Improving Data-driven Heterogeneous Treatment Effect Estimation Under Structure Uncertainty

Christopher Tran
University of Illinois at Chicago
Chicago, IL, USA
ctran29@uic.edu

Elena Zheleva
University of Illinois at Chicago
Chicago, IL, USA
ezheleva@uic.edu

ABSTRACT

Estimating how a treatment affects units individually, known as heterogeneous treatment effect (HTE) estimation, is an essential part of decision-making and policy implementation. The accumulation of large amounts of data in many domains, such as healthcare and e-commerce, has led to increased interest in developing data-driven algorithms for estimating heterogeneous effects from observational and experimental data. However, these methods often make strong assumptions about the observed features and ignore the underlying causal model structure, which can lead to biased HTE estimation. At the same time, accounting for the causal structure of real-world data is rarely trivial since the causal mechanisms that gave rise to the data are typically unknown. To address this problem, we develop a feature selection method that considers each feature's value for HTE estimation and learns the relevant parts of the causal structure from data. We provide strong empirical evidence that our method improves existing data-driven HTE estimation methods under arbitrary underlying causal structures. Our results on synthetic, semi-synthetic, and real-world datasets show that our feature selection algorithm leads to lower HTE estimation error.

CCS CONCEPTS

• **Computing methodologies → Causal reasoning and diagnostics**.

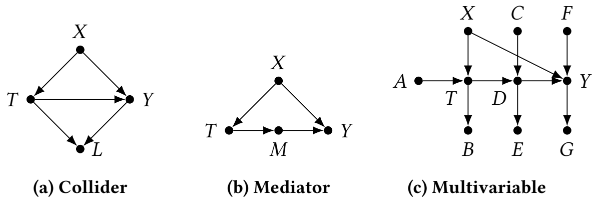
KEYWORDS

heterogeneous treatment effects, causal feature selection, structural causal models

ACM Reference Format:
Christopher Tran and Elena Zheleva. 2022. Improving Data-driven Heterogeneous Treatment Effect Estimation Under Structure Uncertainty. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22), August 14–18, 2022, Washington, DC, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3534678.3539444

1 INTRODUCTION

Estimating heterogeneous treatment effects (HTEs) is essential in many fields. HTE estimation aims to find subpopulations whose causal effects differ from the effects of the population as a whole. For example, if the treatment is a drug, some individuals may have adverse reactions, and some individuals may benefit from treatment [34]. Similarly, public policy may affect different sociodemographic groups differently [13]. HTE analysis allows the discovery of these different subgroups of the population.



(a) Collider (b) Mediator (c) Multivariable

Figure 1: Causal models where data-driven HTE estimation methods may perform poorly. Some variables (e.g., L, M, and E) are not valid adjustment variables for estimating the effect of T on Y.

The accumulation of large amounts of data in many domains has stimulated the increased development of algorithms tailored for automated *data-driven* HTE estimation, relying on supervised machine learning [3, 4, 13, 18–20, 34, 40]. The goal of data-driven HTE estimators is to estimate the conditional average treatment effect using the features available in the data.

While it is appropriate to use all possible features for general supervised learning problems, unbiased estimation of *causal effects* requires knowledge of the causal mechanisms underlying the data and selecting an appropriate subset of variables from the data [25]. One way to select this subset is by modeling the underlying causal mechanisms through structural causal models (SCMs) [25] and selecting an *adjustment set* — a set of variables that are necessary for unbiased estimation of a causal effect of interest [16, 25, 28]. However, the causal model structure is rarely known for real-world data, and data-driven methods do not take it into consideration. Our paper addresses this problem and improves data-driven HTE estimation methods through causal feature selection under such structure uncertainty.

A common assumption used by data-driven HTE estimators is the assumption of *strong ignorability* which states that the treatment assignment is independent of the potential outcomes given the variables in the data [29, 34]. This untestable assumption implies that all potential confounders, variables that can influence both the treatment and the outcome, are observed in the data. In turn, accounting for them would remove spurious associations and allow for causal effect identification. However, hidden confounding is not the only problem leading to biased HTE estimation. Even if

KDD '22, August 14–18, 2022, Washington, DC, USA

ACM ISBN 978-1-4503-9385-0/22/08…$15.00
https://doi.org/10.1145/3534678.3539444

all potential confounders are observed in the data, not all variables should be used for adjustment. Another important and sometimes implicit assumption is that all variables in the data are *pre-treatment* variables, i.e., the variables that were measured before and thus were unaffected by the treatment. When the temporal order of observations is unknown, many practitioners would simply use all variables in the data [22]. To illustrate the problem of using all variables in data-driven methods, consider the SCMs in Figure 1 where X is a confounding variable. According to SCM theory, we need to adjust for X when estimating causal effects, but should not include variables that will bias effect estimation, such as descendants of the treatment or instrumental variables (e.g., L, M, A, B, D, E, G).

To address these deficiencies of data-driven methods, we define the problem of causal feature selection for accurate data-driven HTE estimation under structure uncertainty. One vanilla solution to this problem is to first learn the causal structure from the data [16, 24, 38] and then use an existing identification algorithm to find the adjustment set [16, 28, 35, 41]. However, structure learning is computationally expensive when the space of feature is large, and it may not work well when there is uncertainty in the learned structure [8, 15, 43]. Instead, we develop a feature selection method that reduces this complexity by considering the value of each feature for improving HTE estimation and learning only the relevant parts of the causal structure from data. We provide strong empirical evidence that our method improves existing data-driven HTE estimation methods under arbitrary underlying causal structures. We evaluate the benefits of our feature selection algorithm in reducing HTE estimation error through synthetic, semi-synthetic, and real-world dataset experiments.

2 RELATED WORK

We provide a brief overview of related work on estimating heterogeneous treatment effects and structure learning.

Data-driven methods for HTE estimation. Many methods have been developed for estimating heterogeneous treatment effects (HTEs) from data. Several methods rely on recursive partitioning using tree-based methods, which naturally partition the population into subgroups that may contain heterogeneous effects [3, 4, 39, 40, 48]. Besides tree-based methods, methods have been developed that use machine learning to predict causal effects, such as meta-learners [19, 20] and neural networks [18, 33, 34]. However, these methods do not consider the structure of the underlying causal mechanisms that generate the data, introducing bias in estimation.

Causal structure learning. Many causal structure learning algorithms have been developed for recovering SCMs from data [15, 43]. Heinze-Deml et al. provide a recent survey on causal structure learning algorithms and evaluate their performance in various settings [15]. However, learning the full causal structure from data can be computationally expensive for high-dimensional datasets [8, 43]. Related to causal structure learning is local structure learning [2]. Aliferis et al. presented an algorithmic framework for learning local causal structure around target variables, the parent-children set (*PC*) and the Markov Blanket (*MB*). Yu et al. survey and evaluate several local causal structure learning methods that learn both *PC* and *MB* of target variables [47]. They study local structure learning under the context of causality-based feature selection with the goal of improving supervised learning. In contrast, we study the problem of causal feature selection. Local structure learning has been studied in the context of average treatment effect (ATE) estimation, but not HTE estimation [7]. This work also does not consider excluding mediators or descendants, which we do in our work.

3 BACKGROUND

We provide an introduction to heterogeneous treatment effects under the potential outcomes framework [30] and structural causal models [25].

3.1 Heterogeneous treatment effects

Let $(\mathbf{X}_i, Y_i(0), Y_i(1), T_i) \sim \mathcal{P}$ be a distribution with N units which are independently and identically distributed (i.i.d.). For any unit i, define $\mathbf{X}_i \in \mathbb{R}^d$ be a d-dimensional feature vector, $T_i \in \{0, 1\}$ to be the treatment assignment indicator, $Y_i(0)$ to be the potential outcome when i is not treated, and $Y_i(1)$ to be the potential outcome when i is treated. The individual treatment effect (ITE) is the difference in potential outcomes: $\tau_i = Y_i(1) - Y_i(0)$. However, both potential outcomes $Y_i(1)$ and $Y_i(0)$ cannot be observed at the same time and observed data $\mathcal{D} = (\mathbf{X}_i, Y_i, T_i)$ contains only $Y_i = Y_i(T_i)$. This is known as the fundamental problem of causal inference. Since τ_i can never be measured directly, its estimate is expressed through the conditional average treatment effect (CATE):

$$\hat{\tau}(\mathbf{x}) = E[Y(1) - Y(0) \mid \mathbf{X}_i]. \tag{1}$$

The average treatment effect (ATE) of the population is: ATE $= E[Y(1) - Y(0)]$. If CATE and ATE are not the same, then $\mathbf{X}_i$ induce heterogeneous treatment effects (HTE).

