Improving Local Explainability By Learning Causal Graphs From Data

Anonymous authors
Paper under double-blind review

Abstract

Causal Shapley values take into account causal relations among dependent features to adjust the contributions of each feature to a prediction. A limitation of this approach is that it can only leverage known causal relations. In this work we combine the computation of causal Shapley values with causal discovery, i.e. learning causal graphs from data. In particular, we compute causal explanations across a set of candidate causal graphs learned from observational data, yielding a set of Shapley values that reflects the space of possible explanations consistent with the data. We propose two methods for estimating this list efficiently, drawing on the equivalences of the interventional distributions for a subset of the causal graphs. We evaluate our methods on synthetic and real-world data, showing that they provide explanations that are often closer to the true causal impacts compared to traditional Shapley value approaches that disregard causal relationships. Even when the discovered graph or MEC is imperfect, we on average observe improvements over marginal and conditional Shapley values.

1 Introduction

The advancement of machine learning models has ushered in a new era of AI capabilities, surpassing simpler models in performance. However, this increase in complexity brings a significant challenge: a marked decrease in interpretability with advanced models often resembling a black box (Zhang et al., 2020; Carvalho et al., 2019). This opacity is not just a technical concern but becomes a matter of ethical and legal importance in high-stakes applications, e.g. medical diagnostics or autonomous vehicles, where the inability to explain a model's reasoning can lead to issues of trust and accountability (Doshi-Velez & Kim, 2017).

Shapley values are a notable method used to explain how machine learning models make their decisions (Zhou et al., 2021). They offer a principled local approach to attributing the output of a model to its input features, breaking down the prediction into contributions from each feature. When using Shapley values for explainability, various methods assume independence between the model's input features (Štrumbelj & Kononenko, 2014; Lundberg & Lee, 2017). As a result, importance is only attributed to features that directly contribute to a prediction (Heskes et al., 2020). As a consequence, many inputs to the model will be off the data manifold (Frye et al., 2021) and may lead to misleading explanations (Aas et al., 2021). Relaxing this assumption leads to explanations where correlated features share importance, regardless of their causal effect on the prediction (Aas et al., 2021). As a result, conditional Shapley values may attribute importance to features that are merely correlated with the prediction.

Heskes et al. (2020) integrate causal modeling with the Shapley value framework, leveraging the knowledge of causal relations among the features to adjust the contribution of each feature. This ensures that features only contribute to explanations when they have a causal effect on the prediction, whether direct or indirect. This paper aims to address one of the key challenges in applying causal Shapley values: leveraging only known causal relations. As described in the overview in Fig. 1, we first use causal discovery methods, e.g. PC (Spirtes et al., 2000) and FGES (Chickering, 2003; Ramsey et al., 2017), to learn an equivalence class of causal graphs directly from data and then combine it with the causal Shapley value estimation in an efficient

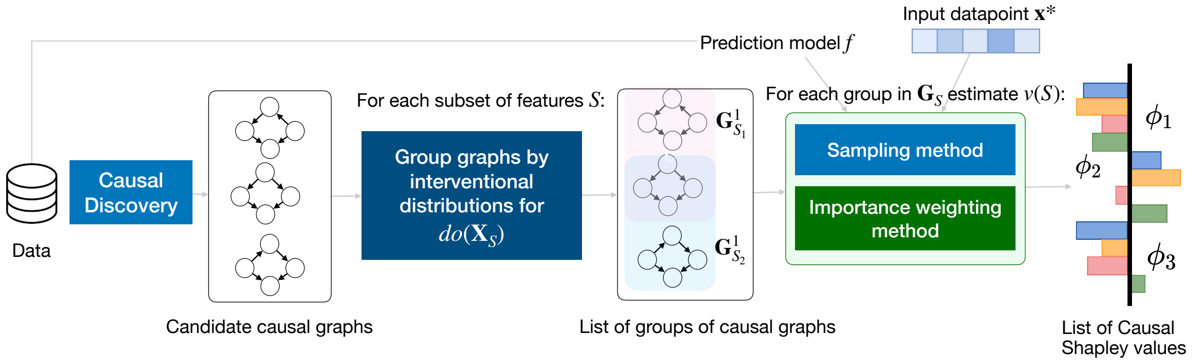

Input datapoint $\mathbf{x}^*$
Prediction model f
Data
Causal Discovery
Candidate causal graphs
For each subset of features S:
Group graphs by interventional distributions for $do(\mathbf{X}_S)$
$\mathbf{G}^1_{S_1}$
$\mathbf{G}^1_{S_2}$
List of groups of causal graphs
For each group in $\mathbf{G}_S$ estimate $v(S)$:
Sampling method
Importance weighting method
ϕ_1
ϕ_2
ϕ_3
List of Causal Shapley values

Figure 1: Overview of our MEC Shapley framework that provides a range of causal Shapley values, describing the causal impact of a feature on an input datapoint, even when the causal graph is unknown. We first apply causal discovery to learn a set of candidate causal graphs directly from the data. For each subset of features S that we consider, we group these causal graphs based on sharing the same interventional distribution for an intervention on $\mathbf{X}_S$ in a list of groups $\mathbf{G}_S$. For each group of graphs, we can then estimate the value function $v(S)$ for the input datapoint $\mathbf{x}^*$ given the prediction model f with one of our two methods, a sampling methods or an importance weighting method, which allows us to provide a list of causal Shapley values that represent a range of explanations for the input datapoint that reflect the different candidate causal relations between the features.

way. This process provides a set of explanations that reflect different possible causal structures, thereby representing causal uncertainty. Our contributions are as follows:

- We introduce a framework for combining causal discovery with causal Shapley values (Heskes et al., 2020), extending their applicability beyond known causal relations. In particular, we address the challenge that causal discovery methods return a set of candidate causal graphs and hence our output is a list of causal Shapley values.
- We propose two methods to efficiently estimate causal Shapley values using only observational data, a sampling method based on simulating interventional data and an importance weighting method. Both of these methods leverage the ID algorithm (Shpitser & Pearl, 2006) to identify the estimand formula for interventional distributions and hence reuse computation across the set of graphs returned by causal discovery methods. Additionally, the importance weighting method only requires the prediction model to be applied on the initial data, thus reducing the computational load.
- We evaluate the performance of our methods using simulated and real-world data. Interestingly, we find that even when the true causal graph or its equivalence class are not learned correctly, the explanations returned by our method tend to be closer on average to the true causal Shapley value than the marginal and conditional Shapley values.

2 Background

In local explainability, the goal is to explain the prediction of a model f for a single input $\mathbf{x}^*$ (Doshi-Velez & Kim, 2017) that is an instantiation of the features $\mathbf{X} = \{X_1, \ldots, X_p\}$ with distribution $P(\mathbf{X})$. A popular method for explaining the influence of each feature $\mathbf{x}_i^*$ on the model output $f(\mathbf{x}^*)$ are Shapley values, (Shapley, 1953), introduced in game theory as a way of distributing a value over a set of p 'players'. The Shapley value of player i for a value function v is:

$$\phi_i(v) = \frac{1}{p} \sum_{S \subseteq [p] \setminus \{i\}} \binom{p-1}{|S|}^{-1} v(S \cup \{i\}) - v(S), \quad i \in [p], \tag{1}$$

where $[p] = \{1, \ldots, p\}$ is the set of all players and each S represents a subset of $[p] \setminus \{i\}$, called a *coalition*. This value can be seen as a weighted average of the marginal contribution of i to every possible coalition of other players S. It can be shown that the contribution of each player sums to the value gained when every player contributes, which ensures that the value produced by the whole group together is distributed over the player's individual Shapley values.

To apply Shapley values in local explainability, we consider each feature a "player". It is not obvious how to choose an appropriate value function v and many options have been proposed (Sundararajan & Najmi, 2020). In particular, we need to choose what happens to features that are in the coalition, denoted by the index set S, and features that are not in the coalition, denoted by the index set $\overline{S}$. A common choice for v is the expectation of our model w.r.t. features that are not in the coalition. If we assume independent features, the value function is the expectation w.r.t. the marginal distribution of the features out of the coalition (Štrumbelj & Kononenko, 2014; Datta et al., 2016; Lundberg & Lee, 2017). The value function is then:

$$v(S) = \mathbb{E}_{P(\mathbf{X}_{\overline{S}})}[f(\mathbf{x}^*_S, \mathbf{X}_{\overline{S}})], \tag{2}$$

where $\mathbf{x}^*_S$ are the features of the input that are in S, while $\mathbf{X}_{\overline{S}}$ represents the features that are not in S over which we take the expectation. We refer to the Shapley values based on this value function as *marginal* Shapley values. Aas et al. (2021) take into account dependence between the features with the following value function, which we refer to as *conditional* Shapley values,

$$v(S) = \mathbb{E}_{P(\mathbf{X}_{\overline{S}}|\mathbf{X}_S=\mathbf{x}^*_S)}[f(\mathbf{x}^*_S, \mathbf{X}_{\overline{S}})]. \tag{3}$$

Heskes et al. (2020) propose *causal* Shapley values, which take into account causal relations between features, represented as a Structural Causal Model (Pearl, 2009; Spirtes et al., 2000), by replacing conditional distributions with *interventional distributions.* This results in choosing v as follows:

$$v(S) = \mathbb{E}_{P(\mathbf{X}_{\overline{S}}|do(\mathbf{X}_S=\mathbf{x}^*_S))}[f(\mathbf{x}^*_S, \mathbf{X}_{\overline{S}})]. \tag{4}$$

Here $P(\mathbf{X}_{\overline{S}}|do(\mathbf{X}_S = \mathbf{x}^*_S))$, represents the distribution after performing an intervention $do(\mathbf{X}_S = \mathbf{x}^*_S)$ that forces $\mathbf{X}_S$ to take the value $\mathbf{x}^*_S$. This distribution is in general not the same as the conditional distribution $P(\mathbf{X}_{\overline{S}}|\mathbf{X}_S = \mathbf{x}^*_S)$ (Pearl, 2009). Heskes et al. (2020) show that with this value function, an indirect effect on the contribution of a feature only occurs when a change in a feature's value has a causal effect on the prediction, which leads to an explanation that more accurately reflects the effect of each feature. We provide a short introduction to causality in App. A.1.

3 MEC Shapley Values

In this paper, we extend the causal Shapley value framework from leveraging known causal relations to the case in which we learn a set of possible causal graphs from observational data, i.e. a Markov Equivalence Class (MEC), using causal discovery methods. This results in a set of causal explanations, one for each Markov-equivalent causal graph that may have generated the data. Considering a set of causal models compounds the computational challenges associated with computing Shapley values. After estimating the MEC, a naive strategy is to iterate over each Directed Acyclic Graph (DAG) in the MEC and compute the corresponding causal Shapley value. However, the size of a MEC grows rapidly with the amount of nodes (He et al., 2015). We therefore want to reuse computation where possible by identifying which DAGs share the same interventional distribution for a subset of features S and by introducing two methods to estimate the interventional expectation: (i) a sampling method and (ii) an importance weighting method. We describe each step of our pipeline in the following.

3.1 Causal discovery and grouping of graphs

Given a dataset $\mathcal{D} = \{\mathbf{x}^{(k)}\}_{k=1}^D$ for features $\{\mathbf{X}\} = \{X_1, \ldots, X_p\}$, our goal is to explain the prediction of a prediction model f for input data point $\mathbf{x}^*$. The first step is learning causal relations between the features from the data $\mathcal{D}$ with causal discovery. We assume that the *causal Markov* and *faithfulness assumptions* hold, i.e. that the conditional independences in the observational data correspond to d-separations in the true causal graph (Spirtes et al., 2000). We assume that we are in a *causally sufficient* setting, in which we do not have unobserved confounding or selection bias and the causal graph is a Directed Acyclic Graph

(DAG). As with most causal discovery methods, our framework relies on these assumptions; when they are violated, results may deviate from the true causal effects. The output of most causal discovery methods is a Markov Equivalence Class (MEC), i.e. all graphs that induce the same conditional independences and dependences as in the observational data. We denote the MEC as $\mathbf{G}$ and define it as a set of candidate graphs $\mathbf{G} = \{G_1, \ldots, G_c\}$. For many sets S, multiple graphs in the MEC share the interventional distribution $P(\mathbf{X}_{\overline{S}}|do(\mathbf{X}_S = \mathbf{x}^*_S))$. By identifying which graphs share this distribution we only need to sample from one of them to estimate $v(S) = \mathbb{E}[f(\mathbf{X}_{\overline{S}}, \mathbf{x}^*_S)|do(\mathbf{X}_S = \mathbf{x}^*_S)]$. So for each intervention target S we rewrite the interventional distributions with the ID algorithm (Shpitser & Pearl, 2006) as an estimand of observational quantities. We group graphs with the same estimand for the intervention on S in a list of set of graphs $\mathbf{G}_S$, where for each group we estimate $v(S)$ once.

