note, candidate models $\mathcal{M}$ may include different CATE estimators (e.g. X-Learner, Doubly Robust), base learners (e.g. Linear Regression, Decision Tree) and hyperparameters (e.g. regularisation strength, hidden layers). Therefore, $\mathcal{M}$ can be defined as a tuple $\{\mathcal{C}, (\mathcal{B}, \mathcal{H})\}$ that consists of causal estimators $\mathcal{C}$ and a pair of base learners and hyperparameter values $(\mathcal{B}, \mathcal{H})$, with $c \in \Omega_{\mathcal{C}}$ and $(b, h) \in \Omega_{\mathcal{B},\mathcal{H}}$. $\mathcal{B}$ and $\mathcal{H}$ must be a pair as not all base learner-hyperparameter combinations are meaningful. Thus, it is clear that changing any of the three will result in a new candidate model. An *S-Learner* with *Linear Regression* as its base learner and a hyperparameter $L1 = 0.1$ could be one possible example of a candidate model, say $\mathcal{M}_1$. Then, using the same CATE estimator and base learner, but changing the hyperparameter to $L1 = 0.5$ now results in a different model $\mathcal{M}_2$. This is because, even though both models have the same $\mathcal{C}_c$ and $\mathcal{B}_b$, they differ in $\mathcal{H}_h$, hence they are different candidate models ($\mathcal{M}_1 \neq \mathcal{M}_2$). Rewriting the problem to include all three search spaces explicitly results in

$$m^* = \{c^*, (b^*, h^*)\} = \underset{\substack{c=1,\dots,C \\ b=1,\dots,B \\ h=1,\dots,H}}{\operatorname{argmin}} \mathcal{L}(\{\mathcal{C}_c, (\mathcal{B}_b, \mathcal{H}_h)\}, \mathcal{D}_{val}) \tag{4.3}$$

Note that many causal estimators allow for multiple base learners to be defined, so a single $\mathcal{B}_b$ may contain a tuple of multiple learners. Similarly, many base learners have multiple tunable hyperparameters, hence it is acceptable for a single $\mathcal{H}_h$ to be defined as a set of multiple hyperparameters. The loss $\mathcal{L}$ is commonly obtained on a single validation set or through CV, where the final loss is an average of losses obtained on each validation fold. For this reason, $\mathcal{D}_{tr} \cap \mathcal{D}_{val}$ is not necessarily $\varnothing$ (i.e. empty) in the case that CV is used. Following Equation (4.2), an example with *MSE* as an evaluation metric would look like:

$$m^*_{MSE} = \underset{m=1,\dots,M}{\operatorname{argmin}} MSE(\mathcal{P}_m, \mathcal{D}_{val}) \tag{4.4}$$

4.2.2 Evaluation of Model Selection

In addition to model evaluation for model selection purposes, assessing the effectiveness of an evaluation metric at the task might also be desirable, especially when comparing multiple metrics. In order to avoid overfitting, the evaluation of model selection metrics should be performed on a separate set of data than model selection itself. For this purpose, let us define validation and test data, $\mathcal{D}_{val}$ and $\mathcal{D}_{te}$ respectively, as well as separate categories of validation and test loss functions

$$\mathcal{L}^m_{val} = \mathcal{L}_{val}(\mathcal{P}_m, \mathcal{D}_{val}) \tag{4.5}$$

$$\mathcal{L}^m_{te} = \mathcal{L}_{te}(\mathcal{P}_m, \mathcal{D}_{te}) \tag{4.6}$$

Note that $(\mathcal{D}_{tr} \cup \mathcal{D}_{val}) \cap \mathcal{D}_{te} = \varnothing$ is always true, even in the case of using CV. Further, we define the performance of a model selection metric $\mathcal{L}_{val}$ as the performance of a model $\mathcal{P}^*_{val}$ selected with $\mathcal{L}_{val}$, but evaluated against the test set with a test metric $\mathcal{L}_{te}$

$$\mathcal{L}^*_{te} = \mathcal{L}_{te}(\mathcal{P}^*_{val}, \mathcal{D}_{te}) \tag{4.7}$$

Continuing the example in Equation (4.4), the model $\mathcal{P}^*_{MSE}$ selected with *MSE* validation metric can now be evaluated via a test metric *PEHE* (see Equation (4.18) in Section 4.3). The value obtained is an assessment of *MSE*'s performance with respect to *PEHE*.

$$\mathcal{L}^*_{PEHE} = PEHE(\mathcal{P}^*_{MSE}, \mathcal{D}_{te}) \tag{4.8}$$

Therefore, the ultimate goal becomes to optimise for the test loss indirectly through model selection performed on validation data. More generally, when considering multiple validation loss functions $\mathcal{L} = \{\mathcal{L}_1, ..., \mathcal{L}_V\}$, and a single test loss function $\mathcal{L}_{te}$ of choice, the task can be defined as a nested optimisation problem, wherein the goal is to select a validation loss function $\mathcal{L}_{v^*}$ that, through its model selection, optimises the test loss the best. That is

$$v^* = \underset{v=1,...,V}{\operatorname{argmin}} \mathcal{L}_{te}(\mathcal{P}^*_v, \mathcal{D}_{te}) \tag{4.9}$$

Acknowledging $\mathcal{L}_{te}$ as the desired ultimate goal, as shown by Equations (4.7) and (4.9), creates the need to incorporate as much information about $\mathcal{L}_{te}$ into $\mathcal{L}_{val}$ as possible. In practice, however, $\mathcal{L}_{te}$ may require access to the type of data unavailable to $\mathcal{L}_{val}$ (e.g. counterfactuals in *PEHE*), making the task extremely challenging. This important realisation of targeting $\mathcal{L}_{te}$ through available data is foundational to *targeted learning* estimators [82] and evaluation metrics [171].

Oracle

The usual practice is to perform model selection on validation data, or a validation task. However, to evaluate model selection metrics, it is also useful to consider model selection on test data directly via test loss functions, completely bypassing validation metrics that could on their own be the source of model selection bias. This is defined as:

$$m^{**} = \underset{m=1,\dots,M}{\operatorname{argmin}} \mathcal{L}_{te}(\mathcal{P}_m, \mathcal{D}_{te}) \tag{4.10}$$

Model $\mathcal{M}_{m^{**}}$, $\mathcal{M}^{**}$ for short, thus is a candidate model that achieves the best test loss $\mathcal{L}_{te}$ among all candidate models $\mathcal{M}$. Note that $\mathcal{M}^{**}$ may differ across different test metrics $\mathcal{L}_{te}$. Furthermore, its loss $\mathcal{L}_{te}^{**}$ with respect to the same test loss function $\mathcal{L}_{te}$ is the best possible among the defined collection of candidate models.

$$\mathcal{L}_{te}^{**} = \mathcal{L}_{te}(\mathcal{P}_{te}^{**}, \mathcal{D}_{te}) \tag{4.11}$$

Note, in some cases, test data $\mathcal{D}_{te}$ can contain ground truth information the validation set does not have, such us true causal effects. Thus, in some sense, $\mathcal{M}^{**}$ can be perceived as the optimal model choice, at least concerning the chosen test metric $\mathcal{L}_{te}$ and available test set. Due to this relative optimality, we refer to Equation (4.10) as *Oracle* model selection, and the loss $\mathcal{L}_{te}^{**}$ as *Oracle* performance. An Oracle defined in such terms would arguably be a *weak* one due to it being derived from sample, not population, data. However, assuming the sample test set $\mathcal{D}_{te}$ closely reflects population data, our weak Oracle is expected to approximate

the true Oracle reasonably well. Thus, for simplicity, we refer to the weak Oracle as just *Oracle* throughout the rest of the text.

An example of *Oracle* performance with respect to test metric *PEHE* would be:

$$\mathcal{L}^{**}_{PEHE} = PEHE(\mathcal{P}^{**}_{PEHE}, \mathcal{D}_{te}) \tag{4.12}$$

Regret

Apart from *Oracle* performance being useful in order to show what performance levels a model can achieve when detached from a possibly biased validation metric, it is also useful in the effectiveness assessment of any validation metric. This is because accurately estimating the test performance $\mathcal{L}_{te}$ is the ultimate goal, thus such an assessment can reveal how far a test loss achieved through a validation metric is from the aforementioned optimal test loss. That is:

$$Regret = |\mathcal{L}_{te}(\mathcal{P}^{**}_{te}, \mathcal{D}_{te}) - \mathcal{L}_{te}(\mathcal{P}^{*}_{val}, \mathcal{D}_{te})| \tag{4.13}$$

Which can be interpreted as a regret of choosing a validation metric $\mathcal{L}_{val}$ that selected model $\mathcal{P}^{*}_{val}$ with respect to test metric $\mathcal{L}_{te}$ and the best model $\mathcal{P}^{**}_{te}$ selected via *Oracle* model selection. With the latter being simply the *Oracle* performance $\mathcal{L}^{**}_{te}$, it can be rewritten to:

$$Regret = |\mathcal{L}^{**}_{te} - \mathcal{L}_{te}(\mathcal{P}^{*}_{val}, \mathcal{D}_{te})| \tag{4.14}$$

With the regret being zero when the validation loss $\mathcal{L}_{val}$ is as accurate in terms of performance estimation as the test loss $\mathcal{L}_{te}$.

Another perspective on *Regret* is that it is a measure of bias introduced by validation loss function $\mathcal{L}_{val}$ as compared to the optimal test performance $\mathcal{L}^{**}_{te}$. An example with *MSE* and *PEHE* validation and test metrics respectively would be as follows:

$$Regret_{PEHE} = |\mathcal{L}^{**}_{PEHE} - PEHE(\mathcal{P}^{*}_{MSE}, \mathcal{D}_{te})| \tag{4.15}$$

This metric has been already introduced in the literature in one form or another, sometimes additionally normalised by the *Oracle* performance $\mathcal{L}^{**}_{te}$, as in [172].

Furthermore, the problem of selecting the best validation score (see Equation (4.9)) could also be rewritten in a way that instead of optimising the test loss function $\mathcal{L}_{te}$, it would minimise regret $Regret_{te}$, though they in principle refer to the same problem. Also, a Regret defined in such a way is in fact a *weak* Regret as it is based on a *weak* Oracle. However, assuming the weak Oracle is a good approximation for the true one, the weak Regret is also expected to be an accurate enough approximation of the true Regret.

Rank Correlation

Regret in Equation (4.13) measures the validation metric's ability to select winning models, but it does not tell how accurately the rest of the candidate models were scored by the metric. Measuring how validation and test scores of all candidate models correlate with each other fills that gap, which is achieved with the Rank Correlation (RC) metric as per below.

$$RC = corr(\mathcal{L}_{te}(\mathcal{P}, \mathcal{D}_{te}), \mathcal{L}_{val}(\mathcal{P}, \mathcal{D}_{val})) \tag{4.16}$$

Higher correlations translate to a validation metric being better at **ranking** (hence 'rank') candidate models from best to worst in relation to each other. An important characteristic of RC is that it potentially exposes mistakes a validation metric makes with respect to scoring non-winning candidate models. This is because changing the rank order given by a validation metric, but only of non-winning candidate models will, not change the *Regret*, but will likely result in a different RC value. Such a characteristic makes rank correlation an important assessment tool, supplementary to *Oracle* and *Regret* when evaluating validation scores and their effectiveness at model selection tasks. An example of using RC for metric evaluation in the literature can be found in [172].

4.2.3 Causal Model Selection Methods

When it comes to model performance evaluation, there are multiple approaches to choose from. Perhaps the most straightforward and most prevalent in supervised

learning is MSE between predicted and factual outcomes (see Equation (4.17) below), also referred to in the literature as μ-risk [168] or *factual validation* [171]. The major shortcoming is the assumption that prediction error on factual outcomes is an accurate proxy for prediction error on CATEs, which, as shown by [170] and in Section 4.3, can have severe limitations. That is, it does not target CATE, only $\mu_0(x)$ and $\mu_1(x)$ separately. Nevertheless, its ease of use and popularity in machine learning make it a common metric of choice, similarly to R^2.

In terms of selection methods tailored specifically to CATE estimation, the usual goal is to work around the issue of missing ground truth (see Equation (4.18) below). Many approaches attempt to synthesise it through another layer of CATE modelling and treat it as a proxy for the final performance, also called *plugin validation* [171]. More concretely, an additional CATE estimator is trained specifically on validation data and then provides CATE predictions $\tilde{\tau}(x)$ on the same subset of data. Cross-fitting is often employed here to avoid overfitting. From now on, those CATE predictions $\tilde{\tau}(x)$ are treated as if they were true causal effects, but only for model selection purposes. Any other candidate CATE estimator that is subject to model evaluation is trained on training data and makes CATE predictions $\hat{\tau}(x)$ on validation data. To evaluate the candidate model, its predictions $\hat{\tau}(x)$ are compared to the synthetic ground truth $\tilde{\tau}(x)$, based on which model selection decisions can be made. The semi-ground truth $\tilde{\tau}(x)$ is essentially *plugged in* instead of true CATEs $\tau(x)$ into, for example, *PEHE* formula (see Equation (4.18)). Moreover, some solutions involve predicting only counterfactual outcomes and obtaining $\tilde{\tau}(x)$ by reusing given factuals, also referred to as *imputed effects* [32], whereas other approaches do not use factual outcomes and predict $\tilde{\tau}(x)$ directly. Many models have been used so far as a source of $\tilde{\tau}(x)$. One example is using matching with a distance measure between data points to identify counterfactuals [170], or any other CATE estimator [173], including neural networks [172], all of which provide synthetic CATEs as a semi-ground truth. Generative models have also been used to generate artificial potential outcomes and subsequently $\tilde{\tau}(x)$ [105, 106], with the major drawback being the notorious instability of such models. However, the idea

of synthesising CATEs shares the same inherent issue with CATE estimation itself, that is, missing counterfactuals. This leaves the problem of model tuning of the CATE synthesiser open. It is also arguably a paradoxical approach, as having a model that accurately synthesises CATEs would solve the main problem in the first place, rendering any further model selection pointless.

A response to possibly biased plugin validation is unplugged validation via influence functions [171]. Another alternative is *R-Loss* derived from the main formula of the R-Learner [87] (see Section 2.2.5), further extended to *R-Score* [86] that penalises constant CATE predictions, similarly to R^2 penalising constant outcome predictions in general. Interestingly, it is also possible to rewrite *R-Loss* as a reweighed ideal metric, such as *PEHE* [167].

4.3 Problem Demonstration

The graphical example in Figure 4.1 depicts some of the challenges of model selection, or model evaluation in general, in causal effect estimation. It is extreme on purpose to demonstrate the issues clearly and make it easier for the reader to appreciate the difficulty of the task. More concretely, we show how systematically missing data in small samples can affect, with a varying degree, goodness-of-fit and causal metrics and possible further consequences of that.

Our example is a straightforward two-dimensional case, consisting of an input feature X, response Y (both continuous), and binary $(0, 1)$ treatment assignment T. The DGP is a non-linear additive noise model where the mean of Y is sinusoidal in X with independent and identically distributed normal errors. Treated cases are often challenging to collect in practice (e.g. high costs, ethical reasons), leading to those units being not fully representative of DGP behind the treated arm. In general, all sorts of imbalances can arise from covariate shift and non-random selection, making the task of modelling the DGP with available data certainly non-trivial. Taken to

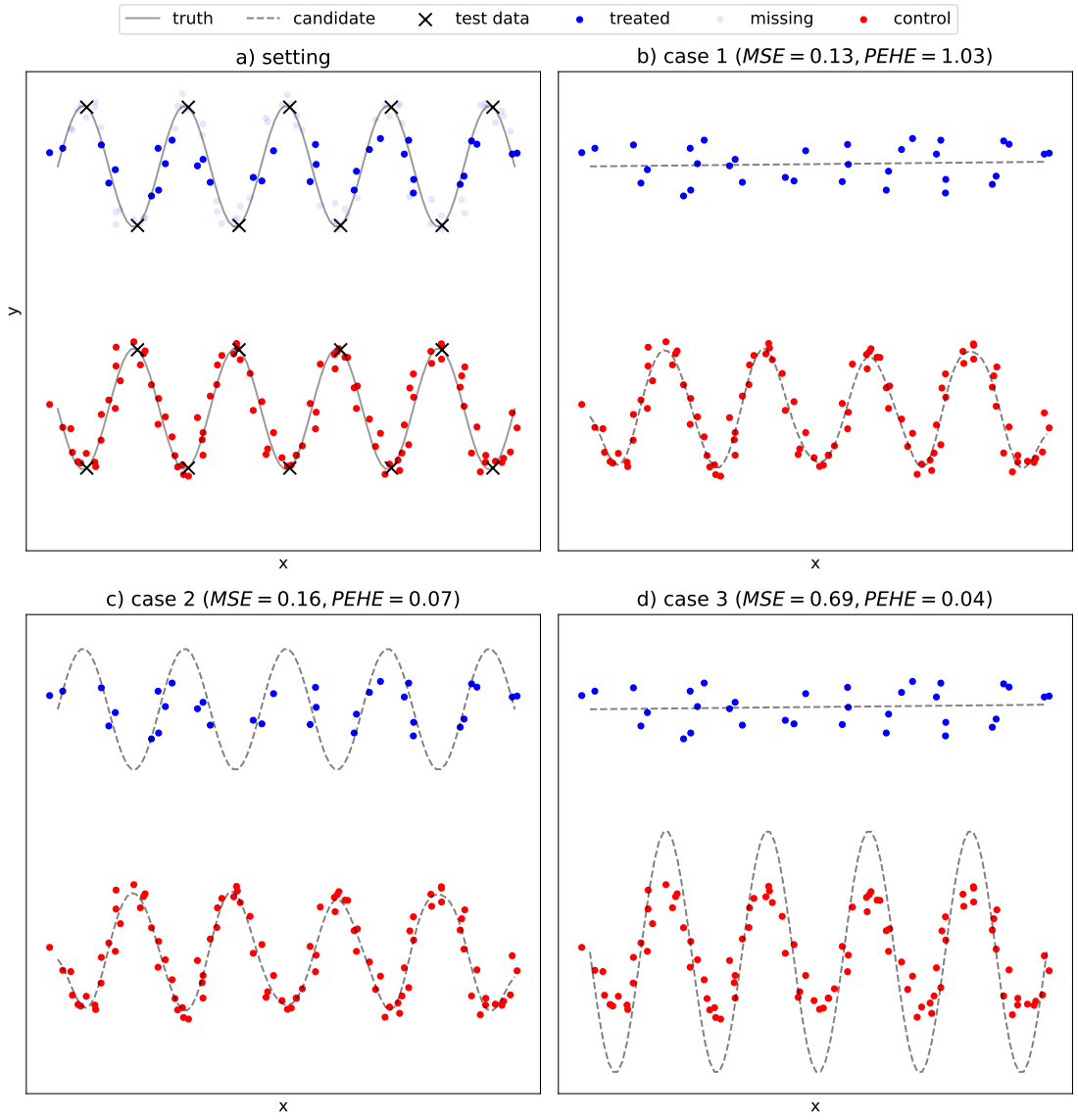


Figure 4.1: MSE - evaluation metric on observed data (validation set/cross-validation) used for model selection purposes (accessible with real datasets; lower is better). PEHE - evaluation metric on unobserved test data (not accessible with real datasets due to missing counterfactuals; lower is better).

the extreme, mostly for demonstration purposes, Subfigure a) depicts a possible scenario we can find ourselves in (note faded treated units). Despite the missing data problem, which in practice can exist in both treatment arms, the goal is to model the underlying DGP as closely as possible and provide accurate treatment effect predictions not only for observed units but also the ones the model has not seen in training (see data points marked with 'X'). The error for this is captured via the PEHE metric (Equation (4.18) below), which is generally inaccessible due to missing

counterfactuals in real datasets, but useful in simulations and benchmark datasets designed specifically to assess the performance of CATE estimators where PEHE practically becomes the metric to beat. The crucial difference between the two is that MSE measures average prediction error on observed outcomes $\mathcal{Y}$, whereas PEHE quantifies errors on predicted CATEs (i.e. difference between outcomes $\mathcal{Y}_1 - \mathcal{Y}_0$). That is, given an outcome prediction model $\hat{\mu}_t$:

$$\text{MSE} = \frac{1}{\text{n}} \sum_{\text{i}=1}^{\text{n}} (\mathcal{Y}_{\text{t}}^{(\text{i})} - (1 - \text{t})\hat{\mu_0}(\text{x}^{(\text{i})}) - \text{t}\hat{\mu_1}(\text{x}^{(\text{i})}))^2 \tag{4.17}$$

$$\text{PEHE} = \sqrt{\frac{1}{\text{n}} \sum_{\text{i}=1}^{\text{n}} [(\hat{\mu}_1(\text{x}^{(\text{i})}) - \hat{\mu}_0(\text{x}^{(\text{i})})) - (\mathcal{Y}_1^{(\text{i})} - \mathcal{Y}_0^{(\text{i})})]^2} \tag{4.18}$$

Note PEHE directly targets $\tau(x) = \mu_1(x) - \mu_0(x)$ and uses both $\mathcal{Y}_0$ and $\mathcal{Y}_1$ (one of them is never observed), whereas MSE targets each of $\mu_0(x)$ and $\mu_1(x)$ individually rather than the difference between them and involves only observed outcomes $\mathcal{Y}_t$. Clearly, PEHE cannot be targeted directly using only the observed data, only MSE; hence, it is tempting to assume that more accurate (in terms of MSE) outcome predictions (better fit) will lead to accurate CATE estimates. However, the disparity between model selection (MSE) and desired performance (PEHE) metrics, combined with the aforementioned missing data issue can lead to undesirable solutions, as demonstrated by Subfigures b) - d) and further discussed in the following paragraphs. Even at this preliminary stage, it becomes relatively apparent that the model selection task, which is perceived as a rather solved problem in supervised ML, is a non-trivial challenge in the treatment effect estimation setting.

Case 1 (Subfigure b) The control data are fitted well, accurately capturing the DGP for this part of the data. The model in the treated arm failed to match the underlying trend due to the severity of data loss. This solution is likely favoured if using standard model selection metrics, such as MSE. Despite having the best MSE compared to the other two candidate solutions (Subfigures c) and d)), this variant is not desirable due to poor generalisation

in the treated arm and overall high CATE prediction errors, as correctly captured by high PEHE.

Case 2 (Subfigure c) Both treatment arm models capture the true DGP very well. Selecting a more complex model that matches the DGP trend in the treated arm despite data loss is especially difficult with conventional performance metrics (see worse MSE than Case 1). One possibility for a performance metric to overcome this is to consider the task in its entirety (CATE estimation) instead of fitting each arm separately. Or put differently, the information about the trend in one treatment arm can be useful for modelling choices of the other arm. This way of thinking leads to targeted learning [82] and in general to more sophisticated causal model selection methods [87, 171]. Due to better generalisation, the Case 2 solution is clearly preferred (better PEHE than Case 1), but certainly non-trivial to identify (worse MSE than Case 1).

Case 3 (Subfigure d) This example is much harder to imagine in real-life situations, but can certainly happen when the goal is to beat SotA methods concerning PEHE on causal benchmark datasets. These datasets are often designed specifically to assess the performance of CATE estimators and thus include true CATEs to enable such tests (PEHE is then possible to calculate). While optimising for PEHE directly during the fitting process of CATE estimators would be an unacceptable violation of good practices, it is practically impossible to eliminate the exposure to PEHE completely to zero. For instance, if one were to develop a new CATE estimator, checking PEHE on the test set throughout the development might be an unavoidable step to ensure low enough PEHE is achieved for the new estimator to be considered worth paying attention to by the community and increase overall chances of it being published. Due to such external judgements, the information about the test PEHE may leak into the design of the new CATE estimator. A possible result is an undesirable solution that misses the main DGP trends and thus generalises poorly (the worst MSE of all 3 cases), but because of

the information leak about test PEHE, it manages to win the competition in terms of PEHE minimisation (and thus becoming a new SotA CATE estimator on the benchmark). Examining the plots of course makes it easy to discard this particular solution, but for high-dimensional X graphical representations quickly become non-trivial and hard to interpret.

Examining the three example cases together, it can be observed that MSE is a rather poor proxy for PEHE, which is quite counterintuitive as a better fit is believed to generally lead to improved CATE estimates. Therefore, simple model selection metrics, such as MSE, may not be sufficient to identify desirable solutions. Focusing too much on benchmark datasets and (indirectly) on PEHE can be dangerous as well. Just because a solution achieves a low level of errors on CATEs does not necessarily mean this is the desirable solution. Let us consider the following hypothetical scenario. A new CATE estimator has been developed (Case 2). Practitioners perform model selection with MSE and thus choose to proceed with the Case 1 model due to lower MSE. On the other hand, the research community deemed the Case 2 model not good enough, as the Case 3 estimator achieved better PEHE on benchmark datasets. Both communities rejected a perfectly sound and desirable Case 2 solution. How do we overcome this situation? Shall we exploit the benchmarks too?

It is straightforward to pick the Case 2 solution by examining the plots. With actual datasets, plotting is not an option due to the high dimensionality of input features, leaving performance metrics as the only feedback signal. These are clearly not ideal, as relying on any of them exclusively may result in undesirable solutions, as shown in the examples. Perhaps an untapped potential lies in a combination of goodness-of-fit and CATE loss measures, or variations of such. This is motivated by an observation that even though the non-trivial Case 2 solution has neither the best MSE, nor the best PEHE, its sum of MSE and PEHE is the lowest. As evidenced by the presented simulation study and analysis, model selection in the causal effect estimation setting is clearly non-trivial and poses many challenges. Performance metrics seem to play an important role, possibly having a much stronger influence on the final performance

of CATE estimation than it is commonly assumed. This motivates further study on causal model selection, CATE estimators, and their interplay, but on datasets closer to reality. This is the subject of the remaining content of this chapter.

4.4 Methodology

The goal of our experimental setup is to empirically investigate: a) the impact of the quality[1] of selected hyperparameter values on causal estimation performance, and b) the effectiveness of observable metrics (oMSE) at approximating ideal but inaccessible target metrics (pMSE) in the task of model selection and tuning. Following Equations (4.5) and (4.6), we further assume that validation data consists of background covariates, treatment and factual outcomes ($\mathcal{D}_{val} = \{X, T, Y_f\}$), whereas test data of the same plus counterfactual outcomes ($\mathcal{D}_{te} = \{X, T, Y_f, Y_{cf}\}$). As a result, validation loss functions become observable metrics ($\mathcal{L}_{val}$ = oMSE) and test loss functions equal to ideal (potential) metrics ($\mathcal{L}_{te}$ = pMSE). Thus, the goal a) is to explore how different hyperparameters across the search space $\mathcal{H}$ affect estimation performance $\mathcal{L}_{te}$ per each individual combination of estimators $\mathcal{C}_c$ and base learners $\mathcal{B}_b$. Whereas goal b) is about comparing performances $\mathcal{L}^*_{te}$ achieved via observable metrics $\mathcal{L}_{val}$ (Equation (4.7)) to potential (or *Oracle*) performances $\mathcal{L}^{**}$ obtained with ideal metrics $\mathcal{L}_{te}$ (Equation (4.11)) at the model selection task across the search space $(\mathcal{C}, \mathcal{B}, \mathcal{H})$.

The setup involves 4 datasets (Section 4.4.1), 6 observable metrics (Section 4.4.2) and multiple CATE estimator combinations (7 CATE estimators, 9 base learners; Section 4.4.3) coupled with various hyperparameter sets. Model selection metrics are also evaluated using different quality measures (Section 4.4.5). Further details about the experiments can be found in Appendix A.2.

[1] We introduce three categories when it comes to HP values: default, the best, and the worst.

4.4.1 Data

We incorporate four benchmark datasets commonly used in the treatment effect estimation literature, which were designed to assess the performance of CATE estimators. These datasets are: IHDP, Jobs, Twins and News. Depending on the characteristics and available information as part of the data, different performance metrics $\mathcal{L}_{te}$ are used. With Jobs, ϵ_{ATT} and $\mathcal{R}_{pol}$ metrics are used; otherwise ϵ_{ATE} and PEHE. Each dataset consists of a number of slightly different variations, often referred to as *realisations* or simply *iterations*. While these numbers vary across datasets (10-1,000), we stick to 10 across all four, mostly due to computational reasons, as we find that 10 iterations are already strongly indicative of the overall performance of CATE estimators. See Sections 2.2.8 and 2.2.9 of Chapter 2 for more details about metrics and datasets involved in this set of experiments.

4.4.2 Model Evaluation Metrics

This study incorporates multiple validation metrics that can be used for model selection purposes. These are also referred to as observable metrics $\mathcal{L}_{val}$.

MSE

The usual MSE measures aggregate squared error on factual outcomes. Lower is better.

$$\mu\text{-risk} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{\mu}_t(x_i))^2 \tag{4.19}$$

R-Squared

R^2 measures the quality of variability in predictions. This is an important improvement over MSE as it penalises models outputting a constant value (or close to it). Higher is better, where 1.0 is the best possible.

$$\mu\text{-risk}_R = 1 - \frac{\sum_i (y_i - \hat{\mu}_t(x_i))^2}{\sum_i (y_i - \frac{1}{n}\sum_{i=1}^{n} y_i)^2} \tag{4.20}$$

Plugin Validation

It involves fitting a CATE model on the validation set and then treating the estimator as a source of (pseudo) ground truth a candidate model trained on the training data is evaluated against. In order to avoid making predictions on the same data set that was used for training, the usual approach is to perform *cross-fitting*. The idea is similar to *cross-validation*, where data is split into K folds of the same size. The training phase is exactly the same as in CV, but instead of performing evaluation on the validation folds, we make only predictions on them, and store them for later use. Through this process, we obtain τ_{plug} estimator which can be used as an alternative source of ground truth when calculating various CATE metrics. The usual approach is to use the formula for PEHE but use τ_{plug} instead of the actual ground truth for individual effects. This gives us the *Plugin PEHE* formula as:

$$\tau\text{-risk}_{plug}^{PEHE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{\tau}(X_i) - \tau_{plug}(X_i))^2} \tag{4.21}$$

Since τ_{plug} can be treated as simply the source of true CATEs, a *Plugin ATE* formula get also be obtained:

$$\tau\text{-risk}_{plug}^{ATE} = \left|\frac{1}{n}\sum_{i=1}^{n}\hat{\tau}(X_i) - \frac{1}{n}\sum_{i=1}^{n}\tau_{plug}(X_i)\right| \tag{4.22}$$

Both of the above are separate model selection strategies as they provide different feedback signals. The following estimators and base learners are used to obtain τ_{plug}. CATE estimators: S-Learner and T-Learner (see Section 2.2.4). Base learners: Decision Tree, Boosted Trees (LightGBM), and Kernel Ridge. Basic model selection is also performed first for base learners. The same hyperparameter search space is used as for the proper CATE estimators. R^2 metric is used to select the best combination via 5-fold CV stratified on treatment T [131]. Once the selection is done, proper learning is performed to obtain τ_{plug}.

Matching

As any CATE estimator can be plugged into τ-risk$_{plug}$ as τ_{plug}, we also experiment with *matching*. For clarity, we denote it as τ-risk$_{match}$ to distinguish it from regular plugins.

We obtain τ_{plug} through *k-Nearest Neighbours* (kNN) algorithm. Data points of each treatment group are stored by a separate kNN instance, and then matching pairs are found using opposite kNN instances. For example, to find counterfactuals for all controls, control units are passed to the kNN instance that stored treated units, resulting in treated units that match the best the control ones. The same process is then repeated for the other treatment group. The overall result is predicted (matched) counterfactuals, which combined with factuals, can be used to obtain CATEs, further used as pseudo ground truth.

We use Euclidean distance in the kNN to find matching pairs. Along with the default single-neighbour matching ($k = 1$), we also experiment with $k = 3$ and $k = 5$. In those cases, the resulting counterfactual prediction is an average of k nearest matches inversely weighted by their distance to the query point (closer points have greater influence). Similarly to τ-risk$_{plug}$, we also investigate the effectiveness of both PEHE and ATE variants.

R-Score

We start by defining two nuisance functions $m(X) = \mathbb{E}[Y \mid X]$ and $e(X) = \mathbb{E}[T \mid X]$, estimates of which are plugged into the *R-Loss* formula [87]:

$$\mathcal{L}(\tau) = \frac{1}{n}\sum_{i=1}^{n}([Y_i - \hat{m}(X_i)] - [T_i - \hat{e}(X_i)]\tau(X_i))^2 \tag{4.23}$$

By inserting any candidate CATE estimator $\hat{\tau}$, the *R-Score*, formulated by [86], is then of the following form:

$$\tau\text{-risk}_R = 1 - \frac{\mathcal{L}(\hat{\tau})}{\min_\tau \mathcal{L}(\tau)} \tag{4.24}$$

With the denominator being a simple linear CATE model that achieves the lowest loss. Similarly to R^2 that penalises constant-valued predictions, τ-risk$_R$ treats the linear CATE model as its baseline, and hence penalises predicting constant effects. Higher is better, with a value of 1.0 being a perfect score. A negative score suggests worse performance than predicting constant effects.

Estimates of the nuisance functions $m(X)$ and $e(X)$ are obtained through *cross-fitting*, same as with τ-risk$_{plug}$. Consequently, the same base learners are used to obtain $\hat{m}(X)$ and $\hat{e}(X)$, that is, Decision Trees, LightGBM and Kernel Ridge. The only difference is the need to use regressors and classifiers to get $\hat{m}(X)$ and $\hat{e}(X)$ respectively. In addition, basic model selection is performed for both nuisance functions via stratified 5-fold CV [131].

The initial idea of the *R-Loss* (Equation (4.23)) comes from [87]. They recommend using *Gradient Boosted Trees* (XGBoost), which we do incorporate but in the form of LightGBM. Further extension to a more robust *R-Score* (here τ-risk$_R$; Equation (4.24)) is credited to [86].

Policy Risk

Whenever it is possible to obtain $\mathcal{R}_{pol}$ metric (see Section 2.2.8), which is the case with the Jobs dataset (see Section 2.2.9), it can also be used for model selection purposes. The only important difference is to obtain it on validation data, instead of the test set.

4.4.3 Estimators

We use the following **CATE estimators**: S-Learner (SL), T-Learner (TL), X-Learner (XL) [32], Doubly Robust (DR) [81], Double Machine Learning (DML) [33], Inverse Propensity Score Weighting (IPSW) [174], Causal Forest (CF) [31]. Many are provided by EconML [86]. See Sections 2.2.4 and 2.2.5 for descriptions.

We then combine with the following **Base learners**: Lasso Lars (L1), Ridge Regression (L2), Decision Trees (DT), Random Forest (RF), Extremely Randomised

Trees (ET) and Kernel Ridge (KR) accessed via scikit-learn [175]. Gradient Boosting Trees as LightGBM (LGBM) [163] and CatBoost (CB) [162]. Feedforward NNs realised via TensorFlow [176].

Each combination of CATE estimators and base learners is treated as a separate model, such as *SL-L1* or *DR-RF*. We combine all learners to obtain the experimental setup. The only exceptions are NNs, combined only with SL and TL for computational reasons, and CF (standalone estimator, no choice of base learners). All CATE estimators, base learners and hyperparameters refer to search spaces $\mathcal{C}$, $\mathcal{B}$ and $\mathcal{H}$ respectively in Equation (4.3). Following standard industry practice, we perform 10-fold cross-validation [131].

4.4.4 Hyperparameters

The hyperparameters and corresponding ranges of values that were explored as part of using various base learners are listed in Table 4.1. Moreover, whenever ensemble learners were used, such as RF, ET, CB, LGBM and CF, the number of inner learners (n_estimators) was set to $1,000$. When it comes to NNs, we used *relu* activations, 0.25 dropout, 0.01 L2 regularisation in the output layer, $10,000$ optimisation steps (not epochs) and Adam optimiser [177]. All hyperparameters refer to the search space $\mathcal{H}$ in Equation (4.3).

4.4.5 Validation of Model Evaluation Metrics

In order to investigate the effectiveness of model evaluation metrics, we incorporate the following quality measures. First, by recognising the fact that the priority is to optimise the accuracy of CATE estimation, we judge a validation metric's $\mathcal{L}_{val}$ quality by the performance of the model $\mathcal{M}^*_{val}$ it selects against the test metrics $\mathcal{L}_{te}$ that are applicable for a particular dataset. See Equations (4.2) and (4.7). Each validation metric is evaluated against all test metrics. *Oracle* performances $\mathcal{L}^{**}$ (Equation (4.11)) are also collected for all test metrics and compared to those achieved via validation metrics. While *Oracle* performances are not accessible when dealing with real datasets (lack of ground truth; missing counterfactuals), it is

base learner	hyperparameter	values
L1 and L2	*alpha*	$\{.001, .01, .1, .5, 1^*, 2, 10, 20\}$
	max_iter	$\{1000^*, 10000\}$
DT, RF, ET and CF	*max_depth*	$\{2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20^*\}$
	min_samples_leaf	$\{.01, .02, .03, .04, .05, 1^*, 2, 3, 4, 5, 6, 7, 8, 9\}$
KR	*alpha*	$\{.001, .01, .1, 1^*\}$
	gamma	$\{.01, .1, 1^*, 10, 100\}$
	kernel	$\{rbf, poly^*\}$
	degree	$\{2, 3^*, 4\}$
CB	*depth*	$\{5, 6, 7, 8, 9, 10^*\}$
	l2_leaf_reg	$\{1^*, 3, 10, 100\}$
LGBM	*max_depth*	$\{5, 6, 7, 8, 9, 10^*\}$
	reg_lambda	$\{0, .1^*, 1, 5, 10\}$
NN	*hidden_layers*	$\{1, 2\}$
	hidden_units	$\{4, 8, 16, 32, 64, 128\}$
	learning_rate	$\{.0001, .001\}$
	batch_size	$\{32, 64, 128, full\}$

Table 4.1: Hyperparameter search spaces defined per base learner. *Default hyperparameter values.

certainly a useful tool for this study to measure how biased different model selection approaches can be (specifically to analyse how far apart observable and potential metrics are). Moreover, using this method can shed some more light on the real capabilities of various CATE estimators, showing what they can potentially achieve if model selection choices are optimised for the ideal target $\mathcal{L}_{te}$.

4.5 Results and Discussion

In order to provide context for the presented numbers, we compare our results to recent literature on causal effect estimation. These include TARNet and CFR-WASS [91], SITE [92], GANITE [95], CEVAE [94], as well as BLR, BNN-4-0 and BNN-2-2 [90]. Instead of picking only the best-performing estimator per dataset and test metric, we select multiple best ones that effectively form a range of values that here we consider as state-of-the-art performance levels. This is represented as grey areas in the following plots. The only exception is the Twins dataset for

which no results with respect to ϵ_{ATE} metric are available. For this reason, we set the range ourselves to $[0.0, 0.1]$.

Our first goal is to analyse how the quality of hyperparameters impacts effect estimation performance across different causal estimators. For each CATE estimator-base learner combination $(\mathcal{C}_c, \mathcal{B}_b)$, and across all hyperparameters $\mathcal{H}$, we select three types of hyperparameters: *default* as defined by code packages, those that achieve the *worst* test metrics values (max $\mathcal{L}_{te}$), and those that attain the *best* test metrics values (min $\mathcal{L}_{te}$, effectively $\mathcal{L}^{**}$, or *Oracle*). Figure 4.2 depicts obtained results. Clearly, changing only hyperparameters and their quality influences estimation test performance by a significant margin. In fact, to the extent that hyperparameters alone can decide whether estimation performance is below or above SotA levels (e.g. green vs. orange dots). Across all datasets, there are very few instances with quality hyperparameters (green) that do not meet SotA levels (above or to the right of grey areas), suggesting that quality hyperparameters are sufficient for great estimation performance. Furthermore, quality hyperparameters (green colour) are evidently necessary to surpass SotA in IHDP and Jobs datasets (below or to the left of grey areas). In Twins, only quality cases are below a high-standard value of 0.01 ϵ_{ATE}. Results on News are less conclusive, though the concentration of green points is still the best (closest to the bottom left corner) among the three groups.

