Which Invariance Should We Transfer? A Causal Minimax Learning Approach

Mingzhou Liu [1 2] Xiangyu Zheng[3] Xinwei Sun [4] Fang Fang[5] Yizhou Wang[1 2 6]

Abstract

A major barrier to deploying current machine learning models lies in their non-reliability to dataset shifts. To resolve this problem, most existing studies attempted to transfer stable information to unseen environments. Particularly, independent causal mechanism (ICM)-based methods were proposed to remove mutable causal mechanisms via the do-operator. Compared to previous methods, the obtained stable predictors are more effective in identifying stable information. However, a key question remains: which subset of this whole stable information should the model transfer, in order to achieve optimal generalization ability? To answer this question, we present a comprehensive minimax analysis from a causal perspective. Specifically, we first provide a graphical condition for the whole stable set to be optimal. When this condition fails, we surprisingly find with an example that this whole stable set, although can fully exploit stable information, is not the optimal one to transfer. To identify the optimal subset under this case, we propose to estimate the worst-case risk with a novel optimization scheme over the intervention functions on mutable causal mechanisms. We then propose an efficient algorithm to search for the subset with minimal worst-case risk, based on a newly defined equivalence relation between stable subsets. Compared to the exponential cost of exhaustively searching over all subsets, our searching strategy enjoys a polynomial complexity. The effectiveness and efficiency of our methods are demonstrated on synthetic data and the diagnosis of Alzheimer's disease.

[1]Sch. of Computer Science, Peking University. [2]Center on Frontiers of Computing Studies, Peking University. [3]Dep. of Statistics, Guanghua Sch. of Management, Peking University. [4]Sch. of Data Science, Fudan University. [5]Sch. of Psychological and Cognitive Sciences, Peking University. [6]Inst. for Artificial Intelligence, Peking University. Correspondence to: Xinwei Sun <xinwei@fudan.edu.cn>.

Proceedings of the 40th International Conference on Machine Learning Honolulu, Hawaii, USA. PMLR 202, 2023.

1. Introduction

Current machine learning systems, which are commonly deployed based on their in-distribution performance, often encounter dataset shifts (Quinonero et al., 2008) such as covariate shift, label shift, etc., due to changes in the data generating process. When such shifts exist in deployment environments, the model may give unreliable prediction results, which can cause severe consequences in safe-critical tasks such as healthcare (Hendrycks et al., 2021). At the heart of this unreliability issue are stability and robustness aspects, which respectively denote the insensitivity of prediction behavior and generalization errors to dataset shifts.

For example, consider the system deployed to predict the Functional Activities Questionnaire (FAQ) score that is commonly adopted (Mayo, 2016) to measure the severity of Alzheimer's disease (AD). During the prediction, the system can only access biomarkers and volumes of brain regions as covariates, with demographic information anonymous for privacy consideration. However, the changes in such demographics can cause shifts in covariates. To achieve reliability for the deployed model, its prediction is desired to be stable against demographic changes, and meanwhile to be constantly accurate across all populations. For this purpose, this paper aims to find the most robust (i.e., minimax optimal) predictor, among the set of stable predictors across all deployed environments.

To achieve this goal, many studies attempted to learn invariance to transfer to unseen data. Examples include ICP (Peters et al., 2016) and (Rojas-Carulla et al., 2018; Liu et al., 2021; Ausset et al., 2022) that assumed the prediction mechanism given causal features or representations to be invariant; or (Subbaswamy et al., 2019; Rothenhäusler et al., 2021) that explicitly attributed the variation to a prior selection diagram or an exogenous variable. Particularly, the recent independent causal mechanism (ICM)-based methods (Subbaswamy et al., 2019; Shpitser et al., 2021) causally factorized the joint distribution into the mutable (M) set and the stable (S) set, which contained variables with changed and unchanged causal mechanisms, respectively. By intervening on M, they obtained a set of stable predictors, with each containing a stable subset of S to transfer. Compared to ICP-related methods (Peters et al., 2016; Bühlmann, 2020), these stable predictors exploited more

Figure 1: FAQ prediction in Alzheimer's disease. (a) Comparison of maximal mean square error (max. MSE) over deployed environments. (b) Max. MSE of subsets that are ranked in ascending order from left to right, respectively according to the estimated worst-case risk of our method (marked by red) and the validation`s loss adopted by (Subbaswamy et al., 2019) (marked by blue).

types of invariance and thus potentially had better prediction power. However, an important question on robustness has not been studied: which subset of S should the model transfer, in order to achieve optimal generalization ability?

In this paper, we give a comprehensive answer from the perspective of the structural causal model. Specifically, we first provide a graphical condition that is sufficient for the whole stable set to be optimal. This condition can be easily tested via causal discovery. When this condition fails, we construct an example that counter-intuitively shows that this whole stable set, although keeps all the stable information, is NOT the optimal one to transfer. Under this case, we propose an optimization scheme over the intervention functions on M, which is provable to identify the worst-case risk for each stable subset. Our key observation is that the source of dataset shifts is governed by M; therefore, the intervention on M, if set appropriately, can well mimic the worst-case deployed environment. Back to the FAQ prediction example, Fig. 1 (b) shows that our method can consistently reflect the maximal mean squared error (max. MSE) of stable subsets; as a contrast, the validation's loss adopted by (Subbaswamy et al., 2019) fails to do so. This explains our advantage in predicting FAQ across patient groups shown in Fig. 1 (a).

To efficiently search for the optimal subset, we define an equivalence relation between stable subsets via d-separation such that two equivalent subsets share the same worst-case risk. We theoretically show that compared to exhaustively searching over all subsets, searching over only equivalence classes can reduce the exponential complexity to a polynomial one. The effectiveness and efficiency of our methods are demonstrated by the improved robustness, stability, and computation efficiency on a synthetic dataset and the diagnosis of Alzheimer's disease.

Contributions. To summarize, our contributions are:

1. We propose to select the optimal subset of invariance to transfer, guided by a comprehensive minimax analysis from the causal perspective. To the best of our knowledge, this is the first work to study the problem of which invariance should we transfer, in the literature of robust learning.
2. We define an equivalence relation between stable subsets, and accordingly propose to search over only equivalence classes. This new search algorithm can be efficiently solved in polynomial time.
3. We achieve better robustness and stability than others on synthetic data and Alzheimer's disease diagnosis.

2. Related work

Causality-based domain generalization. There are emerging works that considered domain generalization from the causal perspective. One line of works (Arjovsky et al., 2019; Liu et al., 2021; Ahuja et al., 2021; Ausset et al., 2022) promoted invariance as a key surrogate feature of causation where the causal graph was more of a motivation. Another line of works (Peters et al., 2016; Rojas-Carulla et al., 2018; Martinet et al., 2022) was based on invariance assumptions regarding the causal mechanisms. The works most relevant to us are (Subbaswamy et al., 2019; Saeed et al., 2021), which followed the principle of independent causal mechanisms (Scholkopf et al., 2012) to identify invariance by removing the mutable causal mechanisms. However, they did not study how to select the optimal subset in terms of robustness on out-of-distribution generalization.

Optimization-based domain generalization. Some recent works, e.g., DRO (Sinha et al., 2018) and (Sagawa et al., 2019; Wu et al., 2022) formulated domain generalization as a minimax optimization problem and optimized the predictor for robustness. For optimization convenience, they usually constrained the dataset shifts to a limited extent, which limited their application in the real world. In contrast, we adopt optimization to estimate the worst-case risks of predictors, then select the best one via comparison. Our

method can generalize well in a broader distribution family, where the extent of dataset shifts can be unbounded.

Heterogeneous causal discovery. Our work benefits from the recent progress in heterogeneous causal discovery (Ghassami et al., 2018; Huang et al., 2020; Perry et al., 2022), a field that seeks to learn the causal graph with data from multiple environments. However, unlike causal discovery that recovers causal relationships, we focus on minimax analysis and robust subset selection.

3. Preliminary

We consider the supervised regression scenario, where the system includes a target variable $Y \in \mathcal{Y}$, covariates $X := [X_1, \cdots, X_d] \in \mathcal{X}$, and data collected from heterogeneous environments. In practice, different "environments" can refer to different groups of subjects or different experimental settings. We denote the set of training environments as $\mathcal{E}_{tr}$, and the broader set of environments for deployment as $\mathcal{E}$. We denote E as the environmental indicator variable with support $\mathcal{E}$. We use $\{D_e\}_{e \in \mathcal{E}_{tr}}$ to denote our training data, with $D_e := \{(x_k^e, y_k^e)\}_{k=1}^{n_e} \overset{i.i.d}{\sim} P^e(X, Y)$ being data collected from environment e. In a directed acyclic graph (DAG) G, we denote the parents, children, neighbors, and descendants of the vertex V_i as $\mathrm{Pa}(V_i)$, $\mathrm{Ch}(V_i)$, $\mathrm{Neig}(V_i)$, and $\mathrm{De}(V_i)$, respectively. We use $\perp\!\!\!\perp_G$ to denote d-separation in G. We denote $G_{\overline{V_i}}$ as the graph attained via deleting all arrows pointing into V_i.

Our goal is to find the most robust predictor for among stable predictors with data from $\mathcal{E}_{tr}$. Here, we say a predictor $f : \mathcal{X} \to \mathcal{Y}$ is stable if it is independent of E. We denote the set of stable predictors as $\mathcal{F}_S$. For robustness, a commonly adopted measurement (Peters et al., 2016; Ahuja et al., 2021) is to investigate a predictor's worst-case risk, which provides a safeguard for deployment in unseen environments. That is, we want f to have the following minimax property:

$$f^*(x) = \operatorname*{argmin}_{f \in \mathcal{F}_S} \max_{e \in \mathcal{E}} \mathbb{E}_{P^e}[(Y - f(x))^2]. \quad (1)$$

Next, we introduce some basic assumptions, which are commonly made in causal inference and learning (Spirtes et al., 2000; Pearl, 2009; Arjovsky et al., 2019).

Assumption 3.1 (Structural causal model). We assume that $P^e(X, Y)$ is entailed by an unknown DAG G over V for all $e \in \mathcal{E}$, where $V := X \cup Y$. Each variable $V_i \in V$ is generated by a structural equation $V_i \leftarrow g_i^e(\mathrm{Pa}(V_i), U_i)$, where U_i denotes an exogenous variable. We assume each g_i^e is continuous and bounded. Each edge $V_i \to V_j$ in G means V_i is a direct cause of V_j. Besides, we assume the model is Markovian which states that $A \perp\!\!\!\perp_G B \mid Z \Rightarrow A \perp\!\!\!\perp B \mid Z$ for disjoint vertex sets $A, B, Z \subseteq V$.

According to the Causal Markov Condition theorem (Pearl, 2009), the joint distribution can be causally factorized into $P^e(V) = \prod_i P^e(V_i \mid \mathrm{Pa}(V_i)))$, where $P^e(V_i \mid \mathrm{Pa}(V_i)))$ is the causal factor of V_i. Based on the principle of independent causal mechanisms (Schölkopf et al., 2012), these causal factors are autonomous of each other. On the basis of this, the interventional distribution is defined as $P(V \mid do(V_i = v_i)) := \prod_{j \neq i} P^e(V_j \mid \mathrm{Pa}(V_j))) 1_{V_i = v_i}$. Here $do(V_i = v_i)$ means lifting V_i from its original causal mechanism $g_i^e(\mathrm{Pa}(V_i), U_i)$ and setting it to a constant value v_i.

In addition to the Markovian assumption, we also assume the causal faithfulness, which enables us to infer the graph structure from probability properties:

Assumption 3.2 (Causal faithfulness). For disjoint vertex sets $A, B, Z \subseteq V$, $A \perp\!\!\!\perp B \mid Z \Rightarrow A \perp\!\!\!\perp_G B \mid Z$.

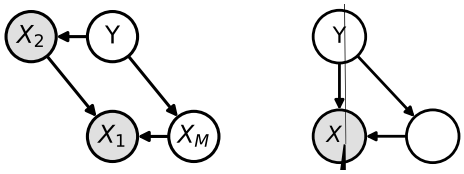


Figure 2: Illustration of the graphical condition in Thm. 4.1. Stable and mutable variables are respectively marked blue and red. In both (a) and (b), we have $\mathbf{X}_M^0 = \{X_M\}; \mathbf{W} = \{X_1\}$.