3.2 Sampling method

After learning the MEC and grouping the graphs based on their interventional estimands, as described in Sec. 3.1, we can now use a sampling method for estimating causal Shapley values, similar to the method used by Heskes et al. (2020) for a single, known causal graph. For each subset of features S and a graph G_i in each group of $\mathbf{G}_S$, we follow three steps: (i) conditional distribution estimation, (ii) simulation of interventional data, and (iii) estimation of the value function.

First, we consider G_i as a potential causal graph and follow the topological order of G_i to estimate $P_{G_i}(X_j|\mathbf{X}_{\mathrm{Pa}_{\mathrm{G_i}}(\mathrm{j})})$ for each variable X_j, where $\mathbf{X}_{\mathrm{Pa}_{\mathrm{G_i}}(\mathrm{j})}$ are the variables in $\mathbf{X}$ that are the causal parents of $\mathbf{X}_j$ in G_i. We then generate simulated interventional samples from the interventional distribution $P_{G_i}(\mathbf{X}|do(\mathbf{X}_S = \mathbf{x}^*_S))$ that assumes that G_i is the true causal graph. In particular, we set all of the values of $\mathbf{X}_S = \mathbf{x}^*_S$ and use the truncated factorization formula (Eq. 11 in App. A.1) for which we sample each factor in a topological order in G_i as follows:

$$\begin{aligned}\{\hat{\mathbf{x}}^{(k,G_i,S)}\}_{k=1}^{n_{mc}} \sim \prod_{j \notin S} & P_{G_i}(X_j|\mathbf{X}_{\mathrm{Pa}_{G_i}(j)})\delta(\mathbf{X}_S = \mathbf{x}^*_S) \\ &= P_{G_i}(\mathbf{X}|do(\mathbf{X}_S = \mathbf{x}^*_S)),\end{aligned}$$

where δ is the Dirac delta function, forcing $\mathbf{X}_S$ to take the values $\mathbf{x}^*_S$.

We then use the n_{mc} simulated samples in a Monte-Carlo approximation of the value function v for the graph G_i as follows:

$$v(S)^{G_i} = \mathbb{E}[f(\mathbf{x}^*_S, \mathbf{X}_{\overline{S}})|G_i] \approx \frac{1}{n_{\mathrm{mc}}} \sum_{k=1}^{n_{\mathrm{mc}}} f(\hat{\mathbf{x}}^{(k,G_i,S)}).$$

Having estimated each interventional expectation, we combine them using Eq. 1 or, when sampling only a subset of all possible subsets, using Kernel SHAP (Lundberg & Lee, 2017; Aas et al., 2021).

3.3 Importance Weighting (IW) method

The sampling method from Sec. 3.2 may incur high computational cost, since it simulates new datapoints for each graph and intervention, each of which must be evaluated by a potentially expensive model. To improve the computational efficiency, we introduce an importance-weighting method that re-uses a single set of predicted samples for an intervention on $\mathbf{x}_S$ across all DAGs in the MEC. The pseudocode for the method is provided in Alg. 1. While this method avoids having to evaluate new simulated data points, we note that this approach can be susceptible to high variance if the target interventional distribution differs significantly from the observational data, a known limitation of importance sampling that the sampling-based method avoids.

We start by defining the set of subsets of feature that we consider $\mathbf{S}$ similar to KernelSHAP (Lundberg & Lee, 2017) (Lines 1-2). We learn the MEC on the initial data $\mathcal{D}$ using standard causal discovery methods and group the graphs based on their interventional estimands, as described in Sec. 3.1 (Lines 3-9). For each subset of features $S \in \mathbf{S}$ we then compute the value function $v(S)$ for each group of graphs (Lines 10-19), which we describe in detail in the following, and apply KernelSHAP (Lundberg & Lee, 2017) following Aas et al. (2021) to compute the final Shapley values (Lines 20-24).

The core of this algorithm is the computation of the value functions for each group of graphs in an efficient way. The first step is to compute the predictions $f(\mathbf{x}^{(k)}_{\overline{S}}, \mathbf{x}^*_S)$ using the prediction model f and a combination of the initial data $\mathcal{D}$ and the input datapoint $\mathbf{x}^*$ (Line 11). We reweight these predictions based on the interventional distribution in each graph G_i and estimate the expected value $\mathbb{E}_{P_{G_i}(\mathbf{X}_{\overline{S}}|do(\mathbf{X}_S=\mathbf{x}^*_S))}[f(\mathbf{x}^*_S, \mathbf{X}_{\overline{S}})|G_i]$, which is the value function $v(S)$ used in causal Shapley values (Heskes et al., 2020). We rewrite $v(S)$ for a graph G_i as a weighted expectation over samples from the marginal distribution $P(\mathbf{X}_{\overline{S}})$, where $\overline{S}$ are the features not in S:

$$
\begin{aligned}
v^{G_i}(S) &= \mathbb{E}_{P_{G_i}(\mathbf{X}_{\overline{S}}|do(\mathbf{X}_S=\mathbf{x}^*_S))}[f(\mathbf{x}^*_S, \mathbf{X}_{\overline{S}})|G_i] && (5)\\
&= \int f(\mathbf{x}^*_S, \mathbf{X}_{\overline{S}}) P_{G_i}(\mathbf{X}_{\overline{S}}|do(\mathbf{X}_S = \mathbf{x}^*_S)) \frac{P(\mathbf{X}_{\overline{S}})}{P(\mathbf{X}_{\overline{S}})} d\mathbf{X}_{\overline{S}} && (6)\\
&= \mathbb{E}_{P(\mathbf{X}_{\overline{S}})}\left[f(\mathbf{x}^*_S, \mathbf{X}_{\overline{S}}) \frac{P_{G_i}(\mathbf{X}_{\overline{S}}|do(\mathbf{X}_S = \mathbf{x}^*_S))}{P(\mathbf{X}_{\overline{S}})} \right], && (7)
\end{aligned}
$$

where in Eq. 6 we introduce $P(\mathbf{X}_{\overline{S}})$ in the denominator and the numerator.

To approximate this expectation, we first estimate the conditional densities $\hat{P}_{G_i}(X_j|\mathbf{X}_{\mathrm{Pa}_{\mathrm{G_i}}(j)})$ from data $\mathcal{D}$ (Line 14). We then re-weight each prediction $f(\mathbf{x}^{(k)}_{\overline{S}}, \mathbf{x}^*_S)$ for each factorization in a Monte Carlo approximation of Eq. 7 as follows:

$$
v^{G_i}(S) \approx \frac{1}{D} \sum_{k=1}^{D} f(\mathbf{x}^{(k)}_{\overline{S}}, \mathbf{x}^*_S) \frac{\hat{P}_{G_i}(\mathbf{X}_{\overline{S}}|do(\mathbf{X}_S = \mathbf{x}^*_S))}{\hat{P}(\mathbf{X}_{\overline{S}})}, \qquad (8)
$$

where

$$
\hat{P}_{G_i}\left(\mathbf{X}_{\overline{S}}|do(\mathbf{X}_S = \mathbf{x}^*_S)\right) = \prod_{j \in \overline{S}} \hat{P}_{G_i}(X_j|\mathbf{X}_{\mathrm{Pa}_{G_i}(j)\cap\overline{S}}, \mathbf{x}^*_{\mathrm{Pa}_{G_i}(j)\cap S})
$$

following the truncated factorization formula (Eq. 11), where we force all of the variables in S to take the value $\mathbf{x}^*_S$, including the parents of each variable X_j in G_i, represented as $\mathrm{Pa}_{G_i}(j) \cap S$. We compute Eq. 8 for a single graph in each group and assign it to the other graphs in the same group (Lines 15-16). For each graph G_i we use $v(S)^{G_i}$ for each set of features S to compute the Kernel SHAP values (Lundberg & Lee, 2017) (Lines 19-22). We return a list of Kernel SHAP values, one for each DAG in the MEC $\mathbf{G}$ (Line 23).

4 Related work

Connections between causality and explainability have been explored in various works (Beckers, 2022; Karimi et al., 2023). Many of these focus on leveraging causal knowledge to improve interpretability (Khademi & Honavar, 2020; Schwab & Karlen, 2019; Oesterle et al., 2023), but they do not use the framework of Shapley values, which provides desirable axiomatic properties and computational tools for local feature attribution.

Some approaches employ causal discovery for explainability outside the Shapley framework. For example, Sani et al. (2020) use causal discovery for explainability of black-box methods. In contrast to our work, they consider unstructured data such as images and allow for potential latent confounding, but do not quantify causal effects. Similarly, Takahashi et al. (2024) apply causal discovery in the context of counterfactual explanations. These approaches highlight the value of integrating discovery with explanation, but do not provide principled feature attributions based on causal effects.

Several works have combined Shapley values with causal reasoning. This paper builds on causal Shapley values as introduced by Heskes et al. (2020). Closely related is Jung et al. (2022) who adopt the same value

Algorithm 1 Importance weighting method for Estimating MEC Shapley Values

```
Input: Dataset $\mathcal{D} = \{\mathbf{x}^{(k)}\}_{k=1}^{D}$, prediction model $f$, target variable $Y$, data point to explain $\mathbf{x}^*$, number of combinations to consider $n_{\text{comb}} \leq |2^{[p]}|$
Initialize set of subsets of features to consider $\mathbf{S} \leftarrow \{\emptyset, [p]\}$ following (Lundberg & Lee, 2017).
Sample $n_{\text{comb}} - 2$ subsets of the power set $2^{[p]} \setminus \{\emptyset, [p]\}$ with probability equal to kernel weights $k(S) = (p-1)/\left(\binom{p}{|S|}|S|(p-|S|)\right)$ and add them to $\mathbf{S}$.
Learn the MEC $\mathbf{G} = \{G_1, \ldots, G_c\}$ by applying a causal discovery method to $\mathcal{D}$
for each subset $S \in \mathbf{S}$ of features do
    for each graph $G_j \in \mathbf{G}$ do
        Identify estimand formula for $P^{G_j}(\mathbf{X}_{\overline{S}} | do(\mathbf{X}_S = \mathbf{x}_S^*))$ with ID algorithm.
    end for
    Group graphs with same estimand formula for the intervention on $S$ in a set of graphs $\mathbf{G}_S^l$.
end for
for each subset $S \in \mathbf{S}$ of features do
    Compute predictions $\{f(\mathbf{x}_{\overline{S}}^{(k)}, \mathbf{x}_S^*)\}_{k=1}^{D}$ with observational data $\{\mathbf{x}_{\overline{S}}^{(k)}\}_{k=1}^{D}$ and input $\mathbf{x}_S^*$.
    for each group of graphs $\mathbf{G}_S^l$ with the same estimand formula for $P(\mathbf{X}_{\overline{S}} | do(\mathbf{X}_S = \mathbf{x}_S^*))$ do
        Choose any $G_j$ in $\mathbf{G}_S^l$.
        Estimate conditional probabilities $\hat{P}_{G_i}(X_j | \mathbf{X}_{\mathrm{Pa}_{\mathrm{G_i}}(j)})$ from observational data $\mathcal{D}$.
        Calculate value of $v$ for graph $G_j$ and set $S$ with Eq. 8:
            $$v^{G_j}(S) \leftarrow \frac{1}{D} \sum_{k=1}^{D} \frac{\prod_{h \notin S} \hat{P}(\mathbf{x}_h^{(k)} | \mathbf{x}_{\mathrm{Pa}_{G_j}(h) \cap \overline{S}}^{(k)}, \mathbf{x}_{\mathrm{Pa}_{G_j}(h) \cap S}^*)}{\hat{P}(\mathbf{x}_{\overline{S}}^{(k)})} f(\mathbf{x}_{\overline{S}}^{(k)}, \mathbf{x}_S^*)$$
        Store the value of $v^{G_j}(S)$ for the value functions of all other graphs in $\mathbf{G}_S^l$.
    end for
end for
for each graph $G_j \in \mathbf{G}$ do
    Collect all value functions for each subset $S \subseteq \mathbf{S}$ in a vector $\mathbf{v}^{G_j} \leftarrow (\mathbf{v}^{G_j}(\emptyset), \ldots, \mathbf{v}^{G_j}([p]))$.
    Compute the Kernel SHAP value $\phi^{G_j}$ for $\mathbf{S}$ with $\mathbf{v}^{G_j}$ following (Lundberg & Lee, 2017).
end for
return List of Kernel SHAP values $\{\phi^{G_j}\}_{G_j \in \mathbf{G}}$ for each DAG in the MEC $\mathbf{G}$.
```

function in the context of causal contribution analysis, and propose a weighted estimator to estimate the value functions in the case of discrete data. Janzing et al. (2020) take a different perspective, arguing that since we explain an algorithm rather than a data-generating process, the features should be seen as independent, making interventional (causal) Shapley values equivalent to marginal Shapley values. Frye et al. (2020) introduce asymmetric Shapley values, incorporating causal knowledge into explanations by weighting subsets according to their agreement with a partial causal ordering. While not intervention-based, their method can be combined with causal Shapley values in chain graphs, but does not extend to full causal structures or discovery. Budhathoki et al. (2022) and Strobl & Lasko (2022) also integrate causality into Shapley values, but in the context of root cause analysis, assuming known casual graphs. An exception is the work by Strobl & Lasko (2022) that learn the causal graph, but assume a linear non-Gaussian model, a more specific setting than ours which leads to a single causal graph instead of a MEC. Concurrently to our work, Ng et al. (2025) propose Causal SHAP, which integrates the PC and IDA algorithms into SHAP to yield a single attribution vector corrected by estimated causal strengths. While effective when a single discovered graph is assumed, their approach does not account for the uncertainty inherent in causal discovery. In contrast, our framework is agnostic to the specific causal discovery method and explicitly considers the entire Markov equivalence class, producing a range of causal Shapley values that reflect multiple plausible causal structures.