In our further analysis, we take a closer look at how estimation performance varies across different causal estimators when picking only quality hyperparameters. This is to investigate if and how the sufficiency of quality hyperparameters differs between specific causal estimators. More specifically, we analyse all combinations of CATE estimators and base learners $(\mathcal{C}_c, \mathcal{B}_b)$, and from all explored hyperparameters $\mathcal{H}$, we select only those instances that achieved the best test metrics values (i.e. Oracle performances). The results are presented in Figure 4.3, which are essentially the green points distilled from Figure 4.2 but grouped by types of CATE estimators and ML base learners. Certainly, the vast majority of presented estimators either meet (all datasets) or even surpass (all datasets except Twins) SotA performance

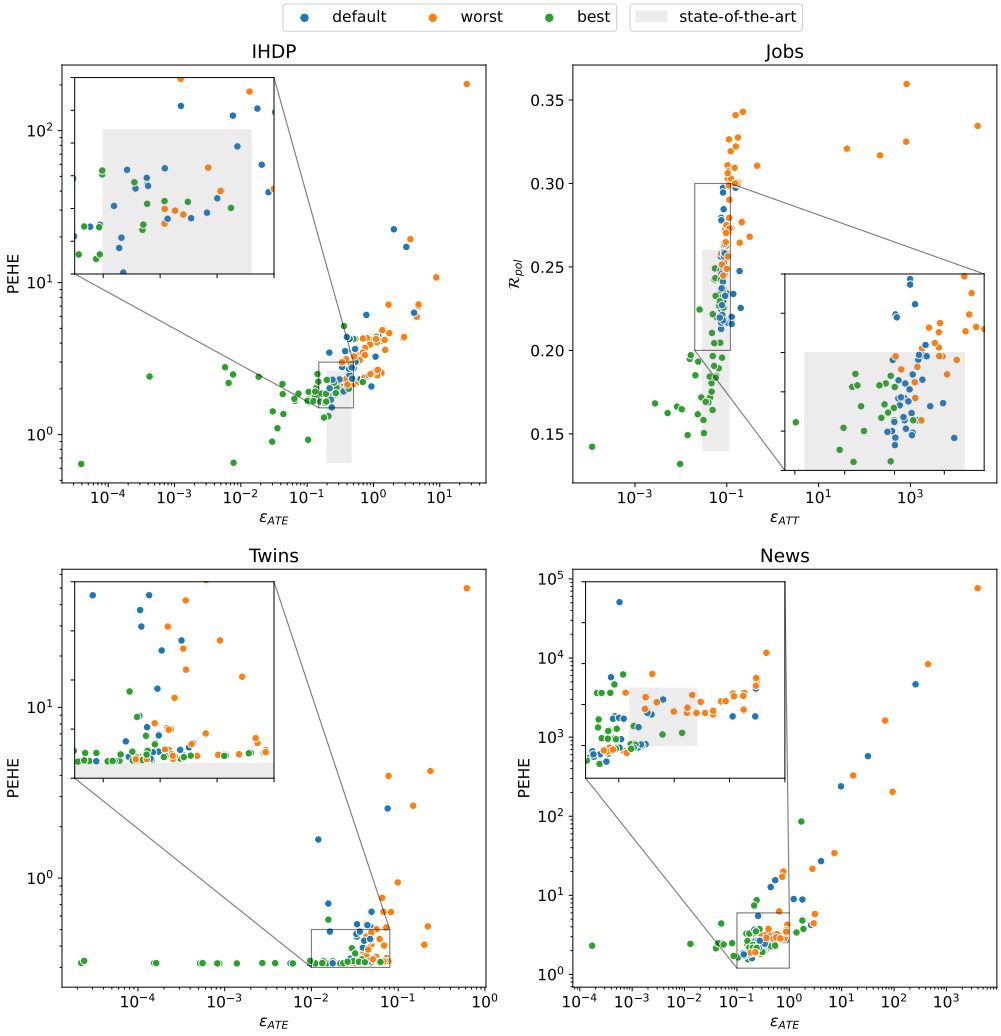


Figure 4.2: Performance of CATE estimators with three different types of quality of hyperparameters across all four datasets. Each data point represents the mean across dataset iterations. Lower is better applies to all metrics, thus the closer to the bottom-left corner, the better. Note the logarithmic scale for all axes to aid visual presentation except policy risk in Jobs. Dots which are below and left of grey SotA boxes beat SotA.

levels (within, below or to the left of the grey area). This is true regardless of CATE estimators and base learners, with very few exceptions. For example, linear models (L1 and L2) combined with SL and IPSW underperform in IHDP, perhaps due to them being inherently too simple to fit the solution. Some instances involving DR performed particularly poorly, which could be due to estimated propensities being too close to zero, a known issue of propensity-based methods [84]. Note, that a similar type of analysis could have been performed to support the necessity of quality hyperparameters (distil orange or blue dots from Figure

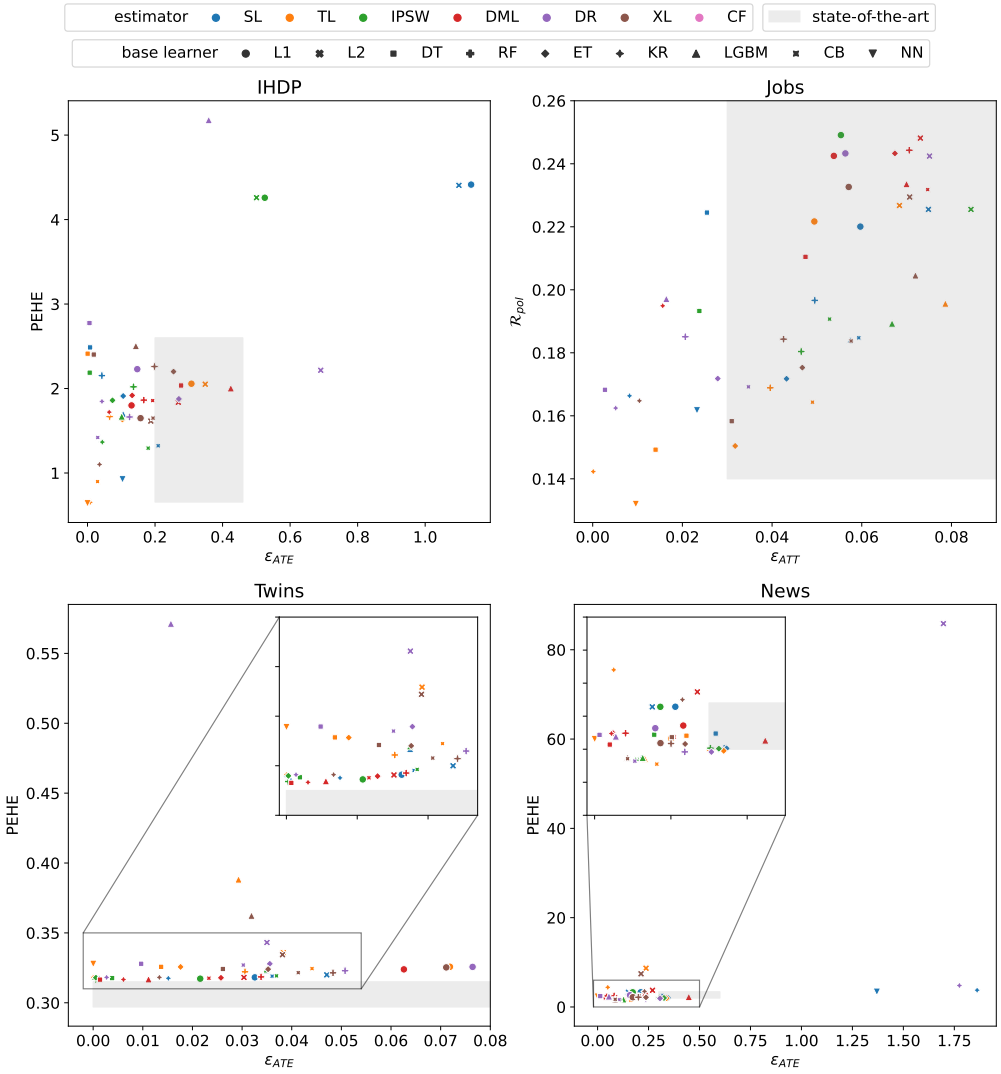


Figure 4.3: Performance of CATE estimators with the best hyperparameters across all four datasets, grouped by types of causal estimators and base learners. Each data point represents the mean across dataset iterations. Lower is better applies to all metrics, thus the closer to the bottom-left corner, the better. Dots which are below and left of grey SotA boxes beat SotA.

4.2). However, hyperparameter tuning is already recognised by the ML community as recommended practice, making any further analysis in this direction beyond Figure 4.2 rather unnecessary.

Our results presented so far show that estimators equipped with the right hyperparameters can perform surprisingly well. However, finding those well-performing hyperparameters in a data-driven way requires an observable metric that can be used in the process of hyperparameter tuning. In Figures 4.2 and 4.3, we used

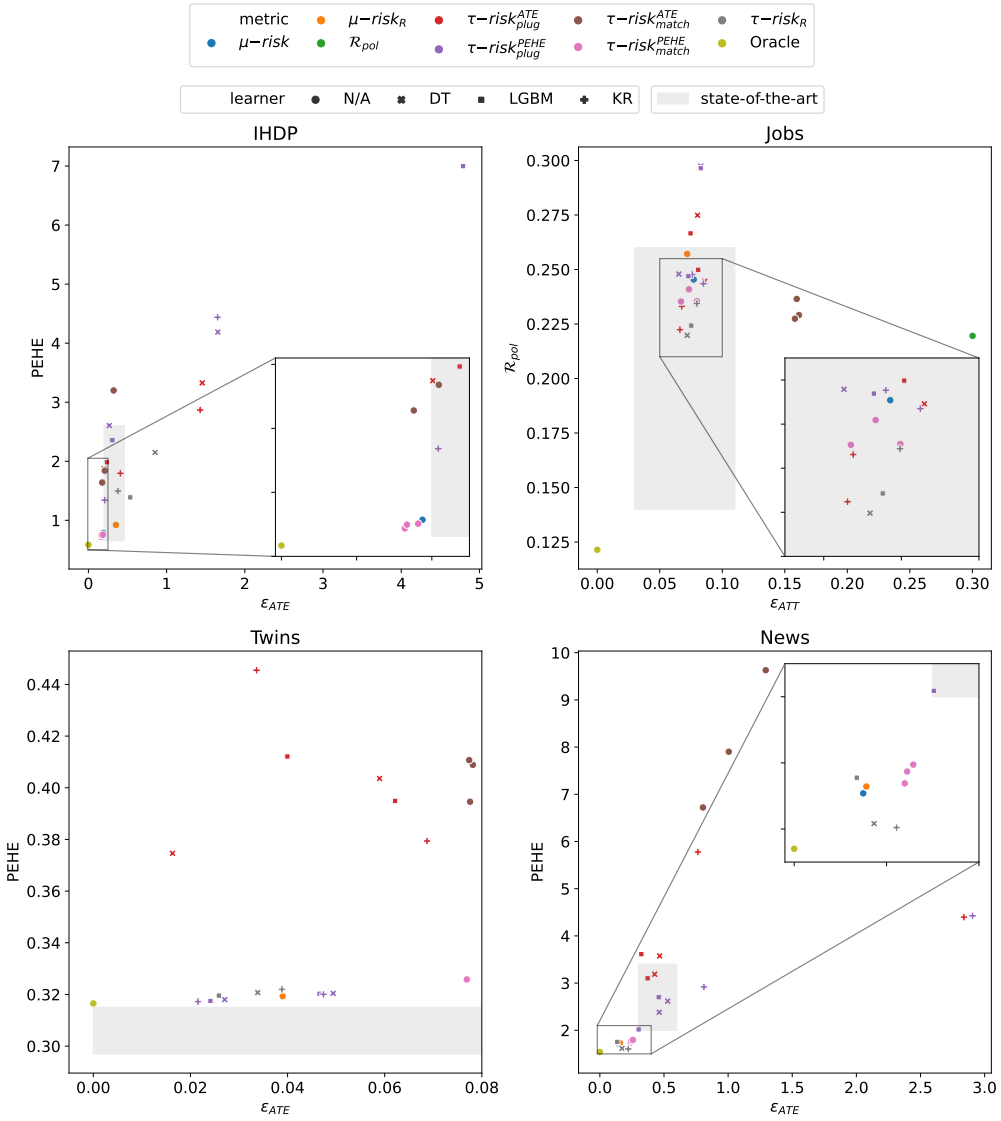


Figure 4.4: Performance of model selection metrics across all four datasets. Each data point represents the mean across dataset iterations. Lower is better applies to all metrics, thus the closer to the bottom-left corner, the better. Dots which are below and left of grey SotA boxes beat SotA.

potential metrics to select the best hyperparameters to show the true capabilities of common causal estimators, but those metrics cannot be used in tuning in practice as they require access to true CATEs (inaccessible in real life scenarios). For this reason, the next step is to investigate how effective are various observable metrics at selecting the best solutions (hyperparameters/models). The solutions they select are evaluated against test metrics, which are effectively a measure of the quality of considered observable metrics. The search space here includes all

CATE estimators $\mathcal{C}$, **base learners** $\mathcal{B}$ and **hyperparameters** $\mathcal{H}$, processed by each metric and its variation. As a consequence, each dataset iteration can be 'won' by a completely different estimator type (as opposed to previous experiments). Also, because the entire search space $(\mathcal{C}, \mathcal{B}, \mathcal{H})$ is now considered, the task at hand is no longer just hyperparameter selection, but rather model selection (choosing among $\mathcal{C}$ and $\mathcal{B}$), though they both use metrics for the same purpose – model evaluation (see also Section 2.5.1). To reveal the bias of observable metrics, we also include potential (*Oracle*) performances achieved when using ideal (potential) metrics instead of observable ones to select winning candidate models (see Oracle). Results obtained are presented in Figure 4.4. First, the performance obtained varies substantially among metrics, especially among those that perform internal learning (same colour but different shapes). Choosing between ϵ_{ATE} and PEHE as feedback signals (see τ-risk$_{plug}$ and τ-risk$_{match}$) also makes a difference. This shows the choice of a selection metric can seriously impact the final estimation performance (meeting or not meeting SotA levels) and makes it a very important choice to make during the modelling process. Another important observation is that, although SotA performance is available through many observable metrics, clearly none of them reach the best possible performance achieved via ideal metrics (Oracle). Indeed, the gap itself is to some degree unsurprising as observable metrics are only imperfect proxies for the potential ones. However, the magnitude of the gap is considerable to the extent that suggests better model selection metrics should be feasible if this problem receives enough attention in the future. This gap also indirectly shows that the quality hyperparameters that suffice for excellent estimation performance (shown in Figures 4.2 and 4.3) might be hardly identified by observable metrics.

4.6 Conclusion

This study provided empirical evidence that the right hyperparameters, with respect to ideal metrics, are sufficient and necessary (empirically speaking) in order to

reach, or even exceed, SotA performance levels in causal effect estimation. In most cases, this holds true across different types of causal estimators and ML base learners, encouraging to prioritise the selection of good hyperparameters over the choice of estimators and learners. Another implication is that both excellent and poor estimation performances can be achieved by only changing hyperparameter values. This places a big question mark on published benchmarks that test various estimators. If hyperparameters account for such a big part of the estimation performance, should claimed performances be credited to the method itself or well-identified hyperparameters? The fact that studies often use different metrics to optimise hyperparameters only adds complexity to this issue. Furthermore, we also show the quality of hyperparameters/models selected with different metrics varies significantly. As a consequence, the choice of model evaluation metric highly impacts the resulting estimation performance. In addition, we show that observable metrics (feasible with observational data) select considerably worse hyperparameters/models as compared to those selected with ideal (but normally inaccessible) metrics. While the discrepancy is unsurprising, mostly due to observable metrics being imperfect proxies for ideal ones, its size encourages the development of more robust model selection methods.

Through substantial empirical evidence, we make a strong case that more focus on metrics is needed for causal learning problems. In this, we connect with the theoretical work on the validation of causal models by [171]. In their framework, oMSE is a special case of plug-in loss in which the counterfactuals are 'synthesised' through elements of the candidate causal model (their Equation (6)), and pMSE is a special case of expected loss under the true causal model used to generate the observed data (their Equation (5)). They propose a targeting procedure based on theoretically derived influence functions to update the plug-in estimate to lie closer to the solution that would have been obtained had pMSE been available (see their Theorem 1). The approach implemented in their paper (based on a first-order Von Mises expansion as set out in their Theorem 2) will work well in terms of choosing the best model if the plug-in estimate lies inside a tight neighbourhood

of the true value, but our results indicate that this cannot be guaranteed to hold. We argue strongly, therefore, that more attention to overcoming this problem is needed from theorists and methodologists (e.g. implementing Alaa and Schaar's influence function approach for higher-order, and thus more accurate, Von Mises expansions) because this is of vital importance for practice.

An overall takeaway is that contrary to common belief, popular causal estimators can provide excellent estimation performance given the right hyperparameters, but currently available observable metrics fail at identifying these miraculous hyperparameters. This, however, does not mean they are unreachable. Highly interpretable models, such as linear regression or decision trees, provide intuitive hyperparameters that can be manually tuned to the problem at hand by domain experts by ingraining their prior knowledge into the model via hyperparameters. This is our practical recommendation until more robust observable metrics are developed.

4.6.1 Limitations

This study is limited in a few ways. First, we make the usual set of assumptions about the data, that is, *SUTVA* and *strong ignorability* (see Section 2.2.2), which may or may not hold in practice. Secondly, the experimental setup has some shortcomings, mostly due to computational reasons. While an effort was put into exploring relatively wide search spaces to make our findings as general as possible, a practical balance had to be found to ensure computational feasibility. This applies to explored hyperparameters of base learners as well as combinations of base learners in CATE estimators. More concretely, if a CATE estimator uses multiple base learners, only one type of learners is explored at a time, never mixing multiple types of learners simultaneously. For instance, in the case of a *T-Learner* with two internal base learners per treatment arm, if the goal is to combine it with Decision Trees then both internal regressors will be Decision Trees. If we explore it with Random Forests, both regressors are then Random Forests. And so on. As some CATE estimators incorporate as many as 5 base learners (e.g. *X-Learner*), we further cut down the search space by assigning the same hyperparameter values for all base

learners while exploring different variations of hyperparameters. This is possible because all base learners are always of the same type within a CATE estimator.

Finally, the setup is limited by our choice of CATE estimators, base learners, model evaluation metrics and datasets. However, regardless of those shortcomings, we believe the proposed setup is general enough to support our findings, and extending it would unlikely alter the main observations fundamentally while unnecessarily increasing the computational burden significantly.

4.6.2 Future Work

In terms of future work, given the excellent performances achieved here with already established estimators, we argue new causal estimators are not needed for meaningful progress in causal effect estimation. Rather a more thorough understanding of the causal model selection and hyperparameter tuning is required, as once recognised in ML [25], that will eventually unlock those great performances we showed are possible but on real-life observational datasets. Recent studies indeed push the frontier in this area (e.g. [87, 171, 172]), but many of the proposed model selection methods involve another layer of learning, on top of causal estimation, which is again subject to hyperparameter tuning, making them very unstable in practice. Perhaps there are alternative ways to be explored for improving causal hyperparameter tuning without relying on learning methods. Moreover, as the number and complexity of causal model selection approaches increase, new tools and frameworks might be needed to facilitate the use of the latest selection methods and help practitioners in the complex task of model evaluation in the causal setting (e.g. [178]). The sensitivity of causal estimators to hyperparameter choices could also be studied further by, for example, investigating whether said sensitivity changes as we modify the sample size of the data. Finally, due to the importance of hyperparameters in the estimation performance and inconsistent practices in published benchmarks, some form of standardisation in model tuning across the field of causal inference might be necessary to overcome this issue in the future. One possibility is to develop a public benchmarking platform that would test and

tune all types of causal estimators equally thoroughly, something that has been done in the causal discovery community [179].

Summary

Major findings:

- The right hyperparameters, with respect to ideal/potential metrics, are **sufficient** and **necessary** (in an empirical sense) to attain excellent causal effect estimation performance, **regardless** of types of causal estimators and base learners, provided that the ML learners are chosen to be flexible enough to fit the problem at hand. *Conclusion: using the right hyperparameter selection methodology is far more important than selecting causal estimators and base learners.*
- For most common causal estimators, there exists a set of suitable hyperparameters for which they **reach** or even **surpass state-of-the-art** performance based on observed metrics for other estimators and learners. *Conclusion: hyperparameters alone can be used to significantly change the estimation performance, questioning recent benchmarks.*
- The quality of hyperparameters/models selected with different commonly used *observable* metrics **varies substantially**. *Conclusion: the choice of the (observable) metric used in hyperparameter tuning/model selection highly impacts achieved estimation performance.*
- Commonly used *observable* metrics select **significantly worse** hyperparameters/models as compared to those selected with ideal but normally inaccessible *potential* metrics. *Conclusion: better observable metrics are needed to unlock, or reduce the gap to, those excellent potential performances.*

Minor findings:

- It is uncommon for a single causal estimator to provide accurate estimates for individual and average effects simultaneously.
- Using an evaluation metric that matches the estimation target may improve ranking, but not necessarily select the winning model.

5

Robustness of Algorithms for Causal Structure Learning to Hyperparameter Choice

The newly discovered importance of hyperparameters in causal inference was the catalyst of renewed interest in causal discovery methods within this project as now hyperparameters have become one of the possible causes of previously encountered issues with using the methods within real-world settings. This forthcoming chapter explores such a possibility by analysing how the causal graph recovery performance is influenced by the choice of hyperparameters across different methods, with a particular emphasis on robustness to misspecified cases due to its relevance in (unsupervised) causal structure learning.

Contents

5.1 Introduction

Uncovering causal graphs is a useful tool in data-driven decision-making as it helps understand the underlying DGP. A large number of CSL algorithms (see Section 2.1.3) incorporate ML methods. These, in turn, heavily rely on HPs (Section 2.5) for accurate predictions [25]. In addition, there has been growing evidence that correctly specified HPs can close the performance gap between SotA and other methods [164, 180–182]. *Are hyperparameters as important in causal structure recovery?*

Hyperparameter optimisation is extremely challenging in CSL as the true graphs (see Section 2.1.1) are inaccessible outside of simulated environments. This inability to reliably tune could be one of the reasons behind the struggle to apply some of the algorithms to real data problems [75], or why HPs are often completely neglected in this area. On the one hand, benchmarks and evaluation frameworks (e.g. [183, 184]) usually focus on finding a learning algorithm that works best under specific circumstances but without considering HPs as part of the problem. On the other hand, studies that address HP tuning (e.g. [185, 186]) consider individual algorithms, but not the impact of tuning (or the lack of it) on selecting the best algorithm for the available data. Understanding how HPs affect algorithm choice, as well as individual methods, is clearly missing but can be a crucial next step toward more stable causal discovery in real data applications. To make matters worse, the evaluation metrics used for tuning can be imperfect and sometimes favour specific learning methods [26]. This brings us to the core questions of this chapter: *Do different algorithms perform similarly given access to a hyperparameter oracle? How robust are they against misspecified hyperparameters?*

In this work, we set out to address these questions and investigate the impact HPs have on the causal graph recovery performance of individual algorithms, as well as

on the best algorithm choice (see Figure 5.1). We start by showing how a single HP plays a crucial role in the simplest causal graph problem (two variables). More extensive experiments strengthen this observation and confirm it as a more general phenomenon. The experimental setup involves many seminal CSL algorithms (see Section 2.1.3) tested against real and simulated datasets.

Contributions. This work offers the following contributions:

- Compare algorithms' performances and their winning percentages across hyperparameters.
- Compare algorithms' performances under well-specified and misspecified hyperparameters.
- Compare algorithms' winning percentages under well-specified and misspecified hyperparameters.

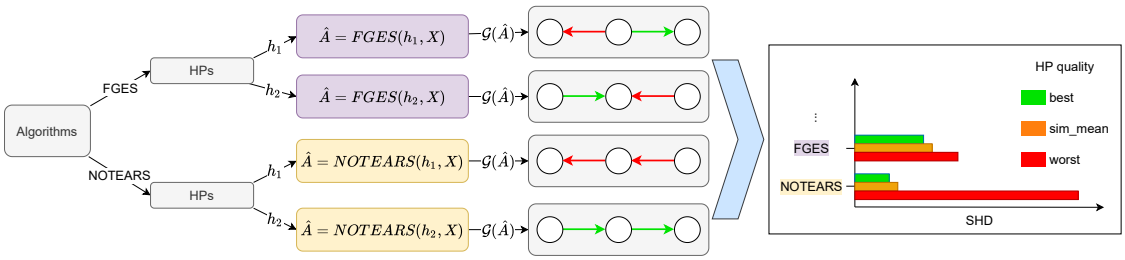


Figure 5.1: Summary of the main idea. We explore various causal structure learning algorithms and investigate how hyperparameters affect their performance. Notation: h_1 and h_2 are different hyperparameter values; X denotes IID data provided to algorithms; $\hat{A}$ is recovered adjacency matrix while $\mathcal{G}(\hat{A})$ is a causal graph based on $\hat{A}$; SHD is structural Hamming distance (lower is better). Note how recovered causal graphs differ between different hyperparameters of the same algorithm (green edges are correct; red incorrect). *sim_mean* are hyperparameters that achieved the best **average** performance across all simulations.

Related work. This work connects with the existing literature mainly through the topics of performance evaluation and HP analysis. The performance of CSL algorithms has been evaluated from several different perspectives, such as mixed data types [183] or time series data [187]. In an attempt to strengthen the evaluation,

there have been efforts to develop testing environments that closely resemble real-life datasets. Some examples include simulators based on gene regulatory networks [188] or neuropathic pain pathology [184]. [189] take evaluation further by proposing to test algorithms on the parts of real-life datasets that are known a priori. Furthermore, to improve reproducibility, [179] developed a benchmarking platform that covers a wide range of learning methods and data scenarios. The importance of hyperparameters and their impact on performance has been mostly studied in other areas outside of CSL. In the offline RL setting, [180] reported, among other aspects, that robustness to hyperparameter choices[1] is an important issue and that careful tuning can deliver close to optimal policies. [181], on the other hand, makes a case for hyperparameter tuning in model-based RL. Similar observations have been reported in causal effect estimation [164], graph learning [182], code classification [190] and evolutionary algorithms [191], where SotA performance levels are attributed to HPs alone. HPs have been also studied in the broader context of "tunability" [132] and optimisation approaches [133]. Some notable recent works even challenge our current understanding of how HPs interact with loss functions [192] and decision boundaries [193]. HPs in CSL have mostly been discussed in the context of tuning. One common approach is to select hyperparameters that result in stable predictions of causal structures across random data samples [185, 194, 195]. Another strand of work performs out-of-sample validation for tuning purposes based on predictive accuracy of models fitted in accordance with the recovered causal graph structure [186], or assigned scores developed specifically for CSL tuning [196]. Metrics based on regression error have been also considered [110], though in the context of two variables.

Chapter structure. In Section 5.3 we demonstrate the importance of hyperparameters in CSL via a bivariate example, further motivating more extensive numerical experiments presented and discussed in Section 5.4. Further sections include extended details about the experiments (Appendix B.3), show supplemental

[1] Robustness is seen as low sensitivity of the final performance to hyperparameter choices.

results (Appendix B.1), and offer additional recommendations for practice (Appendix B.2). Section 5.5 concludes the chapter and offers potential future work directions.

5.2 Causal Structure Learning

Throughout this chapter, we build upon the assumptions and fundamental ideas of CSL introduced in Section 2.1 of Chapter 2. Since the graphical and functional causal graphs from Section 2.1 are of core interest in this chapter, we will now define them more rigorously to aid further technical discussions around them. CSL algorithms that we take into account in this chapter are presented in Section 5.4.2.

Graphical. Let $\mathcal{G} = (V, \mathcal{E})$ be a graph with nodes/vertices $V = \{1, \dots, p\}$ and edges $\mathcal{E} \subseteq V^2$. Edges are pairs of nodes $(j, k) \in \mathbf{V}$ where $(v, v) \notin \mathcal{E}$ to exclude self-cycles. Nodes j, k are **adjacent** in $\mathcal{G}$ if either $(j, k) \in \mathcal{E}$ or $(k, j) \in \mathcal{E}$. An edge is **undirected** if $(j, k) \in \mathcal{E}$ and $(k, j) \in \mathcal{E}$, whereas it is **directed** if only one pair appears in $\mathcal{E}$[2]; if this pair is (j, k) then j is called a **child** of **parent** k. The set of parents of j in $\mathcal{G}$ is denoted by $\mathbf{PA}_j^{\mathcal{G}}$. We call $\mathcal{G}$ undirected if all its edges are undirected; conversely, $\mathcal{G}$ consisting only of directed edges is directed. A **mixed** graph consists of both directed and undirected edges. The **skeleton** of any directed or mixed graph $\mathcal{G}$ is an equivalent graph with all directed edges replaced by undirected ones. A **fully connected** graph $\mathcal{G}$ is one where all pairs of nodes are adjacent. A (directed) **path** is a sequence of nodes connected by (directed) edges. A **partially directed acyclic graph** (PDAG) is a mixed graph such that there is no pair (j, k) such that there are directed paths from j to k and vice versa. Then, $\mathcal{G}$ is a **directed acyclic graph** if it is a PDAG and is directed. Two graphs are *Markov equivalent*, or belong to the same *equivalence class*, when they involve the same sets of *d-separations* (see Section 2.1.1). A **completed PDAG** (CPDAG) can encode such a class of graphs, in which undirected edges mean that the graphs

[2]Clarification: two directed edges between the same nodes that have opposing directions result in a single undirected edge.

within the class may contain a directed edge in either direction; directed edges denote agreement in edge direction in subsumed graphs.

Functional. Another perspective on causal graphs is Structural Causal Models, also called Structural Equation Models (SEMs). In practice, they still encode the same information as the graphical approach, but provide an alternative view via equations. Thus, SCMs can be seen as a complementary technique, not the opposite one. SCMs view the causal relationships among features as functions, which are referred to as *causal mechanisms*. The general idea is that each variable is a function of its parents, including the unobserved variables in the form of noise terms. This can be defined as follows. Let us consider a vector of random variables $\mathbf{X} = (X_1, \ldots, X_p)$ and their corresponding noise terms $E = (\epsilon_1, \ldots, \epsilon_p)$ generated according to an unknown DGP leading to joint distribution $\mathcal{L}(\mathbf{X})$. The node $j \in \mathbf{V}$ represents random variable X_j and the edge between nodes j and k in $\mathcal{E}$ is directed if and only if X_k is used in the DGP to generate X_j. Then, each causal mechanism can be written as:

$$X_j = f_j(X_{\mathbf{PA}_j^{\mathcal{G}}}, \epsilon_j) \text{ for } j = 1, \ldots, p \tag{5.1}$$

5.3 Hyperparameters in Causal Structure Learning

5.3.1 Bivariate Example

An illustrative example, strongly inspired by [110], is the classic *cause-effect pairs* challenge [42] that consists of two (synthetically generated here[3]) variables X and Y, with the goal of establishing the existence and direction of the causal link between them ($X \to Y$, $X \leftarrow Y$, no link) given only observed data. One possible solution is to fit two regressors $y = f(x)$ and $x = g(y)$, and predict the causal direction based on

[3] $X = \epsilon_X$, $Y = \sin(X) + \epsilon_Y$, $\epsilon_X, \epsilon_Y \sim \mathcal{N}(0, 1)$.

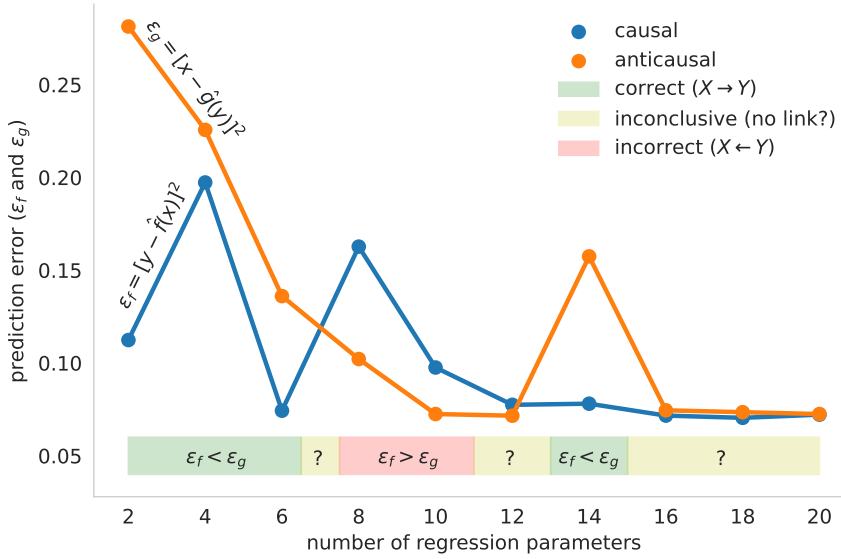


Figure 5.2: The number of allowed regression *parameters*, a **hyperparameter**, clearly affects the prediction error of the two models and can determine the predicted causal direction (see decision colours at the bottom). The algorithm predicts $X \to Y$ if $\epsilon_f < \epsilon_g$, $X \leftarrow Y$ if $\epsilon_f > \epsilon_g$, and is inconclusive otherwise. Note that the true causal direction is unknown in practice.

the lower prediction error of the two models ($\epsilon_f = [y - \hat{f}(x)]^2$ and $\epsilon_g = [x - \hat{g}(y)]^2$; no link if the errors are comparable). As shown in Figure 5.2, changing the hyperparameter that controls the number of regression parameters can result in a different causal direction being predicted. This is precisely what constitutes the problem since the true DGP and the correct hyperparameter value are unknown.

Observation 1. Incorrect hyperparameter values can cause prediction mistakes.

Observation 2. There might be more than one correct and incorrect hyperparameter choice.

The problem grows in complexity as the number of graph nodes and edges increases. This is because each edge is a potentially different function to approximate that will require a different hyperparameter value (function complexity) to obtain the correct answer. In addition, many algorithms provide multiple hyperparameters to tune, making even more room for further mistakes, and effectively increasing the chance of hyperparameter misspecification. In fact, even the bivariate example

can involve more hyperparameters by, for instance, introducing a threshold such that the algorithm predicts 'no link' if $|\epsilon_f - \epsilon_g| < threshold$. To this end, we make the following claim and set up the following key definition:

Claim 1. The existence and direction of an edge in the predicted causal graph strongly depend on the algorithm's hyperparameters.

Definition 2 (Hyperparameter Misspecification)**.** Mistakes in predicted causal graph structure arising from incorrect hyperparameters.

Note that in practice HP misspecification may be not the only source of prediction mistakes, as not all learning algorithms are guaranteed to converge to optimal solutions.

5.3.2 General Form of the Problem

Let $\mathbf{X} \in \mathbb{R}^{n \times p}$ represent IID data of n observations and p features. Furthermore, let $A \in \{0,1\}^{p \times p}$ be a binary **adjacency matrix** of directed graph $\mathcal{G}(A)$ such that the (binary) graph edges $a_{jk} = 1$ if $(j,k) \in \mathcal{E}$ and $a_{jk} = 0$ otherwise (see also Section 5.2 for notation). A **weighted** adjacency matrix $W \in \mathbb{R}^{p \times p}$ is defined such that its corresponding binary edge $A(W)_{jk} = 1$ if the weighted edge $w_{jk} \neq 0$ and zero otherwise, which results in a weighted graph $\mathcal{G}(W)$. Let us also define distance $d(A,B)$ between two adjacency matrices A and B such that $d(A,B) = 0$ if and only if $\mathcal{G}(A) = \mathcal{G}(B)$ are the same graph; otherwise $d(A,B) > 0$. In practice, d can be realised with any specific distance measure of choice, such as Euclidean or sum of squared differences[4]. In our discussion here, this specific choice does not matter as long as the distance metric measures the difference between the two matrices.

From now on, let us denote A as the true adjacency matrix and $\hat{A}$ its estimate obtained by a computer program P (think algorithm) from IID data X using program options O so that

$$\hat{A} = P(O, X), \tag{5.2}$$

[4] $d(A,B) = \sum_j^p \sum_k^p (a_{jk} - b_{jk})^2$

where O generally involves the user specifying algorithm S and hyperparameter values H. Therefore, given K user-specified candidate programs $P = \{P_1, \ldots, P_K\}$ and

$$\hat{A}_k = P(S_k, H_k, X), \tag{5.3}$$

where S_k and H_k are the algorithm and hyperparameter choices associated with program candidate k, then the best program is

$$k^* = \underset{k \in \{1,\ldots,K\}}{\operatorname{argmin}} \; d(A, \hat{A}_k), \tag{5.4}$$

Note that k^* is never identifiable unless A is known. Furthermore, identification of $\mathcal{G}$ does not guarantee $\hat{A}_{k^*} = A$ as $\hat{A}_{k^*}$ depends on algorithms S and their ability to identify A, as well as the choice of their hyperparameters H.

More generally, when considering different algorithms S and hyperparameters H, Equation (5.4) is the standard **model selection** problem, whereas if the choice of algorithms is fixed to a specific value, leaving hyperparameters as the only variable, the task reduces to **hyperparameter tuning**. In practical terms, obtaining the distance $d(A, \hat{A})$ is not feasible, as the true matrix A and its corresponding graph $\mathcal{G}(A)$ are inaccessible outside simulated environments. As algorithms can vary substantially in design, the appropriate way to compare them requires the use of distance measures that incorporate the ground truth A. This renders model selection impractical in CSL problems. Tuning hyperparameters of a single algorithm might be feasible by comparing its relative scores across explored hyperparameter values.

5.3.3 Common Hyperparameters

Despite differences in algorithms, many of them share similar hyperparameters. Commonly used ones are briefly described here.

Significance level of independence tests. Refers to the *p-value* of independence tests and the desired level of certainty. Decreasing the value (increasing certainty) will usually result in fewer predicted edges. Often named as *alpha* and incorporated by traditional and pairwise algorithms.

Sparsity. A penalty term that encourages sparser solutions. Higher values result in fewer predicted edges. Similar in mechanism to *L1 regularisation* which discards less relevant features. Often employed by regression-based solutions, especially if they perform some form of feature selection.

Model complexity. A penalty that encourages simpler models to avoid overfitting (*L2 regularisation*). As shown in the example in Section 5.3.1, its influence on the final prediction is complicated. This usually applies to solutions that model the assumed form of SEMs.

Post-prunning threshold. Many SEM-based methods output weighted adjacency matrices W that need to be converted to the binary form of A (see notation in Section 5.3.2 above). This is usually done by applying a threshold below which all edges are set to zero. That is, $a_{ij} = 1$ if $w_{ij} > w\_thresh$; 0 otherwise.

Note that *alpha* and *w_threshold* are algorithm agnostic and can be transferred between methods, whereas the other two hyperparameters may differ in value between algorithms.

5.4 Experiments

Since our analysis in Section 5.3.1 is based merely on a simple and artificial example, our next step is to study the influence of hyperparameters more rigorously in a more general setting. We devise a set of experiments consisting of diverse data sets of various sizes and difficulty (Section 5.4.1), processed by a representative set of CSL algorithms (Section 5.4.2). Different hyperparameter selection scenarios are also detailed (Section 5.4.4).

The experimental framework is implemented through Benchpress [179], a benchmarking platform to evaluate CSL algorithms. Performances of all algorithms are collected from Benchpress, followed by some mild post-processing of the

results to suit our analysis of hyperparameters. All numerical experiments can be fully replicated using the code and data that are available online at: https://github.com/misoc-mml/hyperparams-causal-discovery.

5.4.1 Graphs and Data

Simulations

We follow recent literature on CSL when it comes to simulating different DGPs. The simulation procedure starts with generating a random DAG $\mathcal{G}$ with p nodes and d edges, built according to a random graph model. The resulting graph is binary, with $A \in \{0,1\}^{p\times p}$. Next, IID data $\mathrm{X} \in \mathbb{R}^{n\times p}$ are sampled from a simulated SEM of choice, with n being the sample size. Each individual combination of settings is repeated for 10 seeds and forms a single experiment. In our experiments, we explore $p \in \{10, 20, 50\}$, $d \in \{1p, 4p\}$, and $n \in \{200, 1000\}$. Included random graph models are Erdös-Rényi (ER) [197] and Barabási-Albert [198], with the latter also known as scale-free (SF). We also explore $n = 10,000$ but only for sparse ER graphs with $p = 50$ nodes due to computational limitations. As for explored SEMs, we include the following:

Linear Non-Gaussian (gumbel). $X = XW^T + z \in \mathbb{R}^p$, with $W \in \mathbb{R}^{p\times p}$ as edge weights assigned independently from $U([-2, -0.5] \cup [0.5, 2])$ and based on A. Noise z follows the Gumbel distribution $z \sim \mathrm{Gumbel}(0, I_{p\times p})$.

Nonlinear Gaussian (gp). $X_j = f_j(X_{\mathbf{PA}_j^{\mathcal{G}}}) + z_j$ for all $j \in [p]$ in the topological order of $\mathcal{G}$. Noise z_j follows Gaussian distribution $z_j \sim \mathcal{N}(0, 1)$, $j = 1, \ldots, p$. Where functions f_j represent a draw from a Gaussian process with a unit bandwidth radial basis function kernel.

Note that both settings have been shown to be identifiable. That is, linear non-Gaussian additive models [50] and nonlinear additive models [51].

Real Datasets

We also tested CSL algorithms against real or semi-real datasets. The most popular ones in the literature are *protein signaling* (Sachs) and *SynTReN*.

Protein signaling (Sachs) comes from [199] which measures protein and phospholipid expression levels in human cells. The ground truth causal graph has been established and accepted by the experts in the field. We use the second dataset that is already logged and standardised and consists of $n = 902$ observations, $p = 11$ nodes and $d = 17$ edges.

SynTReN is a generator of synthetic transcriptional regulatory networks and related gene expression data that simulate a real experiment [188][5]. We use the same data as in [60], which consist of 10 random seeds, $n = 500$ samples and $p = 20$ nodes.

5.4.2 Causal Structure Learning Algorithms

We consider in our setup the following learning algorithms (see also Section 2.1.3 for a refresher). Due to high computational demands, we only focus on well-established and seminal algorithms that, in our view, effectively represent different classes of solutions. More details about the HPs involved can be found in Appendix B.3.1 and B.3.2. Note the algorithms may have different termination criteria due to design differences, but they all produce (CP)DAGs (see Section 5.2).

- **PC** [200]. Peter and Clark algorithm. A constraint-based approach that starts with a fully-connected undirected graph and removes edges based on conditional independence tests. Next, it attempts to orient as many of the remaining edges as possible. The result is a CPDAG.
  Hyperparameters: *alpha* (significance level for conditional independence tests).

- **FCI** [201]. Fast Causal Inference. Constraint-based. An important generalisation of PC to unknown confounding variables.

[5] http://bioinformatics.intec.ugent.be/kmarchal/SynTReN/index.html

Hyperparameters: *alpha* (significance level for conditional independence tests).

- **FGES** [202]. Fast Greedy Equivalence Search. Optimised and parallelised version of the original score-based GES algorithm [46]. It starts with an empty graph and adds an edge that yields maximum score improvement until no significant score gain is achieved. Then it removes edges in the same greedy manner until a plateau.
  Hyperparameters: *penaltyDiscount* (sparsity penalty).