4. Minimax analysis for the optimal subset

In this section, we provide a comprehensive minimax analysis to answer the above question. At a first glance, one may take S as optimal since it keeps all stable information. We shall show that this is not necessarily the case. To this end, we first provide a graphical condition for the whole stable set to be optimal, *i.e.*, $S^* = S$. This graphical condition can be easily tested via causal discovery. Second, when this condition is not met, we offer a counter-example in which S is not optimal. Then, to identify S^* in this case, we propose an optimization scheme that is provable to identify the worst-case risk for each subset, equipped with which we can pick up the S^* as the one with minimal worst-case risk.

Next, we first introduce a graphical condition and show that the whole stable set S is optimal under this condition.

Theorem 4.1 (Graphical condition for $S^* = S$)**.** *Suppose Asm. 3.1 holds. Denote* $\mathsf{X}_M^0 := \mathsf{X}_M \cap \mathsf{Ch}(Y)$ *as mutable variables in* Y*'s children, and* $\mathsf{W} := \mathsf{De}(\mathsf{X}_M^0) \backslash \mathsf{X}_M^0$ *as descendants of* X_M^0*. Then, we have* $S^* = S$ *if* Y *does not point to any vertex in* W*.*

To understand the graphical condition, note that $Y \not\to \mathsf{W}$ enables applying the inference rules (Pearl, 2009) to remove the "do" in $P(Y|\mathsf{X}_S, do(\boldsymbol{x}_M))$ and degenerate it to a conditional distribution $P(Y|\mathsf{X}')$, for some $\mathsf{X}' \subseteq \mathsf{X}$. This degeneration allows us to construct a P^e where any other predictor has a larger quadratic loss than f_S (Rojas-Carulla et al., 2018), thus proving the optimality of S. Formally, we have the following equivalence result:

Proposition 4.2. *Under Asm. 3.1, the graphical condition holds if and only if* $P(Y|\mathsf{X}_S, do(\boldsymbol{x}_M))$ *can degenerate to a conditional distribution without the "do".*

Example 4.3. To understand this equivalence, consider the DAG shown in Fig. 2 (a), where $Y \not\to \mathsf{W}$. We then have $Y \perp\!\!\!\perp_{G_{\overline{X_M}}} X_1; X_M | X_2$ and hence $P(Y|X_1; X_2; do(x_M)) = P(Y|X_2)$. As a contrast, for the DAG shown in Fig. 2 (b), the collider X_1 causes $Y \not\perp\!\!\!\perp_{G_{\overline{X_M}}} X_M | X_1$ and prevents the removing of the "do" in $P(Y|X_1; do(x_M))$.

The graphical condition can be effectively tested via causal discovery, as guaranteed by the following proposition:

Proposition 4.4 (Testability of Thm. 4.1)**.** *Under Asm. 3.1-3.3, we have that* ***i)*** *the* W *is identifiable; and* ***ii)*** *the condition* $Y \not\to \mathsf{W}$ *is testable from* $\{\mathcal{D}_e\}_{e \in \mathcal{E}_{tr}}$*.*

Remark 4.5. To test $Y \not\to \mathsf{W}$, we first learn the skeleton of G, followed by detecting X_M^0 and W with the heterogeneous causal discovery algorithm CD-NOD (Huang et al., 2020). Then, we have $Y \not\to \mathsf{W}$ if and only if Y is not adjacent to W because $\mathsf{W} \subseteq \mathsf{De}(Y)$ by definition. More details are left to Appx. B.

Thm. 4.1 only provides a partial characterization for S to be optimal; it is still unclear whether the whole stable set is optimal in all cases. In the following, we give a negative answer with a counter-example, whose DAG of Fig. 2 (b) does not satisfy the graphical condition and Y, X_M, X_1 are binary variables. We have the following result:

Claim 4.6. There exists $P(Y)$ and $P(X_1|X_M, Y)$, such that $f_S(\boldsymbol{x}) := \mathbb{E}[Y|x_1, do(x_M)]$ has a larger worst-case risk than $f_\emptyset(\boldsymbol{x}) := \mathbb{E}[Y|do(x_M)]$:

$$\max_{e \in \mathcal{E}} \mathbb{E}_{P^e}[(Y - f_S(\boldsymbol{x}))^2] > \max_{e \in \mathcal{E}} \mathbb{E}_{P^e}[(Y - f_\emptyset(\boldsymbol{x}))^2].$$

Remark 4.7. This result seems surprising as intuitively the whole stable set should be optimal since it fully exploits the stable information, according to existing minimax results in (Peters et al., 2016; Rojas-Carulla et al., 2018). To explain, one should note that these results are built on conditional distributions, where one can construct a P^e to make any other subset have a larger quadratic loss than S. However, when the interventional distribution can not degenerate, such construction is generally not feasible. Please refer to Appx. A.2 for details.

Under general cases where the whole stable set may not be optimal, it remains unknown that *which subset of* S *is the optimal one to transfer*. To answer this question, we propose to estimate the worst-case risk $\mathcal{R}_{S^0} := \max_{e \in \mathcal{E}} \mathbb{E}_{P^e}[(Y - f_{S^0}(\boldsymbol{x}))^2]$ for each subset $S' \subseteq S$; then the S^* corresponds to the subset with minimal $\mathcal{R}$.

For this purpose, we consider a distribution family $\{P_h\}_h$, where h maps from $\mathcal{P}a(\mathcal{X}_M)$ to $\mathcal{X}_M$ and $P_h := P(Y, \mathsf{X}_S | do(\mathsf{X}_M = h(\boldsymbol{pa}(\boldsymbol{x}_M))))$. This distribution set keeps the invariant mechanisms of Y and X_S unchanged while allowing the X_M given their parents to vary arbitrarily, which can well mimic the distributional shifts among deployed environments in $\mathcal{E}$. Particularly, we show that the worst-case risk $\mathcal{R}_{S^0}$ can be attained at some P_h, where h is a Borel measurable function. Formally, denote the Borel function set as $\mathcal{B}$, we have:

Theorem 4.8 (Worst-case risk identification)**.** *Let* $\mathcal{L}_{S^0} := \max_{h \in \mathcal{B}} \mathbb{E}_{P_h}[(Y - f_{S^0}(\boldsymbol{x}))^2]$ *be the maximal population loss over* $\{P_h\}_{h \in \mathcal{B}}$ *for subset* S'*. Then, we have* $\mathcal{L}_{S^0} = \mathcal{R}_{S^0}$ *for each* $S' \subseteq S$*. Therefore, we have* $S^* = \operatorname{argmin}_{S^0 \subseteq S} \mathcal{L}_{S^0}$*.*

This result inspires the following optimization scheme over functions $h \in \mathcal{B}$ to estimate $\mathcal{R}_{S^0}$:

$$\max_{h \in \mathcal{B}} \mathcal{L}_{S^0}(h) := \mathbb{E}_{P_h}[(Y - f_{S^0}(\boldsymbol{x}))^2],$$

as the optimality of which is assured to attain $\mathcal{R}_{S^0}$. To implement, we parameterize h with a multilayer perceptron (MLP) h and optimize over θ, due to the ability of MLP to approximate any Borel function (Hornik et al., 1989). To show the tractability of this optimization, we have the following identifiability result for $\mathcal{L}_{S^0}(h)$:

Proposition 4.9. *Under Asm. 3.1-3.3, the P_h, f_{S^0}, and hence $\mathcal{L}_{S^0}(h)$ are identifiable.*

5. Searching S^* among equivalence classes

In this section, we provide Alg. 1 to identify S^*, which combines Thm. 4.1 and Thm. 4.8. Specifically, Alg. 1 returns S as S^* (line 3), if the graphical condition $Y \not\perp\!\!\!\perp \mathsf{W}$ is tested true. Otherwise, it searches over subsets to identify S^* in terms of the estimated worst-case risk $\mathcal{L}$. For this purpose, a simple search method that is commonly adopted in the literature (Peters et al., 2016; Rojas-Carulla et al., 2018; Magliacane et al., 2018; Subbaswamy et al., 2019) is to *exhaustively* search over all subsets of S.

In the following, we provide a new search strategy with better efficiency, by noticing that the exhaustive search can be redundant for subsets that have the same worst-case risk. Formally, we introduce the equivalence relation as follows:

Definition 5.1 (Equivalence relation)**.** Consider a general graph G over the target Y and covariates X. Let $\sim_G$ be an equivalence relation on all subsets of $\{1, ..., \dim(\mathsf{X})\}$. We say $S' \sim_G S''$ if $\exists\, S_\cap \subseteq S' \cap S''$ such that:

$$Y \perp\!\!\!\perp_G \mathsf{X}_{S'^c_\cap} | \mathsf{X}_{S_\cap}, \text{ where } S^c_\cap := (S' \cup S'') \backslash S_\cap. \quad (3)$$

Algorithm 1 Optimal subset S^* selection.

Input: The training data $\{\mathcal{D}_e\}_{e \in \mathcal{E}_{\text{tr}}}$.

1: Learn the skeleton of G; detect X^0_M, W.
2: **if** $Y \not\perp\!\!\!\perp \mathsf{W}$ **then**
3: $\quad S^* \leftarrow S$. # Thm. 4.1
4: **else**
5: $\quad$ Recover $\text{Pow}(S)/\sim_G$ with Alg. 2.
6: $\quad \mathcal{L}_{\min} \leftarrow \infty$.
7: $\quad$ **for** $[S']$ **in** $\text{Pow}(S)/\sim_G$ **do**
8: $\quad\quad$ **if** $\mathcal{L}_{S^0} < \mathcal{L}_{\min}$ **then**
9: $\quad\quad\quad \mathcal{L}_{\min} \leftarrow \mathcal{L}_{S^0}$, $S^* \leftarrow S'$. # Thm. 4.8
10: $\quad\quad$ **end if**
11: $\quad$ **end for**
12: **end if**
13: **return** S^*.

Algorithm 2 Equivalence classes recovery.

function recover(G)
  if $\text{Neig}(Y) = \emptyset$ **then**
    return $\{\text{Pow}(S)\}$.
  else
    $\text{Pow}(S)/\sim_G \leftarrow \emptyset$.
    for $S' \subseteq \text{Neig}(Y)$ **do**
      Construct a MAG M_G over $S \backslash \mathbf{Neig}(Y)$, with S^0 as the selection set, $\mathbf{Neig}(Y) \backslash S^0$ as the latent set.
      $\text{Pow}(S \backslash \mathbf{Neig}(Y)) = \sim_{M_G} \leftarrow$ **recover**(M_G).
      Add S^0 to each subset in $\text{Pow}(S \backslash \mathbf{Neig}(Y)) = \sim_{M_G}$.
      $\text{Pow}(S) = \sim_G$ .append$(\text{Pow}(S \backslash \mathbf{Neig}(Y)) = \sim_{M_G})$.
    end for
    return $\text{Pow}(S)/\sim_G$.
  end if
end function

Input: The causal graph G.

Let G_S the subgraph of G over $\mathsf{X}_S \cup Y$.
return recover(G_S).

We call elements of the quotient space $\text{Pow}(S)/\sim_G$ as equivalence classes. We use $[S'] := \{S'' | S'' \sim_G S'\}$ to denote the equivalence class of S' and $N_G := |\text{Pow}(S)/\sim_G|$ to denote the number of equivalence classes.

Remark 5.2. The causal graph G in Def. 5.1 can be a Maximal Ancestral Graph (MAG) (Spirtes et al., 2000), where bi-directed edges ($\leftrightarrow$) and undirected edges ($-$) exist due to unobserved confounders and selection variables, respectively. Correspondingly, "$\perp\!\!\!\perp_G$" in Eq. (3) refers to m-separation.

In our scenario, we are interested in the $\sim_G$ relation between stable subsets in the subgraph G_S over $\mathsf{X}_S \cup Y$, which corresponds to conditioning on "$do(\boldsymbol{x}_M)$" in G. According to Def. 5.1, two stable subsets S' and S'' are equivalent if they share an intersection set $S_\cap$ that can d-separate $S' \backslash S_\cap$ and $S'' \backslash S_\cap$ from Y. As a result, we have $P(Y|\mathsf{X}_{S^0}, do(\boldsymbol{x}_M)) = P(Y|\mathsf{X}_{S_\backslash}, do(\boldsymbol{x}_M)) = P(Y|\mathsf{X}_{S^{00}}, do(\boldsymbol{x}_M))$ and hence $\mathcal{R}_{S^0} = \mathcal{R}_{S^{00}}$. For example, in Fig. 2 (a), we have $\{X_2\} \sim_G \{X_1, X_2\}$ as $\mathbf{X}_{S_\backslash} = \{X_2\}$ d-separates $\mathbf{X}_{S^c_\cap} = \{X_1, X_2\} \backslash \{X_2\} = \{X_1\}$ from Y in G_S.