Given an observed dataset $\mathcal{D}$, the goal of an *HTE estimator* is to estimate CATE, $\hat{\tau}$, using the available features $\mathbf{X}$ in the data. HTE estimators estimate CATE in a variety of ways. Some HTE estimators rely on recursive partitioning using tree-based methods [3, 4, 40]. Other HTE estimators use predictions from machine learning methods to train a causal effect estimator [19, 20] or to predict counterfactual outcomes for estimation of effect [18, 34]. A common assumption used by data-driven HTE estimators is the assumption of *strong ignorability* which states that the treatment assignment is independent of the potential outcomes given a set of variables $\mathbf{X}$ [29, 34]: $Y_1, Y_0 \perp T \mid \mathbf{X}$, and $0 < p(T = 1 \mid \mathbf{X}) < 1$ for all $\mathbf{X} = \mathbf{x}$. One of the implications of this assumption is that there is no *hidden confounding*. However, hidden confounding is not the only problem that can lead to biased HTE estimation. In practice, it is often assumed that all observed variables in the data are the set of variables $\mathbf{X}$ that meets strong ignorability. This is problematic because some of the variables may make T and Y dependent or introduce new spurious dependencies [25]. Some variables, such as mediators, should not be included in the estimation. Moreover, when the temporal order of variables is unknown, practitioners may consider all features in the data as pre-treatment variables [22]. Selecting the right variables is the goal of our work.

3.2 Structural Causal Models

A structural causal model (SCM), M, consists of two sets of variables, $\mathbf{U}$ and $\mathbf{V}$, and a set of structural equations, $\mathbf{F}$, describing how values are assigned to each *endogenous* variable $V_i \in \mathbf{V}$ based on the values of $\mathbf{v}$ and $\mathbf{u}$ [25]: $v_i = f_i(\mathbf{v}, \mathbf{u})$. A causal graph G is a directed acyclic graph that captures the causal relationships among the variables. The variables in $\mathbf{U}$ are considered *exogenous.* It is assumed that every endogenous variable has a parent in $\mathbf{U}$. The causal graph may omit representing $\mathbf{U}$ explicitly. We denote the parents, children, ancestors, and descendants of X in the graph as: $pa(X), ch(X), an(X), de(X)$. Every node is a descendant and ancestor of itself.

When the SCM underlying the dataset is known, its graph G supports the use of graphical criteria for variable selection for unbiased causal effect estimation, known as adjustment set selection [16, 28, 35, 37, 41]. One such criterion is the backdoor criterion [25], which aims to find a set that blocks the paths from treatment to the outcome, starting with an arrow into treatment. For example, in Figure 1a, there is one backdoor path, namely, $T \leftarrow X \rightarrow Y$, and so the set $\{X\}$ sufficiently satisfies the backdoor criterion. Several adjustment criteria have been developed, such as the adjustment criterion [35, 41], generalized back-door criterion [21], generalized adjustment criterion [28], and optimal adjustment set (O-set) [16]. These adjustment sets defined by the various adjustment criteria can be found through the ID algorithm [36], the framework by Van der Zander et al. [41] or the pruning procedure by Henckel et al. [16]. Once an adjustment set is found, they can be used with an HTE estimator to estimate heterogeneous effects.

When the SCM for a dataset is unknown, and therefore there is *structure uncertainty* of the causal mechanism, the adjustment criteria cannot be applied. An alternative approach is to use causal structure learning [15] to learn a causal graph from the data and then apply an adjustment criterion for feature selection. However, learning the full structure is computationally expensive for high-dimensional datasets [8, 15, 43]. Instead, we propose reducing the feature space by assessing the fit of each variable for HTE estimation and learning the local structure, which we describe in Section 5.

3.2.1 HTEs in SCMs. In SCMs, heterogeneous effects are represented by an *interaction* or *effect modification* through ancestors of Y [25]. For example, suppose Y is generated by the following structural equation corresponding to the SCM in Figure 1c:

$$Y = \alpha \cdot T + \beta \cdot X + \gamma \cdot T \cdot f(X) + U_Y. \tag{2}$$

Here, there is a fixed effect α of the treatment T and a heterogeneous effect γ due to the interaction between T and X. Interaction can occur with any parent of Y (e.g., X, F in Figure 1c) or any parent of a mediator (e.g., C in Figure 1c) [42].

3.2.2 C-specific effects. Shpitser and Pearl provide a nonparametric estimand for HTEs with respect to some feature assignment $C = c$, called c-specific effects [26, 35]. One issue with estimating c-specific effects is that it requires knowledge of the SCM and which variables, C, exhibit heterogeneous effects. When the SCM is known, the O-set [16] can be used on the learned structure since it implicitly includes all potential heterogeneity-inducing variables, while some identification algorithms may miss them. When the SCM is unknown, the identifiability of c-specific effects cannot be established. Moreover, when the set of heterogeneity-inducing variables $\mathbf{C}$ is unknown, then all combinations of features and feature values would need to be checked for heterogeneity in effect, which is exponential in the number of valid features. Circumventing this exponential search is one of the main reasons why data-driven methods are popular in practice.

4 PROBLEM SETUP

Our goal in this work is to augment data-driven methods to accurately predict heterogeneous effects when both the SCM and the variables that explain heterogeneity are unknown. More specifically, the goal is to find a subset $\mathbf{X}^{(\ell)}$, to which we refer as the *causal feature set*, that improves data-driven estimation. In general, the goal in HTE estimation is to obtain accurate estimation of CATE:

Problem 1. *(Heterogeneous treatment effect (HTE) estimation) Given a dataset $\mathcal{D} = (\mathbf{X}_i, Y_i, T_i)$ of N instances with some true ITE τ, estimate CATE $\hat{\tau}(\mathbf{X})$ such that the mean squared error is minimized:*

$$\frac{1}{N}\sum_{i=1}^{N}(\tau_i - \hat{\tau}(\mathbf{X}_i))^2. \tag{3}$$

Since the true ITE is unknown in real-world scenarios, several heuristic HTE evaluation metrics have been developed for evaluating the performance of HTE estimators [31–33]. Instead of minimizing the true estimator error, we can instead minimize one of these metrics, such as the τ-risk [23, 32].

A *causal feature set* is the feature set that minimizes the error in HTE estimation.

Problem 2. *(Causal feature selection (CFS) for HTE estimation) Given a dataset $\mathcal{D} = (\mathbf{X}_i, Y_i, T_i)$ and a data-driven estimator $\hat{\tau}(\mathbf{X})$, find a* causal feature set *$\mathbf{X}_i^{(\ell)} \subseteq \mathbf{X}_i$ such that $\hat{\tau}(\mathbf{X}^{(\ell)})$ results in the lowest MSE:*

$$\underset{\mathbf{X}^{(\ell)}}{\arg\min}\ \frac{1}{N}\sum_{i=1}^{N}\left(\tau_i - \hat{\tau}\left(\mathbf{X}_i^{(\ell)}\right)\right)^2. \tag{4}$$

A data-driven estimator typically estimates $\hat{\tau}(\mathbf{X})$ using all available data. Our goal in this work is to select a causal feature set $\mathbf{X}^{(\ell)}$ such that the error of the estimator is minimized.

In this work, we assume that it is unknown a priori whether all observed features in the data are pre-treatment variables and whether all of them meet the strong ignorability assumption. Instead, we use the following more general assumptions:

1. Causal sufficiency: there are no latent variables.
2. Causal faithfulness: if X_A, X_B are conditionally independent given $\mathbf{X}_\mathbf{Z}$, then A, B are d-separated in the causal graph by $\mathbf{Z}$.
3. Acyclicity: the underlying causal graphs are directed acyclic graphs (DAGs).
4. There is no selection bias.

Understanding how our methods need to be adjusted when these assumptions do not hold is left for future work.

5 CAUSAL FEATURE SELECTION FOR HTE ESTIMATION

When the SCM that generated the data is unknown, a vanilla solution for finding an adjustment set for HTE estimation is to learn the causal structure and then find the optimal adjustment set (O-set) [16]. However, structure learning is computationally expensive in high-dimensional data, both in theory [8] and in practice [15, 43].