- **LiNGAM** [50]. Linear Non-Gaussian Acyclic Model. Assumes linear SEMs and non-Gaussian noise that enters additively: $X_j = \sum_{k \in \mathbf{PA}_j^{\mathcal{G}}} w_{jk} X_k + \epsilon_j$.
  Hyperparameters: *max_iter* (FastICA [203]), *thresh* (post-prunning threshold).

- **ANM** [51]. Additive Noise Model. Assumes nonlinear SEMs and additive noise: $X_j = f_j(X_{\mathbf{PA}_j^{\mathcal{G}}}) + \epsilon_j$.
  Hyperparameters: *alpha* (significance level for the independence test).

- **CAM** [55]. Causal Additive Models. Assumes a generalised additive noise model with additive noise and functions: $X_j = \sum_{k \in \mathbf{PA}_j^{\mathcal{G}}} f(X_k) + \epsilon_j$.
  Hyperparameters: *cutoff* (variable selection threshold).

- **NOTEARS** [56]. Score-based continuous DAG optimisation with a smooth acyclicity regularisation term. Assumes linear SEMs with additive noise.
  Hyperparameters: *lambda1* (sparsity term), *max_iter* (optimisation steps) and *w_threshold* (post-prunning threshold).

- **NOTEARS MLP** [36]. Nonlinear extension of *NOTEARS* by incorporating the Multi-Layer Perceptron (MLP). Assumes nonlinear SEMs with additive noise.
  Hyperparameters: *lambda1* (sparsity term), *lambda2* (regularisation strength), *w_threshold* (post-prunning threshold), *hidden_units* (number of units in the hidden layer).

Many traditional algorithms, such as PC, FCI and FGES, make the standard set of assumptions that involve sufficiency, faithfulness and Markov condition (see Section 2.1.2). These, however, are often not enough to identify a unique DAG as a solution, which is a major drawback of these methods (they output CPDAGs; not all edges oriented). Making assumptions about distributions and functional forms of the DGP seems to be critical to overcoming this issue (all methods above except for PC, FCI and FGES output DAGs; all edges oriented).

5.4.3 Evaluation

We compare algorithms' performances across three commonly used metrics: *structural Hamming distance* (SHD), *false positives* (FPs), and *false negatives* (FNs). All three accommodate for the fact that the ground truth is always a DAG but some of the incorporated algorithms output CPDAGs. The implementation of SHD we use counts not only the number of edge additions, removals and reversals but also the edge orientations, needed to turn the predicted graph into the true DAG. FPs and FNs count false edges and are calculated based on graph skeletons. For this purpose, predicted and true graphs are converted to skeletons to obtain the two metrics. This allows us to include methods that output CPDAGs even though the primary focus of this study is DAG recovery. See below for more detailed definitions of the metrics.

The evaluation metrics we incorporate are provided via Benchpress [179, appendix A.1.]. For the convenience of the reader, we briefly describe here those that are useful for this study.

SHD

Let us define E and E' as a set of edges of the true and predicted DAG respectively. Then, for $e \in E'$, true positives (TP) and false positives (FP) are assigned as follows:

$$TP(e) = \begin{cases} 1 & \text{if } e \in E \text{ and correctly oriented} \\ 0.5 & \text{if } e \in E \text{ and incorrectly oriented} \\ 0 & \text{otherwise} \end{cases} \tag{5.5}$$

$$FP(e) = \begin{cases} 1 & \text{if } e \notin E \\ 0.5 & \text{if } e \in E \text{ and incorrectly oriented} \\ 0 & \text{otherwise} \end{cases} \tag{5.6}$$

where TP and FP are sums of all TP(e) and FP(e) scores respectively. The *structural Hamming distance* (SHD) aggregates the number of additions, removals and reversals in predicted edges so they match the true ones ($E = E'$). It can be defined as:

$$SHD = |E| - TP + FP \tag{5.7}$$

Note the SHD defined as above allows evaluating mixed graphs, that is, comparing DAGs to CPDAGs. If, for instance, a predicted undirected edge exists in E but is supposed to be directed, it will result in $TP = 0.5$ and $FP = 0.5$, ultimately leading to $SHD = 1$. This shows that the need to orient an undirected edge is treated equally as the need to add, remove or reverse an edge so $E = E'$. Such evaluation puts algorithms outputting CPDAGs at a disadvantage compared to DAG-only methods. We justify it on the grounds that the main focus of this study is DAG recovery, hence any predicted undirected edge is treated as any other mistake. The ability to evaluate mixed graphs is an important feature of this study as it allows us to compare algorithms outputting CPDAGs and DAGs.

False Positives and Negatives

The *false positives* (FPs) and *false negatives* (FNs) that we use as standalone performance metrics differ from those defined as part of SHD above. Crucially, the FP and FN metrics we use are always computed based on graph skeletons (i.e. all edges undirected). To achieve this, all directed edges of a predicted or true graph are converted to undirected ones. This modification makes the comparison between algorithms that output CPDAGs and DAGs more fair. Once the graphs in question are converted to skeletons, we can define the metrics as follows.

Let us define E and E' as a set of undirected edges of the true and predicted graph skeleton respectively. Then, for $e \in E$ and $e' \in E'$, false positives (FP) and false negatives (FN) are assigned as follows:

$$FP(e') = \begin{cases} 1 & \text{if } e' \notin E \\ 0 & \text{otherwise} \end{cases} \tag{5.8}$$

$$FN(e) = \begin{cases} 1 & \text{if } e \notin E' \\ 0 & \text{otherwise} \end{cases} \tag{5.9}$$

where FP and FN are sums of all FP(e') and FN(e) scores respectively.

5.4.4 Hyperparameters

All incorporated learning algorithms have at least one HP. We collect performances of algorithms across all HP combinations (exhaustive grid search; see Appendix B.3.1). Note that while many HPs are defined on continuous spaces, we search over a pre-defined set of points chosen to cover the space as completely as possible. To better understand the influence of HPs on CSL performance, we experiment with four different HP selection strategies described below.

BEST. To simulate the choice of the best HPs (as if we had access to the HP oracle), we pick HP values that achieved **the lowest** metric value in that particular data setting. Each data setting can have a different set of the best HPs.

WORST. Identified similarly to 'best' except the criterion here is **the highest** metric value.

DEFAULT. Default HP values recommended by the authors of an algorithm. See Appendix B.3.2.

SIM_MEAN. An alternative to 'default'. We found a single set of HP values per algorithm that achieved **the lowest average** metric value across all simulations. These are *simulation-derived* default values. See Appendix B.3.1 for identified values.

5.4.5 Results

Presented results employ the following naming convention with respect to the DGP: *graph_p* (number of nodes), *graph_d* (edge density), *graph_type* (graph models; ER or SF), *data_n* (sample size), *data_sem* (SEM type; gumbel or gp). Error bars are standard errors unless stated otherwise. Note only the most important results are presented in this section. The rest of the results, that do not change conclusions, are in Appendix B.1.

Performance Distribution Across Hyperparameters

As per Figure 5.3, all algorithms perform similarly when averaged across all simulations and hyperparameters. This confirms that no single algorithm is the best in all conditions; they rather specialise in solving specific challenges that are ingrained in their design via assumptions (see Section 2.1.2). Some minor deviations from this general observation can be noticed when considering FPs (false positives) only (see FCI, PC and ANM), but they become negligible when considering the main metric (SHD).

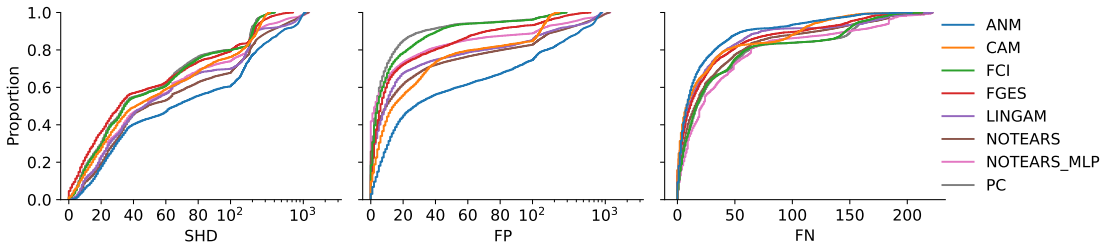


Figure 5.3: Proportions of performances (lower is better) across all HP values and simulations. Interpretation: algorithms perform similarly when averaged over all DGPs and HPs; no algorithm is the best in all conditions.

Performance vs. Hyperparameter Quality

As shown in Figure 5.4, achieved performance is clearly affected by both algorithm and HP choices. Even assuming access to HP oracle (blue colour), selecting different algorithms will have a significant impact on the result (blue bars differ among algorithms). This is because assumptions, either implicit or explicit, about the DGP play a critical role in controlling the performance of each method. It is worth

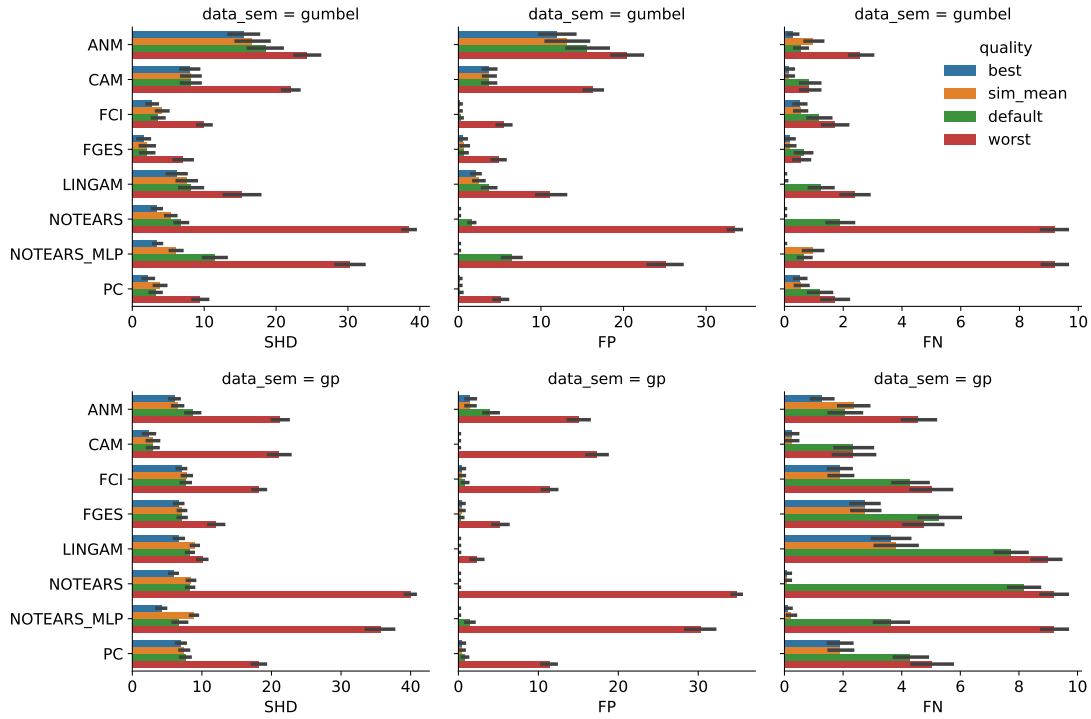


Figure 5.4: Performances (lower is better) for small sparse graphs ($p = 10$, $d = 1$) with linear (*gumbel*) and nonlinear (*gp*) SEMs depending on the **quality** of selected HPs (see the legend). Interpretation: a) fixed HPs (orange and green) perform similarly to the optimal ones (blue), b) algorithm selection is important even with optimal HPs (blue bars differ among algorithms), c) certain algorithms and setups are riskier (differences among red bars), and d) methods with optimised HPs provide only true edges (see blue bars in FPs).

noting that, in contrast to previous studies of HP choice for other causal methods, the absolute performance level achieved by the algorithms is low: no algorithm achieves, nor is very close to, SHD = 0. The complexity of the learning task is such that accurate causal structure discovery currently lies beyond the state of the art, which could be due to limited information within observational data. In this context, fixed HPs (orange and green) seem to be a viable strategy as they are relatively close to the best cases (blue). This shows that HP values transfer well between different DGPs, which can be exploited in practice as a countermeasure to challenging HP optimisation. The differences between simulation-derived and paper-default values (orange and green respectively) are negligible in most cases except for FNs (false negatives) where the former perform better. This indicates that the recommended defaults often prioritise sparsity, which reduces FPs, at

the cost of increased FNs. The worst HPs (red), on the other hand, can result in performances substantially worse than the fixed ones. This shows the risks of HP misspecification, the degree of which clearly varies across algorithms (different robustness), which could be due to again different data assumptions in algorithms or varied degrees of freedom via the algorithm's HPs. Note also how the majority of mistakes under misspecified HPs (red) are due to FPs (red bars in FPs larger than in FNs). However, once HPs are optimised (blue), FPs are mostly eliminated and the remaining mistakes are now due to FNs (blue bars larger in FNs than FPs). This has critical implications for practice: the predicted edges can be trusted (FP small) but the absence of an edge cannot (FN large). Remaining FNs could be also a sign of too-aggressive sparsity/pruning or simply failed identification, with the latter being an indicator for future algorithmic improvements.

Performance vs. Cardinality of Hyperparameters

A speciously interesting comparison that emerges from our study is the relationship between the cardinality[6] of the HPs we explore and algorithm performance. Our study was not set up to explore this issue, so we now clarify what can and cannot be concluded about this relationship from our study. As presented in Figure 5.5, higher cardinalities (81 and 200) lead to only small performance improvements under optimised HPs (blue bars), but they involve significantly higher risks of poor performances if HPs are misspecified (red). This shows that larger HP search spaces should be explored with caution as they provide little gain for a much higher risk of poor performance. Note, however, that different cardinalities presented here may involve different algorithms, making room for coincidental trends. For example, it is unclear given the data if robustness to misspecified HPs is due to HP cardinalities or algorithms, as the two highest cardinalities were explored exclusively with (potentially volatile) neural networks.

[6]We define cardinality as the total number of possible combinations across hyperparameters and values. For example, two HPs, with three possible values each, results in the cardinality of $3 \times 3 = 9$.

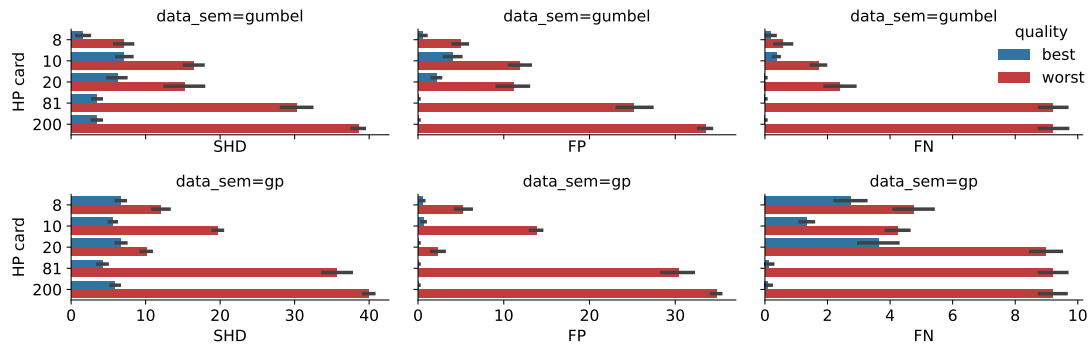


Figure 5.5: The influence of explored HP cardinalities (*HP card*) across algorithms on performance (lower is better), depending on the **quality** of selected HPs (see legend). DGP: small sparse graphs ($p = 10$, $d = 1$) with linear (*gumbel*) and nonlinear (*gp*) SEMs. Note how higher cardinalities lead to small SHD gains under optimal HPs (blue) but at the cost of much worse performances under misspecified HPs (red). Warning: different cardinalities may involve different algorithms (possible coincidental trends).

graph_p	10				20				50			
data_sem	gumbel		gp		gumbel		gp		gumbel		gp	
graph_d	1	4	1	4	1	4	1	4	1	4	1	4
HP best	FGES FCI	FGES	CAM	CAM	FGES	PC N_MLP	CAM	CAM	FGES	PC N_MLP	CAM	CAM
HP sim_mean	FGES PC	FGES CAM ANM	CAM	CAM	FGES	PC	CAM	CAM	FGES	PC	CAM	CAM
HP worst	FGES	FGES PC CAM	LiNGAM FGES	CAM	FGES	PC	LiNGAM	CAM LiNGAM	FGES	PC	LiNGAM	LiNGAM

Table 5.1: Top algorithm choices based on the highest winning percentages, grouped by DGP properties (graph_p, data_sem, graph_d) and quality of selected HPs (best, sim_mean, worst). If there is no clear winner, multiple top performers are reported. Notice how the winners change across DGPs (columns), but also across HPs (rows), showing algorithm selection is nuanced. N_MLP denotes NOTEARS_MLP.

Winning Algorithms vs. Hyperparameter Quality

Previous analysis revealed that the best algorithm choice may depend on specific DGP properties as well as HP choices. To make it clearer, we collect winning algorithms across different DGP properties and HP selection strategies. An algorithm with the lowest SHD in a given setting wins. Table 5.1 presents top performers across different settings based on accumulated winning percentages. The results confirm that no single algorithm wins in all settings. Specific DGP properties may favour certain methods. When looking at how top performers change depending on different HP choices, it is clear that the best algorithm selection depends not only

on DGP properties but also on the type of available HPs. For instance, to minimise the risk of poor performances, one can select algorithms from the 'HP worst' category.

Performance vs. Sample Size

Figure 5.6 confirms that increased sample size generally helps, even with relatively large graphs ($p = 50$), though some algorithms need more data to notice major benefits (see LiNGAM in gumbel and NOTEARS_MLP in gp). Positive effects can be noticed with respect to improved best HP cases (min values) and an increased proportion of good performances (mean values). This case also confirms that relatively large and sparse graphs can be recovered with high accuracy given the right HPs (min values of some green boxes are close to 0).

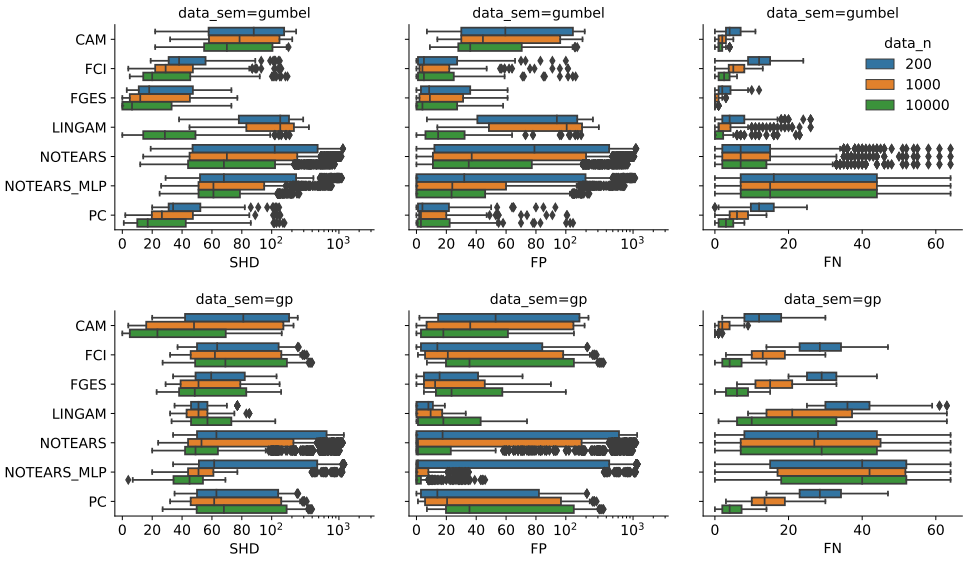


Figure 5.6: The influence of sample size (*data_n*; colours) on performances (lower is better) across all HPs. DGP: sparse ($d = 1$) large ($p = 50$) ER graphs with linear (*gumbel*) and nonlinear (*gp*) SEMs. ANM was excluded due to the long execution time against $10,000$ samples. Note how increased sample size not only improves performances under the best and worst HPs (min and max decrease) but also the proportion of good performances (means decrease as well), suggesting increased sample size helps with HP misspecification.

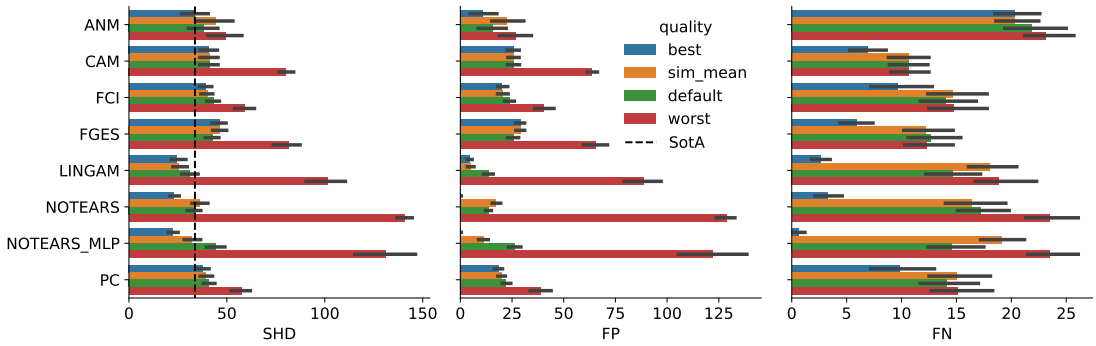
Semi-Synthetic and Real Data

We put our simulation-derived findings to a test by performing CSL on SynTReN and Sachs datasets (Figure 5.7). SHD numbers are compared to SotA performances

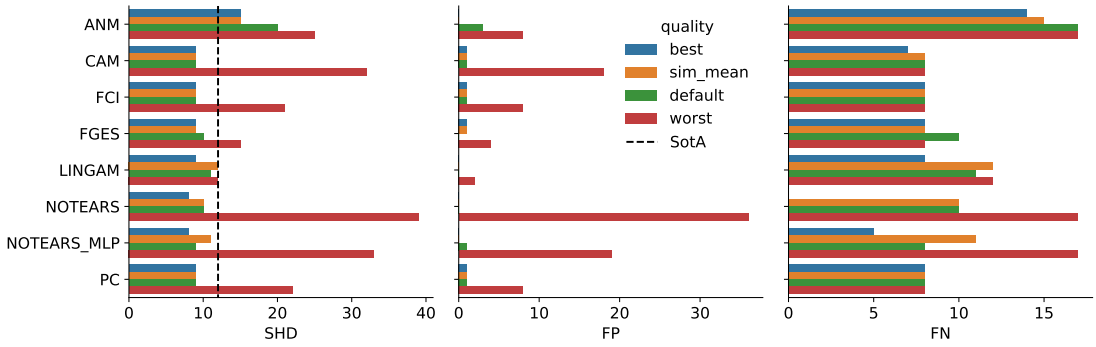
retrieved from [60], which are 33.7 ± 3.7 and 12 SHD for SynTReN and Sachs respectively. Both cases generally confirm that fixed HPs (*sim_mean* and *default*) can work almost as well as the best HPs, and that even the best HPs may not be enough to reach the best possible performance as some algorithms perform better than others under those conditions. It is also clear from both cases that HPs play an important role and, in fact, can decide whether an algorithm reaches or beats SotA. For instance, against SynTReN, both NOTEARS methods and LiNGAM seem to be good options under the best and fixed HPs. But under the worst HPs, NOTEARS methods can be extremely inaccurate, making ANM the safest choice in this case. In the Sachs dataset, this is no longer the case with ANM, showing that the best algorithm pick indeed strongly depends on DGP properties. All algorithms except ANM can, in fact, beat SotA on Sachs data. However, when it comes to robustness to HP misspecification and safety of use, NOTEARS methods appear to be the riskiest, with LiNGAM being extraordinarily robust as it beats SotA even with its worst HPs.

5.5 Conclusion

In this work, we have successfully shown that HPs play an important role in causal structure learning. However, the way HPs influence the methods is somewhat different from recent results from the ML literature. More specifically, [164, 182] found that many learners can reach SotA performance levels with the right HPs, reducing the importance of model class selection. However, in this study, we observe that algorithms still differ significantly in performance even with access to the HP oracle. However, reliable tuning is not always available in CSL, leading to HP misspecification and prediction errors. This is where HPs become important as we showed that different learning algorithms vary in robustness to HP misspecification and that strong performance under the right HPs does not necessarily translate to misspecification robustness. As a consequence, an algorithm



(a) SynTReN dataset. Numbers are averaged across data seeds.



(b) Sachs dataset

Figure 5.7: Performances (lower is better) against SynTReN (top) and Sachs (bottom) datasets depending on the **quality** of selected HPs (colours). Notice that: a) HP values derived from simulations perform well here (orange), and b) the quality of HPs and algorithm choice are both important for beating SotA.

that is the best pick under correct HPs might be a suboptimal choice when its HPs are misspecified; another algorithm with better misspecification robustness might be safer to use, especially in those cases where minimising the consequences of potential misspecification is a priority. Thus, overall, the best algorithm choice may depend not only on the properties of the DGP but also on the quality (as defined in Section 5.4.4) of selected HP values. In terms of secondary findings, default HPs seem to perform surprisingly well in many cases, and hence may constitute a viable alternative to tuning. Moreover, in the case of sparse graphs, predictions with optimised HPs seem to include only true edges (no FPs, some FNs), which has critical implications for practice. Another interesting observation is that relatively large sparse graphs (50 nodes) can be recovered with high accuracy, subject to large sample sizes and the right HPs. It is also important to acknowledge

the possibility that robustness to misspecified HPs might be impacted by the cardinality of explored HPs as larger search spaces may increase the probability of poor performances. Our results show this is indeed possible, but crucially the chances of good performances also slightly rise in this case. This suggests that higher HP cardinalities can be advantageous and disadvantageous, which motivates including the worst performances in this analysis. Furthermore, while we agree that the probability of getting the worst performances is low in practice, it is undeniably greater than zero and hence worth considering.

5.5.1 Recommendations

As for practical advice, we stress the fact that there are no universal answers in CSL; there are many forces at play (DGP properties, algorithms, HPs) that make the choices highly nuanced. Algorithm selection seems to be the most important for performance, though HPs should not be neglected (see Table 5.1 for guidance). Default HPs (from packages or this work) should be a reasonable starting point. For further tuning, one can consider prediction stability across HPs [185, 194, 195], though larger HP search spaces should be considered with care (risk of poor performances). Focusing on "tunable" HPs [132] (i.e. those with greater impact on prediction performance) may help reduce the search space.

5.5.2 Limitations

This study is naturally limited by our choice of explored algorithms, HPs and simulation properties. It is worth noting, however, that we do not intend to identify the best possible learning algorithm or HPs. On the contrary, the objective of this work is to show that the appropriate algorithm choices are nuanced, as also recently shown in the treatment effect estimation domain [26] and that HPs should be part of that subtle decision-making process, further confirming the importance of HPs reported in the literature [164, 180–182]. And while more extensive search spaces are unlikely to negate such conclusions, it is worth pointing out that in our

experiments we explore discrete HP values of continuous search spaces. As for the choice of HPs and values, we believe our setup accurately reflects common practice.

5.5.3 Future Work

As increasing the sample size increases the proportion of good performances across HPs, the next step would be to examine if HP misspecification vanishes with infinite data. More challenging "anterial graphs" [204] with cycles and undirected edges could be another direction. The surprising effectiveness of fixed HPs might be also worth a systematic study, with an emphasis on transfer between DGPs. Further study into the source of robustness to misspecified HPs is also in order (HP cardinality or algorithms). Furthermore, making advanced tuning metrics more accessible could help facilitate better practice. Finally, although some HP tuning metrics are general enough to perform algorithm selection (e.g. [186, 194]), doing so on real-life datasets is still a challenge. One promising direction could be partial validation on graph edges obtained from domain knowledge [189].

Summary

Major findings:

- Causal structure learning algorithms significantly differ in recovery performance even with access to the hyperparameter oracle. Choosing the right algorithm for the problem at hand is still the most important decision to be made.
- Algorithms vary in robustness to hyperparameter misspecification, and strong performance under well-specified hyperparameters does not necessarily translate to robustness under misspecified cases.
- The optimal algorithm choice in ensemble settings requires consideration of the data at hand, but also hyperparameter choices.
- Default hyperparameter values can perform almost as well as near-optimally selected ones.
- The edges in predicted sparse graphs can be trusted as the remaining mistakes are only due to missing edges.

Minor findings:

- Relatively large (50 nodes) sparse graphs can be recovered with good accuracy, assuming a large sample size and correctly specified hyperparameters.
- There are no free lunches in causal discovery, even when assuming access to the tuning oracle.
- Larger sample sizes increase the proportion of good performances regardless of hyperparameters, suggesting they possibly become irrelevant as data size approaches infinity.

6 Discussion

Here we consolidate the main three chapters by revisiting the main discussion points and overlapping topics that connect them. We also comment on the noteworthy findings and contributions of this thesis, including each chapter's involvement to help picture them as a whole thesis. Some important limiting aspects of this work are also discussed.

Contents

6.1 Consolidation

As some selected topics appear in multiple places throughout this thesis, this section discusses them more closely to help understand how the main chapters

relate to each other.

6.1.1 Causal Effect Estimation

Chapters 3 and 4 are both concerned with heterogeneous causal effect estimation (Section 2.2.3), but they approach the topic from different perspectives and hence offer distinct insights and contributions. That is, the former improves CATE estimation by introducing a new method, while the other has a more theoretical focus on hyperparameters and their influence on prediction performance (including CATE estimation). Furthermore, while only one of the chapters explicitly investigates hyperparameters, this topic is in fact highly important in both of them. This is because Chapter 3 also compares performances of existing methods, which requires hyperparameter tuning to ensure fair comparison and meaningful conclusions. However, despite this similarity, there are important tuning differences between the two. Chapter 4 approaches hyperparameter optimisation in a more complete manner as it includes multiple metrics that can be used for hyperparameter selection and crucially also considers optimally selected performances (i.e. oracles). Such considerations make it more theoretical. On the other hand, Chapter 3 is more practical in this regard as it uses a single *observable* (Chapter 4) metric to select the best hyperparameters and naturally cannot access optimal choices. This perspective matches the one of a practitioner.

Different limits in terms of tuning give rise to crucial discrepancies when comparing various estimators and their performances. More specifically, in Chapter 3 we learn that there are non-trivial performance differences due to the choice of the model class, despite performing thorough but practical hyperparameter tuning for each estimator. Chapter 4, however, offers somewhat opposite conclusions that causal estimators perform similarly, subject to optimal hyperparameters. The requirement of optimal hyperparameters is unfortunately unrealistic in practice, making performances obtained in this way only potential and theoretical. This shows that practical tuning, as it is currently done, is very likely to result in larger prediction errors than is theoretically possible. Thus, a remaining open

question is whether the necessary criteria can be identified from the observed data, with influence functions offering one possible way of moving forward due to promising robustness to model choices [171]. This point also shows that the two concluding remarks are not necessarily opposite; they simply consider tuning and performances under different circumstances. In other words, it is still useful to learn that theoretically (in the limit of tuning) many causal estimators can achieve similar performances, while in more challenging and pragmatic settings that are closer to real-world situations, inevitable tuning imperfections (e.g. metric choice [26]) will lead to non-trivial estimator differences. That is, one offers theoretical insights, while the other is more grounded in practice.

More generally, this case also shows that theoretical and practical analyses can lead to rather different conclusions. Both perspectives are certainly necessary to fully comprehend any problem and are arguably rather complementary than opposing. If anything, this case is an example of why any claim should be challenged from theoretical and practical sides, not just one. In addition, it demonstrates that theoretical guarantees, which often imply certain idealistic conditions, do not necessarily translate in full to rather messy practical situations, even to the extent where the conclusions become completely opposite. To clarify, this is not to say that the theoretical works are wrong; they do solve their intended problems they were designed for. The problem is in assumed conditions that may not accurately reflect real data challenges. For example, proving a method converges to the desired solution as sample size approaches infinity is not quite practical if practitioners deal with relatively small samples ($n < 1,000$). Another case is with overlap, where we often assume high common support in the input space between treated and control units (i.e. high overlap), while in practice the coverage might be much lower than anticipated. Some communities also strongly question the ignorability assumption (see Section 2.2.2) that is so central to many estimation methods. Unfortunately, most of the assumed conditions cannot be easily verified with real data, leaving room for expert domain knowledge as a possible alternative at the moment.

6.1.2 Covariate Shift

As covariate shift (see Section 3.2.1) is one of the main issues in effect estimation, it forms yet another topic shared between Chapters 3 and 4. Though the problem affects them in different ways, it all boils down to distributional discrepancies of input features between treated and control units, which leads to fewer samples (or even gaps) in the common support of the inputs across the groups. The consequences of this are two-fold. First, it badly affects modelling and individualised predictions that we are after as point estimates are highly sensitive to sparsely covered data regions. This is covered as part of Chapter 3 and demonstrated in Figure 3.1. Second, sparse data regions also distort performance evaluation, as models can potentially perform well in dense data regions but provide poor predictions in the affected areas that are equally important, but due to error averaging one may fail to detect the problem. Further, inaccurate model evaluation will lead to misspecified hyperparameters as evaluation is a critical part of hyperparameter selection. And as tuning highly influences prediction accuracy, as a result we end up with poor prediction performance. To make matters worse, distorted evaluation also makes the broader model selection much more challenging. Overall, this side of the problem is covered in Chapter 4 and Figure 5.2. Thus, it is clear that covariate shift negatively affects both estimation and model tuning.

Covariate shift is generally a well-recognised problem in estimation, but not so much in model evaluation and tuning. The same seems to be the case in the area of OOD generalisation that deals with more general (i.e. other than covariate) data shifts. Indeed, the OOD literature acknowledges the impact of various distributional changes on prediction performance (e.g. [158, 159]), but does not specifically consider how the shifts may affect the evaluation, tuning and selection of models, indirectly impacting the final performance as well. Given this work showed how shifts in covariates can affect both estimation (Chapter 3) and tuning (Chapter 4), it is not unreasonable to anticipate similar problems under other types of data shifts. Ultimately, even the very best methods will only reduce the errors, as in

the end they will be always limited by the data and information theory. One possible way forward could be to identify regions of weak overlap, for which further data collection would be suggested.

6.1.3 The Role of Hyperparameters

Although all main chapters touch on the topic of hyperparameters in one way or another, analysing them specifically constitutes the core of Chapters 4 and 5. While in both cases we are concerned about the influence of hyperparameters on the estimation performance, the prediction tasks involved are very different; one deals with causal effect estimation (supervised learning; Section 2.2.3) and the other with causal graph recovery (unsupervised learning; Section 2.1.3). This important difference partly explains why the role of hyperparameters, though clearly present in both instances, differs between the two.

That is, as part of Chapter 4 we learnt that HP tuning of ML methods in effect estimation accounts for so much performance change that it makes it more important than model class selection. For clarity, some small performance gains are to be made from selecting an optimal estimator type, but not as large as from optimal tuning. This finding is rather surprising as it implies that most of the estimators explored in this work reach comparable performance levels and hence estimator development might not be as important as better tuning methodologies. Interestingly, Chapter 5 offers somewhat inverse conclusions though in the context of causal discovery – that selecting an algorithm most suitable for the problem at hand accounts for much larger performance gains than tuning. This demonstrates the gravity of data assumptions (see Section 2.1.2) ingrained into CSL methods, and that their role goes well beyond identification results (uniqueness guarantees), such as those based on Gaussian likelihood in [55]. The tight connection to assumptions may also suggest that overall robustness is rather unattainable for such methods. If this is indeed the case, Bayesian approaches may constitute a way forward due to their focus on stating the priors.

However, the above does not mean hyperparameters are irrelevant in causal discovery; their role is simply different from the one seen in causal inference. In CSL, hyperparameters become crucial when we start considering robustness to misspecification, that is, how the prediction performance and choice of best-suited algorithm change between well and poorly-specified hyperparameters. And as CSL is unsupervised in nature, robustness to misspecification only grows in importance as users are more likely to make tuning mistakes (no model evaluation). Thus, overall, hyperparameters prove to be of very high importance in both estimation tasks (causal effects and graphs).

6.1.4 Causality and (No) Free Lunches

Another interesting point that emerges from hyperparameter analysis conducted in Chapters 4 and 5 is the possibility of *free lunches*, or lack thereof. More specifically, Chapter 4 claims that under the right conditions (optimal HP tuning), the model class choice is irrelevant due to very small differences, implying that free lunches[1] are possible under those very specific (admittedly theoretical) circumstances. Such a conclusion is quite astonishing as it goes against the *no free lunch theorem* that is well-established in ML [205] and optimisation [206]. Interestingly, a similar analysis done in the context of causal graph discovery (Chapter 5) showed the opposite – no free lunches.

One possible explanation for such discrepancy between the two strands of work is perhaps the maturity level of the subfields. For one, treatment effect estimation methodologies date back to at least 1974 and Rubin's potential outcomes framework [11] (see also Section 2.2.4), effectively accumulating 50 years of development that resulted in multiple causal estimators now so strong that after identifying the right setup (i.e. tuning) all are comparable. Causal discovery, on the other hand, is a much younger area of research, with some serious treatments of the problem starting only

[1]This implies it is possible to choose any causal estimator and achieve a strong predictive performance, which goes against the logic of selecting a method specifically suited for the data problem at hand. Eliminating this choice, while still getting strong performances, is what constitutes a free lunch here.

in the 1990s [200, 201]. This almost 20 years' worth of research difference could partly explain why CSL algorithms have not yet reached the required level of development to perform universally well and enable free lunches as in effect inference. Perhaps, and hopefully, we will witness causal graph learning greatly improving over the next 20 years, though it may take considerably longer to achieve results similar to effect estimation due to an immensely harder challenge that is unsupervised in nature.

6.1.5 Performance Evaluation

An element found across all main three chapters is the performance evaluation of many methods against multiple datasets and metrics. This approach, also known as *benchmarking*, is central to this work as it provides strong empirical evidence for the claims that would be otherwise difficult to support with theoretical work. For example, Chapter 3 provides extensive benchmarking across many existing causal estimators to prove the effectiveness of the proposed method (Section 3.5.3). While proving certain desirable statistical properties (e.g. convergence, consistency) could explain how the method works in the infinite data regime (and subject to other strict data assumptions), such guarantees do not necessarily hold when sample sizes are finite. Furthermore, it is not unreasonable to assume that estimators can behave quite differently in finite and infinite regimes (see also conflicting conclusions between optimal and practical conditions discussed in Section 6.1.1). On top of that, data in all real-world applications are always limited and may not necessarily exhibit the strict properties assumed during theoretical derivations. Thus, testing methods in more practical settings that benchmarking provides carries a lot of weight in terms of pragmatic expectations of the method's performance levels. This is not to diminish the importance of theory; both strands have their place in science. For instance, theoretical work certainly helps position new developments within the existing body of knowledge. However, it would be not of much use (especially to practitioners) to prove the new method in Chapter 3 has certain theoretical properties if it cannot perform well in practice as compared to existing alternatives.

The above point also touches on a broader problem of developing methods based on data assumptions that cannot be accurately detected. Most methods that follow this development approach solve their intended tasks well, as can be observed with strong performances achieved within simulations that also incorporate said assumptions. The main issue, however, is that we do not know which of the assumptions apply to the real data at hand. For example, as per Chapter 4, causal estimator choice might not be as important if we tune them well, but the evaluation metric we choose for tuning makes a difference. Each penalises different aspects of modelling errors. Which one should we select? Another example, this time from Chapter 5, is that choosing a CSL algorithm will make a difference. Each incorporates different assumptions about the DGP. Which is the right one for our data problem at hand? Ideally, we would answer both questions by detecting DGP's properties to see which data assumptions hold in our real dataset, and select the method that employs them. Unfortunately, this is infeasible, data assumptions cannot be detected with current tools. Whereas current evaluation metrics do not capture the full picture (i.e. metrics are biased towards certain estimators, or are only proxies for true targets). And as a result, such choices require domain expertise as of today.

Benchmarking is also a viable strategy to explore problems for which theory is not developed enough yet to lend satisfactory answers. This is the case with Chapters 4 and 5 which investigate the role of hyperparameters in causal estimation. While theoretical fundamentals in effect estimation are relatively well-developed [43] (also Section 2.2.3), how exactly hyperparameters enter the problem is still not well understood, and the fact that Chapter 4 unveils some very surprising findings (i.e. free lunches; see Section 4.5) only adds weight to this statement. In fact, given the analysis in [192], it can be argued that the way hyperparameters interact with loss functions, in general, does not fit current theories, seriously limiting possible theoretical contributions to hyperparameter analysis at the moment. Not to mention that the aforementioned study in [192] is itself empirical in nature and its design is heavily based on performance evaluation across various experimental setups. Furthermore, as for causal discovery, it can be argued that we are still far from

solving the problem to a satisfactory degree (e.g. sparse graphs of max 10 nodes; see Section 5.4.5) even experimentally, let alone have a solid fundamental theory. Adding hyperparameters to the equation only increases the complexity of an already poorly understood problem. In such circumstances, where theory is of not much help, empirical benchmark-focused studies are often the only means of progression, and this was the case for Chapters 4 and 5 for the above-mentioned reasons.

Despite this immense utility, benchmarking approaches are at times unfairly accused of triviality or lack of technological innovation. However, the art of benchmarking is in a careful study design and execution of experiments, so it is possible to understand problems from perspectives never imagined before while at the same time balancing inevitable computing constraints. And if such in-depth systematic analyses yield critical new insights, they are anything but trivial. Moreover, the strong and rapidly growing track record of unique insights provided by benchmarks, a collection this work strives to support and contribute to, speaks for itself. Some notable examples include benchmark-based reports published in highly regarded venues, such as AISTATS [181], NeurIPS [207, 208] or Journal of Causal Inference [101]. In fact, the need for benchmarking has been already recognised by certain top conferences in the field by, for instance, explicitly mentioning benchmarking as a submission topic[2] or even creating a dedicated submission track[3].