With this $\sim_G$ equivalence, we only need to search equivalence classes, rather than all subsets. To enable this search, we provide Alg. 2 to recover the $\text{Pow}(S)/\sim_G$ in a recursive manner. Specifically, given the input graph G, we first obtain the subgraph G_S by removing X_M in G. Then we find Y's neighbors. Since any two vertices in $\text{Neig}(Y)$ cannot d-separate each other from Y, we go over each subset $S' \subseteq \text{Neig}(Y)$ to construct a MAG over vertices other than $\text{Neig}(Y)$, with S' as the selection set and $\text{Neig}(Y) \backslash S'$ as the latent set. Then it is left to recover equivalence classes in each MAG, and include them to $\text{Pow}(S)/\sim_G$ after append-

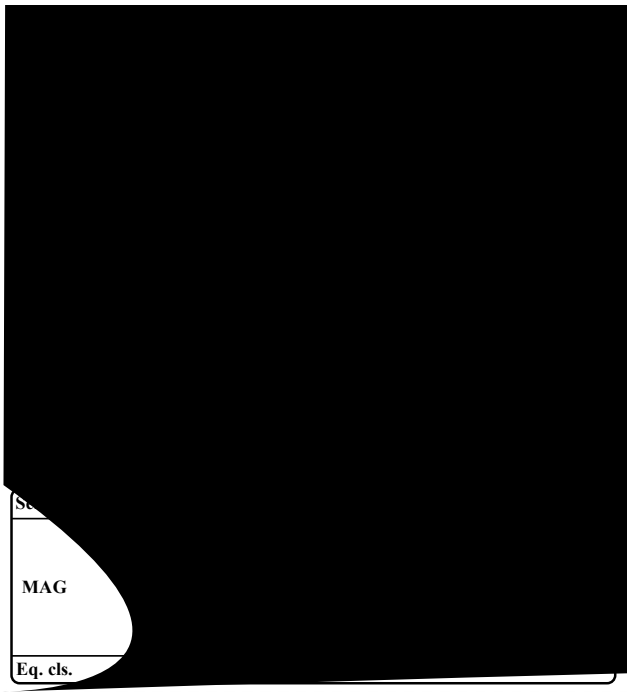


Figure 3: An example to illustrate Alg.2. Stable and mutable variables are respectively marked blue and red.

ing the selection set S' (line 9,10). We recursively repeat the above procedure until $\mathsf{Neig}(Y)$ is empty, which indicates all subsets are equivalent since all of them are d-separated from Y. To illustrate, consider the following Exam. 5.3.

Example 5.3. Consider the causal graph G shown in Fig. 3. We first obtain the G_S over $\mathsf{X}_S \cup Y$, where $\mathbf{Neig}(Y) = \{X_1; X_3\}$. We then take each subset $S^0 \subseteq \{X_1; X_3\}$ as the selection set and $\{X_1; X_3\} \backslash S^0$ as the latent set to respectively construct MAGs (a-d) in the first recursion. For (a) with $\mathbf{Neig}(Y) = \{X_4\}$, we both obtain the MAG in (a.1) when taking $\{X_4\}$ (resp. $\emptyset$) and $\emptyset$ (resp. $\{X_4\}$) as the selection set (resp. latent set). Since $\mathbf{Neig}(Y) = \emptyset$ in (a.1), there is only one equivalence class $[\emptyset] := \mathrm{Pow}(\{X_2; X_5\})$. Following line 9 in Alg. 2, we append X_4 and $\emptyset$ to each subset in equivalence classes of (a.1) to obtain the equivalence classes of (a): $[X_4]$ and $[\emptyset]$. Similarly, after appending the selection set $S^0 = \{X_1; X_3\}$, we include $[X_1; X_3; X_4]$ and $[X_1; X_3]$ to $\mathrm{Pow}(S)/\!\sim_G$. We similarly apply this procedure to (b),(c),(d), which respectively contribute equivalence classes $\{[X_1]\}$, $\{[X_3]; [X_2; X_3]; [X_3; X_4]; [X_2; X_3; X_4]\}$, and $\{[\emptyset]; [X_2]; [X_4]; [X_2; X_4]\}$ to $\mathrm{Pow}(S)/\!\sim_G$.

In practice, we cannot access the true causal graph G but can only recover the graph that is Markovian equivalent to G. The following proposition shows that Alg. 2 can still recover $\mathrm{Pow}(S)/\!\sim_G$ in this case.

Proposition 5.4. *Under Asm. 3.1, 3.2, for each input graph that is Markov equivalent to the ground-truth G, Alg. 2 can correctly recover the* $\mathrm{Pow}(S)/\!\sim_G$.

Besides, we in Appx. E.2 show that the complexity of Alg. 2 is $O(N_G)$, *i.e.*, same as the complexity of searching N_G equivalence classes, which is discussed as follows.

Searching complexity. We show that compared to the exponential cost $O(2^{d_S})$ of exhaustive search, our search strategy enjoys a polynomial cost $\mathrm{P}(d_S)$ when G_S is mainly composited of chain vertices. Here, a chain vertex is a vertex of degree ≤ 2, and a chain is a sequence of connected chain vertices. Specifically, we have the following result:

Proposition 5.5 (Complexity (informal))**.** *Let $d_{\leq 2}$ and $d_{>2} := d_S - d_{\leq 2}$ respectively denote the number of chain vertices and non-chain vertices. When the chain vertices are "distributed intensively", $N_G = \mathrm{P}(d_S)$ if and only if $d_{>2} = O(\log(d_S))$.*

Here, "distributed intensively" means that chain vertices compose only a few chains. Roughly speaking, this is because when the graph is composed of multiple chains that do not intersect each other, N_G is determined by the product of multiple chains' lengths. As a result, the N_G tends to be smaller when the number of chains is small. Formal and more general results are left to Appx. E.

6. Experiment

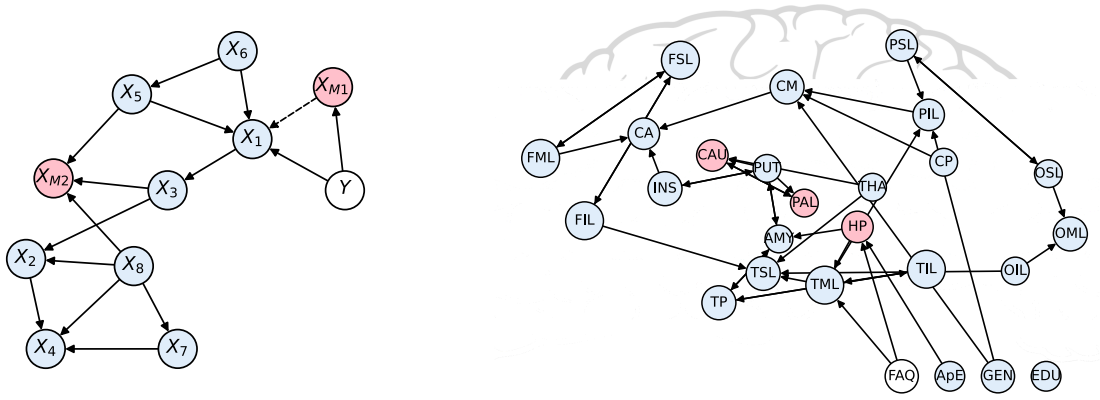
We evaluate our method on synthetic data and a real-world application, *i.e.*, diagnosis of Alzheimer's disease[1].

Compared baselines. i) Vanilla that uses $\mathbb{E}[Y|x]$ to predict Y; **ii) ICP** (Peters et al., 2016) that assumed and used the invariance of parental features $P(Y|\mathsf{Pa}(Y))$; **iii) IC** (Rojas-Carulla et al., 2018) that extended ICP to features beyond $\mathsf{Pa}(Y)$; **iv) DRO** (Sinha et al., 2018) that constrained the distance between training and deployed distributions and conducted optimization for robustness; **v) Surgery estimator** (Subbaswamy et al., 2019) that used validation's loss to identify the optimal subset; **vi) IRM** (Arjovsky et al., 2019) that learned an invariant representation to transfer; **vii) HRM** (Liu et al., 2021) that extended IRM to cases with unknown environmental indices, by exploring the heterogeneity in data via clustering; **viii) IB-IRM** (Ahuja et al., 2021) that leveraged the information bottleneck to supplement the invariance principle in IRM; and **ix) Anchor regression** (Rothenhäusler et al., 2021) that interpolated between ordinary least square (LS) and causal minimax LS.

Evaluation metrics. We use the maximal mean square error (max. MSE) and the standard deviation of MSE (std. MSE) over deployed environments to evaluate the robustness and stability of predictors, respectively.

Implementation details. We use two-layer nonlinear MLPs to implement the f_{S^0} and h . Hyperparameter set-

[1]Code is available at https://github.com/lmz123321/which_invariance.



(a) (b)

Figure 4: (a) The causal graph for synthetic data generation. Stable and mutable variables are respectively marked blue and red. The dashed edge X_{M_1} 99K X_1 does not exist (resp. exist) in setting-1 (resp. setting-2). (b) The learned causal graph on ADNI. The target (FAQ) and biomarkers (ApE, GEN, EDU) are placed in the bottom right. Brain regions are placed at their positions in the brain.

Table 1: Evaluation on synthetic and ADNI datasets. The first column notes the methods we compare. The second and third columns respectively represent the maximal MSE and standard deviation of MSE over deployment environments. The best results are **boldfaced**.

Method	max. MSE (↓)			std. MSE (↓)		
	Syn1	Syn2	ADNI	Syn1	Syn2	ADNI
Vanilla	$1.336_{\pm 0.4}$	$1.861_{\pm 0.4}$	$1.399_{\pm 0.1}$	$0.240_{\pm 0.2}$	$0.481_{\pm 0.1}$	$0.299_{\pm 0.0}$
ICP (Peters et al., 2016)	$1.855_{\pm 0.7}$	$2.331_{\pm 0.2}$	$1.176_{\pm 0.0}$	$0.130_{\pm 0.1}$	$0.230_{\pm 0.0}$	$0.155_{\pm 0.0}$
IC (Rojas-Carulla et al., 2018)	$1.211_{\pm 0.4}$	$1.254_{\pm 0.1}$	$1.165_{\pm 0.2}$	$0.176_{\pm 0.2}$	$0.194_{\pm 0.1}$	$0.198_{\pm 0.1}$
DRO (Sinha et al., 2018)	$1.364_{\pm 0.5}$	$1.495_{\pm 0.1}$	$1.181_{\pm 0.0}$	$0.250_{\pm 0.2}$	$0.326_{\pm 0.0}$	$0.145_{\pm 0.0}$
Surgery (Subbaswamy et al., 2019)	$0.926_{\pm 0.0}$	$1.101_{\pm 0.1}$	$1.069_{\pm 0.1}$	$0.028_{\pm 0.0}$	$0.057_{\pm 0.0}$	$0.129_{\pm 0.0}$
IRM (Arjovsky et al., 2019)	$1.106_{\pm 0.2}$	$1.246_{\pm 0.1}$	$1.223_{\pm 0.0}$	$0.127_{\pm 0.1}$	$0.164_{\pm 0.1}$	$0.177_{\pm 0.0}$
HRM (Liu et al., 2021)	$0.975_{\pm 0.0}$	$1.494_{\pm 0.1}$	$1.272_{\pm 0.1}$	$0.046_{\pm 0.0}$	$0.312_{\pm 0.1}$	$0.194_{\pm 0.1}$
IB-IRM (Ahuja et al., 2021)	$1.076_{\pm 0.0}$	$1.079_{\pm 0.0}$	$1.222_{\pm 0.2}$	$0.056_{\pm 0.0}$	$0.040_{\pm 0.0}$	$0.113_{\pm 0.1}$
AncReg (Rothenhäusler et al., 2021)	$0.938_{\pm 0.0}$	$1.377_{\pm 0.2}$	$1.138_{\pm 0.1}$	$0.033_{\pm 0.0}$	$0.257_{\pm 0.1}$	$0.159_{\pm 0.0}$
Ours (Alg. 1)	$\mathbf{0.926_{\pm 0.0}}$	$\mathbf{1.079_{\pm 0.0}}$	$\mathbf{0.890_{\pm 0.1}}$	$\mathbf{0.028_{\pm 0.0}}$	$\mathbf{0.034_{\pm 0.0}}$	$\mathbf{0.038_{\pm 0.0}}$

Table 2: Comparison of computational cost on ADNI.

Method	Searching cost	Time
Exhaustive (Pow(S))	2^{25}	about 6.4y
Ours (Pow(S)$/\sim_G$)	25307	42h

tings of our method and baselines are left in Appx. F.1.