5 Experiments

We evaluate our methods across synthetic datasets and two real-world datasets. One of the challenges when evaluating methods for explainability is that we lack agreed-upon metrics for assessing the quality of explanations. We therefore focus on a tangible, well-defined objective: evaluating how well and how efficiently our methods approximate the theoretical causal Shapley values derived from a ground-truth causal graph. These values provide a principled framework for attributing a model's prediction to its input features based on their causal effects.

In the synthetic datasets we have access to the true interventional distributions, allowing us to compare the output of our methods to the true causal Shapley values. We evaluate in a linear Gaussian and a non-linear setting. In addition to comparing with the ground truth causal Shapley value defined by the true interventional distributions, we also compare the outcome of our method with the marginal and conditional Shapley values, which do not take any causal relations into account. To implement our method we adapt the `shapr` package by Aas et al. (2021). For causal discovery, we used the `pcalg` package (Kalisch et al., 2012) and the `Tetrad` toolbox (Ramsey et al., 2018). For identifying the interventional estimand formulas we use the `causaleffect` package (Tikka & Karvanen, 2017a;b). For prediction we use XGBoost models (Chen & Guestrin, 2016). We provide the code to reproduce our experiments in the supplementary material.

5.1 Synthetic data

We generate data for two settings, linear Gaussian and non-linear causal models. For each setting, we consider $n_{\text{scm}} = 40$ randomly generated causal models for which we generate synthetic data as follows.

For each model we first sample an Erdos-Renyi graph of size $p = 5, 10, 15$ where we specify the expected number of neighbors $d = 1, 2, 3$ for each node. We add our 'dependent' node Y, the target variable for our prediction model, for which we specify the minimum number of incoming edges ($m_{\min} = 2, 3, 5$) and the expected number of incoming edges ($\bar{m} = 3, 6, 9$).

In the linear Gaussian setting, nodes are parametrized as $X_i = \sum_{j \in \text{Pa}_i} w_{ij} X_j + \epsilon_i$ where $w_{ij} \sim U((-2, -0.5) \cup (0.5, 2))$ and $\epsilon_i = \mathcal{N}(0, 1)$. For the non-linear setting we simulate data with the equations $X_i = w_2^i \sigma(\sum_{j \in \text{Pa}_i} w_1^{ij} X_j) + \epsilon_i$ where σ denotes the sigmoid function, $w_1^{ij} \sim U((-1.5, -0.5) \cup (0.5, 1.5))$, $w_2^i \sim U((-3, 1) \cup (1, 3))$, and ϵ_i follows either a $\mathcal{N}(0, 1)$ or a $U(-1, 1)$ distribution with equal probability. We apply causal discovery to standardized data. For each causal model we simulate $n_{\text{test}} = 40$ data points $\mathbf{x}^*$ for which we compute the Shapley values.

Methods. For each model, following Heskes et al. (2020) we train an XGBoost model (Chen & Guestrin, 2016) on a subset of $n_{\text{train}} = 10K$ samples for 100 rounds $n_{\text{test}} = 40$ data points $\mathbf{x}^*$ which are unseen by the model. We compare the results of our methods with *Marginal* (Lundberg & Lee, 2017) and *Conditional* Shapley values (Aas et al., 2021).

As causal discovery methods, we use the PC algorithm (Spirtes et al., 2000) and the FGES algorithm (Ramsey et al., 2017). We call the combination of our method with PC the *PC MEC* Shapley method, while the combination with FGES is the *FGES MEC* Shapley method. We compare the results of these algorithms with a causal discovery oracle, which we call the *Oracle MEC* method. Crucially, this is a partial oracle: it is given the true underlying graph structure (and thus the true MEC), but it must still estimate the required conditional distributions from the same n_{obs} samples as the other methods. For all methods, we consider a Sampling version, described in Sec. 3.2, and an IW version, described in Sec. 3.3.

We apply the PC algorithm to the linear data using partial correlation tests and for non-linear data we apply kernel-based conditional independence tests (Zhang et al., 2011), both with significance threshold $\alpha = 0.05$. We apply FGES in both settings using the BIC criterion. For the conditional and MEC Shapley methods, we model conditional distributions using a Gaussian parametric assumption. We approximate marginal Shapley values using samples from empirical marginal distribution $p^*(\mathbf{X}_{\overline{S}})$. By default we use $n_{\text{cd}} = 1000$ observational samples for PC and FGES, $n_{\text{obs}} = 1000$ samples to estimate conditional expectations and $n_{\text{mc}} = 1000$ samples to estimate the Monte Carlo approximation. For $p = 5, 10$ we consider the number

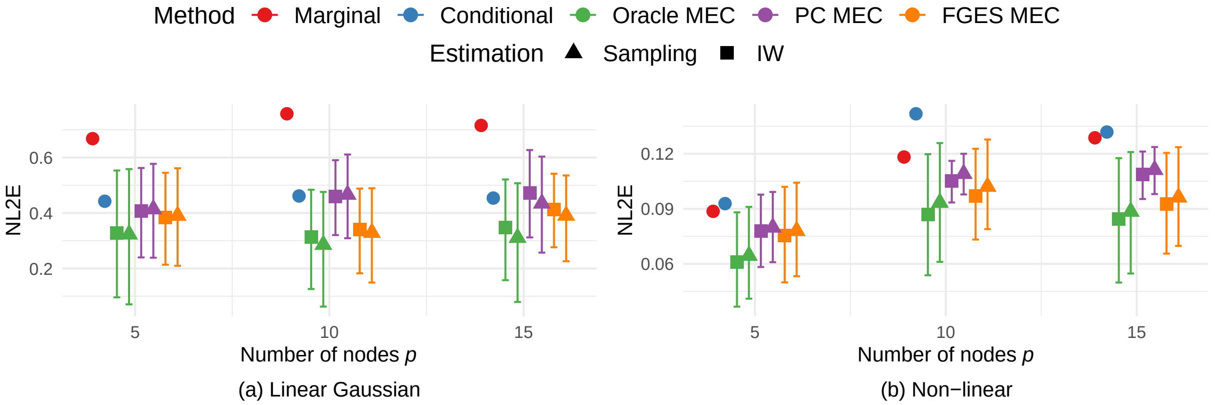


Figure 2: Normalized L2 error between the ground truth causal Shapley value and the Shapley values returned by each method for different parameters: (a) linear Gaussian setting with varying number of features p, (b) nonlinear Gaussian setting with varying number of features p, (c) linear Gaussian setting with varying number of expected neighbors d, and (d) linear Gaussian setting with varying expected number of nodes influencing Y $\bar{m}$. For all MEC Shapley methods we plot the NL2E for the whole range of explanations.

of subsets $n_{\text{comb}} = 2^p$, meaning that we use the full set of of combinations S. As the set of combinations increases exponentially with the amount of nodes, for $p = 15$ we set $n_{\text{comb}} = 8192$ for the ground truth and $n_{\text{comb}} = 4096$ for the other methods. We provide ablations for these parameters in App. A.2.

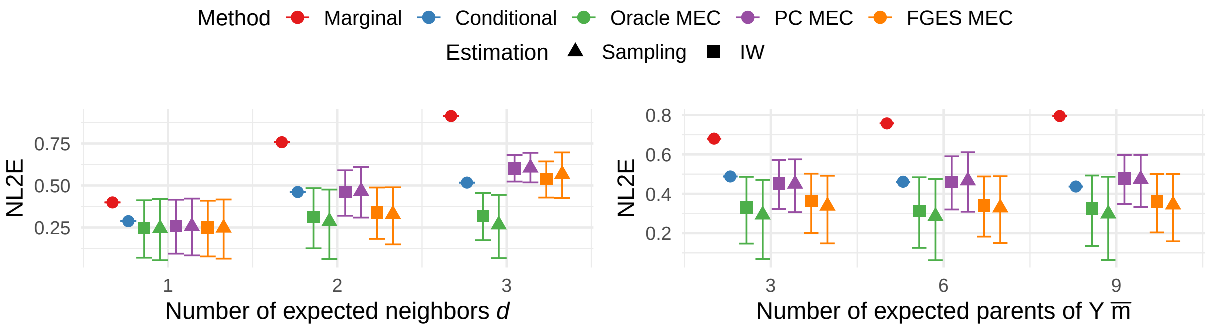
Metrics. We evaluate how close explanations from different methods in terms of their agreement with the ground truth causal Shapley values $\phi^{G^*}_{\text{true}}$, which we compute using the ground truth causal graph G^* and sampling from the true interventional distributions with $n^*_{\text{mc}} = 4000$ Monte Carlo samples for each datapoint $\mathbf{x}^*$.

We then compare on the L2 error between the Shapley value returned by each method and the ground truth causal Shapley value, normalized based on the its value for the datapoint to allow comparison across different data points, defining the normalized L2 error as follows:

$$NL2E_{\text{method}} = \frac{\|\phi^{G^*}_{\text{true}} - \phi_{\text{method}}\|_2}{\|\phi^{G^*}_{\text{true}}\|_2} \tag{9}$$

where $\phi^{G^*}_{\text{true}}$ refers to the ground truth causal Shapley value for $\mathbf{x}^*$ and ϕ_{method} is the Shapley value returned by each method. Since MEC Shapley values methods provide a list of possible explanations, one for each causal graph in the MEC $\mathbf{G} = \{G_1, \ldots, G_c\}$, in the plots we show the NL2E for the range of explanations.

Results. We report the results for all the synthetic data settings in the following. We provide statistics on these numbers in App. A.2. Fig. 2a shows the NL2E between each method and the ground truth causal Shapley as a function of the number of nodes $p = 5, 10, 15$ for linear Gaussian causal models for $d = 2$ and $\bar{m} = 3, 6, 9$. For all MEC Shapley methods we plot the NL2E for whole range of explanations. We observe that as expected the estimated causal Shapley values resulting from the oracle MEC are closest to the ground truth causal Shapley values, but they still often do not include the ground truth causal Shapley values, because the Oracle MEC uses limited data to estimate the conditional distribution. We can see that the average NL2E for all the MEC Shapley methods is still notably lower than the baselines. For the oracle MEC and the FGES MEC we can see that even the worst explanations in the range can have comparable or better performances than the baselines. As expected, the Oracle MEC method consistently has the lowest NL2E. When comparing the performance of FGES and PC, we observe that the resulting explanation quality (NL2E) does mostly, but not always correlate directly with the global graph structure similarity (SHD). For instance, in the linear setting with 16 nodes (p=16), FGES yields explanations with lower error than PC (Fig. 2a), despite having a slightly higher average SHD (Appendix Table 3).



(a) Linear Gaussian

(b) Linear Gaussian

Figure 3: Normalized L2 error between the ground truth causal Shapley value and the Shapley values returned by each method for different parameters: (a) linear Gaussian setting with varying number of expected neighbors d, and (b) linear Gaussian setting with varying expected number of nodes influencing Y $\bar{m}$. For all MEC Shapley methods we plot the NL2E for the whole range of explanations.

For non-linear causal models we can see in Fig. 2b that all our methods provide better explanations on average, and that in some cases, even the worst explanation in our range is better than the baselines. Finally, we see a slight difference between Sampling and IW, with Sampling outperforming IW. Surprisingly, here the marginal Shapley values perform better relative to the true causal Shapley values than the conditional ones.

Fig. 3a shows the NL2E between each method and the ground truth causal Shapley as a function of the expected number of neighbors $d = 1, 2, 3$ for linear Gaussian models with $p = 10$ and $\bar{m} = 6$. As expected the oracle MEC Shapley is closest to the ground truth, and is not significantly affected by the expected number of neighbors, which is as we can see for $d = 3$, not the case for PC and FGES MEC, but they on average still show comparable or closer agreement than the baselines. As the expected number of neighbors d increases, the baselines perform comparatively worse. The difference between Sampling and IW is again minimal. Fig. 3b shows that in the linear setting $\bar{m}$, the expected number of parents, has no strong effect on performance.