6.2 Main Questions and Findings

The research questions and findings initially presented in Chapter 1 are discussed here in more detail. Specifically, how the findings (denoted with F; Section 1.6) offer answers to the previously raised questions (Q; Section 1.5), and through which main chapters.

[2] https://www.cclear.cc/2024/CallforPapers

[3] https://neurips.cc/Conferences/2023/CallForDatasetsBenchmarks

6.2.1 Main Challenges vs. Performance

What are the challenges of causal estimation from observational data? (Q1). While details may differ between causal inference (Section 2.2) and discovery (Section 2.1), the fact that observational data are always in one way or another incomplete (see Section 2.4.1) constitutes the main problem in causal estimation (F1). Some reasons behind it are ingrained into our reality and hence inherently unavoidable, some are potentially addressable through better methodologies.

The constraint that we can observe only one outcome per individual is natural yet fundamentally limiting for data analysis (see Section 2.2.3). Once a selected intervention is applied, we will never learn what would have happened to the individual were they subject to a different intervention. And while collecting data across people may help slightly by administering different treatments to similar subjects, no two individuals are ever truly the same. This is problematic for both causal tasks.

In causal inference, as we learn in Chapters 3 and 4, this issue manifests itself in the form of missing counterfactuals. As a result, computing truly individual effects is inherently impossible; we never observe outcomes for all treatments for the same individual. A way around this problem is to calculate effects within groups of similar units, with similarity captured by a finite set of variables. For example, subjects can be clustered by age, occupation, nationality, etc. This way of approaching things gives rise to conditional average effects (Section 2.2.3). These can get increasingly more individualised by enlarging the conditioning set, but will never become truly individual as this arguably requires a conditioning set of possibly infinite size. Despite this unavoidable limitation, conditional average quantities are still much more informative than plain average parameters obtained across the entire population. And dealing with such individualised estimates constitutes the basis for today's causal effect estimation research field.

Furthermore, the aforementioned strategy of assigning different treatments to similar subjects as a way of overcoming the fundamental problem of missing counterfactuals is also the source of non-trivial issues with observational data. This is because, in the observational setting, we do not control the treatment assignment, just simply collect more data in the hope of better coverage across all treatments that will mimic an otherwise infeasible experiment. However, the lack of control over randomisation leads to selection biases (see Section 2.4.1), which are not necessarily due to data collection mistakes, but simply difficulties in obtaining large quantities of certain subpopulations (e.g. young smokers). As a result, the data exhibits discrepancies in the response function outside the common support of the input variables – covariate shift (F2). As we learn in Chapter 3, this specifically hurts individualised predictions, as point estimates are naturally sensitive to missing overlap. There are usually two ways of overcoming this issue. One is to enrich the data with experimental samples to improve the estimates (as done in [209, 210]), but such data is rarely available. Another way points to the advancement of estimation methods that can withstand such challenging conditions, with our new data augmentation method presented in Chapter 3 being one such example.

In causal discovery, as we learn in Chapter 5, incomplete data means not all interventions are explored for all units. This, in turn, reduces per-variable variance which is critical to accurately establish causal relationships between variables (F3). Such a shortcoming in data is extremely difficult to overcome algorithmically, and the larger and more dense the graphs, the harder the task is. One can even argue that there is a practical limit in terms of graph size that can be recovered accurately from such limited observational data. Our results in Chapter 5 indeed show that graphs with up to 20 nodes can be recovered accurately with a reasonable amount of data ($n = 1,000$), but only assuming sparse connections. However, 50-node (sparse) graphs require an order of magnitude more data ($n = 10,000$) to make the predicted graphs accurate enough. Assuming the data requirement trend continues, the large graphs that are often of interest [202] would need millions of samples for correct predictions – an impractical requirement for practitioners that often have

relatively small datasets ($n < 1,000$). An alternative way of improving things is to supply the observational data with additional interventions. This is of course not always feasible due to the unavailability of experimental data, but it clearly helps with the data-size-to-performance ratio [211, 212].

6.2.2 Main Challenges vs. Hyperparameters

How do identified challenges affect performance evaluation and hyperparameter tuning? (Q2.) All tuning problems take root in performance evaluation that is far from trivial in causal estimation. This is because the identification of the best hyperparameter values across all considered candidates heavily depends on evaluation. That is, the set of hyperparameter values is selected if it achieves the best score across all other combinations of values that are part of the hyperparameter search. But if the evaluation is flawed, suboptimal hyperparameter values will be chosen. Performance evaluation for tuning purposes is generally challenging in causality, but the details differ between the estimation tasks.

In causal inference, as we learn in Chapter 4, covariate shift affects not only individualised estimates (Section 3.5.3) but also model evaluation (F2; Section 4.3). The underrepresented, or entirely missing, subpopulations affected by selection bias (Section 2.4.1) are where the model's performance is misjudged. Consequently, the hyperparameter values that work well on given (broken) observations might be unsuitable for the true DGP that is unbiased. Another issue is rooted in the fundamental problem of missing counterfactuals; we care about the accuracy of effect estimation, but because individual effects are not observed, we cannot measure prediction errors on them. While such perfect metrics are used for benchmarking purposes that take advantage of simulations, such direct quantities cannot be used in real scenarios and tuning. Goodness-of-fit metrics are seemingly attractive alternatives, but as we learn in Chapter 4, those have serious shortcomings as they do not reflect the model's effect estimation capabilities well enough. This motivates the development of specialised causal metrics (e.g. [87, 171]), which clearly do make

a difference as our results in Chapter 4 show performance differences (and hence different hyperparameter choices) across various metrics used for tuning.

As for evaluation in causal discovery (Chapter 5), the situation is even more challenging as CSL is completely unsupervised; the true causal graph is never known. Without the ground truth, accurate performance evaluation is feasible only in simulations. As a result, tuning hardly exists in practice (F3), at least in the conventional sense where the main performance metric of interest would be used for tuning (similar problem to that of effect estimation above). Despite this immense challenge, there have been attempts to derive metrics that would indirectly measure causal graph learning accuracy without the ground truth. Some ideas are specific to tuning and centred around the stability of predictions across explored hyperparameter values [194, 195], though there are no guarantees that the most stable prediction is the correct causal graph. The latest efforts seem to focus on regression accuracy where the structural modelling respects the predicted causal graph structure [186, 196]. Here, the main limitation seems to be the fact that sufficiently (over-)parametrised causal and anti-causal regression models are indiscernible based on only goodness-of-fit metrics [110] (see also Section 5.3.1). A different direction proposed in [189] looks particularly promising as it performs model validation on parts of the graph known in advance by domain experts. While such expertise might be difficult to access, the evaluation, even though limited to known graph parts, directly reflects the type of performance that is of main interest (existence and direction of edges).

6.2.3 Hyperparameters vs. Individual Performance

How does hyperparameter selection affect the performance of causal estimation methods? (Q3.) Through chapters 4 and 5, we learn that the choice of hyperparameters strongly impacts the performance of individual estimation methods. It is expected to some degree as hyperparameters can often control the bias and variance aspects of a model. For example, in linear regression, the analyst can tune the number of coefficients that will control allowed degrees of freedom, but can also

add L1 and L2 regularisation terms that will prevent the model from overfitting. Similar mechanisms are available in most ML methods via hidden neurons and layers in NNs, maximum tree depth in regression trees, and so on (see Section 2.5). As a consequence, one can expect that hyperparameters do have a role in causality as well due to its reliance on ML. But how exactly, and to what extent, do hyperparameters influence the final performance of causal methods? Similarly to the discussion found in the previous two subsections above, the answer is nuanced and depends on the causal task at hand.

Chapter 4 claims that hyperparameters are critical for performance in causal inference in two major ways (F4). First, hyperparameters are *necessary* to achieve SotA as the results show that default hyperparameters perform much worse than after tuning (Figure 4.2). Second, hyperparameters are *sufficient* to attain SotA as even standard estimators can achieve or beat SotA levels with the right hyperparameter values (Figure 4.3). This is a testament to the importance of hyperparameters and the high sensitivity of modern ML methods to them. Such results also support growing evidence in the wider literature on the importance of hyperparameters in achieving SotA [180–182].

However, the caveat is that for the above necessity and sufficiency effects to occur, one needs a near-optimal tuning mechanism (i.e. hyperparameter oracle). Such an idealistic mechanism is unfortunately not available in practice as we can see in our results a significant performance gap between the tuning oracle and currently available evaluation metrics (i.e. the metrics are suboptimal; Figure 4.4). In addition, performance differences when using different metrics for tuning purposes are also non-trivial. This is clear evidence of how performance evaluation is important for tuning – by only changing the metrics we can observe non-trivial performance differences. Thus, the overall practical recommendation is to carefully consider hyperparameter tuning and evaluation metrics used for that purpose as both will strongly impact the final performance. It is also worth pointing out that the advice applies not only to practitioners but also to researchers, so they carefully consider their experimental

details when comparing methods in their studies (i.e. all included methods should be carefully tuned for a fair comparison, ideally with the same metrics if possible, not to mention enough transparency for a successful study replication).

The role of hyperparameters in causal discovery, as we see in Chapter 5, is considerably different from that seen in causal inference as discussed above. First, hyperparameter tuning, even near-optimal one, does not seem to be as important, even to the point that default hyperparameter values perform mostly on par with the optimised ones (F5; Section 5.4.5). Such realisation is on its own very surprising as defaults are rarely as good in ML; tuning usually has at least some effect. It is also good news for practice as tuning in CSL is very limited but in this case avoidable with well-performing default hyperparameters. A possible explanation behind the phenomenon could be that hyperparameters play a much bigger role in regression/line fitting as there they have a direct impact on bias and variance, whereas CSL is a completely different task, much more reliant on statistical dependence among variables, with regression playing only a small part if any at all (i.e. some methods are completely regression free).

Hyperparameters seem to start playing an important role in causal discovery when we consider misspecified cases and the worst performances. According to our results, these clearly differ from the best and default hyperparameter cases, and the extent to which they are worse is rather extreme. Although the chances of facing the worst cases in practice are low, especially when defaults seem to work so well, the probability involved is arguably never zero. Thus, when we combine a non-zero probability of hitting extremely poor performances due to hyperparameter misspecification with the fact that reliable hyperparameter tuning is almost impossible in causal discovery, at least taking it into account as a potential risk factor seems like a reasonable strategy, even more so in high-stakes situations where any degree of prediction mistakes is unacceptable. Furthermore, current data does not say much about how transferable the defaults are to other much different datasets, meaning that using seemingly safe defaults may carry some risks

of hyperparameter misspecification, which is why working on tuning mechanisms for causal discovery algorithms is so important.

6.2.4 Hyperparameters vs. Ensemble Performance

How does the choice of hyperparameter values influence the selection of causal estimation methods in ensemble settings? (Q4.) We know from the previous discussion (Section 6.2.3) that hyperparameters and tuning influence the performance of individual methods, but how about their role in the task of selecting among different model classes (i.e. ensemble settings)? That is, the cases where we select from not just one but multiple different learning methods. The model selection task is arguably very similar to model tuning as both involve model evaluation. Thus, given we are already aware of the importance of hyperparameters on learners, one can anticipate model selection carrying similar importance weight. This expectation is indeed reasonable but only to a certain extent because of possible discrepancies in performance evaluation between the two. More precisely, not all learners support the same evaluation metrics, making it challenging to combine tuning and model class selection across different types of estimators. And because we know that the choice of metrics can heavily influence the model's performance, this detail is in fact a non-trivial issue. For example, MSE as a metric is generally insufficient to penalise all aspects of misspecification in prediction (see Section 4.3). Thus, the choice of the loss function, which usually entails modelling assumptions, is very important.

This aspect is indeed noticeable in causal inference as our results in Chapter 4 show that when we perform model selection (with tuning) with different evaluation metrics, we can expect significantly different results. This observation is further explained in the literature [26] by suggesting that evaluation metrics can favour certain types of learners (hence the term 'biased validation'). It gets even more interesting when we add to this discussion the observation that after optimal tuning performance differences between learners become negligible, meaning that free lunches in causal effect estimation are possible under certain conditions (F4). This leads us to a critical conclusion – that tuning and metrics are much more important

than learners themselves. That is, if after tuning there is no significant difference between learners, but the metrics significantly change the final performance and can work better with certain methods, then metric selection is by far the most important factor here. This has serious implications for practice by emphasising the role of thorough tuning over the model class selection, in the sense that it perhaps might be better to thoroughly tune a single learner as opposed to performing model selection across shallowly tuned models if available resources cannot guarantee both. In research, on the other hand, the choice of metrics used in benchmarks must be carefully considered and justified as it can make certain methods underperform. An explanation for this could be that many effect estimators reached a level of advancement so high that the performance differences between them have become almost indiscernible. It also means that we perhaps do not need new estimators anymore, but rather better evaluation mechanisms that will improve tuning and remove bias from model class selection (i.e. favour no methods).

In causal discovery, algorithm selection seems to be more critical as there are significant performance differences between methods even assuming access to the hyperparameter oracle. This potentially shows the importance of data assumptions factored into the algorithms (Section 2.1.2), which notably can differ between learners quite substantially. Another possible explanation is a lower level of method advancement as compared to effect estimation, which could be due to CSL being a relatively younger research area than effect estimation, but also because causal graph recovery is arguably more challenging due to its unsupervised nature. As a consequence, there are no free lunches in causal discovery. However, as we know from the previous discussion above (Section 6.2.3), hyperparameters in CSL are important when considering the worst performances. This grows in importance in model selection as we show in Chapter 5 that different algorithms display a varied degree of robustness to misspecified hyperparameters (F5). That is, certain algorithms are safer to use if we consider the risk of poor performances under unlucky hyperparameter values. In addition, we can see that different algorithms constitute the best choices depending on the quality of hyperparameter values (i.e. best,

default, worst). Which means that an algorithm under the best hyperparameters might be the best choice for a given DGP, but a different algorithm might perform better if we consider the worst hyperparameters due to its better robustness to misspecified cases (see also Table 5.1). This clearly shows that the capacity to perform well under correctly specified hyperparameters does not necessarily translate to strong robustness to misspecification. It is also worth mentioning that the size of the hyperparameter search space may influence the worst performances and perceived robustness as larger search spaces may make the worst performances more extreme and more likely. Having said that, other reasons for this issue are also possible and equally likely, such as the volatility of neural networks or the fact that some CSL algorithms are heavily based on regression methods. Thus, overall, when selecting CSL algorithms, it is worth considering not only which method performs the best for a specific DGP, but also how they perform under different types of hyperparameters, even more so that tuning in causal discovery is very limited and the transferability of default hyperparameter values across different datasets is not yet well understood and hence not quite ready to use with confidence in production (discussed in Section 6.2.3 above as well).

6.3 Contributions

This work makes several noteworthy original contributions (denoted with C; see Section 1.7) that are briefly discussed in the following paragraphs.

Averages are not enough for effective decision-making; we need more individualised estimates for that purpose. However, standard off-the-shelf methods were mostly designed for average estimates. To address this gap, Chapter 3 is the first to propose a novel data augmentation method based on recently developed generative trees (C1). Through undersmoothing and bias reduction (see Section 2.4.1), it improves the estimation of individualised effects of downstream learners trained on augmented data. This work is partly related to [107] that generates counterfactuals through

generative neural networks for continuous treatments. Our work is significantly different as it: a) uses simple and commonly used decision trees, b) is more general by oversampling underrepresented subpopulations, and c) works with traditional binary treatments.

Through chapters 3 – 5 we learn that the incompleteness of observational data is the main source of challenges in causal estimation (i.e. missing counterfactuals and selection bias; C2). This might be not as surprising, but the impact of this issue on model evaluation and tuning has not been studied in depth before. In this context, Chapter 4 reveals that covariate shift negatively impacts performance evaluation to the point that the selection of metrics used for tuning is the most important component of the modelling process (more important than model class), which is in line with the latest literature on the topic [26]. In Chapter 5, on the other hand, we can see the lack of ground truth is the main source of issues with tuning in CSL, though inconsistent metric support across learning algorithms is also an issue. This shows that further work on reliable hyperparameter optimisation in this area, which recently gathered some interest [186, 189, 196], is of vital importance for meaningful progress in the field.

All three main chapters (3 – 5) have a strong benchmarking component wherein we test estimation performance across many methods and datasets (C3). The main reason behind this is that different methods are often scattered across the literature, but can be resource-intensive to evaluate properly. This makes it difficult to compare them to each other and harms research progress. This work aims to address this issue but is also unique by specifically focusing on standard and seminal methods commonly used among practitioners. For instance, chapters 3 and 4 involve standard causal estimation methods like S and T learners, Double ML (see Sections 2.2.4 and 2.2.5), but importantly we instantiate them with multiple standard regressors, such as decision trees, random forests and linear models. Such an approach gives us a unique opportunity to study the influence of not only meta-learners (S, T, DML) but base learners (types of regressors) as well, which

is rather uncommon in the literature. Further, in this context, Chapter 5 focuses exclusively on seminal algorithms as they are well-established and most likely used in practice but rarely compared in full in the wider literature.

Finally, performance analysis from the perspective of hyperparameter selection constitutes one of the strongest and novel contributions of this thesis (C4 and C5). While metrics and tuning have been studied before in the context of causal inference [87, 168, 169, 171] and discovery [186, 194–196], oracle, default and worst hyperparameter performances have been overlooked, with the hyperparameter analysis done as part of Chapter 5 being the first of its kind in the area of CSL. One of the critical aspects of this approach is that it is disconnected from potentially biased selection metrics and shows the methods' true performance capabilities. With this perspective incorporated, it also becomes clear if default hyperparameters are worth using, and how learners behave if hyperparameters are poorly selected. Moreover, by including existing metrics in the analysis, we were able to reveal their imperfections and the gap they create between potential performances and the ones that are attainable with current tools. Last but not least, this type of design unravelled the sensitivity of estimator selection to the choice of hyperparameters as well as metrics as the optimisation of the former heavily depends on the accuracy of the latter. We hope this work encourages more studies on hyperparameters in the future and contributes towards now growing evidence about their importance [132, 180–182, 192, 193].

6.4 Limitations

A core part of Chapters 3 and 4 are four standard benchmark datasets (Section 2.2.9). One can wonder if larger and more diverse datasets would change the conclusions about causal estimators being so similar in performance and so dependent on hyperparameter tuning.

While this is a valid concern, the four datasets used in this work cover a lot of possibilities in terms of data challenges, in fact more than one can initially anticipate. This is because said datasets are in fact 'meta-datasets' as each of them consists of numerous slightly different data variants, called in the literature *realisations* or *iterations*. Each of the realisations can be biased differently, with some more challenging than others (think Bootstrapping or Monte Carlo simulations for an analogy). All four datasets total $1,070$ such iterations, with IHDP dataset notably accounting for most of them ($1,000$). As a result, the treatment effect estimation experiments involve not just four but in fact $1,070$ different datasets. Furthermore, the datasets cover both regression and classification problems, include continuous and discrete input features, vary in terms of sample size (747–$11,984$) as well as input size (17–$3,477$), and exhibit different ratios between treated and control sample sizes (see Table 2.1 for the details). Thus, in the context of causal effect estimation from observational data with the standard set of assumptions (Section 2.2.2), these benchmarks are a good representation of the problem to be solved, hence their popularity and recognition in the causal inference community. Adding even more datasets to this setup would unlikely enrich the space of problems to solve in a meaningful way, and thus would unlikely alter the conclusions about similar performances of causal estimators.

However, what could possibly change in light of much larger datasets is the dependency of causal estimators on hyperparameters and tuning. This possibility was explored to some extent in Chapter 5 and the CSL task, where the proportion of good performances increased with the sample size regardless of hyperparameters (performances averaged across all HP values), suggesting the effect of HP tuning (and hence its importance) decreases as sample size increases. While this specific aspect was not studied in the treatment effect estimation chapters, a possibility of observing a similar effect to that found in causal discovery cannot be rejected. That is, observing that causal estimators' sensitivity to HP choices decreases as sample size grows significantly. Properly studying this research question is indeed outside

the scope of the datasets incorporated in this work, but which would certainly constitute an interesting future research direction to explore.

Summary

- Some overlapping topics that appear throughout this thesis are: causal effect estimation, covariate shift, hyperparameters, free lunches in causality, and performance evaluation.
- The questions and findings of this work are centred around two main problems: (i) the impact of causal challenges on estimation performance and hyperparameter tuning, (ii) the role of hyperparameters and tuning in causal estimation.
- This work offers a wide range of unique contributions, spanning from a novel method for improved effect estimation, through extensive performance evaluation benchmarks, to the first of their kind hyperparameter analyses in the context of causal estimation.
- The conclusions about causal estimators having similar prediction performances would unlikely change with wider dataset setups, but the dependence of the estimators on hyperparameter tuning could possibly change with significantly larger datasets.

7 Conclusion

This chapter concludes the thesis by briefly summarising the main findings and contributions. The outline of potential future research directions either created or identified by this work finalise this report.

Contents

7.1 Summary of Main Findings

The overall conclusion of this thesis is that hyperparameters can be a blessing and a curse when it comes to causal estimation from observational data. This is because the analysis done in this work showed that prediction performance of causal estimators heavily depends on selected HP values. If done properly, HP tuning can make even standard causal learners reach or surpass SotA performance levels, as evidenced by performances achieved with the best HP values presented in Chapters 4 and 5. However, if HPs are neglected, even the most sophisticated causal estimators become highly unreliable, as evidenced by performances achieved

with default and the worst HP values presented in the same chapters.

The finding about standard causal estimators reaching SotA performance levels implies that all the methods are similar when it comes to prediction performance and that ultimately no causal estimator is stronger than another. A useful conclusion from this observation is that prioritising thorough HP tuning of fewer causal estimators might be a much more effective strategy in practice as opposed to using an extensive number of estimators that were only shallowly (or not at all) tuned.

Further analysis into HP tuning and model class selection in observational causality revealed unacceptable practices employed by the community. First, data-driven HP tuning of causal estimators is almost completely ignored, especially in the fields outside of computer science. Instead, many use default HP values, copy HP values from other projects that used different data, or worse, adjust HP values manually in order to make the causal model confirm preconceived beliefs about the data. In light of the evidence provided in this work implying the critical importance of detailed HP tuning for accurate causal estimation, all the aforementioned current practices are completely unacceptable as without HP tuning the causal estimates are simply untrustworthy. To improve reliability of conclusions based on causal estimates, an in-depth HP tuning must become an integral part of every causal estimation work, which must also be assessed during peer review. A potential reason behind these poor practices could be improper education about good ML practices outside of computer sciences. Another reason could also be the high complexity of HPs that require a non-trivial time investment from users to properly understand them. Software packages that provide implementations for the causal methods could also discourage the use of fixed-valued HPs and shift towards HP tuning performed by default. Perhaps a hidden and less discussed reason behind neglected HP tuning is precisely the fact that no data-driven tuning opens up a possibility to adjust the causal model to a preconceived idea about the data via manually set HP values, which of course would completely break scientific integrity of any work involving such practices. Unfortunately, the level of sensitivity to HP choices demonstrated

by causal estimators certainly makes it possible, which is why it is so imperative to include data-driven HP tuning in observational causality.

Another important issue in HP tuning within observational causal data is unreliable performance evaluation of causal models. This constitutes a critical problem as candidate HP values and models must be evaluated in order to select the combination most suitable for the data. The issue stems from the fact that the true target of causal estimation is never accessible[1]. Instead, we can only use those evaluation metrics that merely approximate the true estimation target, and since these are imperfect proxies, unreliable performance evaluation leads to unreliable HP tuning and model class selection. This is again an unacceptable state of affairs in the causal community as it ultimately leads to unreliable causal estimation from observational data. One potential reason behind the current state of things is heavy reliance on existing ML methodologies for model evaluation in the context of causal HP tuning, but since these are not tailored to causal estimation targets, they do not work properly in causal estimation settings. This very problem of ineffective standard evaluation metrics is supposed to be addressed by bespoke metrics dedicated for causal estimation. However, as evidenced in this work, these special causal metrics also result in unreliable selection of HP values and models. Some major issues with causal metrics used for HP tuning is that they themselves involve data fitting and HP tuning (i.e. room for mistakes), or are biased towards specific types of causal estimators. There is also a question whether strong causal metrics are at all possible given the limited information within purely observational environments, which could be addressed by adding records from experimental data.

The core and ultimate problem with HP tuning here is that causal estimators and metrics may employ different assumptions about the data. Ideally, we would detect which set of assumptions applies to the data at hand, select estimators and metrics accordingly, and perform detailed HP tuning. Unfortunately, detecting data assumptions with current tools is infeasible. In the meantime, the best

[1]We observe outcomes Y but never causal effects $Y_1 - Y_0$. We never observe causal graphs.

workaround in practice might be to use expert domain knowledge when selecting model classes and evaluation metrics.

The final message of this thesis is that HP tuning is critical for reliable causal estimation from observational data, but reliable HP tuning in this context is extremely challenging, resulting in observational causal estimation highly unreliable. This state of things is certainly undesirable; critical improvements are needed if we ever hope to consistently draw meaningful causal conclusions from observational data. A message to practitioners is to prioritise HP tuning despite its imperfections and challenges, but also exploit domain knowledge as much as possible when selecting model classes and evaluation metrics to counteract existing biases of the current methods. When it comes to research frontier, we urge for a shift in efforts from new causal estimators to reliable and unbiased mechanisms for HP tuning and model class selection. In causal inference, the work in the direction of metrics based on influence functions seems to be particularly promising as they show robustness to model class choices [171]. In causal discovery, a meaningful way forward could be to focus on metrics that measure predicted causal graph's usefulness in the next task that builds upon the causal graph, such as causal inference[2].

7.2 Summary of Contributions

This thesis contributes to the wider literature in several distinct ways, spanning across knowledge advancements, methodological developments, and novel approaches to empirical analysis.

Knowledge. This work identifies major challenges in causal observational data and their influence on hyperparameter optimisation, followed by the impact of hyperparameters on estimation performance.

[2]This would be analogous to clustering that precedes classification or recommendation systems.

Methodology. We are the first to propose a novel data augmentation method based on Generative Trees that improves the estimation of individualised effects through undersmoothing and reduced bias of downstream estimators trained on data augmented by our method.

Analysis. This report offers comprehensive performance evaluation benchmarks of causal estimators across a variety of metrics and datasets, with a strong focus on standard and seminal methods commonly used by practitioners. Crucially, these include first-of-their-kind extensive empirical analyses of the role of hyperparameter selection in causal estimation performance within individual learners and ensemble settings.

7.3 Future Work

No research project is ever fully completed; this report is no different in this regard. We discuss viable further extensions of this line of work in Section 7.3.1. Alternative interesting research directions not directly arising from this work but nevertheless related to the topic are described in Section 7.3.2 that also concludes this report.

7.3.1 Generated by This Work

This work is strongly focused on empirical analysis as most of the issues discussed here, especially concerning hyperparameters, are not yet well understood, which is reflected by scarce literature on those specific problems but also through recently published work that fundamentally questions our understanding of hyperparameters with respect to loss functions [192] and explored HP values [193]. Thus, a natural first step was to explore said issues empirically and confirm their existence in practice. Now, with the issues and their impacts confirmed empirically, the next reasonable step would be to analyse them from a theoretical perspective in an attempt to explain the observed mechanisms with mathematical language. This direction would mostly revolve around hyperparameters and how their selection

affects not just specific estimators but the estimation task itself. Such an approach, in turn, could shed some light on the development of tuning procedures and metrics applicable specifically to causal tasks. This movement has already started to some extent in causal inference (e.g. [87, 171]), but none of the current tools is free from model-induced bias [26]. In causal discovery, this is rather new and still requires fundamental work, though research interest seems to be picking up in this challenging area [186, 196, 213].

One specific potential direction could be to find the connection between the likelihood of a recovered causal graph (e.g. similar to that in [55]) and hyperparameters involved in the framework. Alternatively, one could consider graphs through the lens of parametrised edge weights wherein hyperparameters naturally start to occur in the form of the number of allowed coefficients per graph edge. This could be further extended to more complex nonlinear cases where each node is a (parametric) function of its parents, with the number of parameters again controlled by hyperparameters. Admittedly, such a parametric approach might be limited, but nevertheless easier to set up and take initial steps in solving the problem. A similar parametric approach could be taken within effect estimation as well, which should notably transfer well to neural networks since some of their hyperparameters directly control the number of (latent) parameters (e.g. hidden layers and neurons). However, an important point that must be considered in this line of work is the fundamental problem behind covariate shift, which is the lack of information in certain areas of support, with the latter being transferred to evaluation metrics as they are based on observed data. For this reason, it might be useful to consider Bayesian approaches or further data collection in the affected data regions as alternative ways of progression. Another aspect that would benefit from the theoretical approach is the data augmentation method presented in Chapter 3. Mainly, how, why, and through what mechanisms our data augmentation approach differs from traditional (i.e. reweighting) methods.

As for other extensions related to specific projects, our data augmentation can handle underrepresented data regions but not complete data gaps. This is because

the method needs to learn the distributions of those regions. It could also mean there might be a practical threshold of data region density below which our method loses its utility, or an amount of information necessary to reconstruct the distributions well enough for subsequent sampling. Extending the framework to more noisy environments could also increase its applicability to real-world data scenarios. Using generative neural networks (e.g. GANs [69] or VAEs [67, 68]) instead of Gaussian models could also help to deal with high-dimensional data. Regarding the benchmarking framework for CSL that is part of Chapter 5, it would be natural to extend it to more challenging (and perhaps more practical) graphs that are not DAGs, allowing for cycles and undirected edges [204]. The surprising effectiveness of default hyperparameters should also be investigated further, specifically how well they transfer between different DGP settings to confirm their safety of use with real data. Since it is also unclear whether robustness to misspecified hyperparameters should be attributed to the cardinality of explored hyperparameters or learning algorithms themselves, a study focusing on this aspect would be highly valuable.

Another possible strand of future work, motivated by the discovered importance of hyperparameter tuning and its dependence on evaluation metrics, would be centred around causal software packages. These recently have grown in numbers due to the increased popularity of causal methods, but most of them assume that validation methods known from standard ML will be applicable to causal methods as well. Such an assumption is, however, not safe as causal problems require their own dedicated evaluation procedures, which are unfortunately not easily accessible to practitioners without expert knowledge of such bespoke methods. To make a strong positive impact on practice, causal packages should prioritise good practices around model evaluation, selection and tuning that is appropriate to causal methods. This should include accessibility of causal evaluation metrics (notably made possible in [178]) as well as algorithm selection procedures, with a strong emphasis on the ease of use to encourage their widespread adoption in practice. Furthermore, the way most methods are currently implemented does not encourage hyperparameter tuning at all by leaving hyperparameters at default values unless the user is aware

of their importance and puts some extra effort towards tuning. This needs changing, ideally to an extent that tuning occurs effortlessly by default and avoiding it requires additional actions, increasing the chances that the latter is an informed decision, not an accident. While such a design might feel unnatural given the history of writing programming functions and objects in the same established patterns, the status quo should not take precedence over what is right. Some notable approaches, in fact, have started to appear in this space already, with the end goal being Automated Causal Inference (AutoCI [214, 215]) and Automated Causal Discovery (AutoCD [213]).

7.3.2 Further Literature Gaps

There are other interesting research directions not necessarily related to hyperparameters but still potentially crucial for causal estimation. One common aspect that seems underexplored is longitudinal panel data, which opens up at least two interesting future avenues. The first builds on the observation that panel data resembles to some degree the environment seen in offline off-policy reinforcement learning. While the resemblance is not perfect, repeated unit observations over time that occur in the two are the most important mutual factors. A detailed analysis and comparison of the two frameworks could potentially reveal if, and how, methodologies from one could be incorporated into the other. Owing to existing successes of offline RL [216–218], investigating the use of NN-based RL methods with panel data could be specifically useful due to recent interest in personalised nonlinear causal effect estimation and standard panel data methods being centred around mostly linear models. Notably, (recurrent) neural network solutions have been already explored to some extent in similar settings, particularly with counterfactual predictions over time [219], which could constitute a viable starting point for further work in this direction. There is, however, one possible complication worth keeping in mind while exploring this route. More precisely, viewing longitudinal data as RL (i.e. repeated cross-sectional quasi-experiments) may amplify issues known from cross-sectional settings due to growing d-separation sets (see Section 2.1.1) as time passes and more confounding information is collected.

Increased per-unit information found in panel data could be also beneficial to causal discovery algorithms. Causal structure learning from limited cross-sectional observations is indeed extremely challenging; very few methods can recover causal graphs with high accuracy and only under narrow conditions. A viable strategy for further progress could be to increase useful information within the data. For instance, supplying standard cross-sectional observational data with additional interventions opens new possibilities in terms of achievable accuracy and algorithmic design [211]. CSL has been successfully explored also in purely interventional environments found in RL [220]. Thus, since panel data can be seen as limited interventions over time, it might be potentially beneficial for more accurate causal discovery while still remaining in the context of observational data. Some potential obstacles worth considering that may arise from repeated measurements are the increased number of graph nodes and violated sparsity assumptions.

The idea of combining causal discovery and inference into a unified system was initially explored as part of this project (see Section 1.9). While still highly ambitious with current tools, it remains an interesting direction worth future exploration, though perhaps from different perspectives given past fruitless attempts. One critical lesson from this report is that current CSL algorithms cannot fully and reliably reconstruct relatively large graphs that are of interest. However, certain easier cases seem to be within reach, that is, sparse graphs with up to 10-20 nodes. From the perspective of causal inference, existing methods can already achieve high performances, but transfer, or "transportability", across different domains is still a challenge [221]. Truly causal models are meant to be invariant to domain changes via independent causal mechanisms [1] ingrained into them, but it requires the knowledge of the causal structure (to obtain the disentangled factorisation of the joint). Since full causal graph recovery is not quite feasible yet and model-free[3] effect estimation has clear shortcomings, perhaps exploring something in-between

[3]Most effect estimators assume the simplest of causal structures known as the "triangle" wherein all background covariates affect the treatment and the outcome, with the treatment also pointing towards the outcome. Such a simplistic viewpoint rarely reflects reality accurately.

the two would move things forward – localised small sparse CSL, combined with (structured) function approximation. More specifically, the idea revolves around breaking down function approximation into multiple functions of small and sparse causal structures. Note, that in order to break down the target function into smaller ones, another round of (meta-)CSL would be necessary. And since per-function groupings are unknown and collectively create latent variables, causal discovery specifically designed to work with hidden variables would be necessary for this step [222]. In fact, such a structured approach to function approximation, especially taking latent modelling into account, strongly resembles computational hypergraph discovery [223], in which the causal graph is seen as a computational graph, with latent variables as intermittent computational steps.

Domain invariance and model generalisability are, in fact, of high interest outside causal inference as well. The same problem comes up in other areas like domain adaptation and off-policy RL, but also in biomedical signal processing due to the non-stationarity of the data (within and across subjects). This topic, often referred to across ML as the OOD generalisation problem (see [160] for a survey), has been recently investigated with renewed interest owing to ML models failing in production. Some critical issues causing this are essentially data shifts of various kinds that clearly degrade prediction performance, generally categorised into shifts in input covariates X and those affecting the target-input relationship $Y|X$, as per [158, 159] (see also Section 3.2.1). As a result, the fundamental IID assumption is violated (training and deployment data are never truly the same), breaking all methods that build upon it. The fact that this issue is prevalent in so many areas calls for more fundamental research in this direction, which may require not only new estimators more robust to data shifts, but also a fresh perspective on training and evaluation procedures (i.e. test for OOD generalisation and data gaps), as well as better data collection immune to biases.

Appendices

A

Supplementary Material for Chapter 4

Contents

A.1 Extended Results

A.1.1 Baselines

The baseline numbers we compare our results to are presented in Table A.1. We simply present the numbers as reported in the papers. Note the number of dataset iterations used may vary across papers and does not always match ours (10).

A.1.2 CATE Estimators

In order to investigate how different evaluation metrics affect the final performance of individual CATE estimators, we perform model selection over the search space of

method	IHDP		Jobs		Twins	News	
	ϵ_{ATE}	PEHE	ϵ_{ATT}	$\mathcal{R}_{pol}$	PEHE	ϵ_{ATE}	PEHE
TARNet	$.280 \pm .010$	$0.950 \pm .020$	$.110 \pm .040$	$.210 \pm .010$	$.315 \pm .003$	-	-
CFR-WASS	$.270 \pm .010$	$0.760 \pm .020$	$.090 \pm .030$	$.210 \pm .010$	$.313 \pm .008$	-	-
SITE	-	$0.656 \pm .108$	-	$.219 \pm .009$	-	-	-
GANITE	-	$2.400 \pm .400$	-	$.140 \pm .010$	$.297 \pm .016$	-	-
CEVAE	$.460 \pm .020$	$2.600 \pm .100$	$.030 \pm .010$	$.260 \pm .000$	-	-	-
BLR	$.200 \pm .000$	$5.700 \pm .300$	-	-	-	$.600 \pm .000$	$3.300 \pm .200$
BNN-4-0	$.300 \pm .000$	$5.600 \pm .300$	-	-	-	$.300 \pm .000$	$3.400 \pm .200$
BNN-2-2	$.300 \pm .000$	$1.600 \pm .100$	-	-	-	$.300 \pm .000$	$2.000 \pm .100$

Table A.1: State-of-the-art methods in CATE estimation.

base learners and **hyperparameters** for each CATE estimator separately. With respect to Equation (4.3), this translates to a search over $\mathcal{B}$ and $\mathcal{H}$ per each $\mathcal{C}_c$. Optimal performances $\mathcal{L}^{**}$ concerning the dataset's ground truth metrics are also included (*Oracle*). Some metrics cannot be calculated for certain estimators due to not exposing predicted outcomes, hence the '-' entries.

For presentation purposes, only the best-performing variations of τ-risk metrics are included. For τ-risk$_{plug}$ this is τ-risk$_{plug}^{PEHE}$ with *KR* base learner, for τ-risk$_{match}$ it is τ-risk$_{match}^{PEHE}$ with $k = 1$, and for τ-risk$_R$ the *LGBM* learner performs the best. Both μ-risk and μ-risk$_R$ are included here as well. Tables A.2 and A.3 present obtained results.