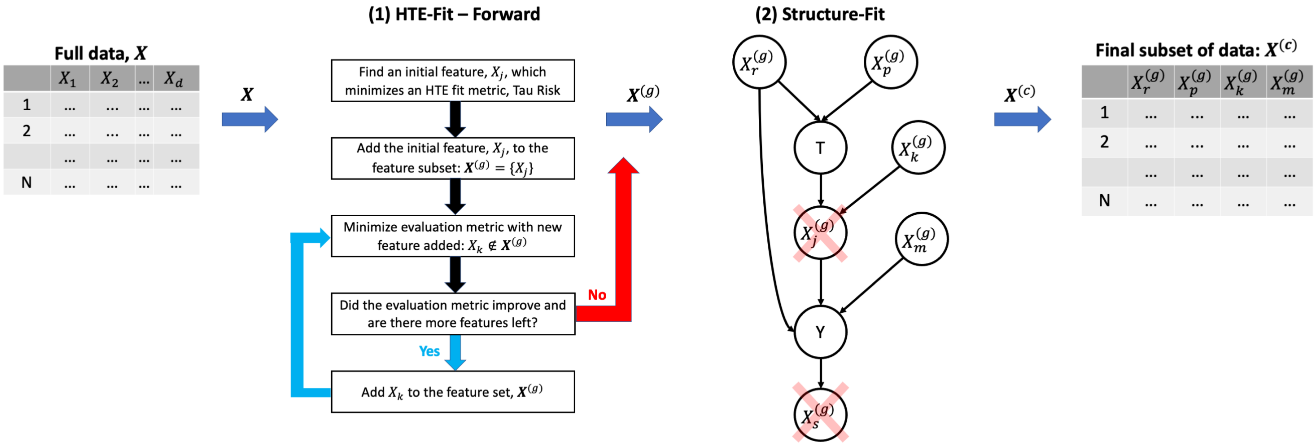

Full data, X
X_1 X_2 ... X_d
1 ... ...
2 ... ...
... ...
N ... ...
X
(1) HTE-Fit – Forward
Find an initial feature, X_j, which minimizes an HTE fit metric, Tau Risk
Add the initial feature, X_j, to the feature subset: $X^{(g)} = \{X_j\}$
Minimize evaluation metric with new feature added: $X_k \notin X^{(g)}$
Did the evaluation metric improve and are there more features left?
No
Yes
Add X_k to the feature set, $X^{(g)}$
$X^{(g)}$
(2) Structure-Fit
$X_r^{(g)}$
$X_p^{(g)}$
T
$X_k^{(g)}$
$X_j^{(g)}$
$X_m^{(g)}$
Y
$X_s^{(g)}$
$X^{(c)}$
Final subset of data: $X^{(c)}$
$X_r^{(g)}$ $X_p^{(g)}$ $X_k^{(g)}$ $X_m^{(g)}$
1 ... ...
2 ... ...
... ...
N ... ...

Figure 2: Our causal feature selection method HTE-FS. Given the full set of features X, the first step in HTE-FS is to select a subset of features that minimizes an HTE fit metric (e.g., Tau risk). Then, given the subset of features found, $\mathbf{X}^{(g)}$, a structure is learned on the path from T to Y. The final feature set is found by removing mediators and descendants of treatment (crossed out in red), which results in the final feature subset of $\mathbf{X}^{(c)} \subseteq \mathbf{X}$.

Instead of doing full structure learning, we propose Heterogeneous Treatment Effect Feature Selection (HTE-FS), which relies on two ideas to make feature selection more practical. First, instead of considering all features, it assesses the value of each feature in reducing the HTE estimation error and selects only the features that contribute to error reduction. Second, instead of learning the full structure, it focuses on learning the parts of the structure that give sufficient information about selecting a valid adjustment set. Figure 2 shows a high-level overview of HTE-FS, and we present its main components, HTE-Fit and Structure-Fit, next.

5.1 HTE-Fit

The first component of our framework, HTE-Fit, proposes the use of *heuristic HTE evaluation* metrics to sequentially select features that are likely to reduce the HTE estimation error. This is shown as the first step in Figure 2. It has two variants: forward (HTE-Fit-F) and backward (HTE-Fit-B).

Since the ground truth of HTE effects in unknown, several heuristic HTE evaluation metrics have been developed to approximate the errors of HTE estimators, including τ-risk [32], Nearest Neighbor Precision in Estimating Heterogeneous Effects (NN-PEHE) [33], Plug-in τ [31], and Counterfactual Cross-validation (CFCV) [31]. The τ-risk takes as input the estimator $\hat{\tau}$ and approximates the objective defined in (3) [32]. The second metric, NN-PEHE, uses nearest neighbor matching for imputing treatment effects for training [33]. The plug-in τ uses an arbitrary estimator for imputing effects for computing error. The fourth metric, CFCV, uses a doubly robust formulation with the plug-in τ for imputing effects for evaluation [31]. These metrics have been shown to correlate with HTE estimator performance [31, 32]. Here, we propose to use these metrics as causal feature selection criteria, which, to the best of our knowledge, has not been studied in previous research.

5.1.1 HTE-Fit — Forward. HTE-Fit-F starts with a feature with the best (lowest) error metric, then iteratively adds features that improve the metric. If there are no more features or improvements, the selection stops. We will use τ-risk as an example metric to make this procedure more concrete. τ-risk has been shown to lead to a consistent high-performing model [32] and to have quasi-oracle error bounds on the CATE [23]. The τ-risk, denoted as R, takes as input the estimator $\hat{\tau}$ and approximates the objective defined in (3) [32]:

$$\tau\text{-risk}(\hat{\tau}) = R(\hat{\tau}) = \frac{1}{N}\sum_{i=1}^{N}\Big((Y_i - \hat{m}(\mathbf{X}_i)) - (T_i - \hat{p}(\mathbf{X}_i))\hat{\tau}(\mathbf{X}_i)\Big)^2. \quad (5)$$

Here, $\hat{m}(\mathbf{X}_i)$ is an outcome estimate for unit i (i.e., $E[Y_i|\mathbf{X}_i]$), and $\hat{p}(\mathbf{X}_i)$ is an estimate of the propensity score, (i.e., $E[T_i|\mathbf{X}_i]$). Both $\hat{m}$ and $\hat{p}$ are learned through supervised techniques and optimized for predictive accuracy.

Suppose the feature vector of individuals has d dimensions, $\mathbf{X} = (X_1, \ldots, X_d)$. Given the τ-risk, we can compute the feature fit for one feature, say $X_j \in \mathbf{X}$, by estimating $\hat{\tau}(\{X_j\})$ and inputting the estimator into the τ-risk. For HTE-Fit-F, the first feature found is the feature that minimizes $R(\hat{\tau}(\{X_j\}))$, and we define the initial set of found features as $\mathbf{X}^{(g)} = \{X_j\}$. Now that the first feature is found, we need to add another to $\mathbf{X}_{(g)}$. For each feature $X_k \in \mathbf{X}, X_k \notin \mathbf{X}^{(g)}$, we estimate a new $\hat{\tau}(\mathbf{X}^{(g)} \cup \{X_k\})$ which is the current set of found features $\mathbf{X}^{(g)}$ combined with a new feature. The next feature added to $\mathbf{X}^{(g)}$ is the feature that minimizes $R(\hat{\tau}(\mathbf{X}^{(g)} \cup \{X_k\})$. This process continues until there is no feature that, when combined with $\mathbf{X}^{(g)}$, minimizes the τ-risk or when there are no more features to add.

5.1.2 HTE-Fit — Backward. Backward feature selection (HTE-Fit-B) works similar to forward feature selection. The difference is that it starts with all features, $\mathbf{X}^{(g)} = \mathbf{X}$, and it iteratively removes features until either the error metric does not improve or there is only one feature left for estimation. Using τ-risk, it would first compute $R(\hat{\tau}(\mathbf{X}))$ to get an initial score on the entire feature set. For each $X_j \in \mathbf{X}$, it would compute $R(\hat{\tau}(\mathbf{X} \setminus \{X_j\})$, and remove the feature that results in the lowest τ-risk when removed.

5.2 Structure-Fit

Now that we have selected the features that can contribute to HTE estimation error reduction, the next step is to select only the features which are a valid adjustment set. The second part of our framework, Structure-Fit, iteratively applies local structure learning [2, 47] to find a path from the treatment, T, to the outcome Y, in order to remove mediators and descendants of treatment. Step 2 of Figure 2 represents this part, Structure-Fit, where local structures from T to Y are learned iteratively, and connected. A local structure is a subgraph of the causal graph centered on a target variable. For example, in Figure 2, the local structure of T are the edges and variables connected to T (e.g., $X_r^{(g)}, X_p^{(g)}, X_j^{(g)}$). Since Y is the outcome variable and we are only interested in paths from T to Y, Structure-Fit terminates and the final feature set is found by removing mediators and descendants (crossed out in red).

This approach has two main advantages compared to full causal structure learning. First, it only considers local structure around causal nodes and does not need to consider all variables if they are not connected to the causal path from T to Y. Second, it aims to only select variables which do not bias causal effect estimation and allow the HTE estimator to use features as appropriate for estimation.

5.2.1 Discovering local structure around target variables. Multiple structures can encode the same conditional independences in the data, making the task of finding the parent sets of T and Y, $pa(T), pa(Y)$, nontrivial. For example, $X_1 \rightarrow T \rightarrow X_2, X_1 \leftarrow T \leftarrow X_2$, and $X_1 \leftarrow T \rightarrow X_2$, all encode that $X_1 \perp X_2 | T$ and in one of them X_1 is a parent of T whereas in the other two X_1 is a child.