In App. A.2 Fig. 7 we show how the methods compare when we consider *only* the cases where PC and FGES output an incorrect CPDAG. We can see that overall, performance is not strongly affected. In almost all settings the average explanation is closer than the baselines, and in many cases this holds even for the worst explanation in a range. We show the mean and standard errors for these results in Fig. 8, showing that the standard error is negligible. We consider d and $\bar{m}$ for the non-linear case in Fig. 9, where we see that in most cases the worst explanation by both PC and FGES still outperforms the baselines. We provide ablations in terms of number of Monte Carlo samples n_{mc} in Fig. 10, which do not seem to influence the results substantially. We also vary the number of samples to estimate the conditional distributions n_{obs} in Fig. 11, the number of combinations used to compute the Shapley values n_{comb} in Fig. 12 and the number of samples used to estimate the MEC n_{cd} in Fig. 13. As expected all methods perform slightly better with more samples and higher numbers of combinations, but the differences between the baselines and the MEC Shapley methods are qualitatively similar as what presented in the previous plots. In App.A.2.9 we show an analysis of the learned MECs in all settings, reporting average sizes for PC and FGES and the quality of the CPDAGs compare to the oracle CPDAGs as measured by the Structural Hamming Distance (SHD).

Figure 4 evaluates the average time to compute the Shapley values for 40 data points, comparing the FGES Sample method against the marginal and conditional baselines as a function of the number of DAGs in the learned MEC. We report per-SCM computation times (scatter points), and overlay LOESS-smoothed curves to highlight the overall trend. The top panel illustrates the absolute computation time. As expected, our full method is more computationally expensive than the baselines, as it must compute explanations for multiple DAGs. Furthermore, it incurs a significant, but fixed, one-time overhead for the ID algorithm to identify opportunities for amortization (`Total w/ ID`). Crucially, however, we observe empirically that the total runtime scales favorably, growing sub-linearly with the MEC size, relative to a naive per-DAG iteration baseline, which would scale linearly with the amount of DAGs in the MEC. This suggests the

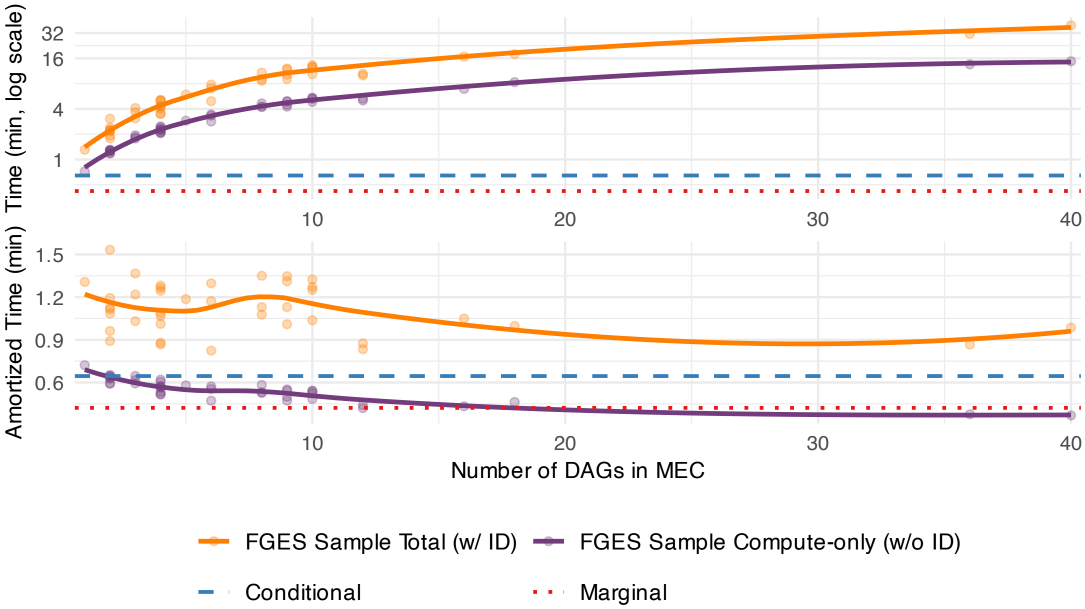


Figure 4: Breakdown of computation time for 40 SCMs (scatter points) in the linear-Gaussian regime. (top) The absolute time for our method (solid lines) compared to baselines (dashed/dotted). (bottom) The amortized time per explanation, showing that our method becomes more efficient than the conditional baseline as the number of DAGs in the MEC increases. The time measured is the computation time (total or amortized) for computing the Shapley values for 40 datapoints. We plot LOESS-smoothed curves to highlight the trend. The plots for the Oracle and PC MECs look similar.

amortization identified by the ID step can materially reduce average cost per explanation as the MEC size grows, though this is an empirical observation rather than a complexity bound. The plot also decomposes the cost, revealing that while the ID algorithm is a major component, the core explanation cost (`Compute-only w/o ID`) scales even more efficiently. The bottom panel reveals the practical benefits of this structure by showing the *amortized time per DAG*. As the MEC size increases, this amortized cost continues to fall and begins to flatten, demonstrating that the marginal cost of explaining an additional causally equivalent graph becomes negligible. Even when including the initial ID overhead, the true average cost per explanation (`Amortized Total w/ ID`) consistently decreases, highlighting the power of amortizing computation.

In summary, our framework presents a computational trade-off: a higher, fixed initial cost to identify shared causal structures is exchanged for per-explanation efficiency and scalability as the causal uncertainty (i.e., MEC size) increases.

5.2 Real-world data

We apply our methods to the Alzheimer's disease dataset obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (`http://adni.loni.usc.edu`) and to the bike rental dataset from (Fanaee-T & Gama, 2014), both used in Heskes et al. (2020). For the Alzheimer's dataset we consider causal Shapley values based on a "gold standard graph" from expert knowledge (Shen et al., 2020) as ground truth, where we trained a prediction model to predict the probability of Alzheimer's diagnosis.

Fig. 5 shows how the outcomes of the different algorithms compare to the ground truth Shapley values, as well as the baselines. The output of both FGES and PC is a MEC with a single graph, leading to a single causal Shapley value for each feature. We see that the FGES MEC methods are closer to the ground truth causal Shapley values than the baselines, especially when using the sampling method. The PC MEC methods are comparable to marginal and conditional Shapley values. This result highlights the critical interplay between the causal discovery and explanation stages of our pipeline. As shown in Appendix Fig. 15, the CPDAG

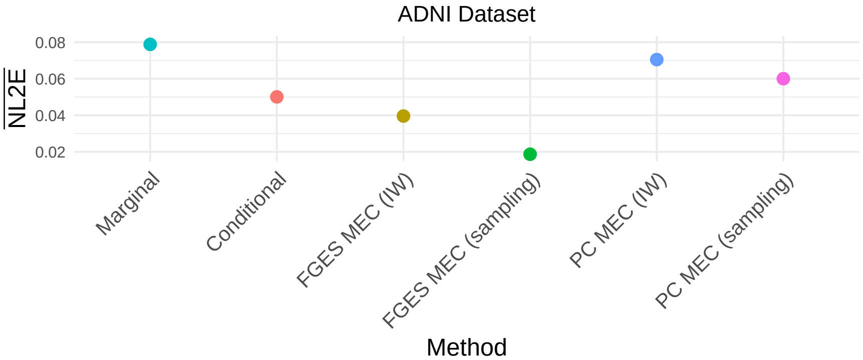


Figure 5: Average NL2E w.r.t the ground truth generated using the “gold standard” graph from Shen et al. (2020) on the ADNI dataset. Standard errors are too small to be visible.

learned by PC incorrectly orients the edge between FDG and ABETA. This error in graph structure likely explains why the subsequent PC MEC Shapley values are less aligned with the ground truth, as the method is reasoning over a flawed causal model. This suggests that the benefit of our framework is directly tied to the ability to learn a reasonably accurate causal structure. When the discovery step is successful (as with FGES), the resulting explanations show notable improvement.

In Fig. 6 we look at the Shapley values for predictions of bike rental counts on two different dates with approximately equal temperature. Here the aim is a qualitative illustration, in line with Heskes et al. (2020), showing how different Shapley formulations can yield distinct explanations for comparable prediction scenarios. While marginal Shapley values provide more or less the same explanation for both days, focusing mostly on the temperature variable, we can see that causal and PC Shapley values assign credit to both season and temperature, resulting in different explanations for the two days. In App. A.3 we show additional sina plots for both datasets that show how the features get assigned different levels of importance by each method, in addition to providing more details on the real-world experiments.

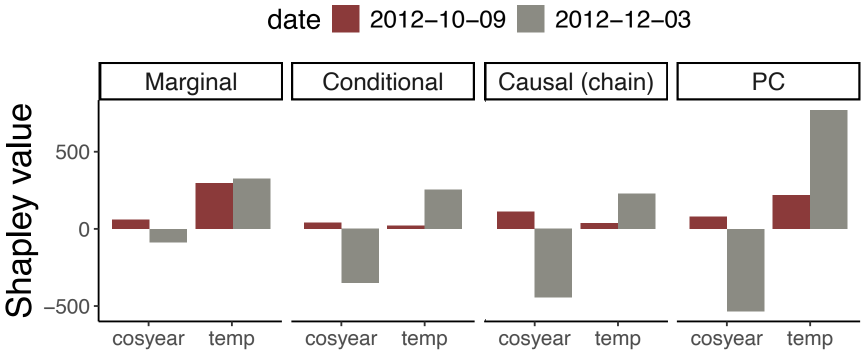


Figure 6: Bar plot for the features *cosyear* and *temp* of the bike rental dataset for two days in October and December with approximately the same temperature. We show the Causal chain Shapley values (Heskes et al., 2020), along with marginal, conditional, and PC Shapley values. FGES Shapley values are similar to PC and have been omitted.

6 Conclusion and limitations

In this paper, we combined causal discovery with causal Shapley values to provide a more nuanced explanation of predictive models, particularly when the causal graph among features is not known. Our experimental results show that our methods provide explanations that are more closely aligned with the true causal impacts of features on predictions, compared to traditional Shapley value approaches that do not incorporate causal reasoning at a limited computational cost. Our results suggest that even when the causal graph is not in the set of learned graphs, the explanations our methods output are often closer to the ground truth causal Shapley values than other methods.

While our approach marks a step forward, several limitations need to be acknowledged. First, our claims are limited to the accuracy and computational efficiency with which we can approximate the true underlying causal Shapley values. Whether causal Shapley values offer a better explanation than those provided by other methods requires interpretation, and will depend on the particular analysis use case under consideration. Since our method returns a set of possible explanations (one per DAG in the MEC), practitioners should interpret these as reflecting causal uncertainty rather than a single definitive attribution.

A computational limitation is the scalability of our pipeline in terms of graph size. In terms of accuracy, we are dependent on the causal discovery methods that is employed, which can be inaccurate under wrong parametric assumptions or for small data samples. Similarly, the accuracy of both the Sampling and IW method depend on the correctness of the estimated conditional distributions. Our methods also assume causal sufficiency, i.e. no unmeasured confounders or selection bias, which may not always hold in real-world settings.

References

Kjersti Aas, Martin Jullum, and Anders Løland. Explaining individual predictions when features are dependent: More accurate approximations to Shapley values. *Artificial Intelligence*, 298:103502, 2021. ISSN 0004-3702. doi: 10.1016/j.artint.2021.103502.

Sander Beckers. Causal explanations and XAI. In Bernhard Schölkopf, Caroline Uhler, and Kun Zhang (eds.), *Proceedings of the First Conference on Causal Learning and Reasoning*, volume 177 of *Proceedings of Machine Learning Research*, pp. 90–109. PMLR, April 2022.

Kailash Budhathoki, Lenon Minorics, Patrick Bloebaum, and Dominik Janzing. Causal structure-based root cause analysis of outliers. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 2357–2369. PMLR, July 2022.

Diogo V. Carvalho, Eduardo M. Pereira, and Jaime S. Cardoso. Machine learning interpretability: A survey on methods and metrics. *Electronicsweek*, 8(832), 2019. ISSN 2079-9292. doi: 10.3390/electronics8080832.

Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Kdd '16, pp. 785–794, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4232-2. doi: 10.1145/2939672.2939785.

David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3(null):507–554, March 2003. ISSN 1532-4435. doi: 10.1162/153244303321897717.

Diego Colombo and Marloes H. Maathuis. Order-independent constraint-based causal structure learning. *Journal of Machine Learning Research*, 15(116):3921–3962, 2014.

Gabor Csardi and Tamas Nepusz. The igraph software package for complex network research. *InterJournal*, Complex Systems:1695, 2006.

Anupam Datta, Shayak Sen, and Yair Zick. Algorithmic Transparency via Quantitative Input Influence: Theory and Experiments with Learning Systems. In *2016 IEEE Symposium on Security and Privacy (SP)*, pp. 598–617, San Jose, CA, May 2016. IEEE. ISBN 978-1-5090-0824-7. doi: 10.1109/SP.2016.42.