As shown in Tables A.2 and A.3, if optimal model selection choices are made (*Oracle* column), the errors with respect to average effect predictions (ϵ_{ATE} and ϵ_{ATT}) are close to perfection across all CATE estimators and datasets. Almost all of these reach, or even surpass, SotA performances (Table A.1). A similar observation can be made with respect to individualised effect predictions (PEHE and $\mathcal{R}_{pol}$), where most CATE estimators are strongly competitive with Table A.1 SotA. Thus, in general, if optimal model selection choices are made, most CATE estimators appear to provide very competitive performances across all metrics and datasets, as compared with the best methods available. This hints at a conclusion that it does not matter that much which particular CATE estimator is chosen if simply a decent performance is desirable, but only under optimal model selection decisions.

name	μ-risk	μ-risk$_R$	τ-risk$_{plug}$	τ-risk$_{match}$	τ-risk$_R$	Oracle
IHDP - ϵ_{ATE}						
SL	0.246 ± 0.098	0.184 ± 0.052	0.264 ± 0.060	0.223 ± 0.094	0.318 ± 0.062	0.001 ± 0.001
TL	0.168 ± 0.098	0.180 ± 0.096	0.151 ± 0.045	0.143 ± 0.070	0.202 ± 0.068	0.000 ± 0.000
IPSW	0.131 ± 0.034	0.261 ± 0.093	0.239 ± 0.058	0.215 ± 0.094	0.315 ± 0.086	0.001 ± 0.000
DR	0.211 ± 0.079	-	0.182 ± 0.041	0.163 ± 0.064	0.533 ± 0.350	0.002 ± 0.001
DML	0.438 ± 0.136	-	0.290 ± 0.047	0.282 ± 0.117	0.508 ± 0.181	0.007 ± 0.003
XL	-	-	0.234 ± 0.085	0.275 ± 0.132	0.270 ± 0.135	0.009 ± 0.007
CF	-	-	0.241 ± 0.129	0.241 ± 0.129	0.240 ± 0.129	0.198 ± 0.131
SL-NN	0.486 ± 0.128	0.486 ± 0.128	0.579 ± 0.270	0.344 ± 0.162	0.342 ± 0.067	0.104 ± 0.038
TL-NN	0.221 ± 0.076	0.198 ± 0.048	0.422 ± 0.138	0.379 ± 0.165	0.507 ± 0.208	0.000 ± 0.000
IHDP - PEHE						
SL	1.373 ± 0.612	1.390 ± 0.644	1.549 ± 0.673	1.296 ± 0.618	1.548 ± 0.697	1.205 ± 0.561
TL	0.701 ± 0.202	0.724 ± 0.199	1.214 ± 0.536	0.747 ± 0.239	1.129 ± 0.379	0.621 ± 0.200
IPSW	1.552 ± 0.726	2.117 ± 0.932	1.517 ± 0.647	1.309 ± 0.616	1.396 ± 0.609	1.204 ± 0.560
DR	1.470 ± 0.624	-	1.688 ± 0.794	1.508 ± 0.755	1.658 ± 0.793	1.275 ± 0.581
DML	1.905 ± 0.826	-	1.890 ± 0.898	1.827 ± 0.893	2.005 ± 0.880	1.679 ± 0.830
XL	-	-	1.317 ± 0.498	1.154 ± 0.441	1.276 ± 0.448	1.067 ± 0.409
CF	-	-	2.300 ± 1.185	2.295 ± 1.185	2.295 ± 1.185	2.290 ± 1.186
SL-NN	1.064 ± 0.212	1.064 ± 0.212	1.490 ± 0.614	1.184 ± 0.340	1.232 ± 0.277	0.925 ± 0.224
TL-NN	0.894 ± 0.183	0.868 ± 0.175	1.535 ± 0.685	1.179 ± 0.459	1.399 ± 0.496	0.641 ± 0.190
Jobs - ϵ_{ATT}						
SL	0.066 ± 0.020	0.076 ± 0.020	0.086 ± 0.030	0.071 ± 0.022	0.076 ± 0.023	0.003 ± 0.001
TL	0.074 ± 0.023	0.077 ± 0.022	0.083 ± 0.023	0.080 ± 0.023	0.081 ± 0.025	0.000 ± 0.000
IPSW	0.077 ± 0.025	0.080 ± 0.024	0.069 ± 0.022	0.079 ± 0.017	0.079 ± 0.020	0.024 ± 0.019
DR	0.075 ± 0.023	-	0.123 ± 0.037	0.080 ± 0.023	0.064 ± 0.019	0.001 ± 0.001
DML	0.078 ± 0.023	-	0.083 ± 0.025	0.079 ± 0.025	0.107 ± 0.031	0.016 ± 0.005
XL	-	-	0.092 ± 0.036	0.078 ± 0.026	0.077 ± 0.025	0.003 ± 0.002
CF	-	-	0.074 ± 0.021	0.077 ± 0.023	0.072 ± 0.021	0.053 ± 0.022
SL-NN	0.068 ± 0.022	0.071 ± 0.022	0.074 ± 0.025	0.066 ± 0.023	0.075 ± 0.025	0.023 ± 0.019
TL-NN	0.066 ± 0.026	0.075 ± 0.025	0.088 ± 0.022	0.092 ± 0.028	0.088 ± 0.025	0.010 ± 0.010
Jobs - $\mathcal{R}_{pol}$						
SL	0.261 ± 0.019	0.262 ± 0.021	0.236 ± 0.018	0.250 ± 0.019	0.211 ± 0.012	0.158 ± 0.011
TL	0.235 ± 0.019	0.237 ± 0.018	0.241 ± 0.015	0.245 ± 0.013	0.241 ± 0.016	0.128 ± 0.012
IPSW	0.245 ± 0.013	0.245 ± 0.011	0.245 ± 0.024	0.249 ± 0.021	0.248 ± 0.019	0.158 ± 0.013
DR	0.240 ± 0.010	-	0.282 ± 0.021	0.249 ± 0.012	0.243 ± 0.015	0.149 ± 0.010
DML	0.264 ± 0.014	-	0.263 ± 0.024	0.249 ± 0.015	0.271 ± 0.021	0.193 ± 0.012
XL	-	-	0.256 ± 0.025	0.236 ± 0.013	0.233 ± 0.019	0.153 ± 0.013
CF	-	-	0.254 ± 0.018	0.246 ± 0.016	0.225 ± 0.016	0.204 ± 0.017
SL-NN	0.254 ± 0.018	0.257 ± 0.017	0.248 ± 0.013	0.250 ± 0.018	0.251 ± 0.014	0.162 ± 0.010
TL-NN	0.226 ± 0.014	0.248 ± 0.022	0.242 ± 0.015	0.239 ± 0.015	0.216 ± 0.015	0.132 ± 0.009

Table A.2: Performance of individual CATE estimators achieved under different model evaluation metrics, grouped by datasets (IHDP and Jobs) and test metrics. Numbers are *mean* $\pm$ *standard error* across dataset iterations. Lower is better.

name	μ-risk	μ-risk$_R$	τ-risk$_{plug}$	τ-risk$_{match}$	τ-risk$_R$	Oracle
Twins - ϵ_{ATE}						
SL	0.039 ± 0.000	0.039 ± 0.000	0.047 ± 0.001	0.047 ± 0.001	0.027 ± 0.002	0.000 ± 0.000
TL	0.051 ± 0.000	0.051 ± 0.000	0.077 ± 0.000	0.077 ± 0.000	0.077 ± 0.000	0.000 ± 0.000
IPSW	0.040 ± 0.000	0.040 ± 0.000	0.045 ± 0.001	0.045 ± 0.001	0.023 ± 0.002	0.000 ± 0.000
DR	0.053 ± 0.001	-	0.077 ± 0.000	0.077 ± 0.000	0.063 ± 0.001	0.001 ± 0.000
DML	0.033 ± 0.001	-	0.043 ± 0.001	0.060 ± 0.003	0.031 ± 0.000	0.001 ± 0.000
XL	-	-	0.077 ± 0.000	0.077 ± 0.000	0.052 ± 0.001	0.013 ± 0.001
CF	-	-	0.068 ± 0.000	0.064 ± 0.001	0.064 ± 0.000	0.063 ± 0.000
SL-NN	0.024 ± 0.001	0.024 ± 0.001	0.038 ± 0.001	0.037 ± 0.001	0.024 ± 0.003	0.001 ± 0.000
TL-NN	0.050 ± 0.002	0.049 ± 0.001	0.064 ± 0.003	0.068 ± 0.002	0.066 ± 0.002	0.000 ± 0.000
Twins - PEHE						
SL	0.319 ± 0.002	0.319 ± 0.002	0.320 ± 0.002	0.320 ± 0.002	0.319 ± 0.002	0.317 ± 0.002
TL	0.328 ± 0.003	0.328 ± 0.003	0.326 ± 0.002	0.326 ± 0.002	0.326 ± 0.002	0.318 ± 0.002
IPSW	0.320 ± 0.002	0.320 ± 0.002	0.320 ± 0.002	0.320 ± 0.002	0.318 ± 0.002	0.317 ± 0.002
DR	0.330 ± 0.003	-	0.326 ± 0.002	0.326 ± 0.002	0.323 ± 0.002	0.318 ± 0.002
DML	0.318 ± 0.002	-	0.320 ± 0.002	0.323 ± 0.001	0.318 ± 0.002	0.317 ± 0.002
XL	-	-	0.326 ± 0.002	0.326 ± 0.002	0.323 ± 0.003	0.318 ± 0.002
CF	-	-	0.324 ± 0.002	0.323 ± 0.002	0.323 ± 0.002	0.323 ± 0.002
SL-NN	0.319 ± 0.002	0.319 ± 0.002	0.320 ± 0.002	0.320 ± 0.002	0.319 ± 0.003	0.317 ± 0.002
TL-NN	0.340 ± 0.004	0.340 ± 0.004	0.332 ± 0.004	0.334 ± 0.004	0.332 ± 0.005	0.328 ± 0.004
News - ϵ_{ATE}						
SL	0.156 ± 0.034	0.156 ± 0.034	0.645 ± 0.102	0.194 ± 0.043	0.144 ± 0.036	0.027 ± 0.011
TL	0.193 ± 0.046	0.193 ± 0.046	0.710 ± 0.173	0.265 ± 0.057	0.190 ± 0.051	0.007 ± 0.004
IPSW	0.149 ± 0.033	0.191 ± 0.045	0.764 ± 0.167	0.174 ± 0.042	0.140 ± 0.029	0.044 ± 0.019
DR	0.140 ± 0.033	-	0.368 ± 0.099	0.233 ± 0.063	0.166 ± 0.037	0.006 ± 0.002
DML	0.392 ± 0.066	-	0.769 ± 0.060	0.806 ± 0.270	0.587 ± 0.086	0.003 ± 0.001
XL	-	-	0.369 ± 0.083	0.169 ± 0.039	0.122 ± 0.029	0.008 ± 0.006
CF	-	-	0.598 ± 0.101	0.540 ± 0.095	0.539 ± 0.095	0.536 ± 0.095
SL-NN	2.395 ± 0.232	2.395 ± 0.232	1.559 ± 0.124	1.419 ± 0.181	1.497 ± 0.164	1.370 ± 0.189
TL-NN	0.714 ± 0.229	0.714 ± 0.229	1.594 ± 0.622	0.540 ± 0.158	1.043 ± 0.597	0.000 ± 0.000
News - PEHE						
SL	1.729 ± 0.127	1.729 ± 0.127	3.115 ± 0.217	1.769 ± 0.125	1.727 ± 0.124	1.688 ± 0.122
TL	1.593 ± 0.129	1.593 ± 0.129	2.704 ± 0.207	1.727 ± 0.126	1.623 ± 0.126	1.549 ± 0.128
IPSW	1.708 ± 0.121	1.733 ± 0.123	3.173 ± 0.218	1.778 ± 0.128	1.723 ± 0.123	1.677 ± 0.122
DR	1.689 ± 0.123	-	2.426 ± 0.251	1.800 ± 0.128	1.730 ± 0.132	1.618 ± 0.128
DML	2.489 ± 0.368	-	2.765 ± 0.318	2.653 ± 0.241	2.399 ± 0.246	2.063 ± 0.157
XL	-	-	2.598 ± 0.214	1.804 ± 0.144	1.784 ± 0.140	1.704 ± 0.134
CF	-	-	2.544 ± 0.191	2.318 ± 0.176	2.321 ± 0.177	2.317 ± 0.177
SL-NN	3.959 ± 0.334	3.959 ± 0.334	3.473 ± 0.298	3.428 ± 0.310	3.426 ± 0.310	3.395 ± 0.308
TL-NN	3.908 ± 0.408	4.057 ± 0.484	4.385 ± 0.748	3.476 ± 0.401	3.760 ± 0.647	2.306 ± 0.122

Table A.3: Performance of individual CATE estimators achieved under different model evaluation metrics, grouped by datasets (Twins and News) and test metrics. Numbers are *mean* $\pm$ *standard error* across dataset iterations. Lower is better.

This is, however, no longer true when using actual, non-optimal, selection metrics (non-*Oracle* columns). Some selection methods appear to be better than others in certain situations, but not consistently. The lack of a clear pattern suggests there is no single best model selection method. A possible explanation is that selection methods make different assumptions about the data or prediction targets, and hence penalise different aspects of prediction mistakes. It is also perhaps a good example demonstrating how much the final CATE performance can vary by changing only model selection metrics, to the extent that a selection metric can decide if the final product is a success or a failure, even when using exactly the same CATE estimator. For an example, see *TL* estimator under *IHDP* - *PEHE* – performances with τ-risk$_{plug}$ and τ-risk$_R$ are relatively mediocre, whereas with all other metrics, performances reach SotA. In other words, model selection metric alone can "make or break" a CATE estimator, which suggests the explored validation approaches are very limited. But even more importantly, there is clearly a significant gap between *Oracle* performances and the ones available via common metrics, showing the latter can be a source of bias too.

A.1.3 Base Learners

We also study the impact different evaluation metrics have from the perspective of individual base learners (Tables A.4 and A.5). Thus, the model selection search space spans across **CATE estimators** $\mathcal{C}$ and **hyperparameters** $\mathcal{H}$ for each individual base learner $\mathcal{B}_b$. Presented τ-risk variants are the same as in Section A.1.2. As μ-risk metrics are unavailable for certain CATE estimators, and the search here is done across $\mathcal{C}$, those metrics are excluded from this particular analysis.

Similar trends can be observed from the perspective of base learners (Tables A.4 and A.5). More precisely, optimal model selection decisions (*Oracle*), lead to performances close to, or surpassing, SotA in Table A.1. Actual model selection metrics again prove to deliver variable results, with a clear gap between them and optimal choices. This suggests the choice of base learners is not that important either for the final CATE performance. Note that by only examining Tables A.2

name	τ-risk$_{plug}$	τ-risk$_{match}$	τ-risk$_{R}$	Oracle
IHDP - ϵ_{ATE}				
L1	0.277 ± 0.100	0.226 ± 0.100	0.340 ± 0.129	0.046 ± 0.016
L2	0.330 ± 0.179	0.215 ± 0.116	0.270 ± 0.142	0.161 ± 0.112
DT	0.620 ± 0.201	0.451 ± 0.293	0.399 ± 0.202	0.000 ± 0.000
RF	0.204 ± 0.054	0.109 ± 0.043	0.253 ± 0.095	0.020 ± 0.011
ET	0.220 ± 0.107	0.281 ± 0.191	0.292 ± 0.129	0.003 ± 0.002
KR	0.190 ± 0.042	0.137 ± 0.072	0.282 ± 0.104	0.001 ± 0.001
CB	0.242 ± 0.061	0.109 ± 0.019	0.267 ± 0.064	0.004 ± 0.002
LGBM	0.351 ± 0.066	0.183 ± 0.069	0.264 ± 0.080	0.016 ± 0.008
NN	0.415 ± 0.139	0.379 ± 0.165	0.507 ± 0.208	0.000 ± 0.000
IHDP - PEHE				
L1	1.820 ± 0.887	1.698 ± 0.904	1.795 ± 0.881	1.643 ± 0.884
L2	1.738 ± 0.859	1.617 ± 0.840	1.810 ± 0.899	1.603 ± 0.841
DT	2.294 ± 0.978	2.189 ± 1.037	2.330 ± 1.028	1.890 ± 0.932
RF	1.816 ± 0.818	1.718 ± 0.822	1.909 ± 0.968	1.529 ± 0.811
ET	1.955 ± 1.026	1.788 ± 0.999	1.999 ± 1.107	1.582 ± 0.915
KR	1.306 ± 0.523	0.746 ± 0.239	1.399 ± 0.623	0.653 ± 0.195
CB	1.428 ± 0.684	0.972 ± 0.410	1.453 ± 0.700	0.893 ± 0.386
LGBM	1.976 ± 0.939	1.418 ± 0.599	1.881 ± 0.907	1.326 ± 0.562
NN	1.531 ± 0.685	1.179 ± 0.459	1.399 ± 0.496	0.641 ± 0.190
Jobs - ϵ_{ATT}				
L1	0.078 ± 0.024	0.075 ± 0.025	0.093 ± 0.024	0.032 ± 0.022
L2	0.077 ± 0.026	0.079 ± 0.025	0.082 ± 0.025	0.061 ± 0.023
DT	0.076 ± 0.022	0.069 ± 0.021	0.062 ± 0.019	0.002 ± 0.001
RF	0.073 ± 0.024	0.074 ± 0.022	0.074 ± 0.021	0.011 ± 0.006
ET	0.082 ± 0.025	0.077 ± 0.026	0.073 ± 0.023	0.014 ± 0.008
KR	0.080 ± 0.023	0.080 ± 0.022	0.085 ± 0.026	0.000 ± 0.000
CB	0.077 ± 0.023	0.073 ± 0.020	0.066 ± 0.021	0.025 ± 0.011
LGBM	0.076 ± 0.022	0.073 ± 0.023	0.076 ± 0.024	0.009 ± 0.003
NN	0.088 ± 0.022	0.085 ± 0.028	0.088 ± 0.025	0.010 ± 0.010
Jobs - $\mathcal{R}_{pol}$				
L1	0.246 ± 0.014	0.243 ± 0.014	0.270 ± 0.024	0.197 ± 0.013
L2	0.245 ± 0.019	0.255 ± 0.014	0.224 ± 0.019	0.199 ± 0.014
DT	0.270 ± 0.025	0.230 ± 0.014	0.233 ± 0.019	0.142 ± 0.010
RF	0.253 ± 0.017	0.241 ± 0.015	0.239 ± 0.017	0.155 ± 0.015
ET	0.244 ± 0.015	0.232 ± 0.012	0.223 ± 0.016	0.148 ± 0.013
KR	0.235 ± 0.016	0.254 ± 0.016	0.262 ± 0.024	0.141 ± 0.009
CB	0.222 ± 0.019	0.211 ± 0.016	0.212 ± 0.015	0.156 ± 0.011
LGBM	0.220 ± 0.016	0.204 ± 0.014	0.221 ± 0.015	0.174 ± 0.011
NN	0.242 ± 0.015	0.247 ± 0.014	0.216 ± 0.015	0.131 ± 0.009

Table A.4: Performance of individual base learners achieved under different model evaluation metrics, grouped by datasets (IHDP and Jobs) and test metrics. Numbers are *mean* $\pm$ *standard error* across dataset iterations. Lower is better.

name	τ-risk$_{plug}$	τ-risk$_{match}$	τ-risk$_{R}$	Oracle
Twins - ϵ_{ATE}				
L1	0.075 ± 0.002	0.077 ± 0.000	0.025 ± 0.003	0.022 ± 0.000
L2	0.047 ± 0.001	0.047 ± 0.001	0.034 ± 0.002	0.030 ± 0.001
DT	0.042 ± 0.001	0.050 ± 0.002	0.030 ± 0.004	0.000 ± 0.000
RF	0.052 ± 0.001	0.063 ± 0.000	0.037 ± 0.003	0.000 ± 0.000
ET	0.043 ± 0.001	0.045 ± 0.000	0.040 ± 0.002	0.000 ± 0.000
KR	0.044 ± 0.000	0.045 ± 0.000	0.027 ± 0.002	0.000 ± 0.000
CB	0.043 ± 0.003	0.046 ± 0.002	0.043 ± 0.001	0.023 ± 0.001
LGBM	0.025 ± 0.001	0.033 ± 0.001	0.026 ± 0.001	0.011 ± 0.000
NN	0.038 ± 0.001	0.037 ± 0.001	0.024 ± 0.003	0.000 ± 0.000
Twins - PEHE				
L1	0.325 ± 0.001	0.326 ± 0.002	0.318 ± 0.002	0.317 ± 0.002
L2	0.320 ± 0.002	0.320 ± 0.002	0.319 ± 0.002	0.318 ± 0.002
DT	0.320 ± 0.002	0.321 ± 0.002	0.320 ± 0.004	0.317 ± 0.002
RF	0.321 ± 0.002	0.324 ± 0.002	0.320 ± 0.003	0.317 ± 0.002
ET	0.320 ± 0.002	0.321 ± 0.003	0.321 ± 0.004	0.318 ± 0.002
KR	0.320 ± 0.002	0.320 ± 0.002	0.319 ± 0.002	0.317 ± 0.002
CB	0.320 ± 0.002	0.321 ± 0.002	0.321 ± 0.002	0.318 ± 0.002
LGBM	0.318 ± 0.002	0.320 ± 0.003	0.319 ± 0.003	0.317 ± 0.002
NN	0.320 ± 0.002	0.320 ± 0.002	0.319 ± 0.003	0.317 ± 0.002
News - ϵ_{ATE}				
L1	0.705 ± 0.136	0.414 ± 0.100	0.403 ± 0.128	0.044 ± 0.015
L2	0.254 ± 0.067	0.184 ± 0.069	0.263 ± 0.074	0.065 ± 0.040
DT	0.694 ± 0.070	0.391 ± 0.095	0.475 ± 0.098	0.002 ± 0.001
RF	0.347 ± 0.097	0.276 ± 0.073	0.260 ± 0.068	0.026 ± 0.020
ET	0.371 ± 0.090	0.341 ± 0.086	0.294 ± 0.075	0.035 ± 0.027
KR	1.700 ± 0.293	0.877 ± 0.345	0.529 ± 0.213	0.022 ± 0.006
CB	0.257 ± 0.063	0.219 ± 0.059	0.136 ± 0.028	0.035 ± 0.015
LGBM	0.722 ± 0.059	0.180 ± 0.035	0.253 ± 0.060	0.029 ± 0.015
NN	1.477 ± 0.155	0.653 ± 0.169	1.497 ± 0.164	0.000 ± 0.000
News - PEHE				
L1	3.070 ± 0.247	2.411 ± 0.208	2.560 ± 0.227	2.140 ± 0.178
L2	3.287 ± 0.303	3.283 ± 0.303	3.289 ± 0.304	3.208 ± 0.270
DT	2.895 ± 0.288	2.735 ± 0.224	2.498 ± 0.155	2.078 ± 0.156
RF	2.555 ± 0.197	1.932 ± 0.168	1.936 ± 0.166	1.911 ± 0.166
ET	2.499 ± 0.188	1.950 ± 0.172	2.000 ± 0.188	1.916 ± 0.176
KR	3.446 ± 0.297	2.986 ± 0.314	3.057 ± 0.575	2.475 ± 0.276
CB	2.132 ± 0.128	1.695 ± 0.133	1.755 ± 0.135	1.544 ± 0.128
LGBM	2.703 ± 0.424	1.801 ± 0.124	1.883 ± 0.100	1.693 ± 0.125
NN	3.208 ± 0.249	3.438 ± 0.405	3.426 ± 0.310	2.306 ± 0.122

Table A.5: Performance of individual base learners achieved under different model evaluation metrics, grouped by datasets (Twins and News) and test metrics. Numbers are *mean* $\pm$ *standard error* across dataset iterations. Lower is better.

and A.3, it could be argued that those great performances under *Oracle* column are due to certain base learners winning most of the time. But by observing Tables A.4 and A.5, this is clearly not the case as most learners achieve very competitive results. Therefore, it seems that it is not CATE estimators, nor base learners, that are important; it is model selection that drives the final CATE performance.

A.1.4 Model Evaluation Metrics

Winner Selection

In Table A.6, we focus exclusively on model evaluation metrics and how their different variants perform at the model selection task. The search space, thus, includes all **CATE estimators** $\mathcal{C}$, **base learners** $\mathcal{B}$ and **hyperparameters** $\mathcal{H}$, processed by each evaluation metric and its variation.

As in Tables A.2 – A.5, it is evident in Table A.6 that the gap between CATE performances obtained via *Oracle* and other metrics is significant. The explored model selection metrics are clearly sub-optimal and biased, across all performance metrics and datasets. No single metric used for model selection attains the desirable standard set by *Oracle*.

Among the practical model selection metrics and their variations, *matching* using PEHE as the feedback signal (τ-risk$^{PEHE}_{match}$) performs the best on the IHDP dataset, followed by μ-risk, with μ-risk$_R$ being somewhat decent but clearly worse. On Jobs data, there is a fairly low variability between the metrics (no clear winner), though τ-risk$_R$ appears to be the most consistent. On Twins, τ-risk$^{PEHE}_{plug}$ with *S-Learner* variation seems to be leading, followed by μ-risk and μ-risk$_R$. Against News, μ-risk and τ-risk$_R$ with tree-based learners (DT and $LGBM$) performed the best.

method	IHDP		Jobs		Twins		News	
	ϵ_{ATE}	PEHE	ϵ_{ATT}	$\mathcal{R}_{pol}$	ϵ_{ATE}	PEHE	ϵ_{ATE}	PEHE
μ-risk*	0.188 ± 0.097	0.786 ± 0.189	0.077 ± 0.025	0.245 ± 0.013	0.039 ± 0.000	0.319 ± 0.002	0.149 ± 0.033	1.708 ± 0.121
μ-risk$_R$*	0.352 ± 0.146	0.922 ± 0.237	0.072 ± 0.022	0.257 ± 0.019	0.039 ± 0.000	0.319 ± 0.002	0.156 ± 0.034	1.729 ± 0.127
$\mathcal{R}_{pol}$	-	-	0.300 ± 0.090	0.220 ± 0.016	-	-	-	-
τ-risk$_{plug}^{ATE}$								
SL-DT	1.455 ± 0.848	3.327 ± 1.454	0.080 ± 0.026	0.275 ± 0.024	0.016 ± 0.005	0.375 ± 0.013	0.466 ± 0.102	3.575 ± 0.456
SL-LGBM	4.791 ± 0.904	6.998 ± 2.261	0.075 ± 0.025	0.267 ± 0.015	0.040 ± 0.003	0.412 ± 0.023	0.372 ± 0.098	3.101 ± 0.406
SL-KR	1.430 ± 0.608	2.868 ± 0.978	0.068 ± 0.022	0.233 ± 0.023	0.034 ± 0.003	0.446 ± 0.024	2.838 ± 0.234	4.397 ± 0.321
TL-DT	0.201 ± 0.057	1.873 ± 0.393	0.086 ± 0.026	0.245 ± 0.017	0.059 ± 0.004	0.404 ± 0.019	0.427 ± 0.096	3.187 ± 0.326
TL-LGBM	0.237 ± 0.082	1.983 ± 0.434	0.081 ± 0.027	0.250 ± 0.016	0.062 ± 0.005	0.395 ± 0.021	0.324 ± 0.117	3.612 ± 0.518
TL-KR	0.409 ± 0.115	1.796 ± 0.690	0.066 ± 0.013	0.222 ± 0.010	0.069 ± 0.004	0.379 ± 0.014	0.765 ± 0.150	5.777 ± 1.186
τ-risk$_{plug}^{PEHE}$								
SL-DT	1.655 ± 1.269	4.188 ± 2.652	0.083 ± 0.026	0.297 ± 0.020	0.027 ± 0.003	0.318 ± 0.002	0.529 ± 0.115	2.619 ± 0.199
SL-LGBM	4.791 ± 0.904	6.998 ± 2.261	0.083 ± 0.026	0.296 ± 0.020	0.024 ± 0.002	0.317 ± 0.002	0.459 ± 0.088	2.701 ± 0.240
SL-KR	1.652 ± 0.788	4.438 ± 2.248	0.076 ± 0.024	0.248 ± 0.018	0.022 ± 0.000	0.317 ± 0.002	2.906 ± 0.251	4.426 ± 0.330
TL-DT	0.267 ± 0.125	2.605 ± 1.433	0.065 ± 0.022	0.248 ± 0.016	0.049 ± 0.003	0.320 ± 0.002	0.463 ± 0.098	2.381 ± 0.184
TL-LGBM	0.306 ± 0.173	2.357 ± 1.014	0.073 ± 0.022	0.247 ± 0.015	0.047 ± 0.000	0.320 ± 0.002	0.302 ± 0.048	2.018 ± 0.143
TL-KR	0.209 ± 0.048	1.341 ± 0.518	0.085 ± 0.023	0.244 ± 0.015	0.047 ± 0.001	0.320 ± 0.002	0.812 ± 0.126	2.918 ± 0.220
τ-risk$_{match}^{ATE}$								
$k = 1$	0.323 ± 0.098	3.199 ± 1.585	0.161 ± 0.076	0.229 ± 0.018	0.078 ± 0.008	0.395 ± 0.014	0.804 ± 0.231	6.721 ± 1.678
$k = 3$	0.176 ± 0.039	1.640 ± 0.352	0.158 ± 0.077	0.227 ± 0.014	0.078 ± 0.005	0.409 ± 0.010	1.005 ± 0.206	7.901 ± 1.545
$k = 5$	0.209 ± 0.065	1.841 ± 0.478	0.160 ± 0.077	0.236 ± 0.021	0.077 ± 0.006	0.411 ± 0.008	1.293 ± 0.459	9.629 ± 2.933
τ-risk$_{match}^{PEHE}$								
$k = 1$	0.164 ± 0.067	0.718 ± 0.239	0.080 ± 0.028	0.235 ± 0.014	0.077 ± 0.000	0.326 ± 0.002	0.239 ± 0.054	1.738 ± 0.125
$k = 3$	0.167 ± 0.068	0.748 ± 0.245	0.067 ± 0.024	0.235 ± 0.010	0.077 ± 0.000	0.326 ± 0.002	0.245 ± 0.074	1.774 ± 0.137
$k = 5$	0.182 ± 0.069	0.755 ± 0.239	0.073 ± 0.028	0.241 ± 0.009	0.077 ± 0.000	0.326 ± 0.002	0.258 ± 0.070	1.795 ± 0.134
τ-risk$_R$								
DT	0.853 ± 0.128	2.149 ± 0.657	0.072 ± 0.019	0.220 ± 0.012	0.034 ± 0.002	0.321 ± 0.003	0.173 ± 0.048	1.616 ± 0.130
LGBM	0.535 ± 0.207	1.389 ± 0.498	0.075 ± 0.019	0.224 ± 0.017	0.026 ± 0.004	0.320 ± 0.003	0.136 ± 0.028	1.755 ± 0.135
KR	0.378 ± 0.106	1.495 ± 0.633	0.080 ± 0.023	0.234 ± 0.014	0.039 ± 0.003	0.322 ± 0.004	0.222 ± 0.049	1.604 ± 0.128
Oracle	0.000 ± 0.000	0.585 ± 0.198	0.000 ± 0.000	0.121 ± 0.011	0.000 ± 0.000	0.317 ± 0.002	0.000 ± 0.000	1.540 ± 0.129

Table A.6: Effectiveness of model evaluation methods. Numbers are *mean* $\pm$ *standard error* across dataset iterations. Lower is better. *Includes only SL, TL and IPSW.

All this shows a lot of variability between selection methods, confirming again that none of the explored selection methods is universally the best on all problems. This is especially true for selection approaches that internally incorporate some form of learning (all τ-risk metrics). No learner is generally the best, and this problem applies to selection methods that involve learning. A solution could be to perform some form of model selection again, that is, select the variation of the model selection method that performs the best. However, this circular logic brings us back to the start of the main problem – model selection for CATE estimation. A "solution" that performs model selection for model selection (double model selection) does not really address the problem, just shifts it to another algorithmic area. Making strong assumptions in the form of priors to manually build an accurate learning-based model selection does not solve the problem either because if such priors are accessible, they can be simply incorporated straight into the CATE estimator, eliminating the very need for any model selection. This is one of the main arguments against selection metrics incorporating learners – the main problem comes back unsolved. Such an issue does not exist when dealing with non-learning metrics, namely μ-risk, which clearly provide more stable CATE performances across all datasets. In addition, plain μ-risk appears to be generally a safer choice over μ-risk$_R$ as the former is rarely worse, but often quite better.

When it comes to plugin validation (τ-risk$_{plug}$), *T-Learning* variations are generally better than *S-Learning* ones on IHDP and Jobs. This indeed makes sense as stronger CATE estimators are expected to be a better guiding force in model selection. Interestingly though, some *S-Learning* variants are better on the Twins dataset. However, such a division between *SL* and *TL* is not that clear with the News dataset. When comparing the plugin variants that provide either ATE or PEHE as the guiding signal, it is not entirely clear which one would be a better choice, though PEHE seems to be a slightly safer choice in most cases, with ATE being as good at times. This is in line with the intuition that more accurate CATEs should lead to better ATEs, but not the other way around (accurate ATEs are not sufficient for good CATEs). If only a single specific plugin variant had to be chosen,

TL-KR providing PEHE appears to be the most consistently good, with an exception for News where tree-based variants appear to perform better, especially *LGBM*.

With matching validation (τ-risk$_{match}$), the ones paired with $PEHE$ are better most of the time, except Jobs and $\mathcal{R}_{pol}$ metric, where the ATE variant dominates. As for the number of neighbours considered in the kNN algorithm, $k = 5$ is often the worst choice, suggesting that spreading the counterfactual estimation into too many data points is detrimental to prediction accuracy. The choice between $k = 1$ and $k = 3$ is not that clear anymore.

Among the three learners used in conjunction with τ-risk$_R$, there is no clear winner, though the one with Decision Trees appears to be the least stable. If stability is indeed the priority, *LGBM* variant is perhaps the safest approach, which admittedly is the recommended option (XGBoost) by [87].

Correlations

Similarly to the previous section, we focus here on various evaluation metrics across the search space of all **CATE estimators** $\mathcal{C}$, **base learners** $\mathcal{B}$ and **hyperparameters** $\mathcal{H}$, but instead of selecting the best model $\mathcal{M}^*$ according to a given metric, we use the metrics to rank all candidate models $\mathcal{M} = (\mathcal{C}, \mathcal{B}, \mathcal{H})$. Because we record the test performances of all candidate models, we can calculate correlations between the aforementioned rankings and ground truth metrics (test performance). This perspective sheds some light on how well evaluation metrics rank candidate models from best to worst. Table A.7 presents obtained results. Note all dataset metrics, as well as most model evaluation ones, indicate better performances if the scores are lower. Thus, positive correlations are expected there. μ-risk$_R$ and τ-risk$_R$ are exceptions as their higher scores indicate better performance, meaning their correlations with dataset metrics are expected to be negative.

	IHDP		Jobs		Twins		News	
method	ϵ_{ATE}	PEHE	ϵ_{ATT}	$\mathcal{R}_{pol}$	ϵ_{ATE}	PEHE	ϵ_{ATE}	PEHE
μ-risk*	$.774 \pm .071$	$.907 \pm .019$	$.051 \pm .007$	$.080 \pm .020$	$.080 \pm .008$	$.968 \pm .011$	$.499 \pm .132$	$.503 \pm .132$
μ-risk$_R$*	$-.908 \pm .019$	$-.737 \pm .096$	$-.048 \pm .006$	$-.079 \pm .020$	$-.076 \pm .007$	$-.969 \pm .010$	$-.405 \pm .124$	$-.410 \pm .124$
$\mathcal{R}_{pol}$	-	-	$.023 \pm .005$	$.289 \pm .074$	-	-	-	-
τ-risk$_{plug}^{ATE}$								
SL-DT	$.749 \pm .148$	$.582 \pm .062$	$.530 \pm .073$	$.014 \pm .008$	$.286 \pm .068$	$.520 \pm .039$	$.314 \pm .132$	$.313 \pm .132$
SL-LGBM	$-.866 \pm .056$	$-.458 \pm .094$	$.530 \pm .073$	$.014 \pm .008$	$.309 \pm .062$	$.538 \pm .030$	$.314 \pm .132$	$.314 \pm .133$
SL-KR	$.718 \pm .107$	$.505 \pm .042$	$.530 \pm .073$	$.014 \pm .008$	$.316 \pm .070$	$.535 \pm .032$	$.314 \pm .132$	$.315 \pm .133$
TL-DT	$.954 \pm .017$	$.613 \pm .049$	$.530 \pm .073$	$.014 \pm .008$	$.283 \pm .048$	$.541 \pm .028$	$.314 \pm .132$	$.313 \pm .132$
TL-LGBM	$.954 \pm .016$	$.611 \pm .048$	$.530 \pm .073$	$.014 \pm .008$	$.294 \pm .056$	$.541 \pm .030$	$.314 \pm .132$	$.313 \pm .132$
TL-KR	$.945 \pm .013$	$.594 \pm .045$	$.530 \pm .073$	$.014 \pm .008$	$.288 \pm .054$	$.546 \pm .024$	$.314 \pm .132$	$.314 \pm .133$
τ-risk$_{plug}^{PEHE}$								
SL-DT	$.544 \pm .101$	$.712 \pm .119$	$.505 \pm .064$	$.011 \pm .006$	$.071 \pm .010$	$.656 \pm .033$	$.311 \pm .132$	$.315 \pm .134$
SL-LGBM	$-.532 \pm .085$	$-.080 \pm .083$	$.505 \pm .064$	$.011 \pm .006$	$.070 \pm .009$	$.657 \pm .033$	$.311 \pm .132$	$.315 \pm .134$
SL-KR	$.408 \pm .076$	$.635 \pm .115$	$.505 \pm .064$	$.011 \pm .006$	$.071 \pm .009$	$.657 \pm .033$	$.311 \pm .132$	$.315 \pm .134$
TL-DT	$.640 \pm .060$	$.810 \pm .061$	$.505 \pm .064$	$.011 \pm .006$	$.068 \pm .008$	$.658 \pm .036$	$.311 \pm .132$	$.315 \pm .134$
TL-LGBM	$.620 \pm .064$	$.786 \pm .071$	$.505 \pm .064$	$.011 \pm .006$	$.067 \pm .008$	$.657 \pm .036$	$.311 \pm .132$	$.315 \pm .134$
TL-KR	$.616 \pm .048$	$.883 \pm .023$	$.505 \pm .064$	$.011 \pm .006$	$.069 \pm .008$	$.658 \pm .034$	$.311 \pm .132$	$.315 \pm .134$
τ-risk$_{match}^{ATE}$								
$k = 1$	$.954 \pm .017$	$.599 \pm .054$	$.530 \pm .073$	$.014 \pm .008$	$.271 \pm .048$	$.541 \pm .029$	$.313 \pm .132$	$.313 \pm .132$
$k = 3$	$.956 \pm .015$	$.608 \pm .050$	$.530 \pm .073$	$.014 \pm .008$	$.274 \pm .049$	$.541 \pm .029$	$.313 \pm .132$	$.313 \pm .132$
$k = 5$	$.957 \pm .015$	$.607 \pm .051$	$.530 \pm .073$	$.014 \pm .008$	$.274 \pm .049$	$.542 \pm .028$	$.313 \pm .132$	$.313 \pm .132$
τ-risk$_{match}^{PEHE}$								
$k = 1$	$.629 \pm .054$	$.895 \pm .020$	$.505 \pm .064$	$.011 \pm .006$	$.067 \pm .009$	$.654 \pm .035$	$.311 \pm .132$	$.315 \pm .134$
$k = 3$	$.639 \pm .055$	$.900 \pm .021$	$.505 \pm .064$	$.011 \pm .006$	$.068 \pm .010$	$.656 \pm .036$	$.311 \pm .132$	$.315 \pm .134$
$k = 5$	$.643 \pm .055$	$.900 \pm .021$	$.505 \pm .064$	$.011 \pm .006$	$.068 \pm .010$	$.656 \pm .036$	$.311 \pm .132$	$.315 \pm .134$
τ-risk$_R$								
DT	$-.161 \pm .037$	$-.556 \pm .058$	$-.566 \pm .112$	$-.003 \pm .002$	$-.035 \pm .008$	$-.397 \pm .046$	$-.209 \pm .087$	$-.229 \pm .097$
LGBM	$-.199 \pm .050$	$-.571 \pm .060$	$-.501 \pm .097$	$-.004 \pm .001$	$-.035 \pm .008$	$-.403 \pm .047$	$-.295 \pm .098$	$-.317 \pm .106$
KR	$-.210 \pm .048$	$-.592 \pm .055$	$-.615 \pm .103$	$-.002 \pm .001$	$-.034 \pm .007$	$-.396 \pm .049$	$-.283 \pm .099$	$-.305 \pm .108$

Table A.7: Correlations between model evaluation and ground truth metrics. Numbers are *mean* $\pm$ *standard error* across dataset iterations. *Includes only SL, TL and IPSW.

In Table A.7 we explore the same model evaluation metrics as in Table A.6, but here we are interested in how they correlate with dataset metrics. In practice, this tells something about an evaluation metric's ability to rank candidate models, the validity of which is tested against dataset metrics that are treated as ground truth.

In terms of specific performances, both μ-risk metrics are roughly on par with each other with plain μ-risk being slightly favourable, especially for PEHE. τ-risk$_R$ seems to be relatively stable across all data as well but is seldom the best choice except for ATT estimation.

T-Learning variants of τ-risk$_{plug}$ seem to rank better than their *S-Learning* counterparts, an observation similar to that of Table A.6. Notably, this is visible for the IHDP dataset only. However, what can be clearly observed across all datasets and both plugin and matching metrics is that τ-risk^{ATE} variants correlate much better with ϵ_{ATE} and ϵ_{ATT} than τ-risk^{PEHE} ones. The opposite is also true, that is, τ-risk^{PEHE} metrics correlate better with PEHE than τ-risk^{ATE}. This shows that choosing the type of feedback signal in the metric, so it matches the estimation target certainly improves ranking. Interestingly, this is not the case when picking only the winning candidate model. Thus, if the plugin or matching metrics are considered, their variant (ATE or PEHE) should be aligned with the estimation target (ATE/ATT or CATE respectively). An additional consequence of this conclusion is that if both average and individualised parameters are to be estimated, they should be tackled separately with different evaluation metrics employed per estimation target. This also indirectly suggests that the models that perform well in the ATE estimation task will unlikely be the top performers when it comes to CATE estimation, and vice versa.

Moving on to $\mathcal{R}_{pol}$ in the Jobs dataset, there is clearly only one correct choice with respect to model evaluation – $\mathcal{R}_{pol}$ itself. All other metrics are surprisingly hardly useful at all at the task given their close proximity to a value of 0 (no correlation). This perhaps also demonstrates the difficulty of the policy recommendation task, and

how much it differs from CATE estimation. It also strengthens the argument that the model evaluation target should be aligned with the final estimation target as much as possible.

Overall, a lot of variability can be observed across the evaluation metrics, again resembling the winner-picking task in this regard. As with estimators and base learners, no single evaluation metric appears to be always the best across various problems. Unsurprisingly, there are no free lunches when it comes to choosing an evaluation metric as well, even more so when a metric involves learning too (τ-risk metrics). As for general recommendations, μ-risk seems to be the most stable for PEHE, τ-risk variants for ATE/ATT, and $\mathcal{R}_{pol}$ when it itself is the estimation target.

There are perhaps very few surprises except for two observations. First, as mentioned before, metrics other than $\mathcal{R}_{pol}$ are unexpectedly unhelpful at ranking models with respect to *policy risk*. Second, the *SL-LGBM* variation of τ-risk$_{plug}$ appears to be extremely incorrect at ranking against IHDP to the extent that its scores correlate negatively with dataset metrics. This anomaly indeed matches poor results in the winner selection task (Table A.6) for this metric variant, suggesting the *SL-LGBM* model did not learn the data very well.

Winner Selection and Ranking

Concluding both winner selection and ranking (Tables A.6 and A.7) at the same time, there is evidently substantial variability across evaluation metrics and datasets, regardless of whether the task is model selection or ranking. As a result, it is extremely difficult to provide general recommendations for practitioners without having a better understanding of the task at hand. The best course of action in this case seems to be to prioritise the stability[1] of a metric over its winning frequency. Metrics not involving learning, such as μ-risk or $\mathcal{R}_{pol}$, appear to share those characteristics, making them favourable and reasonably safe defaults. Learning-based metrics like

[1]Stability in terms of performing well, but never terribly, across different tasks and different metric variants.

τ-risk clearly depend on modelling choices (and hyperparameters) of their inner learners, complicating the main problem of causal estimation even further.