We use two different ways to find the local structure. The first way relies on finding the parent-children (PC) set around our target variables, T and Y [2], collider discovery, and edge orientation methods to find causal directions [5, 9, 11]. Several PC discovery algorithms have been developed [2, 6, 27, 47] and have mainly been used for improving classification methods [47]. To distinguish between parents and children, we first discover the parents of each target variable through collider discovery. If two variables are parents of the target variable (e.g., X and A are parents of T), then they are dependent if conditioning on the child and can be discovered through conditional independence tests. However, if there is zero or one parent, we will not leverage collider discovery. Instead, we orient the remaining edges using an edge orientation algorithm [5, 9, 11] and then check if there is a parent in the set. Using these algorithms, we can check for any variable in PC whether an arrow is pointing in or away from the target variable (i.e., T and Y) and thus identify $pa(T), pa(Y), ch(T)$, and $ch(Y)$.

The second way to find local structures is to utilize more recent local structure learning algorithms, such as PCD-by-PCD [46], MB-by-MB [44], and Causal Markov Blanket [12] These algorithms discover the PC or Markov Blanket (MB) while partially orienting edges. We can find a partially oriented local structure and orient edges using edge orientation methods.

5.2.2 Learning partial structures. Using the process to discover local structure around target variables, we propose an iterative algorithm to remove post-treatment variables from HTE-Fit, which we call Structure-Fit. We apply local structure learning on the treatment variable, T, and then iteratively apply local structure learning on children of T in a breadth-first-search manner until we reach the outcome variable Y. Then, local structures are put together to form a subgraph of the causal graph over the variables selected by HTE-Fit. After learning the local structure, we can apply an identification algorithm to find an adjustment set, such as the O-set [16]. Another option is to only remove all descendants of treatment.

We demonstrate Structure-Fit using Figure 1c. We first find $PC(T) = \{A, X, B, D\}$ and $ch(T) = \{B, D\}$. We proceed to iteratively find the PC on each child, B, D. Since B has no children, we move to D and find $PC(D) = \{T, C, E, Y\}$ and $ch(D) = \{E, Y\}$. Finally, E has no children, and we have reached the outcome variable, Y and find $PC(Y) = \{X, C, F, G\}$. Using the local structures, we find the causal graph shown in Figure 1c. In practice, we only use the output of HTE-Fit in Structure-Fit, so not all variable will be used.

While HTE-Fit and Structure-Fit are both important parts of the framework, each one can be used independently for causal feature selection as well. However, each component has limitations, which can be addressed by combining them into one framework. One limitation of HTE-Fit is that it is based on a heuristic evaluation metric for HTE estimation. While the evaluation metric is correlated with the true performance [32], these metrics may still include non-valid adjustment variables. This issue can be addressed by discovering mediators and descendants of treatment and excluding them from the final estimation. One limitation of using Structure-Fit is that variables that are potentially irrelevant for HTE estimation are considered in the local structure learning, which can lead to noisy independence tests and discovered structures. This is addressed by finding the most important subset of features for HTE estimation. HTE-FS takes advantages of HTE-Fit and Structure-Fit to address limitations of both methods. Pseudocode of HTE-Fit and Structure-Fit is available in the Appendix in Sections 9.1 and 9.2, respectively, and a discussion of complexity is available in Section 9.3.

6 EXPERIMENTS

To evaluate whether and how much feature selection improves existing data-driven HTE algorithms, we generate synthetic datasets which contain ground-truth causal effects. We also use a semi-synthetic dataset and two real-world datasets.

6.1 Experimental setup

In our experiments, we make a distinction between four classes of estimation methods: *base learners*, *baseline feature selection*, *our feature selection*, and *oracle*. A base learner is an HTE estimator without feature selection, and it uses all features for estimation. We consider several prominent HTE estimators as base learners: Causal Tree Learn [40], Causal Forest [4], Meta-learners (S, T, XLearners) [20], Doubly Robust learner (DRLearner) [19], Balancing Neural Network (BNN) [18], and TARNet [34].

The baseline feature selection methods use local structure learning [47] and causal structure learning [15, 16] to select an adjustment set. The first baseline, *PC Simple (Parents)*, uses a local structure learning method on the treatment to find parents. PC Simple has been used for selecting variables for classification [47] but not for HTE estimation. We employ PC Simple to learn the local structure on the treatment variable and select the treatment parents as an adjustment set. Since our setting only considers one treatment variable, the parents of treatment are a valid adjustment set [45]. Two other baselines we consider are based on causal structure learning [15, 16], which we refer to as Full Structure Learning (FSL). We use GFCI [24] to learn a causal structure and apply two adjustment criteria: optimal adjustment set (O-set) and removing post-treatment variables only (Valid). Then a base learner uses the adjustment set for estimation (e.g., TLearner).

For our feature selection methods, we consider our framework HTE-FS and each component, HTE-Fit and Structure-Fit, separately. For HTE-Fit, we use both variants: HTE-Fit-F and HTE-Fit-B. We use four HTE fit metrics: τ-risk [32], Nearest Neighbor Precision in Estimating Heterogeneous Effects (NN-PEHE) [33], plug-in τ [31], and Counterfactual Cross-validation (CFCV) [31]. We report results using HTE-Fit-F and τ-risk since this combination performed the best in our experiments.

For Structure-Fit we used PC-Simple [6], GetPC [27], and HITON-PC [2], CMB [12], and PCD-by-PCD [46]. For pairwise edge orientation, we use Conditional Distribution Similarity Statistic (CDS) [11], Information Geometric Causal Inference (IGCI) [9], and Regression Error based Causal Inference (RECI) [5]. In order to find an adjustment set, we only remove mediators and descendants, and allow data-driven HTE estimators to decide which variables to use in estimation. We report results using PC Simple with RECI (PC, RECI) and CMB since these two combinations represent different variants of Structure-Fit and perform the best in our experiments.

For HTE-FS, we report results using HTE-Fit-F with τ-risk and Structure-Fit with CMB, since this combination performed the best in experiments. We compare against using Oracle (Known Structure), which uses the true optimal adjustment set.

6.2 Datasets

We use three types of datasets. The first dataset is a set of synthetic datasets, in which random SCMs are generated, and data is generated following each SCM. The second dataset is based on the Infant Health and Development Program (IHDP) dataset [17]. The third type are real-world datasets, League of Legends and Cannabis.

6.2.1 Synthetic dataset. We generate SCMs by following a procedure similar to one for evaluating causal structure learning algorithms [15]. SCMs and their corresponding datasets differ in the following characteristics: the number of variables d; the probability of edge creation p_e; the variance in the noise σ; the strength of the noise term in non-source nodes ρ. We add parameters to control confounding γ, mediation m, and heterogeneity in treatment effects p_h, m_p. We initialize a set of parameters as follows:

(1) Number of variables: $d \in [10, 20, 30]$
(2) Edge probability parameter: $p_e \in [0.1, 0.3, 0.5]$
(3) Noise variance: $\sigma \in [0.2, 0.4, 0.6]$
(4) Magnitude in noise: $\rho \in [0.1, 0.5, 0.9]$
(5) Confounder: $\gamma \in$ [True, False]
(6) Mediator chain length: $m \in [0, 1, 2]$
(7) Number of HTE inducing parents: $p_h \in [0, 1, 2]$
(8) HTE from mediating parents: $m_p \in$ [True, False]

For each parameter value, we randomly sample from all other parameter values 500 times to generate SCMs. For each of the 11, 500 SCMs, we create one dataset of size 10, 000. Root node variables are generated from a normal distribution, and non-root nodes are generated linearly from parents. The detailed dataset generation description is available in the Appendix in Section 9.4, and the link for the code is available in Section 9.5.

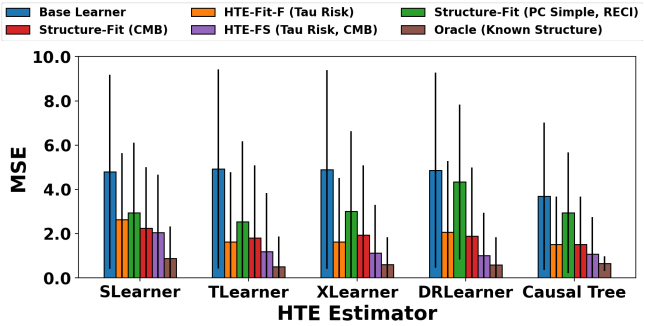


Figure 3: Average MSE for all SCMs generated using: 30 variables, confounding, mediator chain length of 2, 2 HTE inducing parents, and HTE induced by mediating parents (i.e., $(d, \gamma, m, p_h, m_p) = (30, True, 2, 2, True)$). We show how feature selection methods improve several HTE estimators.

6.2.2 Semi-synthetic dataset: IHDP. The covariates in the Infant Health and Development Program (IHDP) dataset come from a randomized experiment studying the effects of specialist home visits on future cognitive test scores [17]. Outcomes can be simulated using the covariates. We simulate 1,000 instances of simulated outcomes and average all results[1].