6.1. Synthetic data

Data generation. We use the DAG in Fig. 4 (a) and the structural equation $V_i = {}^{e}_{i}g_i\left(\sum_{V_j 2\mathbf{Pa}(V_i)} {}_{i,j}V_j\right) + {}''_i$ to generate data, where α_i^e keeps constant, *i.e.*, $\alpha_i^e \equiv \alpha_i$ for all e if V_i is a stable variable; or varies with e if V_i is a mutable variable. For each i, the function g_i is randomly chosen from $\{$*identity*, *tanh*, *sinc*, *sigmoid*$\}$. Each linear parameter $\beta_{i;j}$ is randomly drawn from a uniformed distribution $\mathcal{U}([-2,-0.5]\cup[0.5,2])$ and the noise item $\varepsilon_i \sim \mathcal{N}(0,0.1)$. We generate 20 environments and $n_e = 100$ samples in each environment. To

Figure 5: Results on synthetic data. (a) Setting-1: max. MSE of different subsets, where the whole stable set S is optimal. (b) Setting-2: max. MSE of subsets ranked in the ascending order from left to right, respectively according to the estimated $\mathcal{L}$ of our method and the validation's loss adopted by (Subbaswamy et al., 2019). (c) Comparison of searching cost when $d_{>2}$ increases.

mal max. MSE as expected; in setting-2, the subset with minimal $\mathcal{L}$ also has the minimal max. MSE over deployed environments. Besides, we can observe that the max. MSE shows an approximate increasing trend in subsets ranked by our method; as a contrast, the trend is decreasing in those ranked by the validation's loss adopted by (Subbaswamy et al., 2019). This result suggests that our method can consistently reflect the worst-case risk.

Analysis of $\sim_G$ equivalence. To show the effectiveness of Alg. 2 in recovering equivalence classes, we compute the *intra-class standard deviation*, and compare it with *inter-class std.*, in terms of max. MSE. For intra-class std., we first compute the standard deviation of max. MSE over all subsets in each equivalence class, then take the average over all equivalence classes. For inter-class std, we first compute the average max. MSE over all subsets in each equivalence class; then we compute the std. of the average max. MSE over equivalence classes. In Tab. 3, we observe that the intra-class std. is much smaller than the inter-class std. This result suggests that our Alg. 2 to identify equivalent subsets is effective enough to guarantee the validity of searching over only equivalence classes rather than all subsets.

Searching complexity. We first generate a sequence of causal graphs (Fig. 8) with $d_{>2}$ growing, by deleting/adding edges in the graph shown in Fig. 4 (a) and then compute the searching cost for these graphs. We can see in Fig. 5 (c) that **i**) compared with the exhaustive search, our method can significantly save the searching cost in both sparse and dense graphs; **ii**) the searching cost over equivalence classes decreases when $d_{>2}$ decreases.

6.2. Alzheimer's disease diagnosis

Dataset & preprocessing. We consider the Alzheimer's Disease Neuroimaging Initiative (Petersen et al., 2010) (ADNI) dataset, in which the imaging data is acquired from structural Magnetic Resonance Imaging (sMRI) scans. We apply the Dartel VBM (Ashburner, 2007) for preprocessing and the Statistical Parametric Mapping (SPM) for segmenting brain regions. Then, we implement the Automatic Anatomical Labeling (AAL) atlas (Tzourio-Mazoyer et al., 2002) and region indices provided by (Young et al., 2018) to partition the whole brain into 22 regions (Tab. 4). In addition to brain region volumes, we also include demographics (age, gender (GEN)) and genetic information (the number of ApoE-4 alleles (ApE)). With these covariates, we predict the Functional Activities Questionnaire (FAQ) score (Mayo, 2016) for each patient. We split the dataset into seven environments according to age ($<$60, 60-65, 65-70, 70-75, 75-80, 80-85, $>$85), which respectively contain n_e =27,59,90,240,182,117,42 samples. We repeat 3 times, with each time randomly taking four environments for training and the rest for deployment.

Causal discovery. The learned causal graph is shown in Fig. 4 (b). As we can see, the affection of AD (measured by FAQ score) first shows in the hippocampus (HP) and medial temporal lobe (TML), then propagates to other brain regions, which echos existing studies that the HP and TML are early degenerated regions (Barnes et al., 2009; Duara et al., 2008), Besides, we observe that the caudate (CAU), pallidum (PAL), and hippocampus (HP) are mutable regions, which agrees with the heterogeneity found in different age groups (Cavedo et al., 2014; Fiford et al., 2018).

Equivalence and searching complexity. As shown in Fig. 4 (b), we have FAQ $\rightarrow$ TML, which violates the graphical condition (TML $\in$ W) in Thm. 4.1. We thus search over equivalence classes to find S^*. As shown in Tab. 2, there are only 25307 equivalence classes out of the 2^{25} subsets. Correspondingly, the training time can be saved from about 55,687 hours $\approx$ 6.4 years to only 42 hours.

Results. Fig. 1 (a) shows the max. MSE of our method and baselines. As we can see, our method significantly outperforms the others, which demonstrates the effectiveness of Thm. 4.8 in robust subset selection. Further, Fig. 1 (b) shows that the max. MSE of subsets ranked by our method

appears a positive correlation with the true worst-case risk; as a contrast, the correlation is negative for the max. MSE of subsets ranked by the validation's loss. Particularly, the top subset selected by our method {FSL,TSL,TIL,PSL,OML,CM} reaches a max. MSE of 0.890; while the one selected by the validation's loss {FSL,FML,TSL,TIL,PSL,CA,THA,GEN} only has a max. MSE of 1.069. These results demonstrate the effectiveness of our method in estimating the worst-case risk. The improvements over ICP, IRM, and their extensions can be attributed to the property use of invariance beyond stable causal features/representations. The advantage over DRO may lie in the robustness of our method beyond bounded distributional shifts; while the advantage over Anchor regression can be contributed to the relaxation of the linearity assumption.

7. Conclusion

In this paper, we propose a causal minimax learning approach to identify the optimal subset of invariance to transfer, in order to achieve robustness against dataset shifts. We first provide a graphical condition that is sufficient for the whole stable set to be optimal. When this condition fails, we propose an optimization-based approach that is provable to attain the worst-case risk for each subset. Further, we propose a new search strategy via d-separation, which enjoys better efficiency. The subset selected by our method outperforms the others in terms of robustness on Alzheimer's disease diagnosis. In the future, we are interested to extend our results to cases where the DAG is allowed to vary, which may happen when there are many deployed environments.

Acknowledgements

This work was supported by MOST-2018AAA0102004.

References

Ahuja, K., Caballero, E., Zhang, D., Gagnon-Audet, J.-C., Bengio, Y., Mitliagkas, I., and Rish, I. Invariance principle meets information bottleneck for out-of-distribution generalization. *Advances in Neural Information Processing Systems*, 34, 2021.

Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Ashburner, J. A fast diffeomorphic image registration algorithm. *Neuroimage*, 38(1):95–113, 2007.

Ausset, G., Clémençon, S., and Portier, F. Empirical risk minimization under random censorship. *Journal of Machine Learning Research*, 2022.

Barnes, J., Bartlett, J. W., van de Pol, L. A., Loy, C. T., Scahill, R. I., Frost, C., Thompson, P., and Fox, N. C. A meta-analysis of hippocampal atrophy rates in alzheimer's disease. *Neurobiology of aging*, 30(11): 1711–1723, 2009.

Berge, C. *Topological Spaces*. Oliver and Boyd, London., 1963.

Bühlmann, P. Invariance, causality and robustness. *Statistical Science*, 35(3):404–426, 2020.

Carr, M. W., Roth, S. J., Luther, E., Rose, S. S., and Springer, T. A. Monocyte chemoattractant protein 1 acts as a t-lymphocyte chemoattractant. *Proceedings of the National Academy of Sciences*, 91(9):3652–3656, 1994.

Cavedo, E., Pievani, M., Boccardi, M., Galluzzi, S., Bocchetta, M., Bonetti, M., Thompson, P. M., and Frisoni, G. B. Medial temporal atrophy in early and late-onset alzheimer's disease. *Neurobiology of aging*, 35(9):2004–2012, 2014.

Duara, R., Loewenstein, D., Potter, E., Appel, J., Greig, M., Urs, R., Shen, Q., Raj, A., Small, B., Barker, W., et al. Medial temporal lobe atrophy on mri scans and the diagnosis of alzheimer disease. *Neurology*, 71(24): 1986–1992, 2008.

Fiford, C. M., Ridgway, G. R., Cash, D. M., Modat, M., Nicholas, J., Manning, E. N., Malone, I. B., Biessels, G. J., Ourselin, S., Carmichael, O. T., et al. Patterns of progressive atrophy vary with age in alzheimer's disease patients. *Neurobiology of aging*, 63:22–32, 2018.

Ghassami, A., Kiyavash, N., Huang, B., and Zhang, K. Multi-domain causal structure learning in linear systems. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 6269–6279, 2018.

Goetzl, E., Foster, D., and Payan, D. A basophil-activating factor from human t lymphocytes. *Immunology*, 53(2): 227, 1984.

Gretton, A., Fukumizu, K., Teo, C. H., Song, L., Schölkopf, B., and Smola, A. J. A kernel statistical test of independence. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, pp. 585–592, 2007.

Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8340–8349, 2021.

Hornik, K., Stinchcombe, M., and White, H. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Huang, B., Zhang, K., Zhang, J., Ramsey, J., Sanchez-Romero, R., Glymour, C., and Schölkopf, B. Causal discovery from heterogeneous/nonstationary data. *Journal of Machine Learning Research*, 21(89):1–53, 2020.

Lee, L. E., Pyo, J. Y., Ahn, S. S., Song, J. J., Park, Y.-B., and Lee, S.-W. Clinical significance of large unstained cell count in estimating the current activity of antineutrophil cytoplasmic antibody-associated vasculitis. *International Journal of Clinical Practice*, 75(10):e14512, 2021.

Liu, J., Hu, Z., Cui, P., Li, B., and Shen, Z. Heterogeneous risk minimization. In *International Conference on Machine Learning*, pp. 6804–6814. PMLR, 2021.

Magliacane, S., Van Ommen, T., Claassen, T., Bongers, S., Versteeg, P., and Mooij, J. M. Domain adaptation by using causal inference to predict invariant conditional distributions. *Advances in neural information processing systems*, 31, 2018.

Martinet, G. G., Strzalkowski, A., and Engelhardt, B. Variance minimization in the wasserstein space for invariant causal prediction. In *International Conference on Artificial Intelligence and Statistics*, pp. 8803–8851. PMLR, 2022.

Mayo, A. M. Use of the functional activities questionnaire in older adults with dementia. *Hartford Inst Geriatr Nurs*, 13:2, 2016.

Mitrovic, J., McWilliams, B., Walker, J., Buesing, L., and Blundell, C. Representation learning via invariant causal mechanisms. In *International Conference on Learning Representations (ICLR)*, 2021.

Muñoz-Fuentes, V., Cacheiro, P., Meehan, T. F., Aguilar-Pimentel, J. A., Brown, S. D., Flenniken, A. M., Flicek, P., Galli, A., Mashhadi, H. H., Hrabě de Angelis, M., et al. The international mouse phenotyping consortium (impc): a functional catalogue of the mammalian genome that informs conservation. *Conservation Genetics*, 19(4): 995–1005, 2018.

Pearl, J. *Causality*. Cambridge University Press, 2009.

Perry, R., von Kügelgen, J., and Schölkopf, B. Causal discovery in heterogeneous environments under the sparse mechanism shift hypothesis. *arXiv preprint arXiv:2206.02013*, 2022.

Peters, J., Bühlmann, P., and Meinshausen, N. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, pp. 947–1012, 2016.

Petersen, R. C., Aisen, P., Beckett, L. A., Donohue, M., Gamst, A., Harvey, D. J., Jack, C., Jagust, W., Shaw, L., Toga, A., et al. Alzheimer's disease neuroimaging initiative (adni): clinical characterization. *Neurology*, 74 (3):201–209, 2010.

Quinonero, J., Sugiyama, M., Schwaighofer, A., and Lawrence, N. D. *Dataset shift in machine learning*. Mit Press, 2008.

Rojas-Carulla, M., Schölkopf, B., Turner, R., and Peters, J. Invariant models for causal transfer learning. *The Journal of Machine Learning Research*, 19(1):1309–1342, 2018.

Rothenhäusler, D., Meinshausen, N., Bühlmann, P., and Peters, J. Anchor regression: Heterogeneous data meet causality. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 83(2):215–246, 2021.

Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

Schölkopf, B., Janzing, D., Peters, J., Sgouritsa, E., Zhang, K., and Mooij, J. On causal and anticausal learning. *arXiv preprint arXiv:1206.6471*, 2012.

Schölkopf, B., Locatello, F., Bauer, S., Ke, N. R., Kalchbrenner, N., Goyal, A., and Bengio, Y. Toward causal representation learning. *Proceedings of the IEEE*, 109(5): 612–634, 2021.

Sinha, A., Namkoong, H., and Duchi, J. Certifiable distributional robustness with principled adversarial training. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Hk6kPgZA-.

Spirtes, P., Glymour, C. N., Scheines, R., and Heckerman, D. *Causation, Prediction, and Search*. MIT press, 2000.

Subbaswamy, A., Schulam, P., and Saria, S. Preventing failures due to dataset shift: Learning predictive models that transport. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 3118–3127. PMLR, 2019.

Sun, X., Wu, B., Zheng, X., Liu, C., Chen, W., Qin, T., and Liu, T.-Y. Recovering latent causal factor for generalization to distributional shifts. *Advances in Neural Information Processing Systems*, 34, 2021.

Tzourio-Mazoyer, N., Landeau, B., Papathanassiou, D., Crivello, F., Etard, O., Delcroix, N., Mazoyer, B., and Joliot, M. Automated anatomical labeling of activations in spm using a macroscopic anatomical parcellation of the MNI MRI single-subject brain. *Neuroimage*, 15(1): 273–289, 2002.

Wu, Q., Li, J. Y.-M., and Mao, T. On generalization and regularization via wasserstein distributionally robust optimization. *arXiv preprint arXiv:2212.05716*, 2022.

Young, A. L., Marinescu, R. V., Oxtoby, N. P., Bocchetta, M., Yong, K., Firth, N. C., Cash, D. M., Thomas, D. L., Dick, K. M., Cardoso, J., et al. Uncovering the heterogeneity and temporal complexity of neurodegenerative diseases with subtype and stage inference. *Nature communications*, 9(1):1–16, 2018.

Young, N. Number of vertices that a connected dominating set can reach in densely connected graphs. Theoretical Computer Science Stack Exchange, 2022. URL https://cstheory.stackexchange.com/q/52100.

Zhang, J. On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias. *Artificial Intelligence*, 172(16-17): 1873–1896, 2008.

Appendix

A. Causal minimax theories

A.1. Proof of Thm. 4.1: Graphical condition for $S^* = S$

Thereom 4.1. *Suppose Asm. 3.1 holds. Denote* $\mathbf{X}_M^0 := \mathbf{X}_M \cap \mathsf{Ch}(Y)$ *as mutable variables in* Y*'s children, and* $\mathbf{W} := \mathsf{De}(\mathbf{X}_M^0)\backslash\mathbf{X}_M^0$ *as descendants of* $\mathbf{X}_M^0$*. Then, we have* $S^* = S$ *if* Y *does not point to any vertex in* $\mathbf{W}$*.*

Proof. Define $\mathbf{W}_2 := \mathbf{X}\backslash(\mathbf{X}_M^0 \cup \mathsf{De}(\mathbf{X}_M^0))$ as variables beyond $\mathbf{X}_M^0$ and their descendants, $\mathbf{X}_M^1 := \mathbf{X}_M\backslash\mathbf{X}_M^0$ as mutable variables beyond Y's children.

We first show the equivalence of the following conditions; then show under either of them, we have $S^* = S$.

(1) $Y \perp\!\!\!\perp_{G_{\overline{\mathbf{X}_M^0}}} \mathbf{W}|\mathbf{W}_2$;

(2) Y does not point to any vertex in $\mathbf{W}$;

(3) $P(Y|\mathbf{X}_S, do(\boldsymbol{x}_M))$ can degenerate to the conditional distribution $P(Y|\mathbf{W}_2)$.

We introduce some notations that will be used in the proof. For a vertex V_i, denote $\mathsf{An}(V_i)$ as the set of its ancestors, $G_{\overline{V_i}}$ as the graph obtained by deleting all arrows pointing into V_i, $G_{\underline{V_i}}$ as the graph obtained by deleting all arrows emerging from V_i. To represent the deletion of both pointing (to V_i) and emerging (from V_j) arrows, we use the notation $G_{\overline{V_i}\underline{V_j}}$.

In the following, we will show the equivalence of conditions **(1)**, **(2)**, and **(3)**. Firstly note that **(2)** is equivalent to "Y is not adjacent to $\mathbf{W}$" due to the assumed acyclic of G. Also note that $\mathbf{X}_S \cup \mathbf{X}_M^1 = \mathbf{W} \cup \mathbf{W}_2$.

(1)⇒(2) Prove by contradiction. Suppose Y and $\mathbf{W}$ are adjacent, then they are also adjacent in $G_{\overline{\mathbf{X}_M^0}}$ because $\mathbf{W}\cap\mathbf{X}_M^0 = \emptyset$. As a result, Y and $\mathbf{W}$ can not be *d*-separated by any vertex in $G_{\overline{\mathbf{X}_M^0}}$, which contradicts with **(1)**.

(2)⇒(3) Since $Y \notin \mathsf{Pa}(\mathbf{X}_M^1)$, we have:

$$\begin{aligned} p(y|\boldsymbol{x}_S, do(\boldsymbol{x}_M)) &= \int \frac{p(y|\boldsymbol{pa}(y))}{p(y|\boldsymbol{pa}(y))\prod_{i\in S} p(x_i|\boldsymbol{pa}(x_i))} \\ &= \int \frac{p(y|\boldsymbol{pa}(y))}{p(y|\boldsymbol{pa}(y))\prod_{i\in S} p(x_i|\boldsymbol{pa}(x_i))} \frac{\prod_{X_i\in\mathbf{X}_M^1} p^e(x_i|\boldsymbol{pa}(x_i))}{\prod_{X_i\in\mathbf{X}_M^1} p^e(x_i|\boldsymbol{pa}(x_i))dy} \\ &= \int \frac{p(y, \boldsymbol{x}_S, \boldsymbol{x}_M^1|do(\boldsymbol{x}_M^0))}{p(y, \boldsymbol{x}_S, \boldsymbol{x}_M^1|do(\boldsymbol{x}_M^0))dy} = p(y|\boldsymbol{x}_S, \boldsymbol{x}_M^1, do(\boldsymbol{x}_M^0)), \end{aligned} \tag{4}$$

which indicates $P(Y|\mathbf{X}_S, do(\boldsymbol{x}_M)) = P(Y|\mathbf{X}_S, \mathbf{X}_M^1, do(\boldsymbol{x}_M^0)) = P(Y|\mathbf{W}, \mathbf{W}_2, do(\boldsymbol{x}_M^0))$.

Unfold $P(Y|\mathbf{W}, \mathbf{W}_2, do(\boldsymbol{x}_M^0))$ with the definition of interventional distribution, we have:

$$p(y|\boldsymbol{w}, \boldsymbol{w}_2, do(\boldsymbol{x}_M^0)) = \int \frac{p(y|\boldsymbol{pa}(y))}{p(y|\boldsymbol{pa}(y))\prod_{X_j\in\mathbf{W}} p^e(x_j|\boldsymbol{pa}(x_j))} \frac{\prod_{X_i\in\mathbf{W}_2} p^e(x_i|\boldsymbol{pa}(x_i))}{\prod_{X_i\in\mathbf{W}_2} p^e(x_i|\boldsymbol{pa}(x_i))dy}. \tag{5}$$

Since $\mathsf{Pa}(Y) \cap \{\mathbf{X}_M^0, \mathbf{W}\} = \emptyset$ and $\forall X_i \in \mathbf{W}_2, \mathsf{Pa}(X_i) \cap \{\mathbf{X}_M^0, \mathbf{W}\} = \emptyset$, we further have:

$$p(y|\boldsymbol{w}, \boldsymbol{w}_2, do(\boldsymbol{x}_M^0)) = \int \frac{p^e(y, \boldsymbol{w}_2)}{p^e(y, \boldsymbol{w}_2)} \frac{\prod_{X_j\in\mathbf{W}} p^e(x_j|\boldsymbol{pa}(x_j))}{\prod_{X_j\in\mathbf{W}} p^e(x_j|\boldsymbol{pa}(x_j)dy}. \tag{6}$$

If Y and $\mathbf{W}$ are not adjacent, then $\forall X_j \in \mathbf{W}, Y \notin \mathsf{Pa}(X_j)$. As a result, $p(y|\boldsymbol{w}, \boldsymbol{w}_2, do(\boldsymbol{x}_M^0)) = \int \frac{p(y;w_2)}{p(y;w_2)dy} = p(y|\boldsymbol{w}_2)$, which means $P(Y|\mathbf{X}_S, do(\boldsymbol{x}_M)) = P(Y|\mathbf{W}, \mathbf{W}_2, do(\boldsymbol{x}_M^0)) = P(Y|\mathbf{W}_2)$ can degenerate to a conditional distribution.

(3)

A.2. Details of Claim 4.6: Counter-example of $S^* \neq S$

Counter-example. Consider the DAG in Fig. 6, which is the same as Fig. 2 (b). We set Y, X_s, X_m to binary variables. We will show that there exists $P(Y), P(X_s|X_m, Y)$ such that $f_S := \mathbb{E}[Y|x_s, do(x_m)]$ is not minimax optimal.

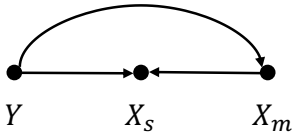


Figure 6: DAG of the counter example.

We show this by proving the predictor f_S has a larger quadratic loss than $f_\emptyset$:

$$\mathbb{E}[(Y - \mathbb{E}[Y|x_s, do(x_m)])^2] > \mathbb{E}[(Y - \mathbb{E}[Y|do(x_m)])^2]. \tag{10}$$

Since we have:

$$\mathbb{E}[(Y - \mathbb{E}[Y|x_s, do(x_m)])^2] = \mathbb{E}[Y^2] + \mathbb{E}\ \mathbb{E}^2[Y|x_s, do(x_m)]\ - 2\mathbb{E}[Y \cdot \mathbb{E}[Y|x_s, do(x_m)]],$$

and $\mathbb{E}[(Y - \mathbb{E}(Y|do(x_m)))^2] = E[Y^2] - E[Y]^2$ due to that $P(Y|do(x_m)) = P(Y)$, Eq. (10) is equivalent to:

$$\mathbb{E}\ \mathbb{E}^2[Y|x_s, do(x_m)]\ > 2\mathbb{E}[Y \cdot \mathbb{E}[Y|x_s, do(x_m)]] - E^2[Y]. \tag{11}$$

Besides, we have:

$$\mathbb{E}\ \mathbb{E}^2[Y|x_s, do(x_m)] = \sum_{x_s, x_m} \Big(\sum_y p(x_s|x_m, y)p(x_m|y)p(y) \Big) \cdot \mathbb{E}^2[Y|x_s, do(x_m)]\ , \tag{12}$$

$$\mathbb{E}[Y \cdot \mathbb{E}[Y|x_s, do(x_m)]] = \sum_{x_s, x_m} \Big(\sum_y p(x_s|x_m, y)p(x_m|y)p(y) \cdot y \Big) \cdot \mathbb{E}[Y|x_s, do(x_m)]\ . \tag{13}$$

Since we have $p(y|x_s, do(x_m)) = \frac{p(y)p(x_s|x_m, y)}{\sum_y p(y)p(x_s|x_m, y)}$, we have:

$$\mathbb{E}[Y|x_s, do(x_m)] = \frac{p(y=1)p(x_s|x_m, y=1)}{\sum_y p(y)p(x_s|x_m, y)}. \tag{14}$$

Substituting Eq. (14) into Eq. (12), (13), we have:

$$\mathbb{E}\ \mathbb{E}^2[Y|X_s, do(X_m)] = \sum_{x_s, x_m} \Big(\sum_y p(x_s|x_m, y)p(x_m|y)p(y) \cdot \frac{p(y=1)p(x_s|x_m, y=1)}{\sum_y p(y)p(x_s|x_m, y)} \Big)^2, \tag{15}$$

$$\begin{aligned}\mathbb{E}[Y \cdot \mathbb{E}[Y|X_s, do(X_m)]] &= \sum_{x_s, x_m} \Big(\sum_y p(x_s|x_m, y)p(x_m|y)p(y) \cdot y \Big) \cdot \frac{p(y=1)p(x_s|x_m, y=1)}{\sum_y p(y)p(x_s|x_m, y)} \\ &= \sum_{x_s, x_m} \Big(\sum_y p(x_s|x_m, y=1)p(x_m|y=1)p(y=1) \Big) \cdot \frac{p(y=1)p(x_s|x_m, y=1)}{\sum_y p(y)p(x_s|x_m, y)}\ .\end{aligned} \tag{16}$$