Another observation across both tables is that the metrics that could be the best choice for selecting a winning model might not be the best at model ranking, and vice versa. This confirms evaluating a metric on how well it selects a winning model differs from evaluating it on how well it ranks all candidate models. Although both tasks require a validation metric to score all candidate models, model ranking requires the correct order of all models, as opposed to picking the correct winner which completely disregards the importance of ordering the non-winning models. Due to those clear differences in winner and ranking evaluations, incorporating both should be considered when evaluating model selection metrics. The type of information they provide can also be used in different ways, depending on set priorities. For instance, rank correlation can possibly serve as a confidence or stability test for any evaluation metric, in addition to its scores based on winner selection. That is, the better the metric's rank correlation, the more confidence one can have that the candidate model selected as the winner is not a random chance but rather a result of a series of correct evaluations of multiple candidate models. Put alternatively, the more models are ordered correctly, the higher the chance the winning model is placed correctly at the top of the ranking. Another example would be to separate winner and ranking measures depending on the task at hand. If selecting the winning model is a priority, evaluating the winner selection becomes more important. Evaluating the ranking, on the other hand, might be useful when the goal is to select not one but top N models from the collection of candidates, for example, to average their responses for improved robustness instead of relying on a single model. Despite the utility of rank correlation, it appears to be uncommon for the purpose of the assessment of evaluation metrics in the literature but was notably included in [172].

A.1.5 Default Hyperparameters vs. Oracle

We are also interested in how estimators with default hyperparameters (no tuning) perform when compared to those with optimal tuning (*Oracle*). The latter is performed only across hyperparameters $\mathcal{H}$, so each data point represents a combination of a CATE estimator $\mathcal{C}_c$ and a base learner $\mathcal{B}_b$ (e.g. *SL-DT*, *DR-KR*, etc.). Figure A.1 presents obtained results.

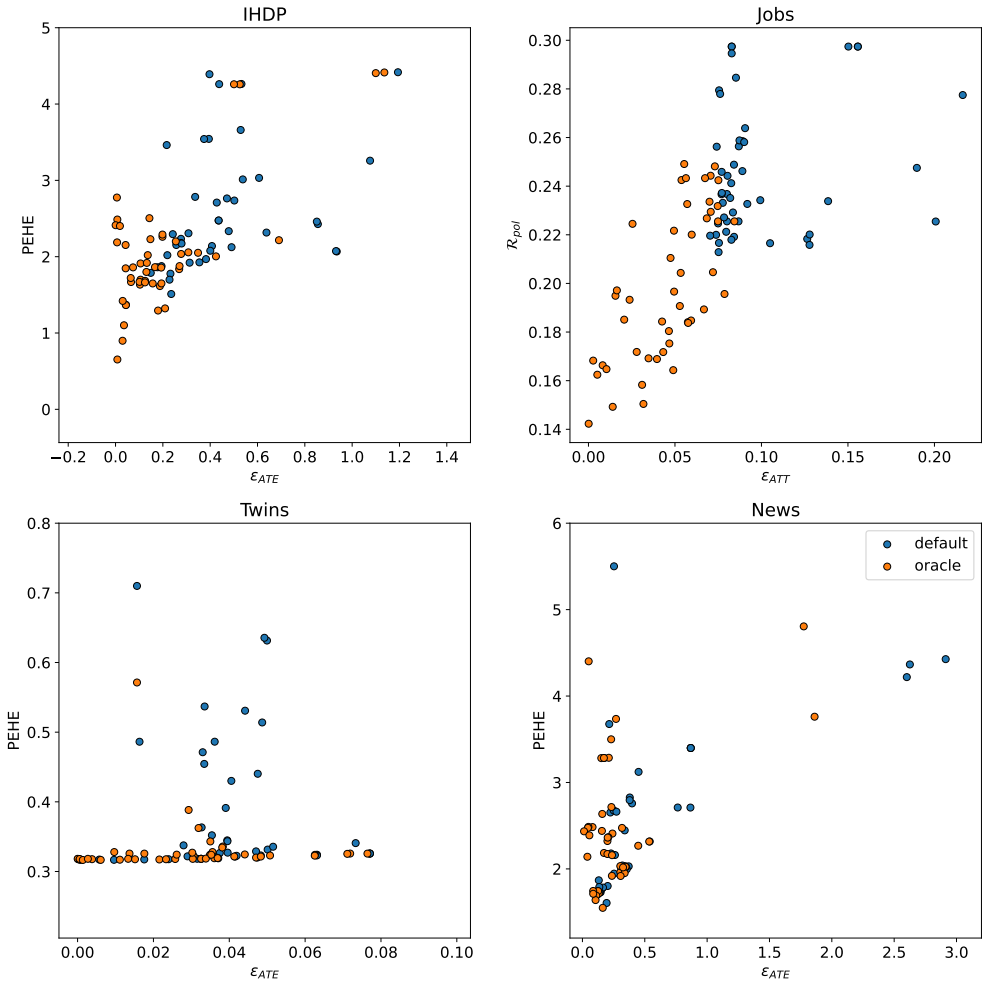


Figure A.1: Performance of CATE estimators with (Oracle) and without (defaults) hyperparameter tuning. Each data point represents the mean across dataset iterations. Lower is better applies to all metrics, thus the closer to the bottom-left corner, the better.

The four plots in Figure A.1 clearly demonstrate the risk associated with using default hyperparameters, and why tuning is of vital importance in order to achieve satisfactory estimation performance. The two top cases against IHDP and Jobs show

the issue perfectly; most of the points representing optimal tuning are clustered near the desirable bottom-left corner of the plots. As we move diagonally from the bottom-left corner of the plots to the top-right, fewer *Oracle* cases, but more default ones, can be observed. The data points linked to default hyperparameters clearly occupy the area representing worse estimation performance as compared to optimal tuning. A similar, but somewhat less clear, picture can be observed with the News dataset. Both groups of points seem to be more mixed, but the main concentration of *Oracle* points still appears to be closer to the desirable bottom-left area. In Twins, the spread across the X axis is particularly inconclusive, whereas, across PEHE, estimators with default hyperparameters perform worse much more frequently.

A general conclusion from the four plots is that not performing any hyperparameter tuning at all can have a severe negative impact on the estimation performance, and the selection of CATE estimators and base learners is not enough to reach optimal results. While the estimation performance also varies across CATE estimators and base learners to some extent (see the spread of orange dots), it is extremely difficult to consistently obtain decent performance with default hyperparameters (blue dots). Evidently, hyperparameter tuning alone can substantially affect the final performance and is arguably more important than the choice of CATE estimators and base learners, as supported by Tables A.2 – A.5 as well.

A.1.6 Correlations Among Dataset Metrics

More accurate CATE predictions are in general expected to lead to more accurate ATE estimates. By extension, a good CATE estimator should do as well in the ATE estimation task, overall encouraging to use of the same instance of an estimator to predict various causal parameters (CATE, ATE) at the same time. This is because CATE estimation is about differences between potential outcomes per individual unit ($\mathcal{Y}_1^{(i)} - \mathcal{Y}_0^{(i)}$), whereas ATE estimation task targets the same quantities, but averaged across the population ($\mathbb{E}[\mathcal{Y}_1 - \mathcal{Y}_0]$). Thus, it is reasonable to expect more accurate individual predictions (CATE) will lead to more accurate prediction of the mean (ATE). We take a closer look at this assumption by investigating

how average-based (e.g. ϵ_{ATE}) and individualised (PEHE) ground truth metrics correlate with each other across various candidate models. Table A.8 presents the results. The top part involves all candidate models across **base learners** and **hyperparameters** $(\mathcal{B}, \mathcal{H})$ per CATE estimator $\mathcal{C}_c$. Lower part includes models across $(\mathcal{C}, \mathcal{H})$ per $\mathcal{B}_b$. The very bottom line refers to correlations obtained across $(\mathcal{C}, \mathcal{B}, \mathcal{H})$. These three different perspectives allow us to see whether modelling choices affect the average-individualised correlations.

	IHDP (ϵ_{ATE},PEHE)	Jobs (ϵ_{ATT},$\mathcal{R}_{pol}$)	Twins (ϵ_{ATE},PEHE)	News (ϵ_{ATE},PEHE)
SL	0.874 ± 0.061	0.056 ± 0.021	0.294 ± 0.047	0.907 ± 0.014
TL	0.757 ± 0.066	0.082 ± 0.035	0.122 ± 0.005	0.980 ± 0.012
IPSW	0.901 ± 0.031	0.127 ± 0.105	0.366 ± 0.011	0.753 ± 0.023
DR	0.644 ± 0.069	0.054 ± 0.024	0.772 ± 0.019	0.989 ± 0.006
DML	0.934 ± 0.037	-0.011 ± 0.047	0.972 ± 0.001	0.522 ± 0.077
XL	0.791 ± 0.031	0.115 ± 0.041	-0.108 ± 0.066	0.746 ± 0.072
CF	0.443 ± 0.155	0.085 ± 0.161	0.799 ± 0.127	0.900 ± 0.028
SL-NN	0.890 ± 0.042	0.212 ± 0.111	0.167 ± 0.026	0.921 ± 0.027
TL-NN	0.238 ± 0.088	0.183 ± 0.114	0.226 ± 0.025	0.662 ± 0.037
L1	0.855 ± 0.085	0.305 ± 0.152	0.906 ± 0.076	0.807 ± 0.029
L2	0.344 ± 0.113	0.142 ± 0.160	-0.136 ± 0.056	0.893 ± 0.035
DT	0.327 ± 0.072	0.117 ± 0.037	0.030 ± 0.018	0.395 ± 0.099
RF	0.466 ± 0.121	0.075 ± 0.096	0.274 ± 0.003	0.902 ± 0.017
ET	0.285 ± 0.125	0.119 ± 0.107	-0.101 ± 0.013	0.925 ± 0.012
KR	0.558 ± 0.035	0.014 ± 0.019	0.032 ± 0.003	0.977 ± 0.013
CB	0.475 ± 0.059	0.037 ± 0.098	0.072 ± 0.021	0.172 ± 0.081
LGBM	0.387 ± 0.087	0.039 ± 0.071	0.023 ± 0.067	0.062 ± 0.120
NN	0.337 ± 0.067	0.182 ± 0.114	0.241 ± 0.023	0.655 ± 0.037
ALL	0.660 ± 0.048	0.022 ± 0.011	0.148 ± 0.003	0.979 ± 0.011

Table A.8: Correlations between ground truth dataset metrics averaged across dataset iterations. Metrics refer to predictions on the test set, collected on all candidate models. Numbers are *mean* $\pm$ *standard error*.

By analysing Table A.8, it is evident that correlations between ground truth metrics can highly depend on both modelling choices and data itself due to considerable variability in correlations across types of estimators as well as datasets. Weak, or in extreme cases negative, correlations are particularly interesting as they challenge the common belief that better CATE predictions translate to better ATEs.

Out of four negative correlations, three are against Twins (see *XL*, *L2* and *ET*), suggesting this task might be particularly challenging to get both CATE and ATE parameters accurate using a single estimator instance. Most correlations for this dataset are indeed weak, though there are four examples with relatively strong correlations (above 0.7) as well, showing the effect is not entirely due to the dataset itself. This is in fact the best example that proves correlations within the same dataset may change drastically (from -0.136 to 0.972), suggesting the ability to handle the estimation of both parameters may depend on a model (which could be due to any of the three in $(\mathcal{C}, \mathcal{B}, \mathcal{H})$). Furthermore, correlations clearly change across datasets as well, even within a single estimator or base learner. The bottom line (see *ALL*) summarises this observation the best, which shows correlations across all candidate models $\mathcal{M}$. In Jobs dataset, all records are particularly consistent in terms of how weak their correlations are (0.022 under *ALL*), further supporting observations from Table A.7 that *policy risk* constitutes a vastly different problem than CATE estimation, pointing towards treatment recommendation and policy optimisation which were already shown to be different from causal effect estimation [34].

As a result of possible multiple sources of variability in correlations (models and datasets), it is difficult to draw any general conclusions with considerable confidence. There are, however, two weaker observations that could offer some useful insights. First, perhaps the row that recorded the most consistently high correlations across all four datasets is *L1*, whereas *CB* and *LGBM* show consistently weak correlations. A possible interpretation of this could be that simpler models are capable of handling both CATE and ATE estimation parameters simultaneously, possibly due to their lower variance. Second, by examining Figure A.1, it can be observed that the degree to which both metrics are correlated (X and Y axis) may depend on the model's relative performance. This is particularly well demonstrated in IHDP, Jobs and News plots – as performances get closer to the desirable bottom-left corner, a stronger correlation between the metrics can be observed. Putting both observations together, a resulting hypothesis would be that relatively simple yet well-performing

models can handle both average and individualised estimation parameters the best. The trick, however, is to find the balance between model simplicity and its performance, where one could possibly come at the cost of another.

While the models that can estimate both CATEs and ATEs with reasonable accuracy and consistency are achievable, they seem to be an exception rather than a rule, especially if specific efforts are not put towards such a balance. Thus, it is generally unreasonable to expect the same model instance to perform well in CATE and ATE estimation simultaneously. Rather, a separate model that targets each estimation parameter separately should be used. A solution could be to perform a model search and tuning in conjunction with an evaluation metric that aligns well with the desired estimation target. This point of view is further supported by Tables A.6 and A.7. Another possible consequence links to how causal estimators are tested on benchmark datasets, where estimators are often validated across multiple ground truth metrics. If separate models, or their variants, should be used per estimation target for optimal results, this practice should be allowed whenever evaluating causal estimators for benchmarking purposes.

A.1.7 Discussion

Putting together observations from Tables A.2 – A.5, it can be seen that all CATE estimators and base learners can achieve great performance levels under optimal model selection choices. As a consequence, the choice of any particular estimator or learner appears to be secondary, especially if pushing CATE performance to the limits is not a priority, with a decent performance being satisfactory. In other words, if most combinations of estimators and learners are equally strong, it does not matter that much which one will end up being selected. It is rather hyperparameter tuning of base learners, which is a form of model selection, that appears to have much more influence on the final CATE performance. The results clearly present that the same combination of estimators and learners but under different model selection schemes can differ in the final CATE performance significantly, ranging from poor to exceptionally good. Given these results and observations, model selection,

particularly base learner hyperparameter tuning, appears to be the deciding factor behind the quality of the final CATE performances, leaving the choices of particular CATE estimators and base learners as less influential.

An important note, however, is that all this is true under a crucial condition of optimal model selection decisions. Can we make optimal choices with model selection methods currently available? As evidenced by the presented results, not quite, due to a clear gap between the performances achieved under optimal and non-optimal choices. Thus, even though many estimators can potentially achieve great performances, they are not accessible with commonly used selection metrics. Going back to the graphical example in Section 4.3, perhaps we can link the performances under the *Oracle* model selection as the desirable but non-trivial to identify solution *Case 2*. This definitely shows the importance of further research into causal model selection, so we can eventually identify those great performances we already know are feasible.

There is also an alternative interpretation of the results from the perspective of the graphical example in Section 4.3. More precisely, it is also possible to interpret the results under the *Oracle* as the undesirable *Case 3*, which seemingly achieves exceptional performance given its low PEHE but clearly did not capture the main trends very well. This possibility highlights an important issue. How can we distinguish between *Case 2* and *Case 3* if the only piece of information we have is PEHE? How can we be sure if the results under the *Oracle* are in fact desirable? Without the aid of plots, as in Section 4.3, this becomes a challenging task. And as demonstrated by the Section 4.3 example, simple metrics like MSE may not be enough for model selection, whereas PEHE might not be enough for final CATE estimation performance evaluation, as both used in separation fail to identify the desirable *Case 2*. This problem yet again shows the importance of further research into causal model selection in order to understand those issues more fundamentally.

Also noteworthy are the performance improvements as the search space of model selection gets bigger, that is, going from Tables A.2 – A.5 (smaller search space) to Table A.6 (the biggest search space), especially when observing the results under the *Oracle*. At the very least, it is a mere confirmation that wider search spaces can potentially deliver better solutions, which of course comes at higher computational costs. But perhaps it is even more important to state the reverse, that smaller search spaces can potentially be detrimental to the final CATE performance. This point of view indirectly stresses the importance of performing model selection as part of the CATE estimation and optimisation process. Taken to the extreme, constraining the solution to a particular model using general defaults (no model selection at all) can be risky. Previous analysis suggests the choice of CATE estimators and base learners is indeed less important under optimal model selection decisions, but this still assumes some form of model selection, hyperparameter tuning of base learners in this case. Thus, tuning hyperparameters at least to some limited extent should be the bare minimum of the CATE modelling process. In fact, this observation is supported by Figure A.1, wherein performances with default hyperparameters are clearly and consistently inferior to those with optimally selected hyperparameters, showing yet again the power of thorough hyperparameter tuning alone.

Finally, model selection methods that themselves perform learning internally (all τ-risk) are subject to tuning, or model selection in general, leaving the main problem of selection unsolved. This is evidenced by significant variability between those selection methods and their variants as they are clearly sensitive to how internal learners respond to provided data. Non-learning selection metrics (all μ-risk) seem to be more stable for this reason, making them a safer choice as a whole. But crucially, none of the selection metrics attain the performance levels obtained by the *Oracle*.

A.2 Experimental Details

The proposed experimental design involves four major stages: data pre-processing (stage 1), learning, prediction, and validation (stage 2), post-processing (stage 3), and compilation of results (stage 4). Figure A.2 depicts the framework and its components. Hyperparameters $\mathcal{H}$ are defined per each base learner (see Section 4.4.4). Validation metrics $\mathcal{L}_V$ are obtained on validation sets $\mathcal{D}_{val}$ (see Section 4.4.2). Test metrics $\mathcal{L}_T = \{\epsilon_{ATE}, \text{PEHE}, \epsilon_{\text{ATT}}, \mathcal{R}_{\text{pol}}\}$ are calculated against test sets $\mathcal{D}_{te}$.

Stage 1 All data splits are performed and stored before any learning to ensure fair comparison between methods. This involves preparing 10 iterations of each of the four datasets (see *iter*), where each iteration is split into training and testing parts (*train* and *test* in the diagram). Furthermore, each training set is split into 10 cross-validation folds (see *fold*), stratified on treatment status T, wherein each fold consists of a training and validation part (*train* and *valid* respectively), following recommendations in [131]. Performing all the splits only once and saving them for later use ensures all estimators are trained on exactly the same training data and then evaluated on the same validation and test sets. The only sources of variability thus are the internal workings of learners and minor differences between dataset iterations.

Stage 2 All τ-risk metrics perform internal learning on validation sets before returning their evaluation scores. Because all validation folds are stored after stage 1, all necessary learning can be performed only once per τ-risk variant and saved for later use. This is why the stage 2 diagram includes an arrow pointing from *valid* to metrics. τ-risk learning is done via 5-fold cross-fitting to avoid overfitting [131]. Once metrics are ready to use, it is possible to start fitting causal estimators. Each candidate model from the $(\mathcal{C}, \mathcal{B}, \mathcal{H})$ search space is trained on training data and provides predictions on validation data that are then used to calculate evaluation metrics $\mathcal{L}_V$ (top half of stage 2). All repeated across 10 CV folds and 10 dataset iterations.

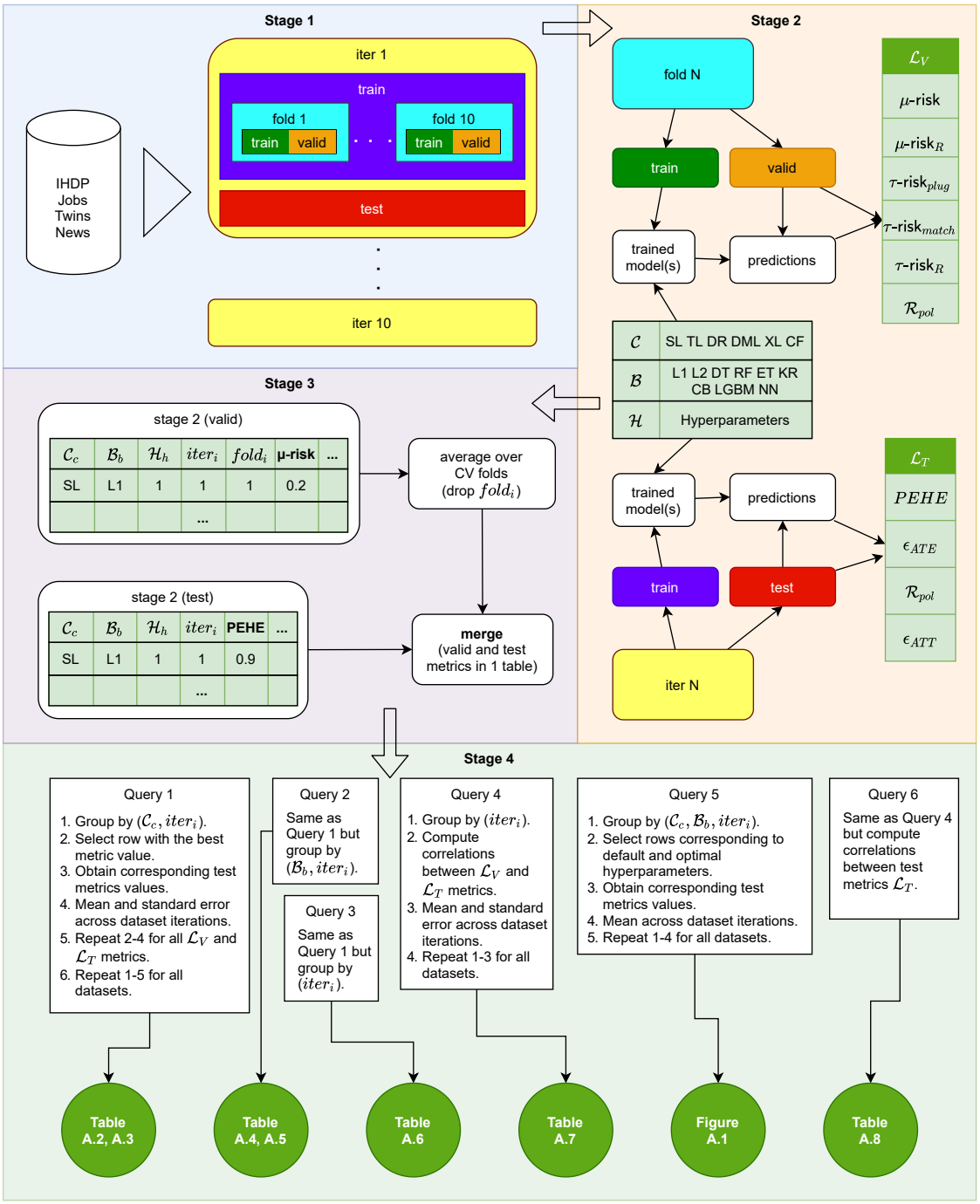


Figure A.2: Diagram of the proposed experimental design with four main stages. *Stage 1*: data pre-processing, *Stage 2*: learning, prediction, and validation, *Stage 3*: post-processing, and *Stage 4*: compilation of results. Note matching colours of different parts of data.

This simulates the standard model selection procedure via cross-validation. Next, all candidate models are trained again but now on the entire training set coming from a dataset iteration (bottom half of stage 2), followed by test predictions and calculation of ground truth metrics $\mathcal{L}_T$. This again mimics standard model selection practice; once a modelling choice is made, the final candidate model is trained on all the data and tested on new data that the model was not exposed to before. In our case, the difference is that we collect test performances of all candidate models, not just the winning ones. Whenever a CATE estimator performs cross-fitting internally (*DR*, *DML*), we set the number of folds to 5 [131]. We explore the search space of candidate models $(\mathcal{C}, \mathcal{B}, \mathcal{H})$ as exhaustively as possible, but with certain limitations due to computational reasons. See Section 4.6.1 for more details.

Stage 3 All validation $\mathcal{L}_V$ and test $\mathcal{L}_T$ metric scores that were obtained in stage 2 are stored together with information about used candidate models, dataset iteration index and CV fold index in the case of validation metrics. To further simulate the usual model selection procedure, it is necessary to average validation metric scores across all 10 CV folds. This is done per each unique combination of $(\mathcal{C}_c, \mathcal{B}_b, \mathcal{H}_h, iter_i)$ across folds $1-10$. Column $fold_i$ now can be dropped. Next, average validation scores can be merged with test scores using $(\mathcal{C}_c, \mathcal{B}_b, \mathcal{H}_h, iter_i)$ as the merging key. The result is a single table containing all candidate models $\mathcal{M}$ and all validation and test metric scores associated with them. This allows us to further analyse how using various validation metrics affects the final test performance of causal estimators. Put differently, it is now possible to simulate using a specific validation metric $\mathcal{L}_V$ for model selection purposes and immediately seeing consequences in test performance $\mathcal{L}_T$. As test performances of all candidate models are stored, it is also possible to identify *Oracle* performances $\mathcal{L}^{**}$ (see Equation (4.11)) with respect to each ground truth metric $\mathcal{L}_T$. Such an all-encompassing table is created per each dataset.

Stage 4 Having all the results in a single table per dataset allows us to explore and analyse them from different perspectives. In order to provide a better intuition as to how the presented tables and figures have been created using obtained numbers, SQL-like pseudocode queries have been included in the diagram, each of them linked to a particular table or figure that is presented in this chapter in Section 4.5 (or Appendix A.1). Note that all queries are repeated for all datasets. Also noteworthy is the fact that the numbers presented in all tables are means and standard errors across dataset iterations, which are stored under the $iter_i$ column in the aforementioned final tables. Finally, when performing model selection across dataset iterations, in each iteration a different candidate model can be selected as a winner. In some sense, each dataset iteration is treated as a separate mini-dataset.

B

Supplementary Material for Chapter 5

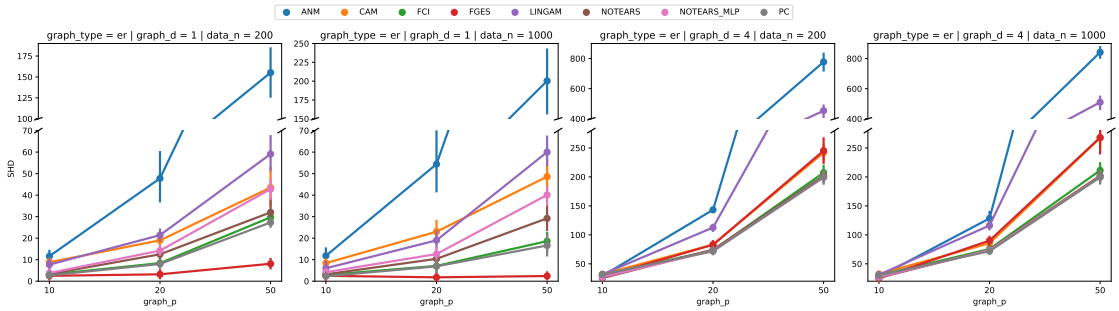
Contents

B.1 Extended Results

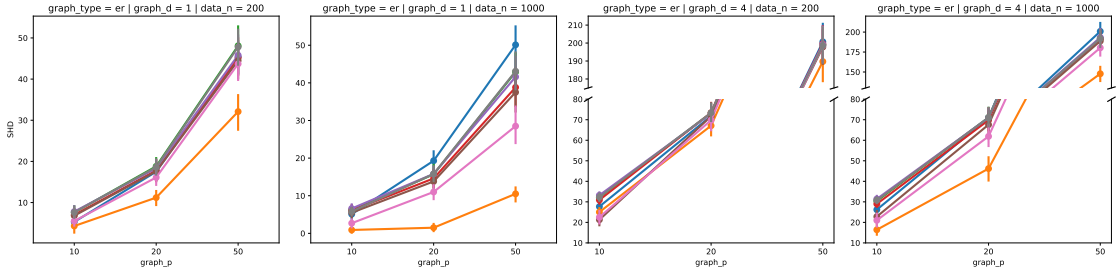
The following results complement the ones presented so far in Section 5.4.5. Although they do not change the overall conclusions of the chapter, they offer additional analysis that may facilitate a deeper understanding of the problem and the final outcomes.

B.1.1 Best Performances

Figures B.1 and B.2 focus on performances achieved specifically with the best (oracle) hyperparameters. These correspond to the best performances presented in Figure 5.4 (blue bars). Some important observations: a) algorithms differ in performance even with access to a hyperparameter oracle, b) number of graph nodes and edge density can significantly impact performance, and c) no algorithm performs best under all conditions (linear vs. nonlinear).

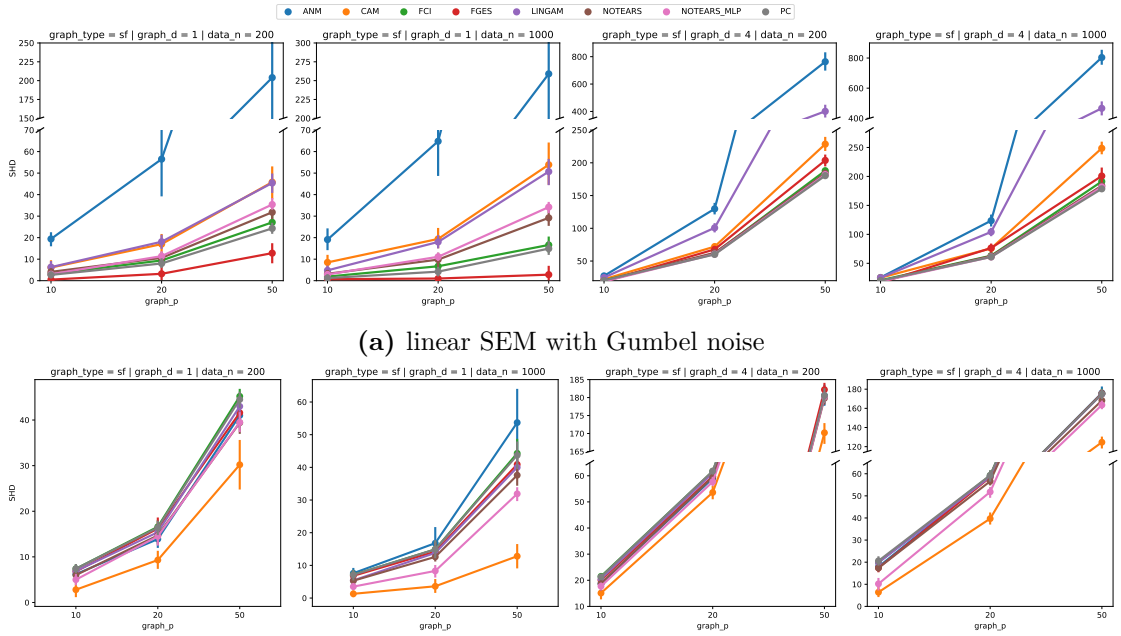


(a) linear SEM with Gumbel noise



(b) nonlinear (GP) SEM with Gaussian noise

Figure B.1: Best performances against ER graphs. Error bars are standard errors.



(a) linear SEM with Gumbel noise

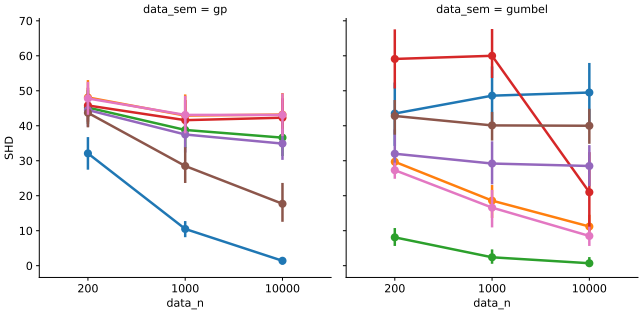
(b) nonlinear (GP) SEM with Gaussian noise

Figure B.2: Best performances against SF graphs. Error bars are standard errors.

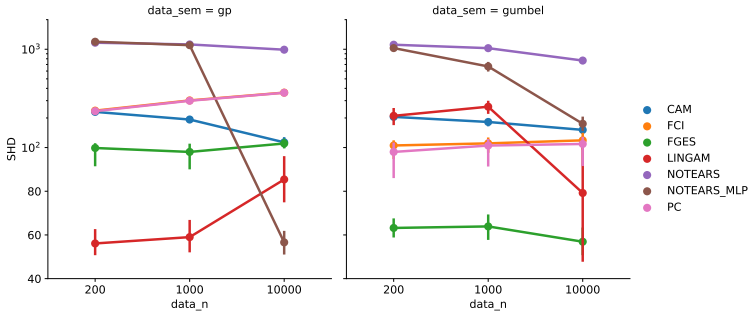
B.1.2 Large Graphs with Large Sample Size

Previous results showed that $p = 50$ graphs are much more challenging than smaller ones. Figure B.3a demonstrates that with enough data samples, even such larger graphs are possible to solve accurately. As presented in subfigure (b), increased sample size can also help with robustness. Note also how the degree of the benefits varies between algorithms.

Figures B.4, B.5 and B.6 further extend the results to performance distributions over all hyperparameters (SHD, FP, and FN respectively). Notice how a larger sample size increases the proportion of good performances (the lines shift to the left and hit higher proportion numbers for lower metric values). If the improvement trend remains for even larger sample sizes, one can wonder if the HP misspecification issue could be solved entirely by larger quantities of data alone.



(a) best performances



(b) worst performances

Figure B.3: Performances for ER1 $p = 50$ graphs.

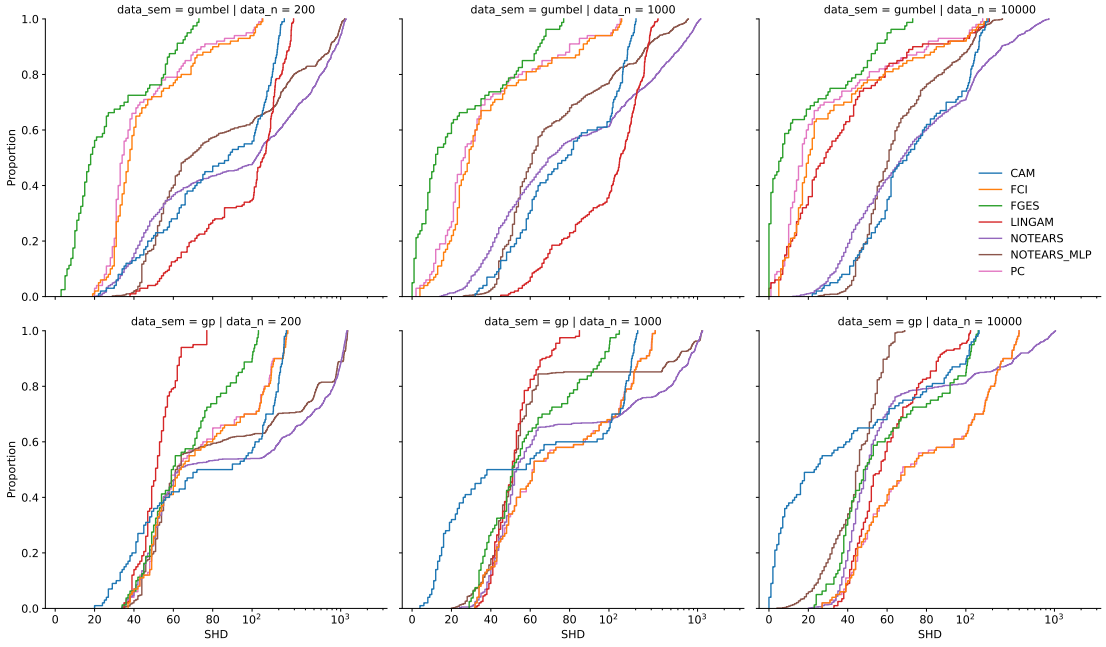


Figure B.4: Distribution of SHD performances across all hyperparameters (ER1 $p = 50$ graphs). A vertical line at $SHD = 0$ is desirable.

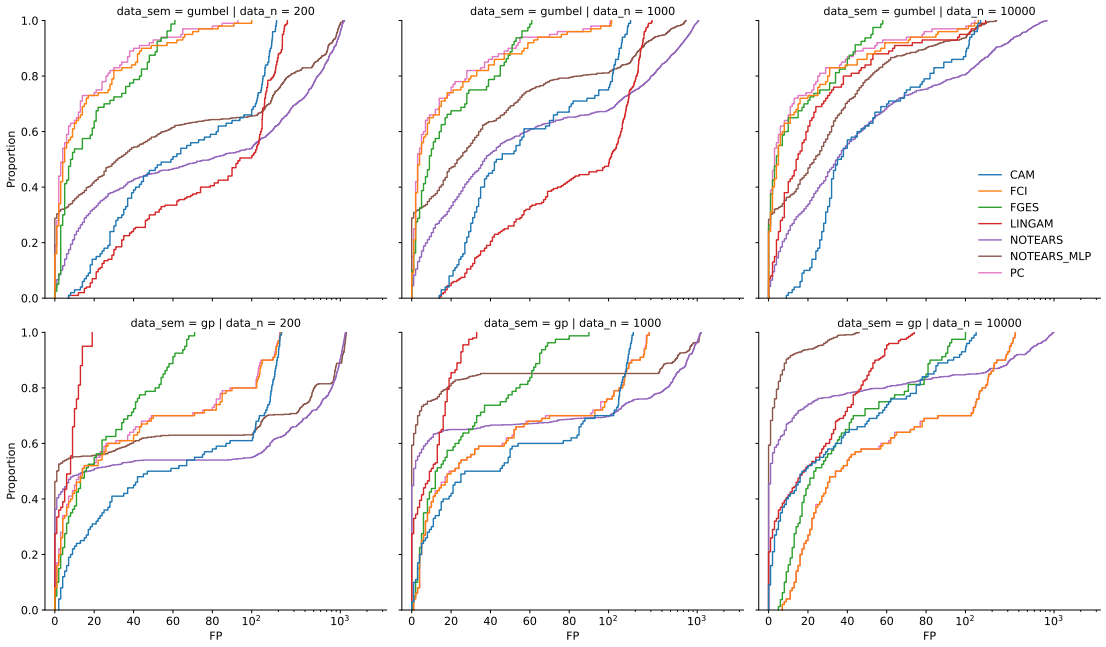


Figure B.5: Distribution of false positives (FPs) across all hyperparameters (ER1 $p = 50$ graphs). A vertical line at $FP = 0$ is desirable.

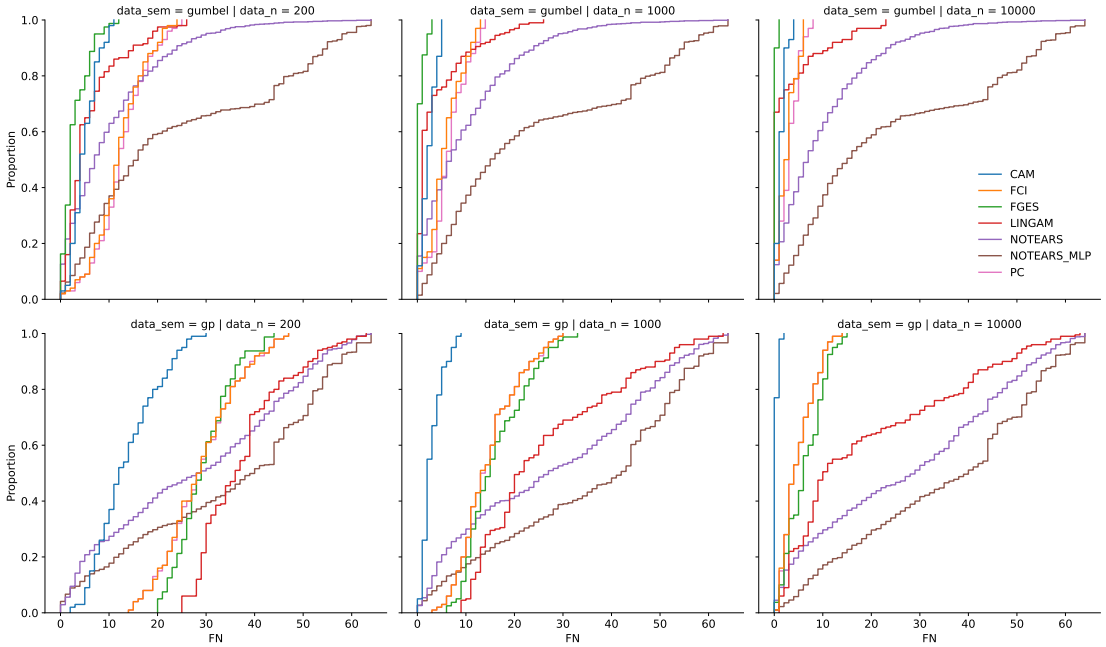


Figure B.6: Distribution of false negatives (FNs) across all hyperparameters (ER1 $p = 50$ graphs). A vertical line at $FN = 0$ is desirable.

B.1.3 Performance vs. Hyperparameter Quality

Figures B.7, B.8 and B.9 complement Figure 5.4 from the main content by showing the results for other types of DGPs.

In Figure B.7, we can see that default HPs perform well across all settings (orange and green bars are close to the blue ones). Furthermore, sparse graphs with 10 and 20 nodes are recovered with good accuracy by most algorithms, but 50-node graphs become much more challenging with the current sample size. Recovery of dense graphs (right half of the figure) is generally unacceptably inaccurate with current tools no matter the size of the graph and learning method.