6.2.3 Real-world datasets. We use two real-world datasets: *League of Legends* and *Cannabis*. League of Legends (LoL) is a popular multiplayer online battle arena (MOBA) game. Riot Games, the developer of LoL, regularly creates software patches to the game that update various parts of the game, affecting balance and how players decide to play the game. We use a dataset consisting of player, match, and patch information over time [14]. The treatment variable of interest is a software patch, and the outcome of interest is the number of *kills* of a team in a single match, which is a post-match statistic. We average results over patches in the dataset.

Cannabis is a dataset consisting of tweets about the e-cigarette Juul and cannabis-related topics [1]. English tweets related to the e-cigarette Juul were collected from 2016 to 2018. For the users who mention Juul, cannabis-related tweets are collected from 2014 to 2018. In this dataset, tweets' stances towards Juul and cannabis are crowdsourced, and a model is trained for stance detection. In this work, we are interested in how users' stance on Juul affects their stance on cannabis. To do this, we find users who have Juul-related tweets occurring before a cannabis-related tweet and detect their stance on Juul and cannabis using the classifier from [1], where features are the word embeddings for tweets using BERT [10]. The treatment variable of interest is a user's stance on Juul, in favor (1) or against (0). The outcome variable is their stance on cannabis, in favor (1) or against (0). Since this dataset contains only one group, we bootstrap the estimators to compute the standard deviation.

[1] Outcomes are generated using the JustCause package:
https://github.com/inovex/justcause

		Mediator Chain Length 0	Mediator Chain Length 1	Mediator Chain Length 2	Contains Confounding	No Confounding	0 HTE Parents	1 HTE Parent	2 HTE Parents	10 Variables	20 Variables	30 Variables
Base Learners	SLearner	10.58 ± 2.25	10.39 ± 2.31	10.38 ± 2.19	10.55 ± 2.08	10.36 ± 2.38	10.55 ± 2.22	10.08 ± 2.38	10.63 ± 2.16	10.56 ± 2.41	10.26 ± 2.30	10.55 ± 2.09
	TLearner	8.18 ± 2.10	8.07 ± 2.17	8.35 ± 2.14	8.32 ± 2.04	8.07 ± 2.22	8.26 ± 2.06	7.93 ± 2.32	8.32 ± 2.04	8.06 ± 2.22	8.06 ± 2.22	8.40 ± 1.98
	XLearner	8.31 ± 2.30	8.37 ± 2.28	8.13 ± 2.20	8.28 ± 2.14	8.30 ± 2.35	8.24 ± 2.24	8.04 ± 2.36	8.51 ± 2.18	8.32 ± 2.46	8.15 ± 2.29	8.44 ± 2.08
	DRLearner	9.64 ± 2.53	9.28 ± 2.72	9.73 ± 2.73	9.57 ± 2.65	9.42 ± 2.71	9.78 ± 2.36	9.14 ± 2.90	9.60 ± 2.66	8.94 ± 2.66	9.30 ± 2.73	10.11 ± 2.53
	Causal Tree	7.89 ± 4.57	7.33 ± 4.23	7.61 ± 4.30	8.04 ± 4.48	7.18 ± 4.21	7.67 ± 4.59	8.02 ± 4.22	7.08 ± 4.25	6.71 ± 4.30	8.06 ± 4.31	7.45 ± 4.32
	Causal Forest	6.92 ± 3.49	8.14 ± 3.63	6.88 ± 3.53	6.90 ± 3.51	7.93 ± 3.64	6.67 ± 3.39	7.15 ± 3.65	8.26 ± 3.59	8.02 ± 3.49	7.07 ± 3.65	7.70 ± 3.63
	BNN	7.28 ± 3.07	7.65 ± 2.98	7.40 ± 3.20	7.49 ± 3.18	7.49 ± 2.98	7.72 ± 3.10	7.65 ± 3.13	7.25 ± 2.98	7.75 ± 2.92	7.61 ± 3.17	7.18 ± 3.01
	TARNet	8.48 ± 3.08	9.01 ± 3.07	8.40 ± 3.34	8.44 ± 3.28	8.91 ± 3.06	8.97 ± 2.92	8.54 ± 3.25	8.71 ± 3.22	9.19 ± 3.02	8.58 ± 3.20	8.57 ± 3.18
Baseline Feature Selection with TLearner	PC Simple (Parents)	6.51 ± 3.01	6.50 ± 3.52	7.29 ± 3.69	7.11 ± 3.23	6.45 ± 3.26	7.18 ± 3.03	6.80 ± 3.56	6.92 ± 2.86	6.74 ± 3.66	7.41 ± 3.33	6.61 ± 3.38
	FSL (O-set)	4.86 ± 2.99	4.37 ± 2.85	4.86 ± 2.98	4.73 ± 2.88	4.54 ± 2.97	4.52 ± 2.93	4.89 ± 3.09	4.45 ± 2.78	4.25 ± 3.01	4.81 ± 3.02	4.64 ± 2.73
	FSL (Valid)	4.82 ± 3.14	4.46 ± 2.95	5.04 ± 3.22	4.81 ± 3.08	4.63 ± 3.21	4.49 ± 3.05	5.04 ± 3.35	4.39 ± 3.03	4.18 ± 3.19	4.72 ± 3.20	4.82 ± 3.00
Our Feature Selection with TLearner	HTE-Fit (Tau Risk)	4.32 ± 1.90	4.33 ± 1.76	4.40 ± 2.05	4.30 ± 1.98	4.37 ± 1.79	4.38 ± 1.92	4.24 ± 1.91	4.43 ± 1.83	4.63 ± 1.92	4.22 ± 1.87	4.33 ± 1.82
	Structure-Fit (PC, RECI)	5.82 ± 3.18	5.74 ± 3.23	5.99 ± 3.37	5.69 ± 3.18	5.93 ± 3.31	5.72 ± 3.19	6.28 ± 3.58	5.53 ± 2.98	5.80 ± 3.23	6.11 ± 3.45	5.48 ± 2.99
	Structure-Fit (CMB)	3.84 ± 3.21	3.51 ± 2.92	3.99 ± 3.03	3.59 ± 3.10	**3.62 ± 3.09**	3.72 ± 2.96	4.12 ± 3.31	3.51 ± 2.80	**3.67 ± 3.10**	4.01 ± 3.05	3.53 ± 2.75
	HTE-FS (Tau, CMB)	**3.67 ± 1.96**	**3.30 ± 2.13**	**3.69 ± 2.32**	**3.56 ± 2.19**	3.69 ± 2.53	**3.26 ± 2.33**	**3.94 ± 2.73**	**3.47 ± 2.40**	3.73 ± 2.59	**3.68 ± 2.61**	**3.30 ± 1.89**
Oracle with TLearner	Known Structure	2.82 ± 3.44	2.42 ± 3.28	2.98 ± 3.45	2.79 ± 3.34	2.58 ± 3.40	2.62 ± 3.21	3.14 ± 3.64	2.33 ± 3.21	2.88 ± 3.44	2.88 ± 3.46	2.26 ± 3.17

Table 1: Average rank for different HTE estimation methods under varying causal structures. Each column shows a fixed parameter value, and averaged ranks over all other settings. The top section shows HTE estimators without feature selection. The second section of the table shows baseline causal feature selection methods, using TLearner as the base HTE estimator. The third section shows our proposed causal feature selection methods. The bottom section shows the Oracle. Bolded results indicate best average rank (ignoring Oracle estimation).

6.3 Evaluation

We use the mean squared error (MSE) for evaluating HTE estimations since the true causal effect is known with synthetic and semi-synthetic data: MSE $= \frac{1}{N}\sum_{i=1}^{N}(\tau(\mathbf{X}) - \hat{\tau}(\mathbf{X}_i))^2$. For each dataset, we split the data into 80/20 training and test sets. Since different SCMs will result in different MSEs, we first rank all estimation methods based on their MSE for a particular SCM. Then we average the rank of each method across SCMs and report on the average rank. To study their performance under specific settings, we also report on average ranks for subsets of parameter settings. For real-world datasets, we report results using τ-risk [23, 32], which we defined in eq (5). Similar to MSE, we report rankings based on τ-risk.

In addition, for datasets in which we can decide which variables *post-treatment* variables, we compute the percentage of included by each feature selection method, which we denote as *inclusion error (IE)*. Given the *post-treatment* set, $\mathcal{F}$, and the predicted causal feature set $\mathbf{X}^{(\ell)}$, we compute the inclusion error as:

$$\text{IE} = \frac{|\mathbf{X}^{(\ell)} \cap \mathcal{F}|}{|\mathcal{F}|}. \tag{6}$$

Some variables occur after treatment for real-world datasets, such as other post-match statistics in LoL and tweets related to cannabis after the initial cannabis-related tweet in the Cannabis dataset.