Finale Doshi-Velez and Been Kim. Towards A Rigorous Science of Interpretable Machine Learning, March 2017.

Hadi Fanaee-T and João Gama. Event labeling combining ensemble detectors and background knowledge. *Progress in Artificial Intelligence*, 2:113–127, June 2014. doi: 10.1007/s13748-013-0040-3.

Matteo Fasiolo. *An Introduction to Mvnfast. R Package Version 0.2.8.*, 2014.

Christopher Frye, Colin Rowat, and Ilya Feige. Asymmetric Shapley values: Incorporating causal knowledge into model-agnostic explainability. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 978-1-71382-954-6.

Christopher Frye, Damien de Mijolla, Tom Begley, Laurence Cowton, Megan Stanley, and Ilya Feige. Shapley explainability on the data manifold. In *International Conference on Learning Representations*, 2021.

Yangbo He, Jinzhu Jia, and Bin Yu. Counting and exploring sizes of Markov equivalence classes of directed acyclic graphs. *Journal of Machine Learning Research*, 16(1):2589–2609, January 2015. ISSN 1532-4435.

Tom Heskes, Ioan Gabriel Bucur, Evi Sijben, and Tom Claassen. Causal shapley values: Exploiting causal knowledge to explain individual predictions of complex models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 978-1-71382-954-6.

Dominik Janzing, Lenon Minorics, and Patrick Bloebaum. Feature relevance quantification in explainable AI: A causal problem. In Silvia Chiappa and Roberto Calandra (eds.), *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 2907–2916. PMLR, 2020.

Yonghan Jung, Shiva Kasiviswanathan, Jin Tian, Dominik Janzing, Patrick Bloebaum, and Elias Bareinboim. On measuring causal contributions via do-interventions. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 10476–10501. PMLR, July 2022.

Markus Kalisch, Martin Mächler, Diego Colombo, Marloes H Maathuis, and Peter Bühlmann. Causal inference using graphical models with the R package pcalg. *Journal of Statistical Software*, 47(11):1–26, 2012.

Amir-Hossein Karimi, Krikamol Muandet, Simon Kornblith, Bernhard Schölkopf, and Been Kim. On the relationship between explanation and prediction: A causal view. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23, , Honolulu, Hawaii, USA,, 2023. JMLR.org.

Aria Khademi and Vasant Honavar. A Causal Lens for Peeking into Black Box Predictive Models: Predictive Model Interpretation via Causal Attribution, August 2020.

Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 4768–4777, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 978-1-5108-6096-4.

Woon Yee Ng, Li Rong Wang, Siyuan Liu, and Xiuyi Fan. Causal shap: Feature attribution with dependency awareness through causal discovery, 2025. URL https://arxiv.org/abs/2509.00846.

Michael Oesterle, Patrick Blöbaum, Atalanti A. Mastakouri, and Elke Kirschbaum. Beyond Single-Feature Importance with ICECREAM, July 2023.

Judea Pearl. *Causality: Models, Reasoning, and Inference.* 2009.

Joseph Ramsey, Madelyn Glymour, Ruben Sanchez-Romero, and Clark Glymour. A million variables and more: The Fast Greedy Equivalence Search algorithm for learning high-dimensional graphical causal models, with an application to functional magnetic resonance images. *International Journal of Data Science and Analytics*, 3(2):121–129, March 2017. ISSN 2364-415X, 2364-4168. doi: 10.1007/s41060-016-0032-z.

Joseph D Ramsey, Kun Zhang, Madelyn Glymour, Ruben Sanchez Romero, Biwei Huang, Savini Samarasinghe, Elizabeth A Barnes, and Clark Glymour. TETRAD - A TOOLBOX FOR CAUSAL DISCOVERY. 2018.

Numair Sani, Daniel Malinsky, and Ilya Shpitser. Explaining the behavior of black-box prediction algorithms with causal learning. *CoRR*, abs/2006.02482, 2020. URL https://arxiv.org/abs/2006.02482.

Patrick Schwab and Walter Karlen. CXPlain: Causal explanations for model interpretation under uncertainty. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Lloyd S Shapley. A Value for n-Person Games. In Harold W. Kuhn and Albert W. Tucker (eds.), *Contributions to the Theory of Games II*, pp. 307–317. Princeton University Press, Princeton, 1953.

Xinpeng Shen, Sisi Ma, Prashanthi Vemuri, Gyorgy Simon, and Alzheimer's Disease Neuroimaging Initiative. Challenges and opportunities with causal discovery algorithms: Application to Alzheimer's pathophysiology. *Scientific Reports*, 10(1):2975, February 2020.

Ilya Shpitser and Judea Pearl. Identification of conditional interventional distributions. In *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*, UAI'06, pp. 437–444, Arlington, Virginia, USA, 2006. AUAI Press. ISBN 0-9749039-2-2.

Peter Spirtes, Clark N. Glymour, and Richard Scheines. *Causation, Prediction, and Search.* Adaptive Computation and Machine Learning. MIT Press, Cambridge, Mass, 2nd ed edition, 2000. ISBN 978-0-262-19440-2.

Eric V. Strobl and Thomas A. Lasko. Identifying patient-specific root causes of disease. In *Proceedings of the 13th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, pp. 1–10, Northbrook Illinois, August 2022. ACM. ISBN 978-1-4503-9386-7. doi: 10.1145/3535508.3545553.

Erik Štrumbelj and Igor Kononenko. Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems*, 41(3):647–665, December 2014. ISSN 0219-1377, 0219-3116. doi: 10.1007/s10115-013-0679-x.

Mukund Sundararajan and Amir Najmi. The many Shapley values for model explanation. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Daisuke Takahashi, Shohei Shimizu, and Takuma Tanaka. Counterfactual explanations of black-box machine learning models using causal discovery with applications to credit rating, 2024. URL https://arxiv.org/abs/2402.02678.

Santtu Tikka and Juha Karvanen. Identifying causal effects with the R package causaleffect. *Journal of Statistical Software*, 76(12):1–30, 2017a. doi: 10.18637/jss.v076.i12.

Santtu Tikka and Juha Karvanen. Simplifying probabilistic expressions in causal inference. *Journal of Machine Learning Research*, 18(36):1–30, 2017b.

Hadley Wickham, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemund, Alex Hayes, Lionel Henry, Jim Hester, Max Kuhn, Thomas Lin Pedersen, Evan Miller, Stephan Milton Bache, Kirill Müller, Jeroen Ooms, David Robinson, Dana Paige Seidel, Vitalie Spinu, Kohske Takahashi, Davis Vaughan, Claus Wilke, Kara Woo, and Hiroaki Yutani. Welcome to the tidyverse. *Journal of Open Source Software*, 4(43):1686, 2019. doi: 10.21105/joss.01686.

Kun Zhang, Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. Kernel-based conditional independence test and application in causal discovery. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, UAI'11, pp. 804–813, Arlington, Virginia, USA, 2011. AUAI Press. ISBN 978-0-9749039-7-2.

Yu Zhang, Peter Tiňo, Aleš Leonardis, and Ke Tang. A survey on neural network interpretability. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 5:726–742, 2020.

Jianlong Zhou, Amir H. Gandomi, Fang Chen, and Andreas Holzinger. Evaluating the quality of machine learning explanations: A survey on methods and metrics. *Electronicsweek*, 10(593), 2021. ISSN 2079-9292. doi: 10.3390/electronics10050593.

A Appendix

A.1 Causality background

We introduce a short background in causality. We assume that the data-generating process can be modelled using a Structural Causal Model (SCM) (Pearl, 2009; Spirtes et al., 2000), such that each variable X_i for $i \in [p]$ has an assignment,

$$X_i \leftarrow f_i(\mathbf{X}_{\mathrm{Pa}(i)}, U_i), \quad i = 1, \dots, p, \tag{10}$$

where $\mathbf{X}_{\mathrm{Pa}(i)}$ are the *parents*, i.e. the variables that cause X_i directly, and U_i is a noise variable, with each U_i being independent of the other noise variables. This SCM induces a joint probability distribution $P(\mathbf{X}) = \prod_{i=1}^{p} P(X_i|\mathbf{X}_{\mathrm{Pa}(i)})$, as well as a graph G. If the graph contains no cycles, we call it a directed acyclic graph (DAG). A do-intervention, denoted by $do(\mathbf{X}_S = \mathbf{x}_S)$, is a replacement of the causal mechanism for the variables $\mathbf{X}_S$ with constant $\mathbf{x}_S$. This results in a new distribution $P(\mathbf{X}|do(\mathbf{X}_S = \mathbf{x}_S))$ that factorizes according to the truncated factorization formula:

$$P(\mathbf{X}|do(\mathbf{X}_S = \mathbf{x}_S) = \prod_{j \notin S} p(X_j|\mathbf{X}_{\mathrm{Pa}(j)})\delta(\mathbf{X}_S = \mathbf{x}_S), \tag{11}$$

where δ is a Dirac delta function that forces the value of $\mathbf{X}_S$ to $\mathbf{x}_S$.

A.2 Complete experiments - synthetic data

Below we present the full set of results for all experiments performed. We investigate what happens when we only consider CPDAGs that were learned incorrectly, the range provided by each MEC Shapley, and ablations for the hyperparameters n_{mc}, n_{obs}, n_{cd}, n_{comb}.

A.2.1 Performance on incorrectly learned CPDAGS

In Fig. 7 we report the same metrics as in Fig. 2 in the main text, but now including only models for which neither PC and FGES learned the correct CPDAG, i.e. the oracle CPDAG. For each setting we report on how many models out of the original 40 are learned incorrectly: for Fig. 7a the number of models analyzed are 12, 18, and 26 for $p = 5, 10, 15$. For Fig. 7b, the numbers are 16, 32, and 36 for $p = 5, 10, 15$. For Fig. 7c, we have 5, 18, 30 models for $d = 1, 2, 3$, Overall this ranges from a third for the simpler settings (linear, small number of nodes or sparse) to half for the default setting and almost all for the non-linear case. We see that in almost all cases results are not strongly affected by excluding the models that were learned perfectly. In almost all settings the average explanation is better than the marginal/conditional Shapley values, and in many cases even the worst explanation is comparable to, and sometimes better than, the baselines.

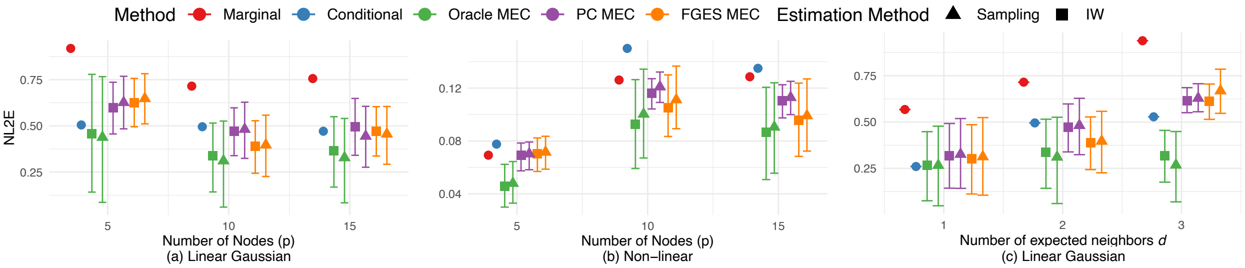


Figure 7: Normalized L2 error between the ground truth causal Shapley value and the Shapley values returned by each method for different parameters, where only models are included where *both* PC and FGES learned an incorrect CPDAG, i.e. a CPDAG not equivalent to the oracle CPDAG. The amount of models included is (a) 12, 18, 26 for $p = 5, 10, 15$ in the linear Gaussian setting, (b) 16, 32, 36 for $p = 5, 10, 15$ in the non-linear setting, (c) 5, 18, 30 for $d = 1, 2, 3$ in the linear Gaussian setting.

A.2.2 Mean and standard error plots for Fig. 2

In Fig. 8 we report the mean and standard error of the NL2E taken over all datapoints of all causal models, for (a) the linear Gaussian setting where we vary p, (b) the non-linear setting with varying p, (c) the linear Gaussian setting with varying d, and (d) the linear Gaussian setting with varying $\bar{m}$. The means that we show here are the same means as in Fig. 2. The error bars are in most cases too small to see, which is the case also for the other experiments. Since we already plot a range of outputs for our methods, for clarity we decided to omit them in the other plots.