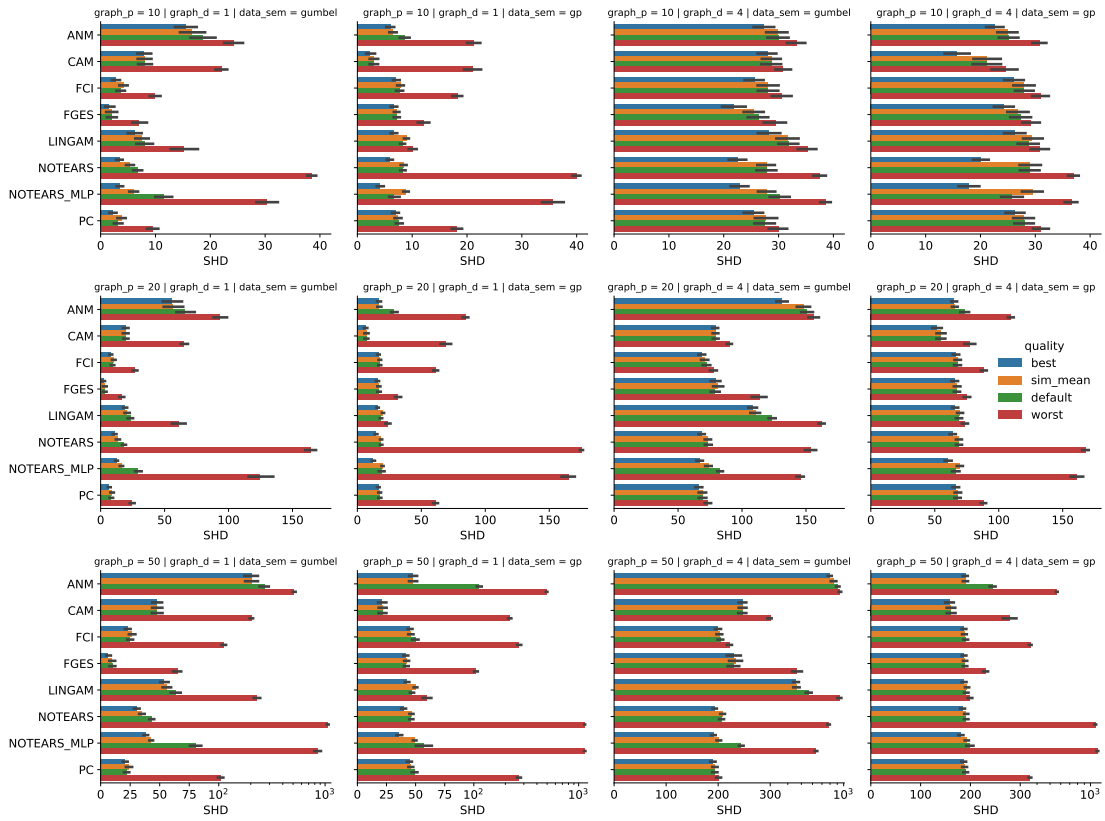


Figure B.7: SHD performances depending on the quality of selected hyperparameters (colours), grouped by DGP properties such as number of nodes (*graph_p*), edge density (*graph_d*) and SEM type (*data_sem*; gumbel is linear, gp nonlinear).

In Figure B.8 that is concerned with false positives, it can be observed that the main conclusions derived from Figure 5.4 still hold across most DGP settings – that FPs can be almost completely reduced given well-specified HPs. A closer examination of the figure suggests that certain DGP properties do play a role here. First, there is a higher chance of reducing FPs in nonlinear cases than in linear ones (gp vs. gumbel). Second, specifically in linear settings, dense graphs are considerably more challenging than sparse ones. Interestingly, in nonlinear cases, neither graph size nor density seems to affect FP reduction.

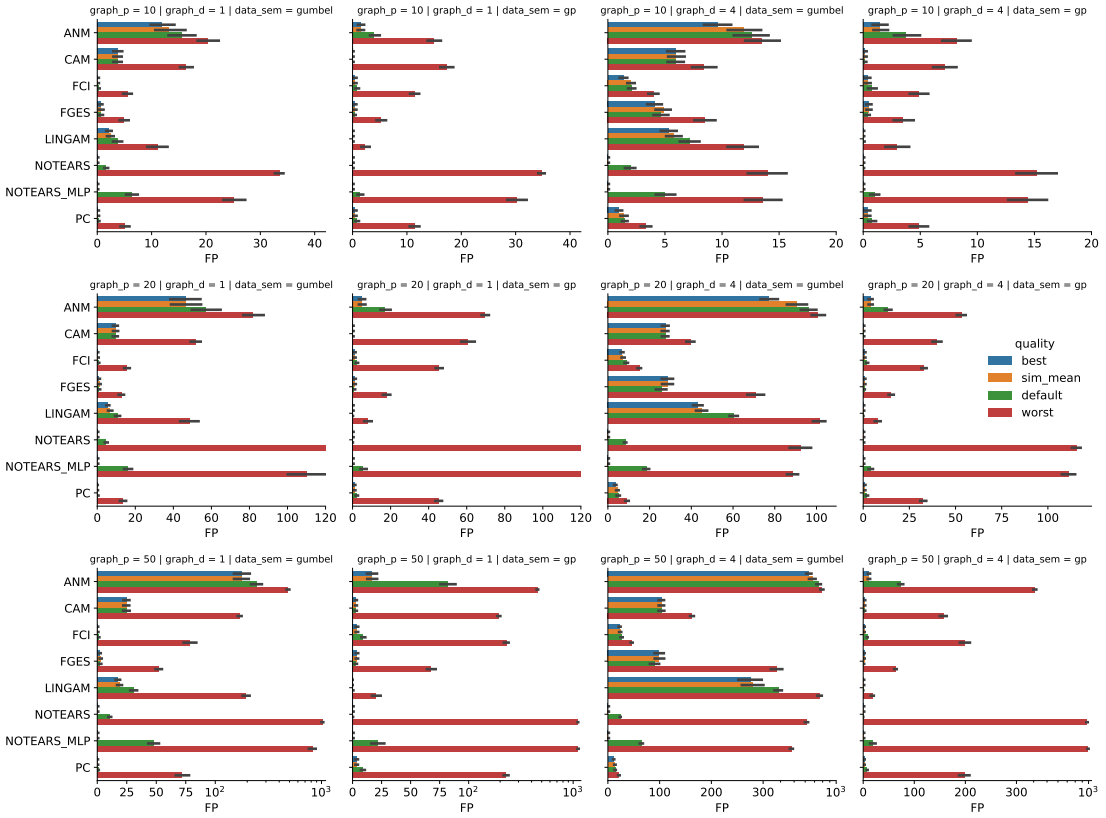


Figure B.8: FP (false positive) performances depending on the quality of selected hyperparameters (colours), grouped by DGP properties such as the number of nodes (*graph_p*), edge density (*graph_d*) and SEM type (*data_sem*; gumbel is linear, gp nonlinear).

As for Figure B.9 and false negatives, only sparse linear cases seem to have acceptable levels of prediction mistakes. As we move to more dense and nonlinear settings, FN errors grow considerably. The latter observation combined with FPs reduced mostly to zero in Figure B.8 suggests that the majority of prediction mistakes in general come from undetected edges (FNs).

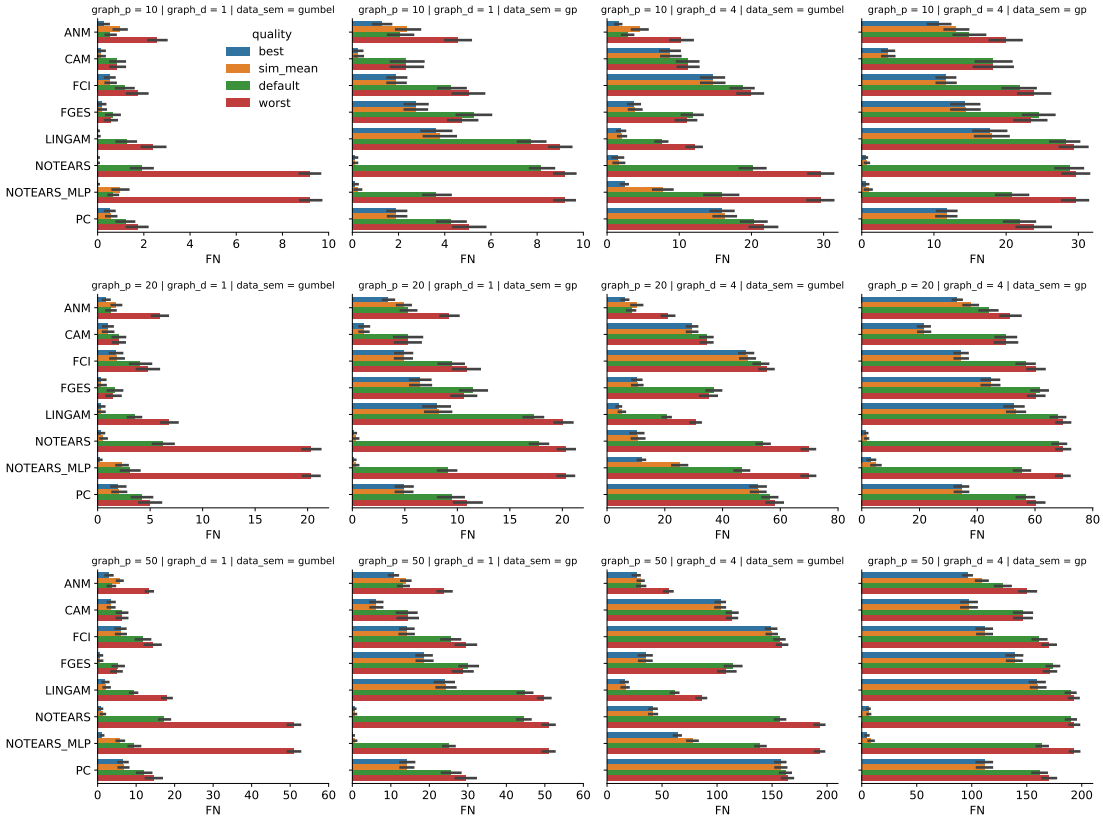
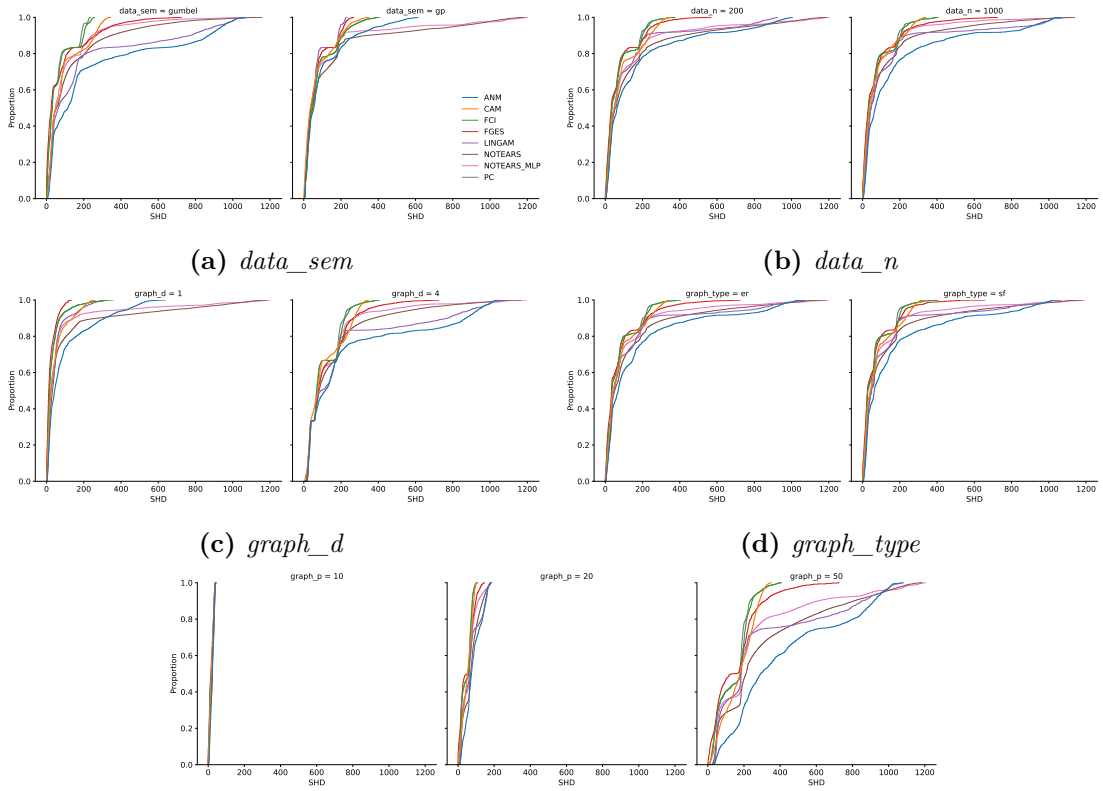


Figure B.9: FN (false negative) performances depending on the quality of selected hyperparameters (colours), grouped by DGP properties such as the number of nodes (*graph_p*), edge density (*graph_d*) and SEM type (*data_sem*; gumbel is linear, gp nonlinear).

B.1.4 Performance Distribution Across Hyperparameters

Figures B.10, B.11 and B.12 complement Figure 5.3 from the main content by showing the results for other types of DGPs. They generally confirm the main claim based on Figure 5.3 – that on average no algorithm dominates the others. The only minor exceptions from this statement can be seen when analysing large graphs ($p = 50$) and different SEMs (gp vs. gumbel), but only to a limited extent.



(a) *data_sem*

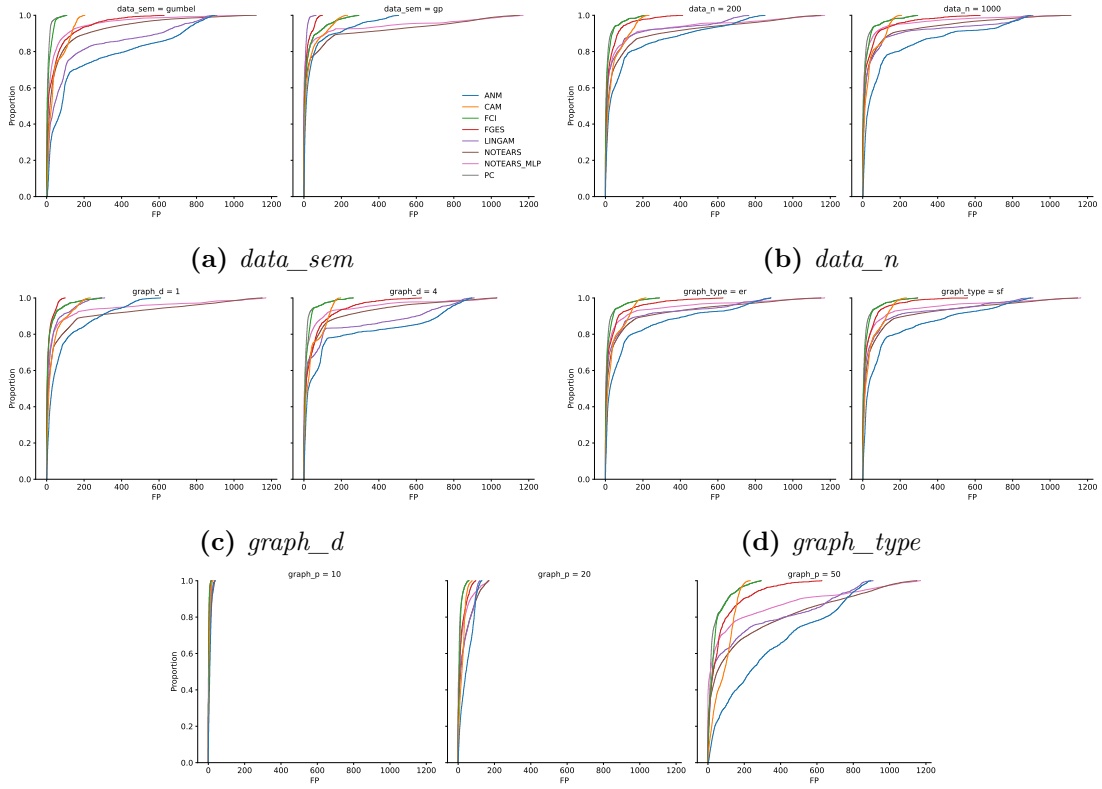
(b) *data_n*

(c) *graph_d*

(d) *graph_type*

(e) *graph_p*

Figure B.10: Distributions of SHD performances across all hyperparameters, grouped by SEM types (*data_sem*), sample size (*data_n*), edge density (*graph_d*), graph type (ER or SF) and number of nodes (*graph_p*).



(a) *data_sem*

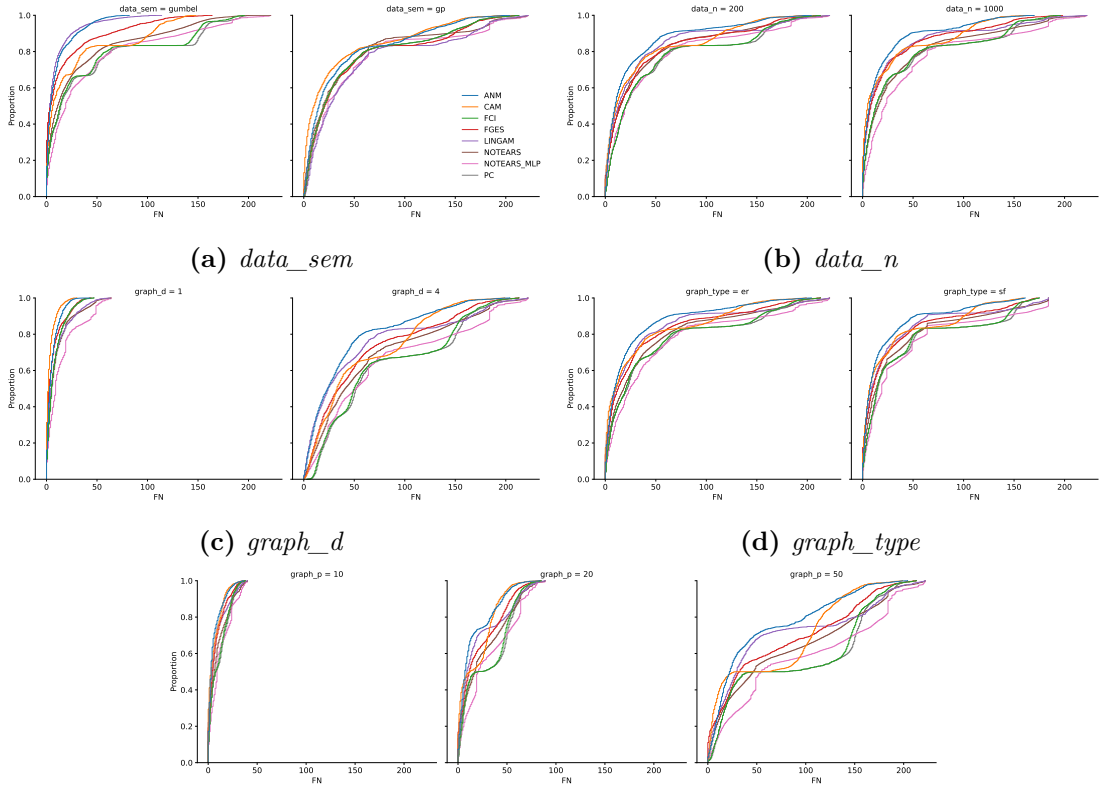
(b) *data_n*

(c) *graph_d*

(d) *graph_type*

(e) *graph_p*

Figure B.11: Distributions of false positive (FP) performances across all hyperparameters, grouped by SEM types (*data_sem*), sample size (*data_n*), edge density (*graph_d*), graph type (ER or SF) and number of nodes (*graph_p*).



(a) *data_sem* **(b)** *data_n*

(c) *graph_d* **(d)** *graph_type*

(e) *graph_p*

Figure B.12: Distributions of false negative (FN) performances across all hyperparameters, grouped by SEM types (*data_sem*), sample size (*data_n*), edge density (*graph_d*), graph type (ER or SF) and number of nodes (*graph_p*).

B.1.5 Winning Algorithms vs. Simulation Properties

Figure B.13 complements Table 5.1 from the main content but removes hyperparameters from the picture in order to analyse how DGP properties alone affect the winning odds of the algorithms. The results presented here involve the use of the best hyperparameter values. From this perspective, it is clear that no algorithm is the best under all conditions, and that SEM types involved and edge density have the strongest impact on the winning odds.

B.1.6 Winning Algorithms vs. Hyperparameter Quality

Figure B.14 forms the basis for Table 5.1 from the main content. It presents winning percentages of algorithms across different DGP types, from which Table 5.1 was derived.

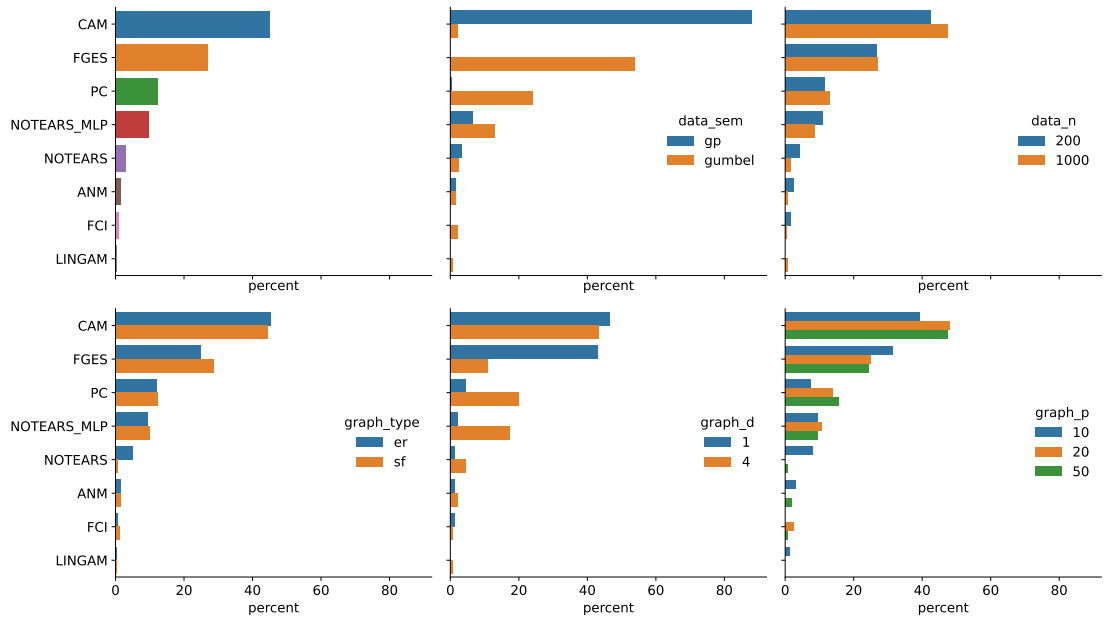


Figure B.13: Percentage of wins per algorithm depending on DGP properties.

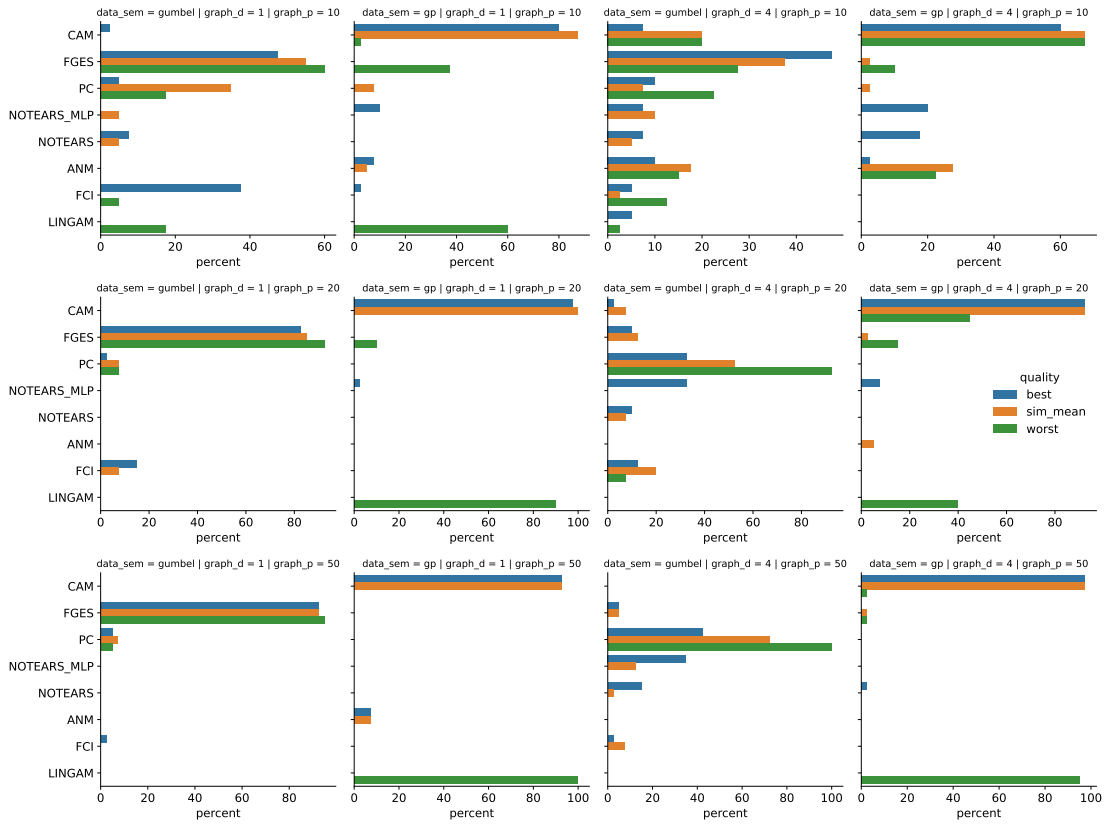


Figure B.14: Percentages of winning algorithms under different data, graph and hyperparameter conditions.

B.2 A Guide To Algorithm Selection

B.2.1 General Recommendations

- Current algorithms seem to work reasonably well for sparse graphs of up to 20 nodes. Bigger graphs (50 nodes) are also possible to solve, but more data might be required ($10,000$ samples) to achieve good accuracy. The accuracy of recovered causal structures drops dramatically for dense graphs.

  Recommendation: Stick to sparse graphs with up to 20 nodes (moderate amount of data) or up to 50 nodes (a lot of data). Avoid dense graphs.

- No single algorithm is the best option for all problems. Some perform the best under very specific conditions.

  Recommendation: Choose an algorithm that is the most likely to accurately solve the problem at hand based on assumptions derived from data.

- The best choice of an algorithm may depend not only on graph and data properties but also on the availability of quality hyperparameters. This is because algorithms vary in robustness to misspecified hyperparameters.

  Recommendation: When selecting the best algorithm for the problem at hand, take into account the type of hyperparameters that are available and algorithm's robustness to misspecified hyperparameters.

B.2.2 Hyperparameter Selection Strategies

Optimal hyperparameters are almost never available in causal structure recovery problems due to inaccessible ground truth. Some methods provide scores that can be used to decide whether a set of hyperparameters is better than others for the same algorithm. However, this strategy cannot be used to compare different algorithms to each other, as they are likely to use different score metrics (while some

use none at all). In addition, in order to use those scores, an algorithm's internal code must be modified in most cases, creating a substantial barrier to practitioners.

Thus, default hyperparameters might be a reasonable selection strategy as they often work almost as well as the optimal ones. This is especially the case if the recommended defaults have been derived from data problems similar in nature to the problem at hand.

If, however, the problem to solve is believed to be fairly unique, blindly using default hyperparameters without considering other factors might not be safe. In this case, the best course of action (assuming hyperparameter tuning is not an option) might be to still use default hyperparameters but choose the algorithm that is the most robust to hyperparameter misspecification under specific graph and data conditions that are believed to apply to the problem at hand.

B.2.3 How to Select Algorithms

Figure B.15 summarises the best algorithm choices based on Figure B.14, which takes into consideration data and graph properties as well as the types of hyperparameters available.

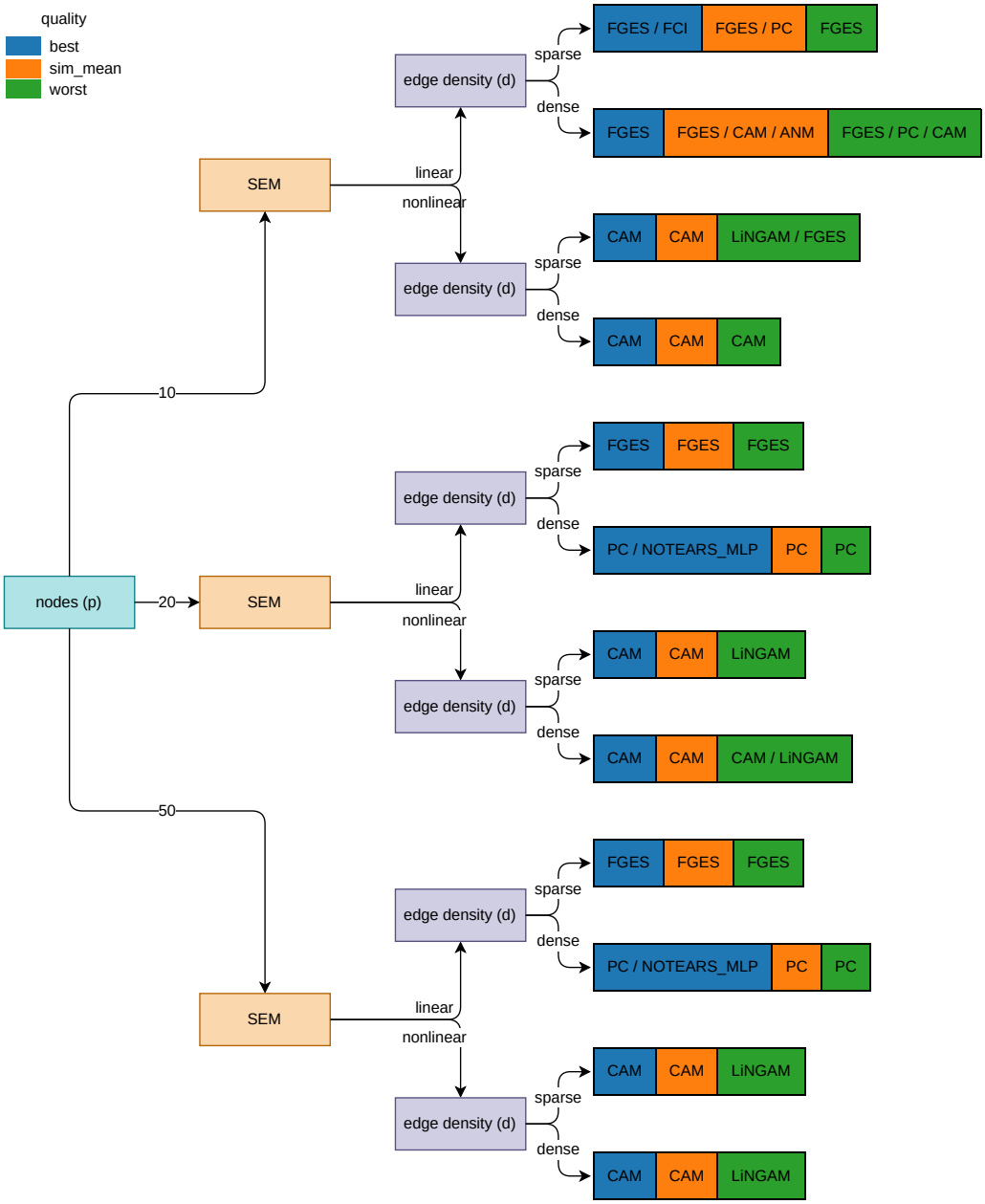


Figure B.15: Recommended algorithm choices based on the number of graph nodes, SEM types and edge density in the graph. The final choice depends also on the type of hyperparameters available – see 'quality' colours. In case there is no clear winner, multiple choices are provided in the order of a higher winning percentage.

B.3 Experimental Details

B.3.1 Hyperparameters

We attempted to explore the hyperparameters as thoroughly as possible to make our findings general enough, while at the same time managing computational demands that increase with each added hyperparameter and explored value. In addition, some methods are considerably more demanding than others, which makes the exploration even more difficult. With these constraints in mind, we believe our hyperparameter exploration accurately reflects common practice.

algorithm	hyperparameters and values	card[a] (total)
ANM	$alpha \in \{0.001^*, 0.002, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.3, 0.5\}$	10 (10)
CAM	$cutoff \in \{0.001^*, 0.002, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.3, 0.5\}$ $score$ = nonlinear $selmethod$ = gamboost $prunmethod$ = gam	10 (10)
FCI	$alpha \in \{0.001^*, 0.002, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.3, 0.5\}$ $test$ = fisher-z-test	10 (10)
FGES	$penaltyDiscount \in \{0.001, 0.01, 0.1, 0.25, 0.5, 0.75, 1, 1.5^*\}$ $score$ = sem-bic	8 (8)
LiNGAM	$thresh \in \{0.001, 0.002, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.3, 0.5^*\}$ $max\_iter \in \{100^*, 1000\}$	10 (20) 2
NOTEARS	$lambda1 \in \{0.001, 0.002, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2^*, 0.3, 0.5\}$ $max\_iter \in \{100^*, 1000\}$ $w\_threshold \in \{0.001, 0.002, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2^*, 0.3, 0.5\}$ $loss\_type$ = l2 $h\_tol = 1e-8$ $rho\_max = 1e+16$	10 (200) 2 10
NOTEARS MLP	$lambda1 \in \{0.001, 0.01^*, 0.1\}$ $lambda2 \in \{0.001, 0.01, 0.1^*\}$ $w\_threshold \in \{0.1, 0.3, 0.5^*\}$ $hidden\_layers = 1$ $hidden\_units \in \{8, 16^*, 32\}$ $max\_iter = 100$ $h\_tol = 1e-8$ $rho\_max = 1e+16$	3 (81) 3 3 3
PC	$alpha \in \{0.001, 0.002^*, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.3, 0.5\}$ $indepTest$ = gaussCItest	10 (10)

Table B.1: Hyperparameter search spaces defined per algorithm. Note that hyperparameters are in most cases continuous, but we explore a discrete space of values. *Found to perform best on average across all simulations (sim_mean). [a]Cardinality of the hyperparameters considered in the experiments.

B.3.2 Summary of Algorithms

algorithm	default hyperparameters	package	paper
ANM	$alpha = 0.05$	gCastle	[51]
CAM	$cutoff = 0.001$ $score =$ nonlinear $selmethod =$ gamboost $prunmethod =$ gam	cdt	[55]
FCI	$alpha = 0.01$	tetrad	[201]
FGES	$penaltyDiscount = 2.0$	tetrad	[202]
LiNGAM	$thresh = 0.3$ $max\_iter = 1000$	gCastle	[50]
NOTEARS	$lambda1 = 0.1$ $max\_iter = 100$ $w\_threshold = 0.3$ $loss\_type =$ l2 $h\_tol = 1e-8$ $rho\_max = 1e+16$	gCastle	[56]
NOTEARS MLP	$lambda1 = 0.01$ $lambda2 = 0.01$ $w\_threshold = 0.3$ $hidden\_layers = 1$ $hidden\_units = 10$ $max\_iter = 100$ $h\_tol = 1e-8$ $rho\_max = 1e+16$	gCastle	[36]
PC	$alpha = 0.01$	pcalg	[200]

Table B.2: Summary of incorporated algorithms and their sources. Recommended default hyperparameters have been derived from respective papers as much as possible. If necessary, they have been further supplemented with defaults suggested within respective packages.

B.3.3 Summary of Packages

package	paper	link
gCastle	[224]	https://github.com/huawei-noah/trustworthyAI/tree/master/gcastle
cdt	[225]	https://fentechsolutions.github.io/CausalDiscoveryToolbox
tetrad	[226]	https://cmu-phil.github.io/tetrad/manual/
pcalg	[227]	https://cran.r-project.org/package=pcalg

Table B.3: Summary of incorporated algorithm packages.

References

[1] Bernhard Schölkopf. “Causality for Machine Learning”. In: *Probabilistic and Causal Inference: The Works of Judea Pearl*. 2022, pp. 765–804.

[2] Edward H. Simpson. “The Interpretation of Interaction in Contingency Tables”. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 13.2 (1951), pp. 238–241.

[3] Sewall Wright. “The Relative Importance of Heredity and Environment in Determining the Piebald Pattern of Guinea-Pigs”. In: *Proceedings of the National Academy of Sciences* 6.6 (June 1920), pp. 320–332.

[4] Sewall Wright. “Correlation and Causation”. In: *Journal of Agricultural Research* 20.7 (1921), p. 557.

[5] Hubert M. Blalock. *Causal Inferences in Nonexperimental Research*. University of North Carolina Press, 1964.

[6] Otis Dudley Duncan. “Path Analysis: Sociological Examples”. In: *American Journal of Sociology* 72.1 (July 1966), pp. 1–16.

[7] Arthur S. Goldberger. “Structural Equation Methods in the Social Sciences”. In: *Econometrica* 40.6 (1972), pp. 979–1001.

[8] Sewall Wright. “On "Path Analysis in Genetic Epidemiology: A Critique".” In: *American Journal of Human Genetics* 35.4 (July 1983), pp. 757–768.

[9] Judea Pearl and Dana Mackenzie. *The Book of Why: The New Science of Cause and Effect*. 1st. USA: Basic Books, Inc., 2018.

[10] Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, Mar. 2000.

[11] Donald B. Rubin. “Estimating Causal Effects of Treatments in Randomized and Nonrandomized Studies.” In: *Journal of Educational Psychology* 66.5 (1974), pp. 688–701.

[12] Peter Spirtes, Clark N. Glymour, Richard Scheines, and David Heckerman. *Causation, Prediction, and Search*. MIT Press, 2000.

[13] Trevor Hastie, Robert Tibshirani, and Jerome H Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Vol. 2. Springer, 2009.

[14] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Vol. 4. 4. Springer, 2006.

[15] Frank Rosenblatt. “The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain.” In: *Psychological Review* 65.6 (1958), p. 386.

[16] Marvin L. Minsky and Seymour A. Papert. *Perceptrons: An Introduction to Computational Geometry*. MIT Press, 1969.

[17] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. "Multilayer Feedforward Networks Are Universal Approximators". In: *Neural Networks* 2.5 (Jan. 1989), pp. 359–366.

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

[19] Yann LeCun, Bernhard Boser, John Denker, Donnie Henderson, R. Howard, Wayne Hubbard, and Lawrence Jackel. "Handwritten Digit Recognition with a Back-Propagation Network". In: *Advances in Neural Information Processing Systems*. Vol. 2. Morgan-Kaufmann, 1989.

[20] Yann LeCun, Koray Kavukcuoglu, and Clément Farabet. "Convolutional Networks and Applications in Vision". In: *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*. IEEE. 2010, pp. 253–256.

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Vol. 25. Curran Associates, Inc., 2012.

[22] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Oct. 2014, pp. 1724–1734.

[23] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. "Mastering the game of Go with deep neural networks and tree search". In: *Nature* 529.7587 (2016), pp. 484–489.

[24] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. "GPT-4 Technical Report". In: *arXiv preprint arXiv:2303.08774* (2023).

[25] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. "Algorithms for Hyper-Parameter Optimization". In: *Advances in Neural Information Processing Systems*. Vol. 24. Curran Associates, Inc., 2011.

[26] Alicia Curth and Mihaela van der Schaar. "In Search of Insights, Not Magic Bullets: Towards Demystification of the Model Selection Dilemma in Heterogeneous Treatment Effect Estimation". In: *arXiv preprint arXiv:2302.02923* (2023).

[27] Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. "Toward Causal Representation Learning". In: *Proceedings of the IEEE* 109.5 (2021), pp. 612–634.

[28] Peng Cui, Zheyan Shen, Sheng Li, Liuyi Yao, Yaliang Li, Zhixuan Chu, and Jing Gao. "Causal Inference Meets Machine Learning". In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD '20. Virtual Event, CA, USA: Association for Computing Machinery, 2020, pp. 3527–3528.

[29] Peng Cui and Susan Athey. "Stable Learning Establishes Some Common Ground between Causal Inference and Machine Learning". In: *Nature Machine Intelligence* 4.2 (Feb. 2022), pp. 110–115.

[30] Claudia Shi, David Blei, and Victor Veitch. "Adapting Neural Networks for the Estimation of Treatment Effects". In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019.

[31] Susan Athey, Julie Tibshirani, and Stefan Wager. "Generalized Random Forests". In: *The Annals of Statistics* 47.2 (Apr. 2019), pp. 1148–1178.

[32] Sören R. Künzel, Jasjeet S. Sekhon, Peter J. Bickel, and Bin Yu. "Meta-Learners for Estimating Heterogeneous Treatment Effects Using Machine Learning". In: *Proceedings of the National Academy of Sciences* 116.10 (Mar. 2019), pp. 4156–4165.

[33] Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins. "Double/Debiased Machine Learning for Treatment and Structural Parameters". In: *The Econometrics Journal* 21.1 (Feb. 2018), pp. C1–C68.

[34] Carlos Fernández-Loria and Foster Provost. "Causal Decision Making and Causal Effect Estimation Are Not the Same... and Why It Matters". In: *INFORMS Journal on Data Science* 1.1 (2022), pp. 4–16.

[35] Susan Athey and Stefan Wager. "Policy Learning With Observational Data". In: *Econometrica* 89.1 (2021), pp. 133–161.

[36] Xun Zheng, Chen Dan, Bryon Aragam, Pradeep Ravikumar, and Eric Xing. "Learning Sparse Nonparametric DAGs". In: *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*. Vol. 108. Proceedings of Machine Learning Research. PMLR, Aug. 2020, pp. 3414–3425.

[37] Diviyan Kalainathan, Olivier Goudet, Isabelle Guyon, David Lopez-Paz, and Michèle Sebag. "Structural Agnostic Modeling: Adversarial Learning of Causal Graphs". In: *Journal of Machine Learning Research* 23.219 (2022), pp. 1–62.