Denote $a_y := p(y=1)$, $p(x_m=1|y) := a_{my}$, $p(x_s=1|x_m, y) = a_{smy}$. Because X_s, X_m are both binary variables, the summation over them traverses over four indicator functions $\mathbb{1}(x_s=0, x_m=0)$, $\mathbb{1}(x_s=0, x_m=1)$, $\mathbb{1}(x_s=1, x_m=0)$, and $\mathbb{1}(x_s=1, x_m=1)$, which means the left side of Eq. (11) is:

$$\mathbb{E}\ \ \mathbb{E}^2[Y|x_s, do(x_m)]\ = \mathbb{1}(x_s=1,x_m=1)\,(a_{s11}a_{m1}a_y + a_{s10}a_{m0}(1-a_y))\ \ \frac{a_y a_{s11}}{a_y a_{s11}+(1-a_y)a_{s10}}^{\,2} +$$
$$\mathbb{1}(x_s=1,x_m=0)\,[a_{s11}(1-a_{m1})a_y + a_{s10}(1-a_{m0})(1-a_y)]\ \ \frac{a_y a_{s01}}{a_y a_{s01}+(1-a_y)a_{s00}}^{\,2} +$$
$$\mathbb{1}(x_s=0,x_m=1)\,[(1-a_{s11})a_{m1}a_y + (1-a_{s10})a_{m0}(1-a_y)]\ \ \frac{a_y(1-a_{s11})}{a_y(1-a_{s11})+(1-a_y)(1-a_{s10})}^{\,2} +$$
$$\mathbb{1}(x_s=0,x_m=0)\,[(1-a_{s01})(1-a_{m1})a_y + (1-a_{s00})(1-a_{m0})(1-a_y)]\ \ \frac{a_y(1-a_{s01})}{a_y(1-a_{s01})+(1-a_y)(1-a_{s00})}^{\,2}. \quad (17)$$

Similarly, the right side of Eq. (11) is:

$$\begin{aligned}2\mathbb{E}\,[Y\mathbb{E}[Y|x_s,do(x_m)]]-\mathbb{E}[Y^2] =& 2\ \mathbb{1}(x_s=1,x_m=1)\frac{a_y^2a_{s11}^2a_{m1}}{a_y a_{s11}+(1-a_y)a_{s10}}+\\ &\mathbb{1}(x_s=1,x_m=0)\frac{a_y^2a_{s01}^2(1-a_{m1})}{a_y a_{s01}+(1-a_y)a_{s00}}+\\ &\mathbb{1}(x_s=0,x_m=1)\frac{a_y^2(1-a_{s11})^2a_{m1}}{a_y(1-a_{s11})+(1-a_y)a_{s10}}+\\ &\mathbb{1}(x_s=0,x_m=0)\frac{a_y^2(1-a_{s01})(1-a_{m1})}{a_y(1-a_{s01})+(1-a_y)(1-a_{s00})}\ -a_y^2. \end{aligned} \quad (18)$$

Let $a_{s10}=0.001, a_{s11}=0.999, a_{s00}=a_{s01}=a_{s10}=0.5, a_{m0}-2a_{m1}=1, a_y=0.001$, Eq. 11 becomes "$994>-1$", which means Eq. 10 holds and $S^*\neq S$. □

A.3. Proof of Thm. 4.8: Worst-case risk identification

Theorem 4.8. *Let $\mathcal{L}_{S^0} := \max_{h\in\mathcal{B}}\mathbb{E}_{P_h}[(Y-f_{S^0}(\boldsymbol{x}))^2]$ be the maximal population loss over $\{P_h\}_{h\in\mathcal{B}}$ for subset S'. Then, we have $\mathcal{L}_{S^0}=\mathcal{R}_{S^0}$. Therefore, we have $S^* := \operatorname{argmin}_{S^0\subseteq S}\mathcal{L}_{S^0}$.*

Proof. Recall that $P_h := P(Y, \mathbf{X}_S|do(\mathbf{X}_M = h(\boldsymbol{pa}(\boldsymbol{x}_M))))$, where h is a Borel measurable function from $\mathcal{P}a(\mathcal{X}_M)$ to $\mathcal{X}_M$.

To prove the theorem, we show that the worst-case risk $\mathcal{R}_{S^0}$ is attained when the causal factor $P^e(\mathbf{X}_M|\mathbf{Pa}(\mathbf{X}_M))$ degenerates to a delta function $\mathbb{1}(\mathbf{X}_M = h^*(\boldsymbol{pa}(\boldsymbol{x}_M)))$, for some Borel function $h^*:\mathcal{P}a(\mathcal{X}_M)\to\mathcal{X}_M$.

First, consider the case where $\mathbf{X}_M=\{X_m\}$. The $\mathcal{R}_{S^0}$ expands into:

$$\mathcal{R}_{S^0} = \max_{e\in\mathcal{E}}\int_y\int_x [y-f_{S^0}(\boldsymbol{x})]^2 p(y|\boldsymbol{pa}(y))p^e(x_m|\boldsymbol{pa}(x_m))\prod_{i\in S}p(x_i|\boldsymbol{pa}(x_i))dyd\boldsymbol{x}. \quad (19)$$

Let $\tilde{\mathbf{X}} := \mathbf{X}\backslash(X_m\cup\mathbf{Pa}(X_m))$ be variables beyond X_m and its parents. Split the integral in Eq. 19 into three parts: the integral over x_m, the integral over $\boldsymbol{pa}(x_m)$, and the integral over $y,\boldsymbol{x}$. Denote the last part as:

$$l(x_m,\boldsymbol{pa}(x_m)) := \int_y\int_{\tilde{x}} [y-f_{S^0}(\boldsymbol{x})]^2 p(y|\boldsymbol{pa}(y))\prod_{X_i\in\tilde{\mathbf{X}}}p(x_i|\boldsymbol{pa}(x_i))dyd\boldsymbol{x}. \quad (20)$$

Since in Eq. 21, the only item that varies with e is $p^e(x_m|\boldsymbol{pa}(x_m))$, we can move the $\max_{e\in\mathcal{E}}$ into the inner integral and have:

$$\mathcal{R}_{S^0} = \int_{pa(x_m)} \max_{e\in\mathcal{E}} \int_{x_m} l(x_m, \boldsymbol{pa}(x_m)) p^e(x_m|\boldsymbol{pa}(x_m)) dx_m \prod_{X_i\in \mathbf{Pa}(\mathbf{X}_m)} p(x_i|\boldsymbol{pa}(x_i)) dpa(x_m). \tag{22}$$

Let $h^*(\boldsymbol{pa}(x_m)) := \arg\max_{x_m} l(x_m, \boldsymbol{pa}(x_m))$ be a function from $\mathcal{P}a(\mathcal{X}_M)$ to $\mathcal{X}_M$, we have:

$$\mathcal{R}_{S^0} = \int_{pa(x_m)} l(h^*(\boldsymbol{pa}(x_m)), \boldsymbol{pa}(x_m)) \prod_{X_i\in \mathbf{Pa}(\mathbf{X}_m)} p(x_i|\boldsymbol{pa}(x_i)) dpa(x_m), \tag{23}$$

which means the worst-case risk is attained when the causal factor $P(X_m|\mathrm{Pa}(X_m))$ degenerates to a delta function $\mathbb{1}(X_m = h^*(\boldsymbol{pa}(x_m)))$. In addition, under Asm. 3.1, $l(x_m, \boldsymbol{pa}(x_m))$ is a continues function. By the Maximum Theorem (Berge, 1963), $h^* := \arg\max_{x_m} l(x_m, \boldsymbol{pa}(x_m))$ is upper semi-continuous and thus a Borel function.

When $\mathbf{X}_M$ contains multiple mutable variables, we can consider the maximization according to the topology order $\{X_{M;1}, X_{M;2}, ..., X_{M;d_M}\}$, where $X_{M;i}$ is a mutable variable that is not the ancestor of any other variable in $\{X_{M;j}|j<i\}$. That is, we consider the $\max_{e\in\mathcal{E}} \int_{x_{M;i}} l(x_{M;i}, \boldsymbol{pa}(x_{M;i})) p^e(x_{M;i}|\boldsymbol{pa}(x_{M;i})) dx_{M;i}$ sequentially for $i = 1, 2, ..., d_M$.

Such a sequential maximization is plausible because the topology order of mutable variables is identifiable. Please refer to the discovery of $\mathrm{De}(X_i)$ for $X_i \in \mathbf{X}_M$ in Appx. B.1 for details. □

B. Causal discovery and structural identifiability

Minimax theories in Sec. 4 rely on the identifiability of specific causal structures, such as X_M, W. In this section, we will prove the structural identifiability by offering causal discovery algorithms to recover them, with data from $\mathcal{E}_{\text{tr}}$. Specifically, we first show the discovery of several *basic* causal structures, then use them to prove Prop. 4.4 and Prop. 4.9.

B.1. Basic causal structures

In this section, we show the discovery of several basic causal structures: X_M, X_M^0, $\mathsf{X}_M^0 \cup \mathsf{De}(\mathsf{X}_M^0)$, $\mathsf{De}(X_i)$ for $X_i \in \mathsf{X}_M$, and $\mathsf{Pa}(X_i)$ for $X_i \in \mathsf{X}_M \cup \mathsf{De}(\mathsf{X}_M)$. Our algorithms are inspired by (Huang et al., 2020).

We first introduce some notations. We use the subscript $X_i, X_j \in \mathsf{X}$, $V_i, V_j \in \mathsf{V}$ to denote vertices; the superscript $\mathsf{V}^i, \mathsf{V}^j \subseteq \mathsf{V}$ to denote vertex sets. Denote E_{tr} as the environmental indicator variable with support $\mathcal{E}_{\text{tr}}$. Let G_{aug} be the augmented graph (Huang et al., 2020) over $\mathsf{V} \cup E_{\text{tr}}$. We consider the causal DAG G as the induced subgraph of G_{aug} over V. We have the following .9626 Tf 122.64eTd 933 -1.495in6 Tf 1 0 0 1 232.427 586.065d [703. We

Discovery of $\mathsf{Pa}(X_i)$ **for** $X_i \in \mathsf{X}_M \cup \mathsf{De}(\mathsf{X}_M)$**.** This structure has been identified in lines 11 and 19 of Alg. 5.

Algorithm 4 Recovery of $\mathsf{X}^0_M \cup \mathsf{De}(\mathsf{X}^0_M)$

Start with $\mathsf{A}, \mathsf{B} \leftarrow \mathsf{X}^0_M$ and $\mathsf{visited}(X_i) \leftarrow$ false.
while $\mathsf{B} \neq \emptyset$ **do**
  for $X_i \in \mathsf{B}$ **do**
    for $X_j \in \mathsf{Neig}(X_i)$ **do**
      if $X_j \notin \mathsf{X}_M$ and $X_j \perp\!\!\!\perp E_{\text{tr}} | (\mathsf{Z}^{e;j} \cup X_i) \backslash \mathsf{D}^{j;e}$ **then**
        $\mathsf{A} \leftarrow \mathsf{A} \cup X_j$.
        if $\mathsf{visited}(X_j) =$ false **then**
          $\mathsf{B} \leftarrow \mathsf{B} \cup X_j$.
        end if
      end if
      if $X_j \in \mathsf{X}_M$ and $X_j \notin \mathsf{Neig}(Y)$ and $X_j \perp\!\!\!\perp Y | (\mathsf{Z}^{j;y} \cup X_i) \backslash \mathsf{D}^{y;i}$ **then**
        $\mathsf{A} \leftarrow \mathsf{A} \cup X_j$.
        if $\mathsf{visited}(X_j) =$ false **then**
          $\mathsf{B} \leftarrow \mathsf{B} \cup X_i$.
        end if
      end if
    end for
    $\mathsf{B} \leftarrow \mathsf{B} \setminus \{X_i\}$.
  end for
end while

Algorithm 5 Recovery of $\mathsf{De}(X_i)$ for $X_i \in \mathsf{X}_M$.