6.4 Evaluation on synthetic data

We first present the results on synthetic datasets. We first show how our feature selection method improves multiple base learners. Then we show how feature selection improves base learners across multiple structural configurations.

6.4.1 Feature selection helps all base learners. Figure 3 shows the MSE comparisons between several HTE estimators and feature selection methods on SCMs which are generated using: 30 variables, confounding, mediator chain length of 2, 2 HTE inducing parents, and HTE induced by mediating parents (i.e., $(d, \gamma, m, p_h, m_p) = (30, True, 2, 2, True)$). We average MSE over all other parameters (i.e., p_e, σ, ρ). We show the best performing Structure-Fit, HTE-Fit-F, HTE-FS. This figure shows that adding feature selection improves MSE for all HTE estimators. In addition, feature selection reduces the error bars for all base learners. In general, HTE-FS (Tau risk, CMB) performs the best, compared to using HTE-Fit-F or Structure-Fit alone. Structure-Fit with CMB performs better than Structure-Fit with PC Simple and RECI, although the latter still reduces error overall compared to using the base learner on all features.

6.4.2 Feature selection improves base learners for different types of structures. Table 1 ranks the feature selection methods using TLearner as the base learner vs. several other HTE estimators (without using feature selection) under a range of structural scenarios. We choose TLearner as the base learner since it is a relatively simple model and performs consistently in all settings. One experimental parameter that affects the causal structure is fixed in each column while averaging over the other parameters. The top section shows the average rank for base learners without feature selection, the second section shows baseline feature selection methods, the third section shows our feature selection methods, and the bottom section shows the Oracle method. Bolded results indicate the best rank in the column (ignoring the Oracle estimator). The large variety of underlying structures explains the high variance in rank.

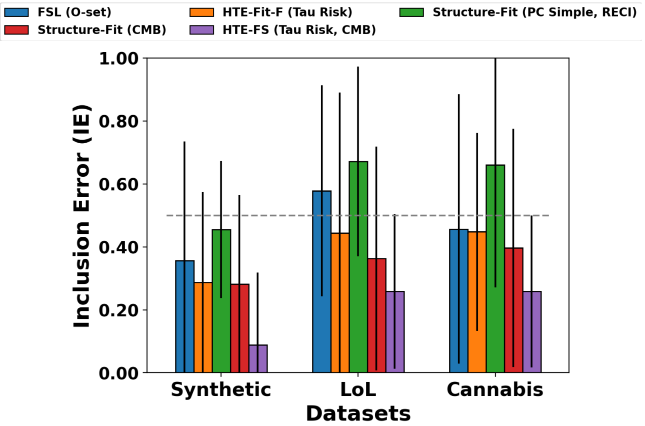


Figure 4: How often feature selection methods include descendants of treatment (lower is better). The grey line denotes 50%, or selected at random.

The table shows that our feature selection methods improve base learners for different structures, perform better than baseline feature selection methods, and perform better than other base learners without feature selection. HTE-FS performs the best overall. In two settings, with no confounding and a low number of variables, Structure-Fit with CMB performs the best. While our methods perform better than baseline methods, baseline methods still improve base learners, compared to no feature selection.

6.4.3 Feature selection reduces inclusion of wrong variables. In addition to performance in terms of MSE, we investigate the *inclusion error (IE)* of feature selection methods. Figure 4 shows the inclusion error for each feature selection method, averaged over all experiments, where lower is better. The grey line represents the case of randomly selected features. The first set of bars shows the average percentage of overall synthetic datasets. All feature selection methods do better than random feature selection. HTE-FS has the lowest chance to include wrong variables, at about 8% inclusion rate. This result supports the results shown in Table 1, that HTE-FS has the lowest error, followed by Structure-Fit with CMB and HTE-Fit-F.

6.5 Evaluation on IHDP dataset

Table 2 shows the average rankings on the IHDP dataset using MSE and τ-risk. We used TLearner as the base learner and showed the same variations in feature selection as in Table 1. We see that all feature selection methods perform better on average than the base estimators, with HTE-FS performing the best. Since TLearner is a linear model close to the data generation process for this dataset, it already performs very well on it. Since this dataset contains pre-treatment features only, we do not evaluate inclusion error.

6.6 Evaluation on real-world datasets

Finally, we show results on real-world datasets. Since these datasets do not contain ground truth effects, we evaluate methods using τ-risk rankings.

		MSE Rank	τ-risk Rank	LoL τ-risk Rank	Cannabis τ-risk Rank
Base Learners	SLearner	7.82 ± 2.07	9.01 ± 2.31	12.02 ± 0.51	14.06 ± 0.80
	TLearner	4.65 ± 1.63	5.16 ± 1.79	4.97 ± 0.50	7.86 ± 1.03
	XLearner	5.03 ± 1.84	5.36 ± 1.70	4.72 ± 0.36	6.22 ± 0.97
	DRLearner	9.29 ± 2.33	9.44 ± 2.05	10.74 ± 0.45	14.58 ± 0.58
	Causal Tree	8.84 ± 2.51	9.14 ± 3.02	8.48 ± 0.58	7.01 ± 0.92
	Causal Forest	10.01 ± 2.46	7.12 ± 3.01	10.93 ± 0.46	9.68 ± 0.99
	BNN	8.51 ± 1.74	8.55 ± 2.71	10.45 ± 0.69	13.01 ± 0.85
	TARNet	9.24 ± 1.61	8.08 ± 2.79	9.09 ± 0.54	12.02 ± 0.82
Baseline Feature Selection with TLearner	PC Simple (Parents)	4.61 ± 1.67	4.19 ± 1.82	6.51 ± 0.64	11.07 ± 0.76
	FSL (O-set)	3.48 ± 1.94	3.99 ± 2.34	3.47 ± 0.47	2.82 ± 1.56
	FSL (Valid)	4.23 ± 1.70	5.17 ± 1.32	4.44 ± 0.37	5.02 ± 0.61
Our Selection with TLearner	HTE-Fit-F (Tau Risk)	3.30 ± 1.89	3.84 ± 2.37	3.97 ± 0.44	2.76 ± 0.88
	Structure-Fit (PC, RECI)	3.53 ± 1.94	4.01 ± 2.34	5.66 ± 0.49	9.20 ± 0.76
	Structure-Fit (CMB)	3.25 ± 1.75	3.45 ± 1.88	3.27 ± 0.37	2.74 ± 0.90
	HTE-FS (Tau, CMB)	**2.77 ± 1.70**	**3.37 ± 1.47**	**2.48 ± 0.44**	**1.95 ± 1.27**

Table 2: Average rank of estimation methods for the IHDP, LoL, and Cannabis datasets.

6.6.1 LoL dataset. The average rank for each method in terms of τ-risk for the LoL dataset is shown in Table 2. Overall, HTE-FS achieves the lowest τ-risk rank among all estimators. We see that TLearner performs the best among base learners and has a lower rank than some feature selection methods. One reason could be that some methods are not suitable for high dimensional datasets, such as Structure-Fit with PC Simple. However, Structure-Fit with CMB performs much better than PC Simple, HTE-Fit-F, and FSL.

In the LoL dataset, there are pre-match and post-match features. Since our outcome *kills* is a post-match feature, other post-match features should not be included in the estimation, which is the *post-treatment* set in this setting. The second set of bars in Figure 4 show the inclusion error for the LoL datasets. Overall, each feature selection method has a higher IE than in the synthetic dataset. HTE-FS, Structure-Fit with CMB, and HTE-Fit-F reduce the inclusion error rate on average, while Structure-Fit with PC-Simple and FSL do not reduce the inclusion error more than random selection.

6.6.2 Cannabis. The rank in terms of τ-risk for the Cannabis dataset is shown in Table 2. Since this dataset contains only one set of data, we bootstrap each estimator. In this dataset, HTE-FS achieves the lowest τ-risk, followed by FSL and HTE-Fit-F. Here, XLearner achieves the lowest τ-risk among base learners and performs close to our proposed methods.

In the Cannabis dataset, the post-treatment features are tweets after the first cannabis-related tweet. The features that should be included are only tweets related to the stance towards Juul and the tweet related to the first cannabis tweet. Figure 4 shows the inclusion error for the Cannabis dataset in the rightmost set of bars. In most cases, our feature selection methods reduce the number of wrong variables included, except Structure-Fit with PC-Simple.

7 CONCLUSION

In this work, we addressed the problem of improving data-driven heterogeneous treatment effect (HTE) estimation when the underlying structural causal model (SCM) is unknown. We developed HTE-FS for causal feature selection for HTE estimation which consists of two components: HTE-Fit and Structure-Fit. We show that HTE-FS reduces HTE estimation error of existing data-driven methods under a variety of causal model assumptions when compared to using their vanilla versions with all available features. Our research suggests several interesting avenues of future research, including the development of causal feature selection algorithms in the presence of latent variables and the consideration of the full do-calculus, not just backdoor adjustment.