[38] Muhan Zhang, Shali Jiang, Zhicheng Cui, Roman Garnett, and Yixin Chen. "D-VAE: A Variational Autoencoder for Directed Acyclic Graphs". In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019.

[39] Frederick Eberhardt. "Introduction to the Foundations of Causal Discovery". In: *International Journal of Data Science and Analytics* 3.2 (Mar. 2017), pp. 81–91.

[40] Clark Glymour, Kun Zhang, and Peter Spirtes. "Review of Causal Discovery Methods Based on Graphical Models". In: *Frontiers in Genetics* 10 (2019).

[41] Jonas Peters, Dominik Janzing, and Bernhard Schlkopf. *Elements of Causal Inference: Foundations and Learning Algorithms*. The MIT Press, 2017.

[42] Isabelle Guyon, Alexander Statnikov, and Berna Bakir Batu, eds. *Cause Effect Pairs in Machine Learning*. The Springer Series on Challenges in Machine Learning. Springer International Publishing, 2019.

[43] Judea Pearl. *Causality*. Cambridge University Press, 2009.

[44] Judea Pearl, Madelyn Glymour, and Nicholas P. Jewell. *Causal Inference in Statistics: A Primer*. John Wiley & Sons, Mar. 2016.

[45] Thuc Duy Le, Tao Hoang, Jiuyong Li, Lin Liu, Huawen Liu, and Shu Hu. "A Fast PC Algorithm for High Dimensional Causal Discovery with Multi-Core PCs". In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 16.5 (Sept. 2019), pp. 1483–1495.

[46] David Maxwell Chickering. "Optimal Structure Identification With Greedy Search". In: *Journal of Machine Learning Research* 3.Nov (2002), pp. 507–554.

[47] Thomas S. Richardson. "Models of Feedback: Interpretation and Discovery". PhD Thesis. Carnegie-Mellon University, 1996.

[48] Christopher Meek. "Strong Completeness and Faithfulness in Bayesian Networks". In: *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*. UAI'95. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., Aug. 1995, pp. 411–418.

[49] Dan Geiger and Judea Pearl. "On the Logic of Causal Models". In: *Uncertainty in Artificial Intelligence*. Vol. 9. Machine Intelligence and Pattern Recognition. North-Holland, 1990, pp. 3–14.

[50] Shohei Shimizu, Patrik O. Hoyer, Aapo Hyvärinen, and Antti Kerminen. "A Linear Non-Gaussian Acyclic Model for Causal Discovery". In: *Journal of Machine Learning Research* 7.Oct (2006), pp. 2003–2030.

[51] Patrik Hoyer, Dominik Janzing, Joris M. Mooij, Jonas Peters, and Bernhard Schölkopf. "Nonlinear Causal Discovery with Additive Noise Models". In: *Advances in Neural Information Processing Systems*. Vol. 21. Curran Associates, Inc., 2008.

[52] Jonas Peters, Joris M. Mooij, Dominik Janzing, and Bernhard Schölkopf. "Causal Discovery with Continuous Additive Noise Models". In: *Journal of Machine Learning Research* 15.58 (2014), pp. 2009–2053.

[53] Jonas Peters and Peter Bühlmann. "Identifiability of Gaussian Structural Equation Models with Equal Error Variances". In: *Biometrika* 101.1 (Mar. 2014), pp. 219–228.

[54] Kun Zhang and Aapo Hyvärinen. "On the Identifiability of the Post-Nonlinear Causal Model". In: *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. UAI '09. Arlington, Virginia, USA: AUAI Press, June 2009, pp. 647–655.

[55] Peter Bühlmann, Jonas Peters, and Jan Ernest. "CAM: Causal Additive Models, High-Dimensional Order Search and Penalized Regression". In: *The Annals of Statistics* 42.6 (2014), pp. 2526–2556.

[56] Xun Zheng, Bryon Aragam, Pradeep K. Ravikumar, and Eric P. Xing. "DAGs with NO TEARS: Continuous Optimization for Structure Learning". In: *Advances in Neural Information Processing Systems*. Vol. 31. Curran Associates, Inc., 2018.

[57] Ignavier Ng, Shengyu Zhu, Zhitang Chen, and Zhuangyan Fang. "A Graph Autoencoder Approach to Causal Structure Learning". In: *arXiv preprint arXiv:1911.07420* (2019).

[58] Yue Yu, Jie Chen, Tian Gao, and Mo Yu. "DAG-GNN: DAG Structure Learning with Graph Neural Networks". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 7154–7163.

[59] Xiongren Chen. "Gradient-Based Causal Structure Learning with Normalizing Flow". In: *arXiv preprint arXiv:2010.03095* (2020).

[60] Sébastien Lachapelle, Philippe Brouillard, Tristan Deleu, and Simon Lacoste-Julien. "Gradient-Based Neural DAG Learning". In: *International Conference on Learning Representations*. Sept. 2019.

[61] Ignavier Ng, Shengyu Zhu, Zhuangyan Fang, Haoyang Li, Zhitang Chen, and Jun Wang. "Masked Gradient-Based Causal Structure Learning". In: *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM)*. SIAM. 2022, pp. 424–432.

[62] Eric Jang, Shixiang Gu, and Ben Poole. "Categorical Reparameterization with Gumbel-Softmax". In: *International Conference on Learning Representations*. 2017.

[63] Olivier Goudet, Diviyan Kalainathan, Philippe Caillou, Isabelle Guyon, David Lopez-Paz, and Michèle Sebag. "Causal Generative Neural Networks". In: *arXiv preprint arXiv:1711.08936* (2017).

[64] Erik P. Hoel, Larissa Albantakis, and Giulio Tononi. "Quantifying Causal Emergence Shows That Macro Can Beat Micro". In: *Proceedings of the National Academy of Sciences of the United States of America* 110.49 (Dec. 2013), pp. 19790–19795.

[65] Krzysztof Chalupka, Pietro Perona, and Frederick Eberhardt. "Visual Causal Feature Learning". In: *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*. UAI'15. Amsterdam, Netherlands: AUAI Press, July 2015, pp. 181–190.

[66] Krzysztof Chalupka, Frederick Eberhardt, and Pietro Perona. "Multi-Level Cause-Effect Systems". In: *Artificial Intelligence and Statistics*. May 2016, pp. 361–369.

[67] Diederik P. Kingma and Max Welling. "Auto-Encoding Variational Bayes". In: *arXiv preprint arXiv:1312.6114* (2013).

[68] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. "Stochastic Backpropagation and Approximate Inference in Deep Generative Models". In: *International Conference on Machine Learning*. PMLR, June 2014, pp. 1278–1286.

[69] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems*. Vol. 27. Curran Associates, Inc., 2014.

[70] Mengyue Yang, Furui Liu, Zhitang Chen, Xinwei Shen, Jianye Hao, and Jun Wang. "CausalVAE: Disentangled Representation Learning via Neural Structural Causal Models". In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 9588–9597.

[71] Xinwei Shen, Furui Liu, Hanze Dong, Qing Lian, Zhitang Chen, and Tong Zhang. "Weakly Supervised Disentangled Generative Causal Representation Learning". In: *Journal of Machine Learning Research* 23.241 (2022), pp. 1–55.

[72] Michael Park. "Moment-Matching Graph-Networks for Causal Inference". In: *arXiv preprint arXiv:2007.10507* (2020).

[73] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15.56 (2014), pp. 1929–1958.

[74] Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle. "MADE: Masked Autoencoder for Distribution Estimation". In: *International Conference on Machine Learning*. PMLR, June 2015, pp. 881–889.

[75] Marcus Kaiser and Maksim Sipos. "Unsuitability of NOTEARS for Causal Graph Discovery when Dealing with Dimensional Quantities". In: *Neural Process. Lett.* 54.3 (June 2022), pp. 1587–1595.

[76] Alexander Reisach, Christof Seiler, and Sebastian Weichwald. "Beware of the Simulated DAG! Causal Discovery Benchmarks May Be Easy to Game". In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, pp. 27772–27784.

[77] Ruocheng Guo, Lu Cheng, Jundong Li, P. Richard Hahn, and Huan Liu. "A Survey of Learning Causality with Data: Problems and Methods". In: *ACM Computing Surveys* 53.4 (July 2020), 75:1–75:37.

[78] Liuyi Yao, Zhixuan Chu, Sheng Li, Yaliang Li, Jing Gao, and Aidong Zhang. "A Survey on Causal Inference". In: *ACM Trans. Knowl. Discov. Data* 15.5 (May 2021).

[79] Guido W. Imbens. "Potential Outcome and Directed Acyclic Graph Approaches to Causality: Relevance for Empirical Practice in Economics". In: *Journal of Economic Literature* 58.4 (Dec. 2020), pp. 1129–79.

[80] Paul R. Rosenbaum and Donald B. Rubin. "The Central Role of the Propensity Score in Observational Studies for Causal Effects". In: *Biometrika* 70.1 (Apr. 1983), pp. 41–55.

[81] James M. Robins, Andrea Rotnitzky, and Lue Ping Zhao. "Estimation of Regression Coefficients When Some Regressors Are Not Always Observed". In: *Journal of the American Statistical Association* 89.427 (Sept. 1994), pp. 846–866.

[82] Mark J. van der Laan and Daniel Rubin. "Targeted Maximum Likelihood Learning". In: *The International Journal of Biostatistics* 2.1 (Dec. 2006).

[83] Kosuke Imai and Marc Ratkovic. "Covariate Balancing Propensity Score". In: *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* 76.1 (2014), pp. 243–263.

[84] Brian K. Lee, Justin Lessler, and Elizabeth A. Stuart. "Weight Trimming and Propensity Score Weighting". In: *PLOS ONE* 6.3 (Mar. 2011), e18174.

[85] Rom Gutman, Ehud Karavani, and Yishai Shimoni. "Propensity Score Models Are Better When Post-Calibrated". In: *arXiv preprint arXiv:2211.01221* (2022).

[86] Keith Battocchi, Eleanor Dillon, Maggie Hei, Greg Lewis, Paul Oka, Miruna Oprescu, and Vasilis Syrgkanis. *EconML: A Python Package for ML-Based Heterogeneous Treatment Effects Estimation*. https://github.com/py-why/EconML. Version 0.x. 2019.

[87] Xinkun Nie and Stefan Wager. "Quasi-Oracle Estimation of Heterogeneous Treatment Effects". In: *Biometrika* 108.2 (June 2021), pp. 299–319.

[88] Dylan J. Foster and Vasilis Syrgkanis. "Orthogonal Statistical Learning". In: *The Annals of Statistics* 51.3 (2023), pp. 879–908.

[89] Damian Machlanski, Spyridon Samothrakis, and Paul Clarke. "Undersmoothing Causal Estimators With Generative Trees". In: *IEEE Access* 12 (2024), pp. 38562–38574.

[90] Fredrik D. Johansson, Uri Shalit, and David Sontag. "Learning Representations for Counterfactual Inference". In: *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*. ICML'16. New York, NY, USA: JMLR.org, June 2016, pp. 3020–3029.

[91] Uri Shalit, Fredrik D. Johansson, and David Sontag. "Estimating Individual Treatment Effect: Generalization Bounds and Algorithms". In: *International Conference on Machine Learning*. PMLR, July 2017, pp. 3076–3085.

[92] 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*. Vol. 31. Curran Associates, Inc., 2018.

[93] Spyridon Samothrakis, Ana Matran-Fernandez, Umar Abdullahi, Michael Fairbank, and Maria Fasli. "Grokking-like Effects in Counterfactual Inference". In: *2022 International Joint Conference on Neural Networks (IJCNN)*. July 2022, pp. 1–8.

[94] Christos Louizos, Uri Shalit, Joris M. Mooij, David Sontag, Richard Zemel, and Max Welling. "Causal Effect Inference with Deep Latent-Variable Models". In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017.

[95] Jinsung Yoon, James Jordon, and Mihaela van der Schaar. "GANITE: Estimation of Individualized Treatment Effects Using Generative Adversarial Nets". In: *International Conference on Learning Representations*. Feb. 2018.

[96] Toon Vanderschueren, Jeroen Berrevoets, and Wouter Verbeke. "NOFLITE: Learning to Predict Individual Treatment Effect Distributions". In: *Transactions on Machine Learning Research* (2023).

[97] Ahmed M. Alaa, Zaid Ahmad, and Mark van der Laan. "Conformal Meta-learners for Predictive Inference of Individual Treatment Effects". In: *Advances in Neural Information Processing Systems*. Vol. 36. Curran Associates, Inc., 2023, pp. 47682–47703.

[98] Yingying Zhang, Chengchun Shi, and Shikai Luo. "Conformal Off-Policy Prediction". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2023, pp. 2751–2768.

[99] Miruna Oprescu, Jacob Dorn, Marah Ghoummaid, Andrew Jesson, Nathan Kallus, and Uri Shalit. "B-Learner: Quasi-Oracle Bounds on Heterogeneous Causal Effects Under Hidden Confounding". In: *Proceedings of the 40th International Conference on Machine Learning*. Vol. 202. Proceedings of Machine Learning Research. PMLR, July 2023, pp. 26599–26618.

[100] Toon Vanderschueren, Alicia Curth, Wouter Verbeke, and Mihaela Van Der Schaar. "Accounting For Informative Sampling When Learning to Forecast Treatment Outcomes Over Time". In: *Proceedings of the 40th International Conference on Machine Learning*. Vol. 202. Proceedings of Machine Learning Research. PMLR, July 2023, pp. 34855–34874.

[101] Kara E. Rudolph, Nicholas T. Williams, Caleb H. Miles, Joseph Antonelli, and Ivan Diaz. "All Models Are Wrong, but Which Are Useful? Comparing Parametric and Nonparametric Estimation of Causal Effects in Finite Samples". In: *Journal of Causal Inference* 11.1 (Jan. 2023).

[102] Paul Clarke and Annalivia Polselli. "Double Machine Learning for Static Panel Models with Fixed Effects". In: *arXiv preprint arXiv:2312.08174* (2023).

[103] Kevin Xia, Kai-Zhan Lee, Yoshua Bengio, and Elias Bareinboim. "The Causal-Neural Connection: Expressiveness, Learnability, and Inference". In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, pp. 10823–10836.

[104] Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. "Grokking: Generalization Beyond Overfitting on Small Algorithmic Datasets". In: *arXiv preprint arXiv:2201.02177* (2022).

[105] Susan Athey, Guido W. Imbens, Jonas Metzger, and Evan Munro. "Using Wasserstein Generative Adversarial Networks for the design of Monte Carlo simulations". In: *Journal of Econometrics* 240.2 (2024), p. 105076.

[106] Brady Neal, Chin-Wei Huang, and Sunand Raghupathi. "RealCause: Realistic Causal Inference Benchmarking". In: *arXiv preprint arXiv:2011.15007* (2020).

[107] Ioana Bica, James Jordon, and Mihaela van der Schaar. "Estimating the Effects of Continuous-valued Interventions using Generative Adversarial Networks". In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020, pp. 16434–16445.

[108] Alvaro Correia, Robert Peharz, and Cassio P de Campos. "Joints in Random Forests". In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020, pp. 11404–11415.

[109] Xiaofang Wang, Dan Kondratyuk, Eric Christiansen, Kris M. Kitani, Yair Alon, and Elad Eban. "Wisdom of Committees: An Overlooked Approach To Faster and More Accurate Models". In: *International Conference on Learning Representations*. 2022.

[110] Alexander Marx and Jilles Vreeken. "Identifiability of Cause and Effect Using Regularized Regression". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD '19. New York, NY, USA: Association for Computing Machinery, July 2019, pp. 852–861.

[111] Dominik Janzing. "Causal Regularization". In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019.

[112] Damian Machlanski. *Treatment Effect Estimation Benchmarks*. IEEE Dataport. 2023. URL: https://dx.doi.org/10.21227/0v4q-nn37.

[113] Jennifer L. Hill. "Bayesian Nonparametric Modeling for Causal Inference". In: *Journal of Computational and Graphical Statistics* 20.1 (Jan. 2011), pp. 217–240.

[114] Jeanne Brooks-Gunn, Fong-ruey Liaw, and Pamela Kato Klebanov. "Effects of Early Intervention on Cognitive Function of Low Birth Weight Preterm Infants". In: *The Journal of Pediatrics* 120.3 (Mar. 1992), pp. 350–359.

[115] Jeffrey A. Smith and Petra E. Todd. "Does Matching Overcome LaLonde's Critique of Nonexperimental Estimators?" In: *Journal of Econometrics* 125.1-2 (2005), pp. 305–353.

[116] Robert J. LaLonde. "Evaluating the Econometric Evaluations of Training Programs with Experimental Data". In: *The American Economic Review* 76.4 (1986), pp. 604–620.

[117] Rajeev H. Dehejia and Sadek Wahba. "Propensity Score-Matching Methods For Nonexperimental Causal Studies". In: *The Review of Economics and Statistics* 84.1 (2002), pp. 151–161.

[118] Douglas Almond, Kenneth Y. Chay, and David S. Lee. "The Costs of Low Birth Weight". In: *The Quarterly Journal of Economics* 120.3 (Aug. 2005), pp. 1031–1083.

[119] Ricardo Pio Monti, Ilyes Khemakhem, and Aapo Hyvarinen. "Autoregressive Flow-Based Causal Discovery and Inference". In: *arXiv preprint arXiv:2007.09390* (2020).

[120] Arshdeep Sekhon, Zhe Wang, and Yanjun Qi. "Relate and Predict: Structure-Aware Prediction with Jointly Optimized Neural DAG". In: *arXiv preprint arXiv:2103.02405* (2021).

[121] Jerzy Neyman. "On the Application of Probability Theory to Agricultural Experiments. Essay on Principles. Section 9." In: *Statistical Science* (1990), pp. 465–472.

[122] Miguel Hernan and James M. Robins. *Causal Inference: What If*. Boca Raton: Chapman & Hall/CRC, 2020.

[123] RL Piedmont. "Bias, statistical". In: *Encyclopedia of Quality of Life and Well Being Research.,. Dordrecht: Springer* (2014).

[124] Elias Bareinboim, Jin Tian, and Judea Pearl. "Recovering from Selection Bias in Causal and Statistical Inference". In: *Probabilistic and Causal Inference: The Works of Judea Pearl*. 1st ed. New York, NY, USA: Association for Computing Machinery, 2022, pp. 433–450.

[125] Maurice H Quenouille. "Notes on bias in estimation". In: *Biometrika* 43.3/4 (1956), pp. 353–360.

[126] James Bergstra and Yoshua Bengio. “Random Search for Hyper-Parameter Optimization”. In: *Journal of Machine Learning Research* 13.10 (2012), pp. 281–305.

[127] Oded Berger-Tal, Jonathan Nathan, Ehud Meron, and David Saltz. “The Exploration-Exploitation Dilemma: A Multidisciplinary Framework”. In: *PLOS ONE* 9.4 (Apr. 2014), pp. 1–8.

[128] Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. “Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization”. In: *Journal of Machine Learning Research* 18.185 (2018), pp. 1–52.

[129] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. “Neural Architecture Search: A Survey”. In: *Journal of Machine Learning Research* 20.55 (2019), pp. 1–21.

[130] Kenneth O. Stanley and Risto Miikkulainen. “Evolving Neural Networks through Augmenting Topologies”. In: *Evolutionary Computation* 10.2 (June 2002), pp. 99–127.

[131] Ron Kohavi. “A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection”. In: *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*. IJCAI’95. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., Aug. 1995, pp. 1137–1143.

[132] Philipp Probst, Anne-Laure Boulesteix, and Bernd Bischl. “Tunability: Importance of Hyperparameters of Machine Learning Algorithms”. In: *Journal of Machine Learning Research* 20.53 (2019), pp. 1–32.

[133] Tong Yu and Hong Zhu. “Hyper-Parameter Optimization: A Review of Algorithms and Applications”. In: *arXiv preprint arXiv:2003.05689* (2020).

[134] Xin He, Kaiyong Zhao, and Xiaowen Chu. “AutoML: A Survey of the State-of-the-Art”. In: *Knowledge-Based Systems* 212 (2021), p. 106622.

[135] Junfeng Wen, Chun-Nam Yu, and Russell Greiner. “Robust Learning under Uncertain Test Distributions: Relating Covariate Shift to Model Misspecification”. In: *International Conference on Machine Learning*. PMLR, 2014, pp. 631–639.

[136] Halbert White. “Consequences and Detection of Misspecified Nonlinear Regression Models”. In: *Journal of the American Statistical Association* 76.374 (1981), pp. 419–433.

[137] Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, and Whitney K. Newey. *Double Machine Learning for Treatment and Causal Parameters*. Tech. rep. CWP49/16. Centre for Microdata Methods and Practice, Institute for Fiscal Studies, Sept. 2016.

[138] Whitney Newey, Fushing Hsieh, and James Robins. *Undersmoothing and Bias Corrected Functional Estimation*. Working Paper 98-17. Massachusetts Institute of Technology (MIT), Department of Economics, Oct. 1998.

[139] Christian Hansen, Damian Kozbur, and Sanjog Misra. “Targeted Undersmoothing: Sensitivity Analysis for Sparse Estimators”. In: *Review of Economics and Statistics* 105.1 (2023), pp. 101–112.

[140] Randall Balestriero, Ishan Misra, and Yann LeCun. "A Data-Augmentation Is Worth A Thousand Samples: Analytical Moments And Sampling-Free Training". In: *Advances in Neural Information Processing Systems*. Vol. 35. Curran Associates, Inc., 2022, pp. 19631–19644.

[141] Luis Perez and Jason Wang. "The Effectiveness of Data Augmentation in Image Classification Using Deep Learning". In: *arXiv preprint arXiv:1712.04621* (2017).

[142] Yikun Hou and Miguel Navarro-Cia. "A Computationally-Inexpensive Strategy in CT Image Data Augmentation for Robust Deep Learning Classification in the Early Stages of an Outbreak". In: *Biomedical Physics & Engineering Express* 9.5 (2023), p. 055003.

[143] A Kiran and S Saravana Kumar. "A Comparative Analysis of GAN and VAE based Synthetic Data Generators for High Dimensional, Imbalanced Tabular data". In: *2023 2nd International Conference for Innovation in Technology (INOCON)*. IEEE. 2023, pp. 1–6.

[144] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. "SMOTE: Synthetic Minority Over-Sampling Technique". In: *Journal of Artificial Intelligence Research* 16.1 (June 2002), pp. 321–357.

[145] Haibo He, Yang Bai, Edwardo A. Garcia, and Shutao Li. "ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning". In: *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*. June 2008, pp. 1322–1328.

[146] György Kovács. "An empirical comparison and evaluation of minority oversampling techniques on a large number of imbalanced datasets". In: *Applied Soft Computing* 83 (2019), p. 105662.

[147] Tingting Pan, Junhong Zhao, Wei Wu, and Jie Yang. "Learning imbalanced datasets based on SMOTE and Gaussian distribution". In: *Information Sciences* 512 (2020), pp. 1214–1233.

[148] Maximilian Ilse, Jakub M. Tomczak, and Patrick Forré. "Selecting Data Augmentation for Simulating Interventions". In: *Proceedings of the 38th International Conference on Machine Learning*. Vol. 139. Proceedings of Machine Learning Research. PMLR, July 2021, pp. 4555–4562.

[149] Ananth Balashankar, Xuezhi Wang, Yao Qin, Ben Packer, Nithum Thain, Ed Chi, Jilin Chen, and Alex Beutel. "Improving Classifier Robustness through Active Generative Counterfactual Data Augmentation". In: *Findings of the Association for Computational Linguistics: EMNLP 2023*. 2023, pp. 127–139.

[150] Mohammed Temraz and Mark T. Keane. "Solving the Class Imbalance Problem Using a Counterfactual Method for Data Augmentation". In: *Machine Learning with Applications* 9 (2022), p. 100375.

[151] Silviu Pitis, Elliot Creager, Ajay Mandlekar, and Animesh Garg. "MoCoDA: Model-based Counterfactual Data Augmentation". In: *Advances in Neural Information Processing Systems*. Vol. 35. Curran Associates, Inc., 2022, pp. 18143–18156.

[152] Fiona Anting Tan, Devamanyu Hazarika, See Kiong Ng, Soujanya Poria, and Roger Zimmermann. “Causal Augmentation for Causal Sentence Classification”. In: *Proceedings of the First Workshop on Causal Inference and NLP*. 2021, pp. 1–20.

[153] Sindhu C. M. Gowda, Shalmali Joshi, Haoran Zhang, and Marzyeh Ghassemi. “Pulling Up by the Causal Bootstraps: Causal Data Augmentation for Pre-Training Debiasing”. In: *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. CIKM ’21. Virtual Event, Queensland, Australia: Association for Computing Machinery, 2021, pp. 606–616.

[154] Yaowei Hu, Yongkai Wu, Lu Zhang, and Xintao Wu. “A Generative Adversarial Framework for Bounding Confounded Causal Effects”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 13. 2021, pp. 12104–12112.

[155] Yunzhe Li, Kun Kuang, Bo Li, Peng Cui, Jianrong Tao, Hongxia Yang, and Fei Wu. “Continuous Treatment Effect Estimation via Generative Adversarial De-confounding”. In: *Proceedings of the 2020 KDD Workshop on Causal Discovery*. Vol. 127. Proceedings of Machine Learning Research. PMLR, 24 Aug 2020, pp. 4–22.

[156] Mouad El Bouchattaoui, Myriam Tami, Benoit Lepetit, and Paul-Henry Cournède. “Causal Dynamic Variational Autoencoder for Counterfactual Regression in Longitudinal Data”. In: *arXiv preprint arXiv:2310.10559* (2023).

[157] Steffen Bickel, Michael Bruckner, and Tobias Scheffer. “Discriminative Learning for Differing Training and Test Distributions”. In: *Proceedings of the 24th International Conference on Machine Learning*. ICML ’07. Corvalis, Oregon, USA: Association for Computing Machinery, 2007, pp. 81–88.

[158] Tiffany Tianhui Cai, Hongseok Namkoong, Steve Yadlowsky, et al. “Diagnosing Model Performance Under Distribution Shift”. In: *arXiv preprint arXiv:2303.02011* (2023).

[159] Haoran Zhang, Harvineet Singh, Marzyeh Ghassemi, and Shalmali Joshi. “"Why did the Model Fail?": Attributing Model Performance Changes to Distribution Shifts”. In: *Proceedings of the 40th International Conference on Machine Learning*. Vol. 202. Proceedings of Machine Learning Research. PMLR, July 2023, pp. 41550–41578.

[160] Jiashuo Liu, Zheyan Shen, Yue He, Xingxuan Zhang, Renzhe Xu, Han Yu, and Peng Cui. “Towards Out-Of-Distribution Generalization: A Survey”. In: *arXiv preprint arXiv:2108.13624* (2021).

[161] Pierre Geurts, Damien Ernst, and Louis Wehenkel. “Extremely Randomized Trees”. In: *Machine Learning* 63.1 (Apr. 2006), pp. 3–42.

[162] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. “CatBoost: Unbiased Boosting with Categorical Features”. In: *Advances in Neural Information Processing Systems*. Vol. 31. Curran Associates, Inc., 2018.

[163] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. “LightGBM: A Highly Efficient Gradient Boosting Decision Tree”. In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017.

[164] Damian Machlanski, Spyridon Samothrakis, and Paul Clarke. "Hyperparameter Tuning and Model Evaluation in Causal Effect Estimation". In: *arXiv preprint arXiv:2303.01412* (2023).

[165] James M. Robins and Sander Greenland. "The role of model selection in causal inference from nonexperimental data". In: *American Journal of Epidemiology* 123.3 (Mar. 1986), pp. 392–402.

[166] Min Qian and Susan A. Murphy. "Performance Guarantees for Individualized Treatment Rules". In: *The Annals of Statistics* 39.2 (Apr. 2011), pp. 1180–1210.

[167] Matthieu Doutreligne and Gaël Varoquaux. "How to select predictive models for causal inference?" In: *arXiv preprint arXiv:2302.00370* (2023).

[168] Alejandro Schuler, Michael Baiocchi, Robert Tibshirani, and Nigam Shah. "A Comparison of Methods for Model Selection When Estimating Individual Treatment Effects". In: *arXiv preprint arXiv:1804.05146* (2018).

[169] Divyat Mahajan, Ioannis Mitliagkas, Brady Neal, and Vasilis Syrgkanis. "Empirical Analysis of Model Selection for Heterogeneous Causal Effect Estimation". In: *The Twelfth International Conference on Learning Representations*. 2024.

[170] Craig A. Rolling and Yuhong Yang. "Model Selection for Estimating Treatment Effects". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 76.4 (2014), pp. 749–769.

[171] Ahmed Alaa and Mihaela van der Schaar. "Validating Causal Inference Models via Influence Functions". In: *Proceedings of the 36th International Conference on Machine Learning*. PMLR, May 2019, pp. 191–201.

[172] Yuta Saito and Shota Yasui. "Counterfactual Cross-Validation: Stable Model Selection Procedure for Causal Inference Models". In: *Proceedings of the 37th International Conference on Machine Learning*. PMLR, Nov. 2020, pp. 8398–8407.

[173] Alejandro Schuler, Ken Jung, Robert Tibshirani, Trevor Hastie, and Nigam Shah. "Synth-Validation: Selecting the Best Causal Inference Method for a Given Dataset". In: *arXiv preprint arXiv:1711.00083* (2017).

[174] Keisuke Hirano, Guido W. Imbens, and Geert Ridder. "Efficient Estimation of Average Treatment Effects Using the Estimated Propensity Score". In: *Econometrica* 71.4 (2003), pp. 1161–1189.

[175] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. "Scikit-Learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12.85 (2011), pp. 2825–2830.

[176] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015.

[177] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations, ICLR*. 2015.

[178] Yishai Shimoni, Ehud Karavani, Sivan Ravid, Peter Bak, Tan Hung Ng, Sharon Hensley Alford, Denise Meade, and Yaara Goldschmidt. "An Evaluation Toolkit to Guide Model Selection and Cohort Definition in Causal Inference". In: *arXiv preprint arXiv:1906.00442* (2019).

[179] Felix L. Rios, Giusi Moffa, and Jack Kuipers. "Benchpress: A Scalable and Versatile Workflow for Benchmarking Structure Learning Algorithms". In: *arXiv preprint arXiv:2107.03863* (May 2021).

[180] Tom Le Paine, Cosmin Paduraru, Andrea Michi, Caglar Gulcehre, Konrad Zolna, Alexander Novikov, Ziyu Wang, and Nando de Freitas. "Hyperparameter Selection for Offline Reinforcement Learning". In: *arXiv preprint arXiv:2007.09055* (2020).

[181] Baohe Zhang, Raghu Rajan, Luis Pineda, Nathan Lambert, André Biedenkapp, Kurtland Chua, Frank Hutter, and Roberto Calandra. "On the Importance of Hyperparameter Optimization for Model-based Reinforcement Learning". In: *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*. PMLR, Mar. 2021, pp. 4015–4023.

[182] Jan Tönshoff, Martin Ritzert, Eran Rosenbluth, and Martin Grohe. "Where Did the Gap Go? Reassessing the Long-Range Graph Benchmark". In: *The Second Learning on Graphs Conference*. 2023.

[183] Vineet K. Raghu, Allen Poon, and Panayiotis V. Benos. "Evaluation of Causal Structure Learning Methods on Mixed Data Types". In: *Proceedings of 2018 ACM SIGKDD Workshop on Causal Discovery*. PMLR, Aug. 2018, pp. 48–65.

[184] Ruibo Tu, Kun Zhang, Bo Bertilson, Hedvig Kjellstrom, and Cheng Zhang. "Neuropathic Pain Diagnosis Simulator for Causal Discovery Algorithm Evaluation". In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019.

[185] Eric V. Strobl. "Automated Hyperparameter Selection for the PC Algorithm". In: *Pattern Recognition Letters* 151 (Nov. 2021), pp. 288–293.

[186] Konstantina Biza, Ioannis Tsamardinos, and Sofia Triantafillou. "Out-of-Sample Tuning for Causal Discovery". In: *IEEE Transactions on Neural Networks and Learning Systems* (2022), pp. 1–11.

[187] Charles K Assaad, Emilie Devijver, and Eric Gaussier. "Survey and Evaluation of Causal Discovery Methods for Time Series". In: *Journal of Artificial Intelligence Research* 73 (2022), pp. 767–819.

[188] Tim Van den Bulcke, Koenraad Van Leemput, Bart Naudts, Piet van Remortel, Hongwu Ma, Alain Verschoren, Bart De Moor, and Kathleen Marchal. "SynTReN: A Generator of Synthetic Gene Expression Data for Design and Analysis of Structure Learning Algorithms". In: *BMC Bioinformatics* 7 (Jan. 2006), p. 43.

[189] Daniel Grünbaum, Maike L. Stern, and Elmar W. Lang. "Quantitative Probing: Validating Causal Models with Quantitative Domain Knowledge". In: *Journal of Causal Inference* 11.1 (Jan. 2023).

[190] Lei Shen, Wangshu Liu, Xiang Chen, Qing Gu, and Xuejun Liu. "Improving Machine Learning-Based Code Smell Detection via Hyper-Parameter Optimization". In: *2020 27th Asia-Pacific Software Engineering Conference (APSEC)*. Dec. 2020, pp. 276–285.

[191] Agoston E. Eiben and Selmar K. Smit. "Parameter Tuning for Configuring and Analyzing Evolutionary Algorithms". In: *Swarm and Evolutionary Computation* 1.1 (Mar. 2011), pp. 19–31.

[192] Mingyu Huang and Ke Li. "On the Hyperparameter Landscapes of Machine Learning Algorithms". In: *arXiv preprint arXiv:2311.14014* (2023).

[193] Jascha Sohl-Dickstein. "The boundary of neural network trainability is fractal". In: *arXiv preprint arXiv:2402.06184* (2024).

[194] Han Liu, Kathryn Roeder, and Larry Wasserman. "Stability Approach to Regularization Selection (StARS) for High Dimensional Graphical Models". In: *Advances in Neural Information Processing Systems*. Vol. 23. Curran Associates, Inc., 2010.

[195] Wei Sun, Junhui Wang, and Yixin Fang. "Consistent Selection of Tuning Parameters via Variable Selection Stability". In: *Journal of Machine Learning Research* 14.107 (2013), pp. 3419–3440.

[196] Kiattikun Chobtham and Anthony C. Constantinou. "Tuning structure learning algorithms with out-of-sample and resampling strategies". In: *arXiv preprint arXiv:2306.13932* (June 2023).

[197] P. Erdös and A. Rényi. "On Random Graphs". In: *Publ. math. debrecen* 6.290-297 (1959), p. 18.

[198] Albert-László Barabási and Réka Albert. "Emergence of Scaling in Random Networks". In: *Science* 286.5439 (1999), pp. 509–512.

[199] Karen Sachs, Omar Perez, Dana Pe'er, Douglas A. Lauffenburger, and Garry P. Nolan. "Causal Protein-Signaling Networks Derived from Multiparameter Single-Cell Data". In: *Science* 308.5721 (2005), pp. 523–529.

[200] Peter Spirtes and Clark Glymour. "An Algorithm for Fast Recovery of Sparse Causal Graphs". In: *Social Science Computer Review* 9.1 (Apr. 1991), pp. 62–72.

[201] Peter Spirtes, Clark Glymour, and Richard Scheines. "Discovery Algorithms for Causally Sufficient Structures". In: *Causation, Prediction, and Search*. New York, NY: Springer New York, 1993, pp. 103–162.

[202] 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". In: *International Journal of Data Science and Analytics* 3.2 (Mar. 2017), pp. 121–129.

[203] Aapo Hyvarinen. "Fast and Robust Fixed-Point Algorithms for Independent Component Analysis". In: *IEEE Transactions on Neural Networks* 10.3 (May 1999), pp. 626–634.

[204] Kayvan Sadeghi. "Marginalization and Conditioning for LWF Chain Graphs". In: *The Annals of Statistics* 44.4 (2016), pp. 1792–1816.

[205] David H. Wolpert. "The Lack of A Priori Distinctions Between Learning Algorithms". In: *Neural Computation* 8.7 (Oct. 1996), pp. 1341–1390.

[206] David H. Wolpert and William G. Macready. "No Free Lunch Theorems for Optimization". In: *IEEE Transactions on Evolutionary Computation* 1.1 (1997), pp. 67–82.

[207] Leo Grinsztajn, Edouard Oyallon, and Gael Varoquaux. "Why Do Tree-Based Models Still Outperform Deep Learning on Typical Tabular Data?" In: *Advances in Neural Information Processing Systems*. Vol. 35. Curran Associates, Inc., 2022, pp. 507–520.

[208] Duncan McElfresh, Sujay Khandagale, Jonathan Valverde, Vishak Prasad C, Ganesh Ramakrishnan, Micah Goldblum, and Colin White. "When Do Neural Nets Outperform Boosted Trees on Tabular Data?" In: *Advances in Neural Information Processing Systems*. Vol. 36. Curran Associates, Inc., 2023, pp. 76336–76369.

[209] Piersilvio De Bartolomeis, Javier Abad, Konstantin Donhauser, and Fanny Yang. "Hidden yet quantifiable: A lower bound for confounding strength using randomized trials". In: *arXiv preprint arXiv:2312.03871* (2023).

[210] Yonghan Jung, Ivan Diaz, Jin Tian, and Elias Bareinboim. "Estimating Causal Effects Identifiable from a Combination of Observations and Experiments". In: *Advances in Neural Information Processing Systems*. Vol. 36. Curran Associates, Inc., 2023, pp. 46446–46490.

[211] Philippe Brouillard, Sébastien Lachapelle, Alexandre Lacoste, Simon Lacoste-Julien, and Alexandre Drouin. "Differentiable Causal Discovery from Interventional Data". In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020, pp. 21865–21877.

[212] Panagiotis Tigas, Yashas Annadani, Andrew Jesson, Bernhard Schölkopf, Yarin Gal, and Stefan Bauer. "Interventions, Where and How? Experimental Design for Causal Models at Scale". In: *Advances in Neural Information Processing Systems*. Vol. 35. Curran Associates, Inc., 2022, pp. 24130–24143.

[213] Konstantina Biza, Antonios Ntroumpogiannis, Sofia Triantafillou, and Ioannis Tsamardinos. "Towards Automated Causal Discovery: a case study on 5G telecommunication data". In: *arXiv preprint arXiv:2402.14481* (2024).

[214] Timo Debono, Julian Teichgraber, Timo Flesch, Edward Zhang, Guy Durant, Wen Hao Kho, Mark Harley, and Egor Kraev. *CausalTune: A Python package for Automated Causal Inference model estimation and selection.* https://github.com/py-why/causaltune. 2022.

[215] Huy Nguyen, Prince Grover, and Devashish Khatwani. "OpportunityFinder: A Framework for Automated Causal Inference". In: *Causal Inference and Machine Learning in Practice: Use cases for Product, Brand, Policy and Beyond* (2023).

[216] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. "Conservative Q-Learning for Offline Reinforcement Learning". In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020, pp. 1179–1191.

[217] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. "Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems". In: *arXiv preprint arXiv:2005.01643* (2020).

[218] Rishabh Agarwal, Dale Schuurmans, and Mohammad Norouzi. "An Optimistic Perspective on Offline Reinforcement Learning". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 104–114.

[219] Ioana Bica, Ahmed M. Alaa, James Jordon, and Mihaela van der Schaar. "Estimating Counterfactual Treatment Outcomes over Time Through Adversarially Balanced Representations". In: *International Conference on Learning Representations*. 2020.

[220] Ishita Dasgupta, Jane Wang, Silvia Chiappa, Jovana Mitrovic, Pedro Ortega, David Raposo, Edward Hughes, Peter Battaglia, Matthew Botvinick, and Zeb Kurth-Nelson. "Causal Reasoning from Meta-reinforcement Learning". In: *arXiv preprint arXiv:1901.08162* (2019).

[221] Elias Bareinboim and Judea Pearl. "Meta-Transportability of Causal Effects: A Formal Approach". In: *Artificial Intelligence and Statistics*. PMLR, Apr. 2013, pp. 135–143.

[222] Sina Akbari, Ehsan Mokhtarian, AmirEmad Ghassami, and Negar Kiyavash. "Recursive Causal Structure Learning in the Presence of Latent Variables and Selection Bias". In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, pp. 10119–10130.

[223] Théo Bourdais, Pau Batlle, Xianjin Yang, Ricardo Baptista, Nicolas Rouquette, and Houman Owhadi. "Computational Hypergraph Discovery, a Gaussian Process framework for connecting the dots". In: *arXiv preprint arXiv:2311.17007* (2023).

[224] Keli Zhang, Shengyu Zhu, Marcus Kalander, Ignavier Ng, Junjian Ye, Zhitang Chen, and Lujia Pan. "gCastle: A Python Toolbox for Causal Discovery". In: *arXiv preprint arXiv:2111.15155* (2021).

[225] Diviyan Kalainathan, Olivier Goudet, and Ritik Dutta. "Causal Discovery Toolbox: Uncovering Causal Relationships in Python". In: *Journal of Machine Learning Research* 21.37 (2020), pp. 1–5.

[226] Joseph D Ramsey, Kun Zhang, Madelyn Glymour, Ruben Sanchez Romero, Biwei Huang, Imme Ebert-Uphoff, Savini Samarasinghe, Elizabeth A Barnes, and Clark Glymour. "TETRAD—A Toolbox for Causal Discovery". In: *8th International Workshop on Climate Informatics*. 2018, p. 29.

[227] Markus Kalisch, Martin Mächler, Diego Colombo, Marloes H. Maathuis, and Peter Bühlmann. “Causal Inference Using Graphical Models with the R Package pcalg”. In: *Journal of Statistical Software* 47.11 (2012), pp. 1–26.