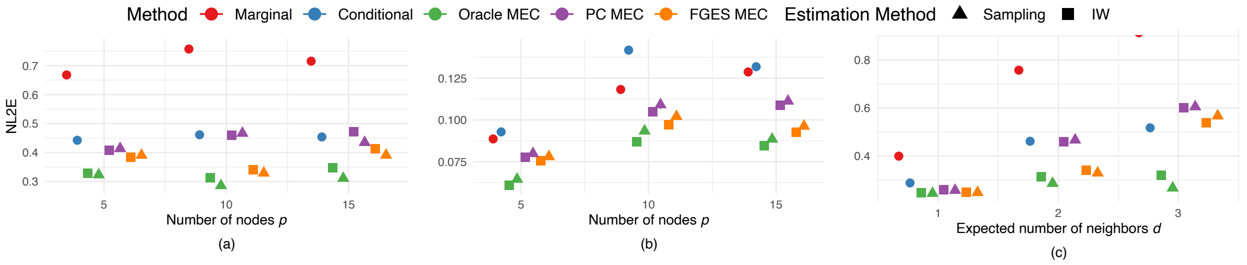


Figure 8: Mean and standard error for NL2E between the ground truth causal Shapley value and the Shapley values returned by each method for different parameters and settings: (a) linear Gaussian setting with varying number of features p, (b) nonlinear Gaussian setting with varying number of features p, (c) linear Gaussian setting with varying number of expected neighbors d, and (d) linear Gaussian setting with varying expected number of nodes influencing Y $\bar{m}$. For all MEC Shapley methods we plot the average NL2E over all the explanations.

A.2.3 Varying the amount of neighbors and parents

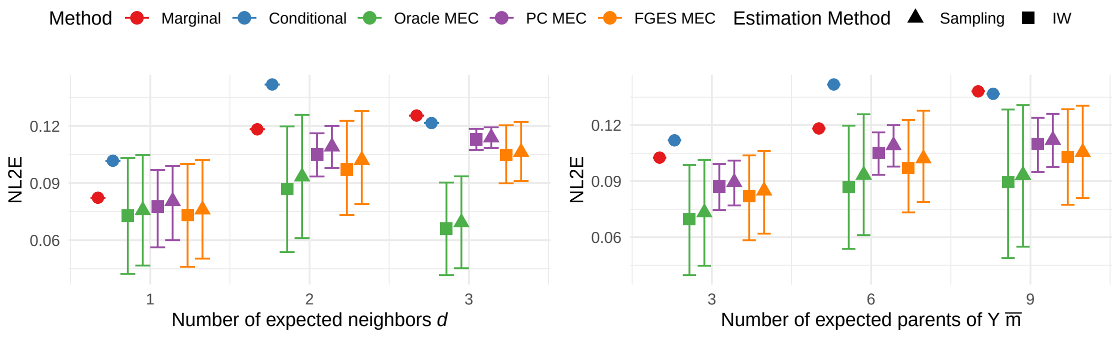
In Fig. 9 we vary (a) the expected number of neighbors for each node d and (b) the number of expected parents of Y $\bar{m}$, both in the non-linear setting. We can see that the number of expected neighbors has a strong influence on causal discovery performance in the non-linear setting, as the MEC Shapley values for PC get progressively worse. The difference between the range resulting from the oracle MEC and from the PC MEC shows that the true graph is on average not contained in the MEC, and the best graph on average leads to explanations deviating far from the one of the true graph. Increasing the expected number of parents of Y $\bar{m}$ seems to affect all methods equally, with the results getting worse as $\bar{m}$ increases. Here we fix the amount of nodes $d = 10$. We use the default settings $n_{\text{cd}} = n_{\text{mc}} = n_{\text{obs}} = 1000$, with $n_{\text{mc}} = 4000$ for the ground truth causal Shapley values.

A.2.4 Amount of Monte Carlo samples

In Fig. 10 we perform an ablation where we vary n_{mc}, the amount of Monte Carlo samples used to estimate expectations for (a) the linear Gaussian setting and (b) the non-linear setting. We can observe that the results in terms of error relative to the ground truth are, after averaging over all datapoints, not significantly influenced by the number of Monte Carlo samples n_{mc}. Here we fix $n_{\text{obs}} = 1000$. For the ground truth causal Shapley values we used $n_{\text{mc}} = 4000$. In this ablation the graphs contain $p = 10$ nodes with $d = 2$ and $\bar{m} = 6$. We use the full set of combinations S.

A.2.5 Amount of data for estimation of distributions

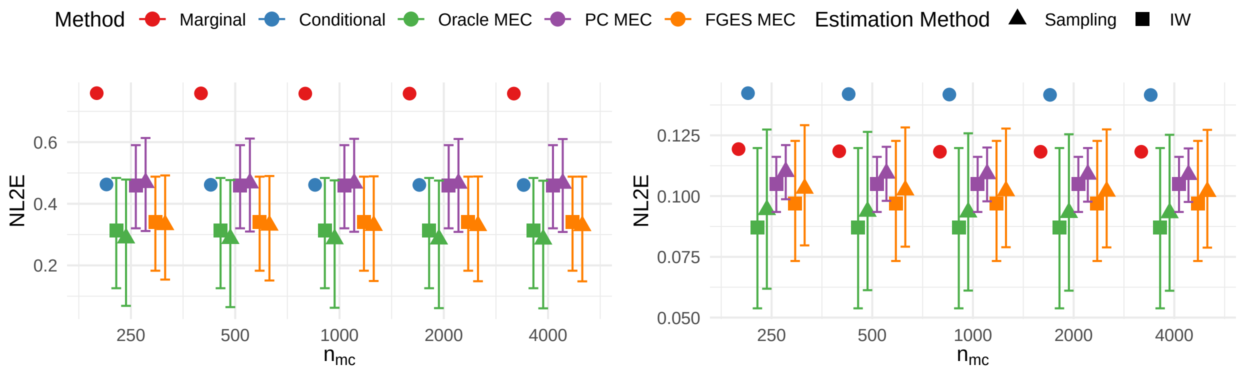
Fig. 11 shows the average $NL2E$ as a function of n_{obs}, the number of observational samples used to estimate marginal and conditional distributions, for (a) the linear Gaussian setting and (b) the non-linear setting. Here we can see that the amount of samples used to estimate distributions affects all methods equally. A low amount of samples increases the error for all explanations, but we can see that as n_{obs} increases, this effect starts to diminish. Interestingly, this holds also for the non-linear case, where the parametric assumptions for the conditional and MEC Shapley values fails to hold.



(a)

(b)

Figure 9: $NL2E$ as a function of (a) the expected number of neighbors of each node d and (b) the expected number of parents of Y in the non-linear setting.



(a) Linear Gaussian

(b) Non−linear

Figure 10: $NL2E$ between the ground truth causal Shapley values and other methods in (a) the linear Gaussian setting and (b) non-linear setting as a function of number of Monte Carlo samples n_{mc} used to estimate each expectation.

A.2.6 Amount of combinations

In Fig. 12 we investigate what occurs when we use only a subset of $[p]$. Unsurprisingly, the error relative to the ground truth decreases as we use more combinations. We can see that the differences between methods stay consistent, with the average error decreasing with the amount of combinations. Here we also include the ground truth computed using only a subset of the power set of $[p]$, denoted *Ground truth (limited)*, which uses the same number of combinations n_{comb} as the other methods. We can see that the error between the methods and the ground truth computed using a lower number of combinations stays consistent, which is important given that we do not use the full power set of subsets for the results on 15 nodes.

A.2.7 Amount of data for causal discovery

In Fig. 13 we vary the amount of samples available for causal discovery for the linear Gaussian setting (a) and the non-linear setting (b). We can see that as the number of samples grows, the difference between the Oracle MEC and PC and FGES decreases, explained by the fact that the outcome of the causal discovery algorithms gets closer to the Oracle MEC.

A.2.8 Effect of number MC samples on computation time

In Fig. 14 we plot the computation time for a single datapoint for each method. We show how n_{mc} affects computation time, where we report both the time taken including the identification of equivalent interventional

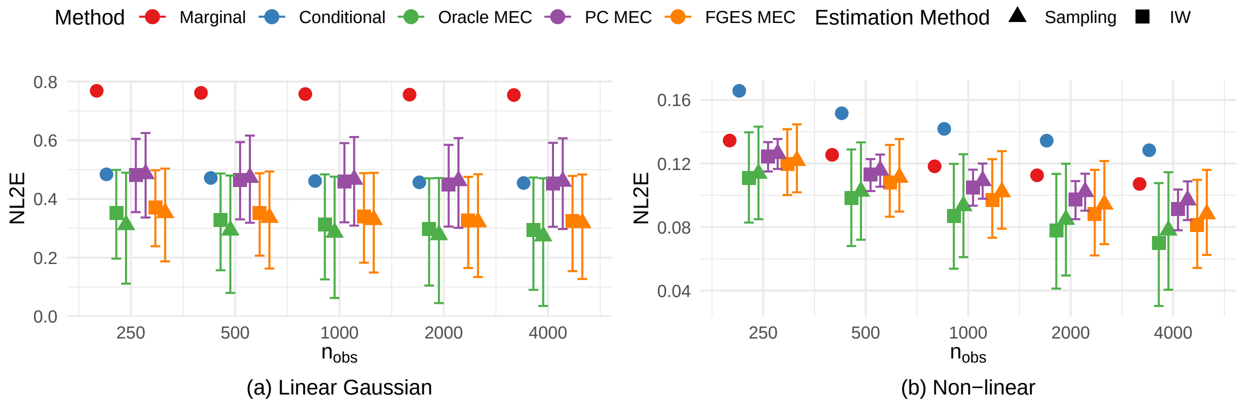


Figure 11: *NL2E* between the ground truth causal Shapley values and other methods in (a) the linear Gaussian setting and (b) non-linear setting as a function of number of observational samples n_{obs} used to estimate each marginal or conditional distribution.

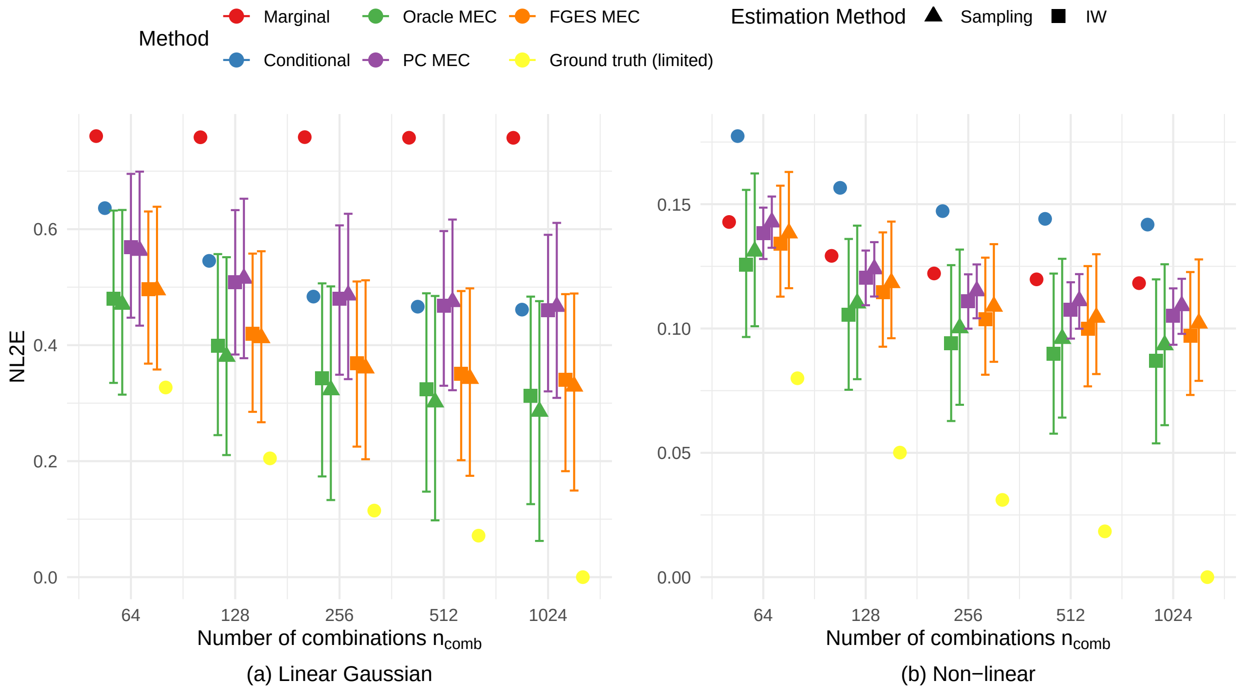
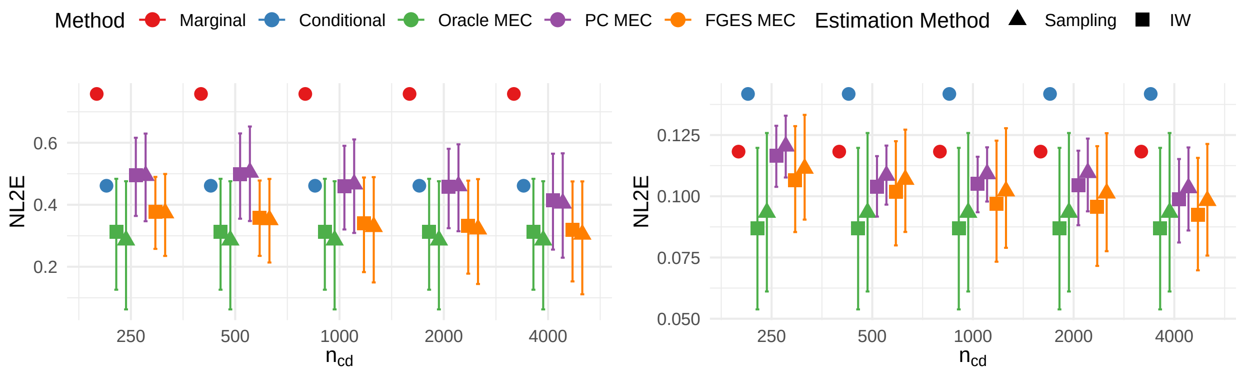


Figure 12: *NL2E* between the ground truth causal Shapley values and other methods in (a) the linear Gaussian setting and (b) non-linear setting as a function of number of combinations n_{comb} used to estimate each Shapley value.

distributions and the time taken just for computation of the Shapley values denoted as `(w/o id)`. While the time taken to identify equivalent distributions is significant, once performed it markedly decreases computation time. The IW method also speeds up computation time. To compare MECs across graphs, we divide total computation time by the amount of graphs in each MEC. As expected the higher the n_{mc}, the higher the average running time for the MEC Shapley methods and the Conditional Shapley values, with the running time for higher numbers of MC samples comparable between Conditional Shapley and the MEC methods.