8 ACKNOWLEDGMENTS

This material is based on research sponsored in part by NSF under grant No. 2047899, DARPA under contract number HR001121C0168, and an Adobe Data Science Research Award.

REFERENCES

[1] Shishir Adhikari, Akshay Uppal, Robin Mermelstein, Tanya Berger-Wolf, and Elena Zheleva. 2021. Understanding the Dynamics between Vaping and Cannabis Legalization Using Twitter Opinions. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 15. 14–25.

[2] Constantin F Aliferis, Alexander Statnikov, Ioannis Tsamardinos, Subramani Mani, and Xenofon D Koutsoukos. 2010. Local causal and markov blanket induction for causal discovery and feature selection for classification part i: Algorithms and empirical evaluation. *JMLR* (2010).

[3] Susan Athey and Guido Imbens. 2016. Recursive partitioning for heterogeneous causal effects. *PNAS* (2016).

[4] Susan Athey, Julie Tibshirani, and Stefan Wager. 2019. Generalized random forests. *The Annals of Statistics* 47, 2 (2019), 1148–1178.

[5] Patrick Blöbaum, Dominik Janzing, Takashi Washio, Shohei Shimizu, and Bernhard Schölkopf. 2018. Cause-effect inference by comparing regression errors. In *AISTATS*.

[6] Peter Bühlmann, Markus Kalisch, and Marloes H Maathuis. 2010. Variable selection in high-dimensional linear models: partially faithful distributions and the PC-simple algorithm. *Biometrika* (2010).

[7] Debo Cheng, Jiuyong Li, Lin Liu, Jixue Liu, and Thuc Duy Le. 2020. Local search for efficient causal effect estimation. *arXiv preprint arXiv:2007.10014* (2020).

[8] David Maxwell Chickering, David Heckerman, and Christopher Meek. 2004. Large-sample learning of Bayesian networks is NP-hard. *JMLR* (2004).

[9] Povilas Daniušis, Dominik Janzing, Joris Mooij, Jakob Zscheischler, Bastian Steudel, Kun Zhang, and Bernhard Schölkopf. 2010. Inferring deterministic causal relations. (2010).

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 4171–4186.

[11] José AR Fonollosa. 2019. Conditional distribution variability measures for causality detection. In *Cause Effect Pairs in Machine Learning*. Springer.

[12] Tian Gao and Qiang Ji. 2015. Local causal discovery of direct causes and effects. *Advances in Neural Information Processing Systems* 28 (2015), 2512–2520.

[13] Justin Grimmer, Solomon Messing, and Sean J Westwood. 2017. Estimating heterogeneous treatment effects and the effects of heterogeneous treatments with ensemble methods. *Political Analysis* (2017).

[14] Yuzi He, Christopher Tran, Julie Jiang, Keith Burghardt, Emilio Ferrara, Elena Zheleva, and Kristina Lerman. 2021. Heterogeneous Effects of Software Patches in a Multiplayer Online Battle Arena Game. In *The 16th International Conference on the Foundations of Digital Games (FDG) 2021*. 1–9.

[15] Christina Heinze-Deml, Marloes H Maathuis, and Nicolai Meinshausen. 2018. Causal structure learning. *Annual Review of Statistics and Its Application* (2018).

[16] Leonard Henckel, Emilija Perković, and Marloes H Maathuis. 2019. Graphical criteria for efficient total effect estimation via adjustment in causal linear models. *arXiv preprint arXiv:1907.02435* (2019).

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

[18] Fredrik D Johansson, Uri Shalit, and David Sontag. 2016. Learning Representations for Counterfactual Inference. In *ICML*.

[19] Edward H Kennedy. 2020. Optimal doubly robust estimation of heterogeneous causal effects. *arXiv preprint arXiv:2004.14497* (2020).

[20] Sören R Künzel, Jasjeet S Sekhon, Peter J Bickel, and Bin Yu. 2019. Metalearners for estimating heterogeneous treatment effects using machine learning. *PNAS* (2019).

[21] Marloes H Maathuis and Diego Colombo. 2015. A generalized back-door criterion. *The Annals of Statistics* 43, 3 (2015), 1060–1088.

[22] Jacob M Montgomery, Brendan Nyhan, and Michelle Torres. 2018. How conditioning on posttreatment variables can ruin your experiment and what to do about it. *American Journal of Political Science* 62, 3 (2018), 760–775.

[23] Xinkun Nie and Stefan Wager. 2021. Quasi-oracle estimation of heterogeneous treatment effects. *Biometrika* 108, 2 (2021), 299–319.

[24] Juan Miguel Ogarrio, Peter Spirtes, and Joe Ramsey. 2016. A hybrid causal search algorithm for latent variable models. In *Conference on Probabilistic Graphical Models*. PMLR, 368–379.

[25] Judea Pearl. 2009. *Causality*. Cambridge university press.

[26] Judea Pearl. 2017. Detecting latent heterogeneity. *Sociological Methods & Research* (2017).

[27] Jose M Peña, Johan Björkegren, and Jesper Tegnér. 2005. Scalable, efficient and correct learning of Markov boundaries under the faithfulness assumption. In *ECSQARU*. Springer.

[28] Emilija Perković, Johannes Textor, Markus Kalisch, and Marloes H Maathuis. 2015. A complete generalized adjustment criterion. In *UAI*.

[29] Paul R Rosenbaum and Donald B Rubin. 1983. The central role of the propensity score in observational studies for causal effects. *Biometrika* (1983).

[30] Donald B Rubin. 1974. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology* (1974).

[31] Yuta Saito and Shota Yasui. 2020. Counterfactual Cross-Validation: Stable Model Selection Procedure for Causal Inference Models. *ICML* (2020).

[32] Alejandro Schuler, Michael Baiocchi, Robert Tibshirani, and Nigam Shah. 2018. A comparison of methods for model selection when estimating individual treatment effects. *arXiv preprint arXiv:1804.05146* (2018).

[33] Patrick Schwab, Lorenz Linhardt, and Walter Karlen. 2018. Perfect match: A simple method for learning representations for counterfactual inference with neural networks. *arXiv preprint arXiv:1810.00656* (2018).

[34] Uri Shalit, Fredrik D Johansson, and David Sontag. 2017. Estimating individual treatment effect: generalization bounds and algorithms. In *ICML*.

[35] Ilya Shpitser and Judea Pearl. 2006. Identification of conditional interventional distributions. In *22nd Conference on Uncertainty in Artificial Intelligence, UAI 2006*.

[36] Ilya Shpitser and Judea Pearl. 2008. Complete Identification Methods for the Causal Hierarchy. *Journal of Machine Learning Research* 9, 9 (2008).

[37] Ilya Shpitser, Tyler VanderWeele, and James M Robins. 2010. On the validity of covariate adjustment for estimating causal effects. *UAI* (2010).

[38] Peter Spirtes, Clark N Glymour, Richard Scheines, and David Heckerman. 2000. *Causation, prediction, and search*. MIT press.

[39] Xiaogang Su, Chih-Ling Tsai, Hansheng Wang, David M Nickerson, and Bogong Li. 2009. Subgroup analysis via recursive partitioning. *Journal of Machine Learning Research* 10, Feb (2009), 141–158.

[40] Christopher Tran and Elena Zheleva. 2019. Learning triggers for heterogeneous treatment effects. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 5183–5190.

[41] Benito Van der Zander, Maciej Liskiewicz, and Johannes Textor. 2014. Constructing Separators and Adjustment Sets in Ancestral Graphs.. In *CI@ UAI*. 11–24.

[42] Tyler J VanderWeele and James M Robins. 2007. Four types of effect modification: a classification based on directed acyclic graphs. *Epidemiology* (2007).

[43] Matthew J Vowels, Necati Cihan Camgoz, and Richard Bowden. 2021. D’ya like dags? a survey on structure learning and causal discovery. *arXiv preprint arXiv:2103.02582* (2021).

[44] Changzhang Wang, You Zhou, Qiang Zhao, and Zhi Geng. 2014. Discovering and orienting the edges connected to a target variable in a DAG via a sequential local learning approach. *Computational Statistics & Data Analysis* 77 (2014), 252–266.

[45] Elizabeth J Williamson, Zoe Aitken, Jock Lawrie, Shyamali C Dharmage, John A Burgess, and Andrew B Forbes. 2014. Introduction to causal diagrams for confounder selection. *Respirology* 19, 3 (2014), 303–311.

[46] Jianxin Yin, You Zhou, Changzhang Wang, Ping He, Cheng Zheng, and Zhi Geng. 2008. Partial orientation and local structural learning of causal networks for prediction. In *Causation and Prediction Challenge*. PMLR, 93–105.