Start with $\mathsf{B} \leftarrow \mathsf{X}_M$ and $\mathsf{visited}(X_i) \leftarrow$ false.
while $\mathsf{B} \neq \emptyset$ **do**
  for $X_i \in \mathsf{B}$ **do**
    for $X_j \in \mathsf{Neig}(X_i)$ **do**
      if $X_j \notin \mathsf{X}_M$ and $X_j \perp\!\!\!\perp E_{\text{tr}} | (\mathsf{Z}^{e;j} \cup X_i) \backslash \mathsf{D}^{j;e}$ **then**
        orient $X_i - X_j$ as $X_i \to X_j$.
        if $\mathsf{visited}(X_j) =$ false **then**
          $\mathsf{B} \leftarrow \mathsf{B} \cup X_j$.
        end if
      else
        orient $X_i - X_j$ as $X_i \leftarrow X_j$.
      end if
      if $X_j \in \mathsf{X}_M$ and $\mathsf{b}_{i \to j} < \mathsf{b}_{j \to i}$ **then**
        orient $X_i - X_j$ as $X_i \to X_j$.
        if $\mathsf{visited}(X_j) =$ false **then**
          $\mathsf{B} \leftarrow \mathsf{B} \cup X_j$.
        end if
      else
        orient $X_i - X_j$ as $X_i \leftarrow X_j$.
      end if
    end for
    $\mathsf{B} \leftarrow \mathsf{B} \backslash X_i$.
  end for
end while

B.2. Proof of Prop. 4.4: Testability of Thm. 4.1

Proposition 4.4. *Under Asm. 3.1-3.3, we have that **i)** the* W *is identifiable; and **ii)** the condition* $Y \not\to \mathsf{W}$ *is testable from* $\{\mathcal{D}_e\}_{e\in\mathcal{E}_{\text{tr}}}$.

Proof. $\mathsf{W} = (\mathsf{X}\setminus\mathsf{X}_M^0)\cap \mathsf{De}(\mathsf{X}_M^0) = (\mathsf{X}\setminus\mathsf{X}_M^0)\cap \quad \mathsf{X}_M^0 \cup \mathsf{De}(\mathsf{X}_M^0)$ is identifiable because X_M^0 and $\mathsf{X}_M^0 \cup \mathsf{De}(\mathsf{X}_M^0)$ are identifiable, as shown in Appx. B.1. Since all vertices in W are descendants of Y, we have $Y \not\to X_i, X_i \in \mathsf{W}$ iff X_i is not adjacent to Y in the causal skeleton of G_{aug}. □

B.3. Proof of Prop. 4.9: Identifiability of Thm. 4.8

Proposition 4.9. *Under Asm. 3.1-3.3, the* P_h, f_{S^0}, *and hence* $\mathcal{L}_{S^0}(h)$ *are identifiable.*

Proof. To identify P_h, we need to use $h(\mathsf{Pa}(\mathsf{X}_M))$ to replace X_M, followed by regenerating X_i from $\mathsf{Pa}_{G_{\overline{\mathbf{X}}_M}}(X_i)$ for $X_i \in \mathsf{De}_{G_{\overline{\mathbf{X}}_M}}(\mathsf{X}_M)$. Here, $\mathsf{Pa}_{G_{\overline{\mathbf{X}}_M}}(X_i)$ denotes the parents of X_i in the graph $G_{\overline{\mathbf{X}}_M}$.

To identify f_{S^0}, we need to sample from $P(Y, \mathsf{X}_{S^0}|do(\boldsymbol{x}_M))$, which involves intervening X_M and regenerating X_i from $\mathsf{Pa}_{G_{\overline{\mathbf{X}}_M}}(X_i)$ for $X_i \in \mathsf{De}_{G_{\overline{\mathbf{X}}_M}}(\mathsf{X}_M)$.

These structures, *i.e.*, X_M, $\mathsf{De}(X_i)$ for $X_i \in \mathsf{X}_M$, and $\mathsf{Pa}(X_i)$ for $X_i \in \mathsf{X}_M \cup \mathsf{De}(\mathsf{X}_M)$ are readily identified in Appx. B.1. □

C. Empirical estimation methods

C.1. Estimation of f_{S^0}

We adopt soft-intervention to replace $P^e(\mathbf{X}_M|\mathbf{Pa}(\mathbf{X}_M))$ with $P(\mathbf{X}_M)$ and hence define:

$$P'(\mathbf{X},Y) = P(Y|\mathbf{Pa}(Y))P(\mathbf{X}_M)\prod_{i\in S}P(X_i|\mathbf{Pa}(X_i)), \tag{24}$$

which converts the estimation of f_{S^0} to a regression problem, *i.e.*, $f_{S^0}(\boldsymbol{x}) = \mathbb{E}_{P^0}[Y|\boldsymbol{x}_{S^0},\boldsymbol{x}_M]$. To generate data distributed as P', we first randomly permute $\mathbf{X}_M$ in a sample-wise manner to generate data from $P(\mathbf{X}_M)$. We then regenerate data for $X_i \in \mathbf{De}_{G_{\overline{\mathbf{X}}_M}}(\mathbf{X}_M)$ from $\mathbf{Pa}_{G_{\overline{\mathbf{X}}_M}}(X_i)$ via estimating the structural equation.

Indeed, we only need to regenerate $\mathbf{De}_{G_{\overline{\mathbf{X}}_M}}(\mathbf{X}_M)\cap\mathbf{Blanket}_{G_{\overline{\mathbf{X}}_M}}(Y)$ since $P'(Y|\mathbf{X}) = P'(Y|\mathbf{Blanket}_{G_{\overline{\mathbf{X}}_M}}(Y))$. Here, $\mathbf{Blanket}_{G_{\overline{\mathbf{X}}_M}}(Y))$ is the Markovian blanket of Y in the graph $G_{\overline{\mathbf{X}}_M}$. Following this intuition, we consider intervening on another variable set $X^*_{do} := \mathbf{X}^0_M \cup \{\mathbf{De}(\mathbf{X}^0_M)\backslash\mathbf{Ch}(Y)\}$ and regenerate $X_i \in \mathbf{De}_{G_{\overline{\mathbf{X}}_{do}}}(\mathbf{X}^*_{do})$. We show $\mathbf{De}_{G_{\overline{\mathbf{X}}_{do}}}(\mathbf{X}^*_{do})$ is the minimum regeneration set in Prop. C.1.

Proposition C.1. *For any admissible set $\mathbf{X}_{do}$, we have $\mathbf{De}_{G_{\overline{\mathbf{X}}_{do}}}(\mathbf{X}^*_{do}) \subseteq \mathbf{De}_{G_{\overline{\mathbf{X}}_{do}}}(\mathbf{X}_{do})\cap\mathbf{Blanket}_{G_{\overline{\mathbf{X}}_{do}}}(Y)$, which means $\mathbf{De}_{G_{\overline{\mathbf{X}}_{do}}}(\mathbf{X}^*_{do})$ is the minimum regeneration set.*

Proof. We first prove a set $\mathbf{X}_{do}$ is admissible, *i.e.*, $P(Y|\mathbf{X}\backslash\mathbf{X}_{do},do(\boldsymbol{x}_{do})) = P(Y|\mathbf{X}_S,do(\boldsymbol{x}_M))$ if and only if $\mathbf{X}^0_M \subseteq \mathbf{X}_{do}$ and $\{\mathbf{X}_S\cap\mathbf{Ch}(Y)\}\cap\mathbf{X}_{do} = \emptyset$. Note that:

$$\begin{aligned} p(y|\boldsymbol{x}\backslash\boldsymbol{x}_{do},do(\boldsymbol{x}_{do})) &= \frac{p(y|\boldsymbol{pa}(y))}{p(y|\boldsymbol{pa}(y))}\frac{\prod_{X_i\in\{\mathbf{X}\backslash\mathbf{X}_{do}\}}p(x_i|\boldsymbol{pa}(x_i))}{\int\prod_{X_i\in\{\mathbf{X}\backslash\mathbf{X}_{do}\}}p(x_i|\boldsymbol{pa}(x_i))dy} \\ &= \frac{p(y|\boldsymbol{pa}(y))}{p(y|\boldsymbol{pa}(y))}\frac{\prod_{X_i\in\{\mathbf{X}\backslash\mathbf{X}_{do}\}\cap\mathbf{Ch}(Y)}p(x_i|\boldsymbol{pa}(x_i))}{\int\prod_{X_i\in\{\mathbf{X}\backslash\mathbf{X}_{do}\}\cap\mathbf{Ch}(Y)}p(x_i|\boldsymbol{pa}(x_i))dy}, \end{aligned} \tag{25}$$

and

$$\begin{aligned} p(y|\boldsymbol{x}_S,do(\boldsymbol{x}_M)) &= \frac{p(y|\boldsymbol{pa}(y))}{p(y|\boldsymbol{pa}(y))}\frac{\prod_{i\in S}p(x_i|\boldsymbol{pa}(x_i))}{\int\prod_{i\in S}p(x_i|\boldsymbol{pa}(x_i))dy} \\ &= \frac{p(y|\boldsymbol{pa}(y))}{p(y|\boldsymbol{pa}(y))}\frac{\prod_{X_i\in\mathbf{X}_S\cap\mathbf{Ch}(Y)}p(x_i|\boldsymbol{pa}(x_i))}{\int\prod_{X_i\in\mathbf{X}_S\cap\mathbf{Ch}(Y)}p(x_i|\boldsymbol{pa}(x_i))dy}. \end{aligned} \tag{26}$$

Together, Eq. 25 and Eq. 26 indicate $P(Y|\mathbf{X}\backslash\mathbf{X}_{do},do(\boldsymbol{x}_{do})) = P(Y|\mathbf{X}_S,do(\boldsymbol{x}_M))$ if and only if $\{\mathbf{X}\setminus\mathbf{X}_{do}\}\cap\mathbf{Ch}(Y) = \mathbf{X}_S\cap\mathbf{Ch}(Y)$, which can be re-written as:

$$\{\mathbf{X}^0_M\cap\mathbf{X}^c_{do}\}\cup\{\mathbf{X}_S\cap\mathbf{Ch}(Y)\cap\mathbf{X}^c_{do}\} = \mathbf{X}_S\cap\mathbf{Ch}(Y), \tag{27}$$

where $\mathbf{X}^c_{do}$ is the complementary set of $\mathbf{X}_{do}$. Eq. 27 holds if and only if $\mathbf{X}^0_M \subseteq \mathbf{X}_{do}$ and $\{\mathbf{X}_S\cap\mathbf{Ch}(Y)\}\cap\mathbf{X}_{do} = \emptyset$.

We then prove X^*_{do} is an admissible set and $\mathbf{De}_{G_{\overline{\mathbf{X}}_{do}}}(\mathbf{X}^*_{do}) = \mathbf{De}(\mathbf{X}^0_M)\cap(\mathbf{X}_S\cap\mathbf{Ch}(Y))$. X^*_{do} is admissible as the conditions $\mathbf{X}^0_M \subseteq \mathbf{X}^*_{do}$ and $\{\mathbf{X}_S\cap\mathbf{Ch}(Y)\}\cap\mathbf{X}^*_{do} = \emptyset$ hold by definition. We show $\mathbf{De}_{G_{\overline{\mathbf{X}}_{do}}}(\mathbf{X}^*_{do}) = \mathbf{De}(\mathbf{X}^0_M)\cap(\mathbf{X}_S\cap\mathbf{Ch}(Y))$ by showing **i)** $\mathbf{De}_{G_{\overline{\mathbf{X}}_{do}}}(\mathbf{X}^*_{do}) \subseteq \mathbf{De}(\mathbf{X}^0_M)\cap(\mathbf{X}_S\cap\mathbf{Ch}(Y))$ and **ii)** $\mathbf{De}_{G_{\overline{\mathbf{X}}_{do}}}(\mathbf{X}^*_{do}) \supseteq \mathbf{De}(\mathbf{X}^0_M)\cap(\mathbf{X}_S\cap\mathbf{Ch}(Y))$.