(a) Linear Gaussian

(b) Non−linear

Figure 13: *NL2E* between the ground truth causal Shapley values and other methods in (a) the linear Gaussian setting and (b) non-linear setting as a function of number of causal discovery samples n_{cd} used to estimate each MEC. Marginal, conditional and Oracle Shapley values are not affected by this parameter.

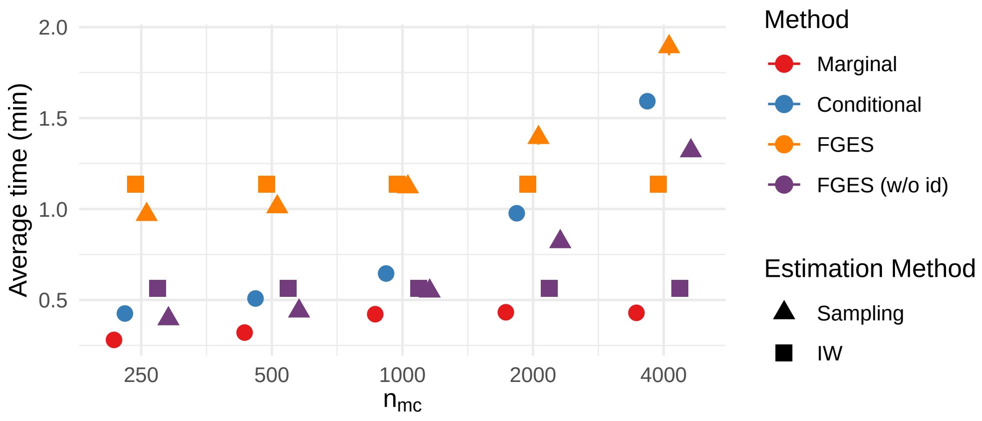


Figure 14: The average runtime for the linear Gaussian setting as a function of the number of MC samples n_{mc}.

A.2.9 Analysis of MECs

In Table 1 we report the average number of DAGs in each MEC as well as the standard deviation for the linear Gaussian and non-linear setting, where we vary the number of nodes p. In Table 2 we report the same, now varying d, the expected number of neighbors for each node. We do not report these tables for the expected number of parents of Y $\bar{m}$, as Y is not included in the causal discovery phase and as such this parameter does not influence the size of each MEC. We can see that the size of the MECs varies substantially across the different models even within the same setting.

In Tables 3 and 4 we report the average Structural Hamming Distance (SHD) for each learned CPDAG with respect to the Oracle CPDAG as well as the standard deviation for the linear Gaussian and non-linear setting. In Table 3 we report these values for varying numbers of nodes d, and in Table 4 we vary the expected number of neighbors $\bar{m}$ for each node.

A.3 Complete experimental results - real world data

A.3.1 Alzheimer's data set

Following Heskes et al. (2020), we consider the Alzheimer's disease data set obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (`http://adni.loni.usc.edu`), which focuses on identifying

Nr. of Nodes p	**Type**	**MEC size**
6	FGES	3.00 ± 2.03
6	Oracle	4.9 ± 5.33
6	PC	3.15 ± 2.25
11	FGES	7.82 ± 8.08
11	Oracle	10.2 ± 13.0
11	PC	7.42 ± 8.20
16	FGES	15.2 ± 20.3
16	Oracle	17.9 ± 14.8
16	PC	13.9 ± 10.5

Nr. of Nodes p	**Type**	**MEC size**
6	FGES	4.55 ± 5.10
6	Oracle	4.9 ± 5.33
6	PC	2.7 ± 2.40
11	FGES	5.5 ± 4.11
11	Oracle	10.2 ± 13.0
11	PC	3.18 ± 4.25
16	FGES	11.2 ± 12.0
16	Oracle	17.9 ± 14.8
16	PC	4.97 ± 5.48

Table 1: Tables reporting MEC sizes for the linear Gaussian setting (left) and non-linear setting (right) where the number of nodes p varies. We report mean and standard deviation for each setting for the Oracle, FGES, and PC MECs.

Nr. of Neighbors d	**Type**	**MEC Size**
1	FGES	7.35 ± 5.26
1	Oracle	7.15 ± 5.30
1	PC	6.92 ± 5.23
2	FGES	7.82 ± 8.08
2	Oracle	10.2 ± 13.0
2	PC	7.42 ± 8.20
3	FGES	6.72 ± 13.0
3	Oracle	8.82 ± 14.2
3	PC	3.92 ± 2.67

Nr. of Neighbors d	**Type**	**MEC Size**
1	FGES	6.78 ± 4.10
1	Oracle	7.15 ± 5.30
1	PC	5.12 ± 5.41
2	FGES	5.5 ± 4.11
2	Oracle	10.2 ± 13.0
2	PC	3.18 ± 4.25
3	FGES	6.3 ± 8.66
3	Oracle	8.82 ± 14.2
3	PC	2.55 ± 3.52

Table 2: Tables of MEC sizes for the linear Gaussian setting (left) and non-linear setting (right) where the number of neighbors d varies. We report mean and standard deviation for each setting for the Oracle, FGES, and PC MECs.

biomarkers for early Alzheimer's detection and progression tracking. Our analysis incorporates the same features as in (Shen et al., 2020; Heskes et al., 2020), age (AGE), education level (EDU), gender (SEX), amyloid beta ($ABETA$), fudeoxyglucose (FDG), phosphorylated tau ($PTAU$), and the number of apolipoprotein alleles ($APOE41$, $APOE42$). As ground truth for the causal discovery we employ the gold standard causal graph from Shen et al. (2020), which includes dummy-encoded variables for the apolipoprotein alleles ($APOE41$ and $APOE42$), and which we show in Fig. 15 (left). For causal discovery we discretized the continuous variables with a maximum of 4 values for each variable using the Freedman-Diaconis rule. Values that are encoded as being larger or smaller than a cutoff value, such as $ABETA > 1700$, are re-encoded at their cutoff value. Rows containing missing values were removed.

Nr. of Nodes p	**Type**	**SHD**
6	FGES	1.73 ± 2.75
6	PC	1.58 ± 2.10
11	FGES	2.00 ± 3.66
11	PC	4.03 ± 3.50
16	FGES	4.72 ± 7.17
16	PC	4.62 ± 4.26

Nr. of Nodes p	**Type**	**SHD**
6	FGES	1.58 ± 2.04
6	PC	2.4 ± 2.07
11	FGES	3.85 ± 3.08
11	PC	5.65 ± 3.04
16	FGES	5.65 ± 4.41
16	PC	8.48 ± 3.72

Table 3: Tables of SHD (Structural Hamming Distance) with respect to the Oracle CPDAG for linear Gaussian setting (left) and non-linear setting (right). Mean and standard deviations are reported for FGES and PC CPDAGs across varying numbers of nodes p.

Nr. of Neighbors d	**Type**	**SHD**
1	FGES	0.6 ± 0.98
1	PC	0.7 ± 1.12
2	FGES	2.0 ± 3.66
2	PC	4. ± 3.50
3	FGES	9.5 ± 9.46
3	PC	9.3 ± 5.16

Nr. of Neighbors d	**Type**	**SHD**
1	FGES	1.27 ± 1.65
1	PC	2.6 ± 2.53
2	FGES	3.85 ± 3.08
2	PC	5.65 ± 3.04
3	FGES	7.38 ± 5.29
3	PC	9.1 ± 3.74

Table 4: Tables of SHD (Structural Hamming Distance) with respect to the Oracle CPDAG for linear Gaussian setting (left) and non-linear setting (right). Mean and standard deviations are reported for FGES and PC CPDAGs across varying expected numbers of neighbors per node d.

After these preprocessing steps, we apply the PC algorithm with the G-square test with $\alpha = 0.05$, and the FGES algorithm using the discrete BIC score with penalty discount $\lambda = 1$. We prohibit edges from biomarkers into demographic variables following (Shen et al., 2020). The TETRAD library was used to run both algorithms (Ramsey et al., 2018). In Fig. 15 on the right we show the learned CPDAGs, with blue edges identified by both FGES and PC, green edges exclusively by FGES, and red edges only by PC. The blue and green edges agree with the gold standard graph, whereas the red edge ($FDG \rightarrow ABETA$) is oriented incorrectly. Neither of the methods include the edge $AGE \rightarrow ABETA$. The two MECs represented by the CPDAGs each contain only a single DAG.

Like (Heskes et al., 2020), we consider a binary classification task, grouping together patients with mild cognitive impairment and Alzheimer's disease, dropping patients with the diagnosis of significant memory concerns. We train an XGBoost model (Chen & Guestrin, 2016) for 100 rounds on a subset of 80% randomly selected training points, with the other 20% unseen by the prediction model and used for evaluation. In the sina plot in Fig. 16 we show for each feature the marginal (left), gold-standard causal (middle), and FGES Shapley values (right), respectively. We can see that FGES and ground truth Causal Shapley values both capture the indirect effect of APOE41 on the prediction, while the marginal Shapley values do not, as well as attributing a larger importance to APOE42.

A.3.2 Bike rental dataset

We consider the bike rental dataset of Fanaee-T & Gama (2014), with the same features and preprocessing as used by Heskes et al. (2020): the number of days since January 2011 (*trend*), two cyclical variables to represent season (*cosyear, sinyear*), the temperature (*temp*), feeling temperature (*atemp*), wind speed (*windspeed*), and humidity (*hum*). We use the partial order ({*trend*},{*cosyear, sinyear*}, {all weather variables}) from Heskes et al. (2020) as background information for the PC and FGES algorithm, both ran using Tetrad. For PC we apply Fisher's Z test with $\alpha = 0.05$, and for FGES the BIC criterion with $\lambda = 1$. The resulting CPDAGs are shown in Fig.17. The CPDAG discovered by FGES contains a single undirected edge between humidity and

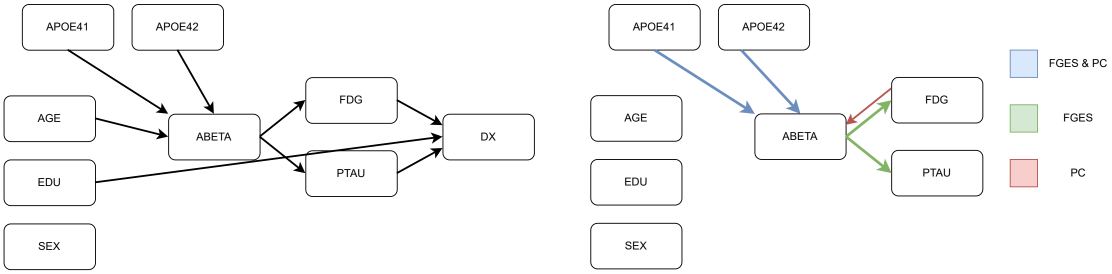


Figure 15: Comparison of gold standard graph recreated from Shen et al. (2020) (left) with CPDAGs discovered by FGES and PC algorithms (right). Edges are color-coded to indicate discovery: blue for edges identified by both FGES and PC, green exclusively by FGES, and red only by PC. The blue and green edges agree with the gold standard graph, whereas the red edge is oriented incorrectly.