[47] Kui Yu, Xianjie Guo, Lin Liu, Jiuyong Li, Hao Wang, Zhaolong Ling, and Xindong Wu. 2020. Causality-based feature selection: Methods and evaluations. *ACM Computing Surveys (CSUR)* (2020).

[48] Achim Zeileis, Torsten Hothorn, and Kurt Hornik. 2008. Model-based recursive partitioning. *Journal of Computational and Graphical Statistics* 17, 2 (2008).

9 APPENDIX

Here we provide additional information for Heterogeneous Treatment Effect Feature Selection (HTE-FS) and the synthetic data generation.

9.1 HTE-Fit

Here we present pseudocode for the two HTE-Fit algorithms. The HTE-Fit — Forward algorithm is shown in Algorithm 1. Lines 1-8 first find the one feature that returns the largest score (computed by R) and adds it to $\mathbf{X}^{(g)}$. Then, for each remaining feature $X^{(j)}$, we compute the score when combining $X^{(j)}$ to $\mathbf{X}^{(g)}$, and concatenate the feature that returns the best score. This repeats until the score does not change, or there are no more features to add.

HTE-Fit — Backward works similarly to HTE-Fit-F and the pseudocode is shown in Algorithm 2. We start by computing the score using all features in Line 1. In Lines 9-15, we iteratively add features to **K**, which are features that when removed improve the score. The final set returned is the whole feature set without **K**.

Algorithm 1 HTE-Fit — Forward

```
Require: A heuristic evaluation metric $R$ and features $\mathbf{X}$
Ensure: An adjustment set $\mathbf{X}^{(g)}$.
 1: bestScore = $\infty$
 2: bestFeature = None
 3: for each $X^{(j)} \in \mathbf{X}$ do
 4:     Compute tempScore = $R(X^{(j)})$
 5:     if tempScore < bestScore then
 6:         bestFeature = $X^{(j)}$
 7:         bestScore = tempScore
 8: $\mathbf{X}^{(g)}$ = (bestFeature)
 9: while bestScore changes do
10:     for each $X^{(j)} \in \mathbf{X} \setminus \mathbf{X}^{(g)}$ do
11:         Compute tempScore = $R(\mathbf{X}^{(g)} \oplus X^{(j)})$
12:         if tempScore < bestScore then
13:             bestFeature = $X^{(j)}$
14:             bestScore = tempScore
15:     $\mathbf{X}^{(g)} = \mathbf{X}^{(g)} \oplus$ (bestFeature)
16: return $\mathbf{X}^{(g)}$
```

Algorithm 2 HTE-Fit — Backward

```
Require: A heuristic evaluation metric $R$ and features $\mathbf{X}$
Ensure: An adjustment set $\mathbf{X}^{(g)}$.
 1: bestScore = $R(\mathbf{X})$
 2: $\mathbf{K}$ = () // Features to remove
 3: while bestScore changes do
 4:     for each $X^{(j)} \in \mathbf{X}$ do
 5:         Compute tempScore = $R(\mathbf{X} \setminus \mathbf{K} \setminus (X^{(j)}))$
 6:         if tempScore < bestScore then
 7:             $\mathbf{K} = \mathbf{K} \oplus (X^{(j)})$
 8:             bestScore = tempScore
 9: $\mathbf{X}^{(g)} = \mathbf{X} \setminus \mathbf{K}$
10: return $\mathbf{X}^{(g)}$
```

9.2 Structure-Fit

Algorithm 3 details Structure-Fit. In lines 1-2, we initialize two variables: dn, which are discovered nodes, **Z**, a queue of nodes to traverse. The discovered nodes are initialized to $ch(Y)$, so we do not traverse past Y. In lines 3-5, we initialize a graph $\hat{G}$ which will be the local structure from T to Y, and **Z** is initialized to be the children of T. Lines 6-10 show the iterative process in which we consider the nodes in **Z**. For every node $Z \in \mathbf{Z}$, find $pa(Z)$ and $ch(Z)$. Z is added to the discovered nodes, is removed from **Z**, and the children of Z are added to **Z**. Then we connect Z to its children. In the end, a graph, $\hat{G}$ with causal paths from T to Y is constructed, and we can construct the valid set.

Algorithm 3 Structure-Fit

```
Require: Parent and children sets of $T$ and $Y$: $pa(T)$, $ch(T)$,
    $pa(Y), ch(Y)$
 1: Initialize $dn = ch(Y)$
 2: Initialize $\mathbf{Z} = ch(T)$
 3: $\mathbf{V} = \{T\} \cup ch(T)$
 4: $\mathbf{E} = \{(T, Z) \mid Z \in ch(T)\}$
 5: $\hat{G} = (\mathbf{V}, \mathbf{E})$
 6: for $Z \in \mathbf{Z}$ do
 7:     Find $pa(Z), ch(Z)$
 8:     $dn = dn \cup \{Z\}$
 9:     $\mathbf{Z} = (\mathbf{Z} \cup ch(Z)) \setminus dn$
10:     $\mathbf{V} = \mathbf{V} \cup \{Z\}$
11:     $\mathbf{E} = \mathbf{E} \cup \{(Z, Z_c \mid Z_c \in ch(Z)\}$
12: $forb(T, Y, \hat{G}) = de(T)$
13: return $\mathbf{V} \setminus forb(T, Y, \hat{G})$
```

9.3 Complexity of HTE-FS

The complexity of HTE-FS depends on the complexity of training and prediction of the HTE estimator and the local structure learning algorithm. Given an HTE estimator with complexity $O(H)$ training and $O(P)$ prediction, HTE-Fit takes $O(d^2 \cdot (H + P))$. If HTE-Fit outputs s variables and we use CMB for Structure-Fit, then the worse case scenario happens when we find local structures for all s variables: $O(s \cdot 2^{|PC|}|U|^2|PC|)$ [12].

9.4 Synthetic data generation

Here, we describe how we generate synthetic datasets used in the experiments. We first generate an adjacency matrix, **A** over the d variables. Assuming that the variables $\{1, \ldots, d\}$ are causally ordered. For each pair of nodes i, j, where $i < j$ in the causal order, we sample from Bernoulli(p_e) to determine if there is an edge from i to j. With the adjacency matrix, we sample the corresponding coefficients, C, from Unif($-1, 1$). Noise terms ϵ_i are generated by sampling from a d-dimensional zero-mean Gaussian distribution with covariance matrix Σ, where $\Sigma_{i,i} = 1$ and $\Sigma_{i,j} = \sigma$. We use ρ to tune the magnitude of the noise.

Next, we choose the treatment and outcome nodes based on two parameters, γ and m. γ controls whether there is confounding in the generated dataset and m controls the length of the mediating chain (0 means no mediator). First, we select all pairs of nodes which have a *directed m* hop path. Then, if γ is True, we filter all pairs that do not contain a *backdoor path.* Otherwise we filter out pairs that contain a backdoor path. Finally, we sample a pair of nodes from the filtered set to obtain the treatment and outcome nodes. If no pairs of nodes satisfy the criteria for the given parameter setting, we resample the graph up to 100 times until a graph is found, or we discard the parameter setting.

A source node i (node with no parents) takes the value of the sampled noise terms, ϵ_i. For a non-source node, i, we consider several additive factors. First is the base term from the coefficient matrix: $C_{pa(i),i}X_{pa(i)}$, where $C_{pa(i),i}$ are the coefficients from $pa(i)$ to i and $X_{pa(i)}$ are the values generated for $pa(i)$. Next, are the interaction terms, which are determined by the mediating chain length m, number of HTE inducing parents p_h, and whether there is HTE from parents of mediators m_p. For mediators, we have: $\mathbb{1}_m(i)\mathbb{1}_{m_p}X_{pa_m(i)}X_{pa_n(i)}$, where $\mathbb{1}_m$ is an indicator for if i is a mediator, $\mathbb{1}_{m_p}$ is an indicator for if there is HTE induced from mediator parents, and pa_m and pa_n represent mediating and non-mediating parents. For outcomes, the interaction is defined similarly: $\mathbb{1}_y(i)X_{pa_m(i)}X_{pa_n(i)}$, where $\mathbb{1}_y(i)$ indicates whether the node is the outcome node. The number of HTE parents determines how many non-mediating parents (pa_n) the interaction occurs with. A value of zero means there is no heterogeneity. So the final data generating equation is:

$$\begin{aligned} X_i = &\ C_{pa(i),i}X_{pa(i)} \\ &+ \mathbb{1}_m(i)\mathbb{1}_{m_p}X_{pa_m(i)}X_{pa_n(i)} \\ &+ \mathbb{1}_y(i)X_{pa_m(i)}X_{pa_n(i)} + \epsilon_i \end{aligned} \tag{7}$$

9.5 Code

Code for HTE-FS and the synthetic data generation is available at https://github.com/edgeslab/causal_feature_selection.