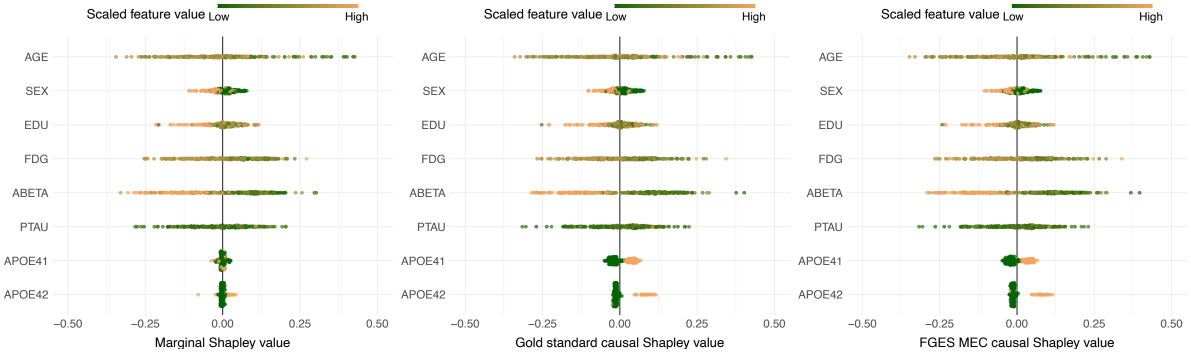


Figure 16: Sina plots for a binary classification task using XGBoost on the Alzheimer's dataset. We plot marginal (left), gold-standard causal (middle) and FGES Shapley values (right).

windspeed, while this edge is oriented *windspeed* → *hum* in the CPDAG discovered by PC. In both graphs, some of the edges are missing, e.g. *trend* → *cosyear*, possibly because *sinyear* and *cosyear* are deterministic functions of the trend.

We split the data into 80% training data and 20% test data, and trained an XGBoost model for 100 rounds, predicting the amount of bike rentals based on the previously mentioned features. In Fig 18 we show sina plots for the marginal Shapley values, the conditional Shapley values and the PC Shapley values (right). We see that the PC Shapley values better capture the contribution of cosyear than marginal and conditional Shapley values, which represents the seasonality of the bike rental (i.e. higher bike rental in summer than in winter). The figure also shows that there is little difference in the attribution between temperature and perceived temperature when using conditional Shapley values, unlike with PC Shapley values where the difference is more noticeable. We omit the sina plots for the two DAGs in the MEC discovered by FGES, as these are very similar to the sina plot for PC. In Fig. 6 we show the bar plot of the Shapley values for two days with temperature about 13 degrees Celsius in October and December. While this temperature represents an average day in October, it is a warm day for December, hence the contribution of the temperature to the prediction of bike rental is substantially more positive for December for both Causal chain and PC Shapley values. Similarly, there is a significant seasonal drop in bike rentals from October to December, which can be seen in both Causal Chain and PC Shapley values, while this does not hold true for marginal Shapley values. Finally, similarly to what is shown in the sina plot, PC Shapley values attribute a bigger impact to temperature than Causal chain Shapley values.

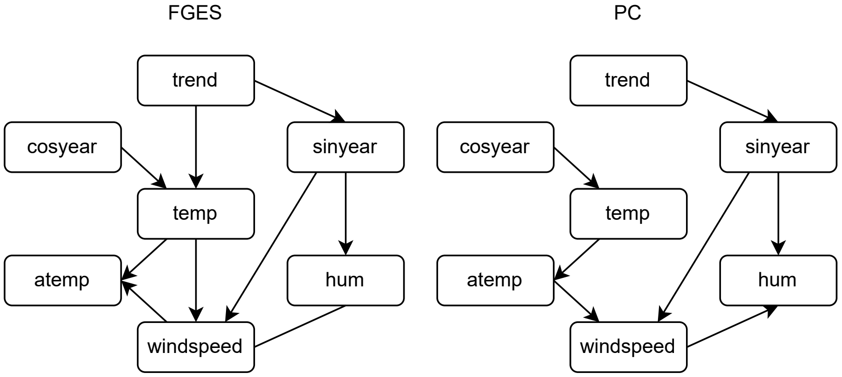


Figure 17: The learned CPDAGs on the bike rental dataset for FGES (left) and the PC algorithm (right).

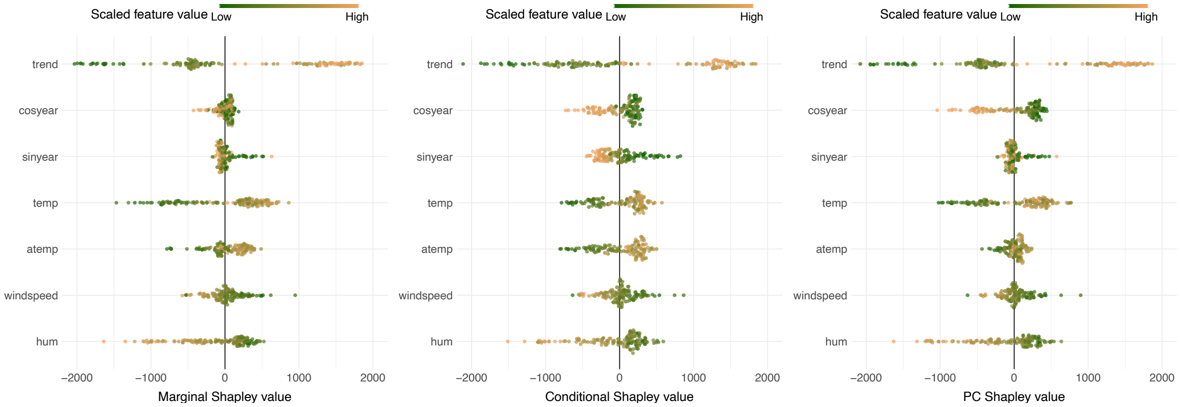


Figure 18: Sina plots for a regression task using XGBoost on the bike rental dataset. We plot marginal (left), conditional (middle) and PC Shapley values (right). The sina plots for the two DAGs in the MEC discovered by FGES are similar to the one by PC.

A.4 Implementation and compute

To implement our method, we adapted the ‘shapr‘ package by Aas et al. (2021) (`https://github.com/NorskRegnesentral/shapr`) released under the MIT license. The causal discovery methods we used were implemented in the ‘pcalg‘ package (Kalisch et al., 2012) released under GPL-2, the ‘kpcalg‘ package (Zhang et al., 2011) released under GPL2, and the Tetrad package (Ramsey et al., 2018) released under the GPL-2 license. We make use of the ‘igraph‘ package (Csardi & Nepusz, 2006), released under the GPL license, to represent graphs. To identify causal effects we make use of the ‘causaleffect‘ package (Tikka & Karvanen, 2017a) released under GPL-2. The prediction models we trained were xgboost models (Chen & Guestrin, 2016) released under the Apache License v2.0. For estimating conditional Gaussian distributions we make use of the ‘mvnfast‘ package (Fasiolo, 2014), released under GPL-2. For analysis of results and the creation of plots we make use the ‘tidyr‘, ‘dplyr‘ and ‘ggplot2‘ packages, all part of the tidyverse (Wickham et al., 2019), released under the MIT license. We provide the code to reproduce all our experiments in the supplementary material.

A.4.1 Hyperparameters

In this section we list each hyperparameter used in the experiments, along with an explanation, how we vary them, and their default value in bold.

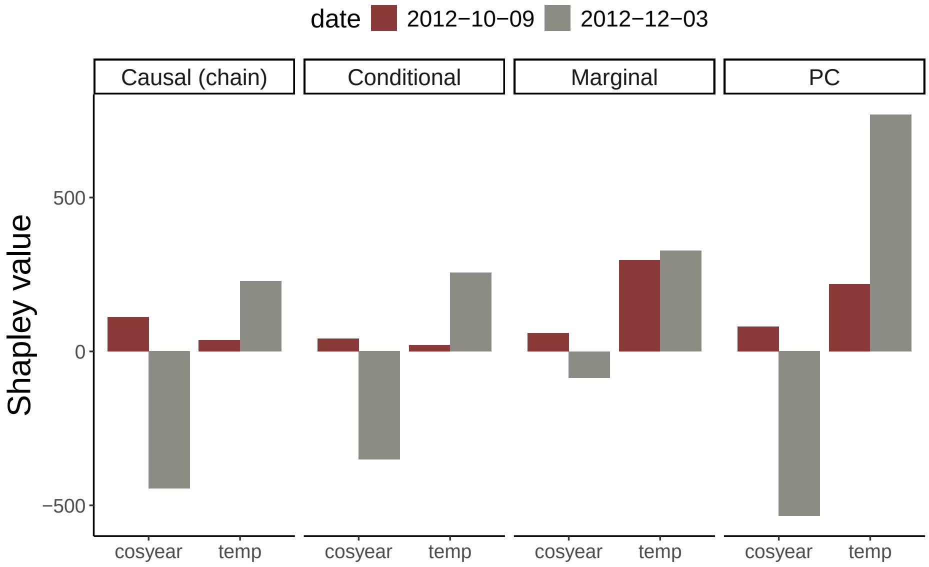


Figure 19: Bar plot of Shapley values for the features *cosyear* and *temp* of the bike rental dataset for two days in October and December with the same temperature. We show the Causal chain Shapley values from Heskes et al. (2020), along with marginal, conditional, and PC Shapley values. Both FGES Shapley values are similar to PC and have been omitted.

When sampling random DAGs, we specify p the amount of nodes excluding the dependent variable (Y), d the expected number of neighbors for each node X_i (representing features), and $\bar{m}$, the expected amount of parents of Y, i.e. the expected number of features directly affecting the dependent variable. We use $p = 5, \mathbf{10}, 15$, $d = 1, \mathbf{2}, 3$, and $\bar{m} = 3, \mathbf{6}, 9$ for $p = 10$.

We consider two parameterizations for each graph: linear Gaussian and non-linear. In the linear Gaussian setting, nodes are parametrized as $X_i = \sum_{j \in \text{Pa}_i} w_{ij} X_j + \epsilon_i$ where $w_{ij} \sim U((-2, -0.5) \cup (0.5, 2))$ and $\epsilon_i = \mathcal{N}(0, 1)$.

For the non-linear setting we consider parameterizations of the form $X_i = w_2^i \sigma(\sum_{j \in \text{Pa}_i} w_1^{ij} X_j) + \epsilon_i$ where σ denotes the sigmoid function, $w_1^{ij} \sim U((-1.5, -0.5) \cup (0.5, 1.5))$, $w_2^i \sim U((-3, 1) \cup (1, 3))$, and ϵ_i follows either a $\mathcal{N}(0, 1)$ or a $U(-1, 1)$ distribution with equal probability. As such, for each DAG we get two different SCMs, one linear Gaussian and one non-linear. For each and each parameterization we sample $n_{\text{scm}} = 40$ random SCMs.

For each setting of hyperparameters we generate $n_{\text{scm}} = 40$ random causal models.

For causal discovery, we use the PC method implemented in the 'pcalg' package (Kalisch et al., 2012). We set a default significance level of $\alpha = 0.05$. We use the default settings, which uses PC-stable by Colombo & Maathuis (2014). For the linear Gaussian parameterizations we use partial correlation conditional independence tests ('gaussCI') and for the non-linear setting kernel-based conditional independence tests implemented by the 'kpcalg' package (Zhang et al., 2011), specifically the Hilbert-Schmidt Independence Criterion gamma test. For FGES we use the TETRAD toolbox (Ramsey et al., 2018) using the BIC score with default penalty discount 1. We use the following values for number of observational samples that we apply PC and FGES to $n_{\text{cd}} = 250, 500, \mathbf{1000}, 2000, 4000$.

Prediction models were trained using XGBoost (Chen & Guestrin, 2016) using $\eta = 0.1$, and 'subsample = 0.8' for a 100 rounds. We used $n_{\text{train}} = 10K$ observational samples to train each model.

For computing Shapley values we consider the hyperparameters n_{comb}, n_{mc}, n_{obs}. n_{comb} specifies the number of combinations used by all methods, where we use the full set of combinations for $p = 5, 10$, and for $p = 15$ $n_{\text{comb}}^{\text{true}} = 8192$ for the ground truth and $n_{\text{comb}} = 4096$ for the other methods. n_{mc} specifies the number of

Monte Carlo samples used to estimate each expectation. We use the values $n_{\text{mc}} = 250, 500, \mathbf{1000}, 2000, 4000$, with $n_{\text{mc}}^{\text{true}} = 4000$ fixed for the ground truth. n_{obs} specifies the amount of observational datapoints used to estimate marginal and conditional distributions, which we vary $n_{\text{obs}} = 250, 500, \mathbf{1000}, 2000, 4000$.

For each SCM we evaluate performance on $n_{\text{test}} = 40$ data points unseen by the prediction model.

A.4.2 Computing infrastructure

The full experiment suite was run in parallel on a cluster of AMD Rome CPUs with 16 cores each. Total compute time for all experiments listed was approximately 88 days. Individual runs in the default setting per 40 data points varied from 30 seconds (marginal, conditional), up to 4 hours for MECs of the largest size. Runs for 15 nodes could take up to 8 hours for 40 datapoints.