i) $\mathbf{De}_{G_{\overline{\mathbf{X}}_{do}}}(\mathbf{X}^*_{do}) \subseteq \mathbf{De}(\mathbf{X}^0_M)\cap(\mathbf{X}_S\cap\mathbf{Ch}(Y))$. Note that $\mathbf{X}^*_{do} \subseteq \mathbf{X}^0_M\cup\mathbf{De}(\mathbf{X}_M)^0$, which means $\mathbf{De}(\mathbf{X}^*_{do}) \subseteq \mathbf{De}(\mathbf{X}^0_M)$. Then, we have:

$$\begin{aligned} \mathbf{De}_{G_{\overline{\mathbf{X}}_{do}}}(\mathbf{X}^*_{do}) &= \mathbf{De}(\mathbf{X}^*_{do})\cap(\mathbf{X}^*_{do})^c = \mathbf{De}(\mathbf{X}^*_{do})\cap(\mathbf{X}^0_M)^c\cap\{\mathbf{De}(\mathbf{X}^0_M)\setminus\mathbf{Ch}(Y)\}^c \\ &= \mathbf{De}(\mathbf{X}^*_{do})\cap\{\mathbf{X}^c_M\cup\mathbf{Ch}(Y)\}\}\cap\{\mathbf{De}(\mathbf{X}^0_M)^c\cup\mathbf{Ch}(Y)\} \\ &\subseteq \mathbf{De}(\mathbf{X}^0_M)\cap\{\mathbf{X}^c_M\cup\mathbf{Ch}(Y)^c\}\cap\{\mathbf{De}(\mathbf{X}^0_M)^c\cup\mathbf{Ch}(Y)\} \\ &= \mathbf{De}(\mathbf{X}^0_M)\cap\mathbf{X}^c_M\cap\mathbf{Ch}(Y) = \mathbf{De}(\mathbf{X}^0_M)\cap\mathbf{X}_S\cap\mathbf{Ch}(Y) \\ &\subseteq \mathbf{De}(\mathbf{X}^0_M)\cap(\mathbf{X}_S\cap\mathbf{Ch}(Y)). \end{aligned} \tag{28}$$

ii) $\mathrm{De}_{G_{\overline{\mathbf{X}}_{do}}}(\mathrm{X}^*_{do}) \supseteq \mathrm{De}(\mathrm{X}^0_M) \cap (\mathrm{X}_S \cap \mathrm{Ch}(Y))$. Since $\mathrm{X}^0_M \subset \mathrm{X}^*_{do}$, $\mathrm{De}(\mathrm{X}^0_M) \subseteq \mathrm{De}(\mathrm{X}^*_{do})$. As a result, we have $\mathrm{De}(\mathrm{X}^0_M) \cap (\mathrm{X}_S \cap \mathrm{Ch}(Y)) \subseteq \mathrm{De}(\mathrm{X}^0_M) \subseteq \mathrm{De}(\mathrm{X}^*_{do})$ and hence $\{\mathrm{De}(\mathrm{X}^0_M) \cap (\mathrm{X}_S \cap \mathrm{Ch}(Y)) \setminus \mathrm{X}^*_{do}\} \subseteq \{\mathrm{De}(\mathrm{X}^*_{do}) \setminus \mathrm{X}^*_{do}\}$. Besides, note that $\mathrm{X}^*_{do} \cap \mathrm{De}(\mathrm{X}^0_M) \cap (\mathrm{X}_S \cap \mathrm{Ch}(Y)) = \emptyset$, which indicates $\mathrm{De}(\mathrm{X}^0_M) \cap (\mathrm{X}_S \cap \mathrm{Ch}(Y)) \setminus \mathrm{X}^*_{do} = \mathrm{X}^*_{do}$ and $\mathrm{De}_{G_{\overline{\mathbf{X}}_{do}}}(\mathrm{X}^*_{do}) = \mathrm{De}(\mathrm{X}^*_{do}) \setminus \{\mathrm{De}(\mathrm{X}^0_M) \cap (\mathrm{X}_S \cap \mathrm{Ch}(Y))\}$. As a result, we have $\mathrm{De}(\mathrm{X}^0_M) \cap (\mathrm{X}_S \cap \mathrm{Ch}(Y)) \subseteq \mathrm{De}_{G_{\overline{\mathbf{X}}_{do}}}(\mathrm{X}^*_{do})$.

Given that any X_{do} needs to satisfy the two conditions, we have:

$$\begin{aligned} \mathrm{X}^0_M \subseteq \mathrm{X}_{do} &\Rightarrow \mathrm{De}(\mathrm{X}^0_M) \subseteq \mathrm{De}(\mathrm{X}_{do}), \\ \mathrm{X}_{do} \subseteq \{\mathrm{X}_S \cap \mathrm{Ch}(Y)\}^c &\Rightarrow \{\mathrm{X}_S \cap \mathrm{Ch}(Y)\} \subseteq \mathrm{X}^c_{do}. \end{aligned} \tag{29}$$

Therefore, we have:

$$\mathrm{De}(\mathrm{X}^0_M) \cap \{\mathrm{X}_S \cap \mathrm{Ch}(Y)\} \subseteq \mathrm{De}(\mathrm{X}_{do}) \cap \mathrm{X}^c_{do}, \tag{30}$$

which means $\mathrm{De}_{G_{\overline{\mathbf{X}}_{do}}}(\mathrm{X}^*_{do}) = \mathrm{De}(\mathrm{X}^0_M) \cap (\mathrm{X}_S \cap \mathrm{Ch}(Y)) \subseteq \mathrm{De}_{G_{\overline{\mathbf{X}}_{do}}}(\mathrm{X}_{do})$ for any admissible set X_{do}. □

Remark C.2. The X^*_{do}, $\mathrm{De}_{G_{\overline{\mathbf{X}}_{do}}}(\mathrm{X}^*_{do})$, and $\mathrm{Pa}(X_i)$ for X_i in $\mathrm{De}_{G_{\overline{\mathbf{X}}_{do}}}(\mathrm{X}^*_{do})$ are identifiable according to Appx. B.1.

C.2. Estimation of $\mathcal{L}_{S^0}$

We first sample from P_h. Specifically, we replace X_i with $h(\mathrm{Pa}(X_i))$ for $X_i \in \mathbf{X}_M$ and regenerate data for $X_i \in \mathrm{De}_{G_{\overline{\mathbf{X}}_M}}$ from $\mathrm{Pa}_{G_{\overline{\mathbf{X}}_M}}(X_i)$ via estimating the structural equation. We then maximize $\mathbb{E}_{P_h}[(Y - f_{S^0}(\boldsymbol{x}))^2]$ over h to obtain $\mathcal{L}_{S^0}$.

D. Equivalence relation and the recovery algorithm

We first introduce some notations that will be used in this section. We use the subscript $X_i, X_j \in \mathbf{X}$, $V_i, V_j \in \mathbf{V}$ to denote variables and vertices; the superscript $S', S^i, S^j \subseteq S$, $\mathbf{V}', \mathbf{V}^i, \mathbf{V}^j \subseteq \mathbf{V}$ to denote variable and vertex subsets. A path $p :=< V_1, V_2, ..., V_l >$ is a sequence of distinct vertices with V_i being adjacent to V_{i+1} for $i = 1, 2, ..., l-1$. We use l to denote the length of the path. The path p can be *blocked* by a vertex set $\mathbf{V}'$ means it can be *d*-separated by $\mathbf{V}'$ when G is a DAG, and *m*-separated by $\mathbf{V}'$ when G is a Maximal Ancestral Graph (MAG). For a vertex V_i, denote $\deg(V_i) := |\mathbf{Neig}(V_i)|$ as its degree. In a MAG, we use $\mathbf{C}$, $\mathbf{L}$ to denote the selection set and the latent set, respectively.

D.1. Details of Def. 5.1: Equivalence relation

We first introduce the following lemma, which studies the property of *d*-separation and *m*-separation in the difference set.

Lemma D.1. *Consider two vertex sets $\mathbf{V}^1, \mathbf{V}^2$, and a path p. If p can be blocked by $\mathbf{V}^1 \cup \mathbf{V}^2$ but can not be blocked by the difference set $(\mathbf{V}^1 \cup \mathbf{V}^2) \backslash \mathbf{V}^2 = \mathbf{V}^1$, then the set $\mathbf{V}^2$ contains a non-collider on p.*

Proof. We first show p contains at least one non-collier. Prove by contradiction. Suppose all vertices on p are colliders. Since p can not be blocked by $\mathbf{V}^1$, we have $\forall V_i \in p$, $V_i \in \mathbf{V}^1$ or $\exists V_j \in \mathbf{De}(V_i)$ such that $V_j \in \mathbf{V}^1$. This means p can not be blocked $\mathbf{V}^1 \cup \mathbf{V}^2$, which is a contradiction.

We then prove the lemma by considering two cases: **i)** p contains only non-colliders; **ii)** p contains both colliders and non-colliders. For **i)**, since p can not be blocked by $\mathbf{V}^1$, all vertices on p are not in $\mathbf{V}^1$. Since p can be blocked by $\mathbf{V}^1 \cup \mathbf{V}^2$, at least a vertex on p is in $\mathbf{V}^2$, thus proving the lemma. For **ii)**, since p can not be blocked by $\mathbf{V}^1$, $\forall V_i \in p$, we have: if V_i is a non-collider on p, $V_i \notin \mathbf{V}^1$; otherwise V_i is a collider on p, $V_i \in \mathbf{V}^1$ or $\exists V_j \in \mathbf{De}(V_i)$ such that $V_j \in \mathbf{V}^1$, thus in the set $\mathbf{V}^1 \cup \mathbf{V}^2$. Therefore, the set $\mathbf{V}^2$ must contain a non-collider on p, otherwise, p will not be blocked by $\mathbf{V}^1 \cup \mathbf{V}^2$. □

Definition 5.1. Consider a general causal graph G over an output Y and covariates $\mathbf{X}$. Let $\sim_G$ be an equivalence relation on all subsets of $\{1, ..., \dim(X)\}$. We say $S^i \sim_G S^j$ if $\exists\, S^{ij} \subseteq S^i \cap S^j$ such that:

$$Y \perp\!\!\!\perp_G \mathbf{X}_{(S^{ij})^c} | \mathbf{X}_{S^{ij}}, \text{ where } (S^{ij})^c := (S^i \cup S^j) \backslash S^{ij}. \tag{31}$$

Proof. It is obvious that the $\sim_G$ is reflective ($S^i \sim_G S^i$) and symmetric ($S^i \sim_G S^j \Rightarrow S^j \sim_G S^i$). In the following, we will show it is also transitive, *i.e.*, $S^i \sim_G S^j, S^j \sim_G S^k \Rightarrow S^i \sim_G S^k$. We show this by constructing an intersection set $S^{ik} \subseteq S^i \cap S^k$ such that $Y \perp\!\!\!\perp_G \mathbf{X}_{(S^{ik})^c} | \mathbf{X}_{S^{ik}}$.

Since $S^i \sim_G S^j$, we have $\exists\, S^{ij}$ s.t. $Y \perp\!\!\!\perp_G \mathbf{X}_{(S^{ij})^c} | \mathbf{X}_{S^{ij}}$. Similarly for $S^j \sim_G S^k$, we have $\exists\, S^{jk}$ s.t. $Y \perp\!\!\!\perp_G \mathbf{X}_{(S^{jk})^c} | \mathbf{X}_{S^{jk}}$. In the following, we will construct the intersection set S^{ik} from $S^{ij} \cap S^{jk}$.

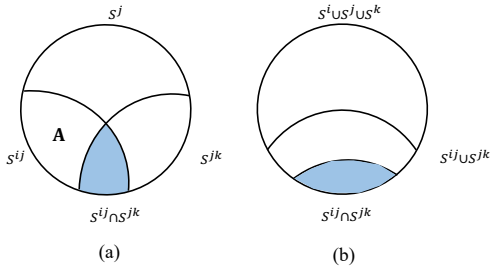


Figure 7: Illustration of union and intersection of $\mathbf{X}_{S_{ij}}$ and $\mathbf{X}_{S_{jk}}$.

Denote $\mathbf{A} := \mathbf{X}_{S^{ij} \backslash (S^{ij} \cap S^{jk})}$ and $\mathbf{B} := \mathbf{X}_{S^{jk} \backslash (S^{ij} \cap S^{jk})}$, as shown by Fig. 7 (a). We first show $Y \perp\!\!\!\perp_G \mathbf{A} | \mathbf{X}_{S^{ij} \cap S^{jk}}$ and $Y \perp\!\!\!\perp_G \mathbf{B} | \mathbf{X}_{S^{ij} \cap S^{jk}}$. We show this by proving that any path between Y and $\mathbf{A}$ (similarly $\mathbf{B}$) can be blocked by $\mathbf{X}_{S^{ij} \cap S^{jk}}$. Prove by contradiction. Suppose there is a path $p_0 :=< Y, X_1, ..., X_{l_0} >$ between Y and $X_{l_0} \in \mathbf{A}$ such that p_0 can not be blocked by $\mathbf{X}_{S^{ij} \cap S^{jk}}$. We have p_0 can be blocked by the set $\mathbf{X}_{S^{jk}}$. This is because $X_{l_0} \in \mathbf{A} \subseteq \mathbf{X}_{S^j \backslash S^{jk}} \subseteq \mathbf{X}_{(S^{jk})^c}$ and $Y \perp\!\!\!\perp_G \mathbf{X}_{(S^{jk})^c} | \mathbf{X}_{S^{jk}}$. Therefore, by Lemma D.1, the set $\mathbf{B} = \mathbf{X}_{S^{jk}} \backslash \mathbf{X}_{S^{ij} \cap S^{jk}}$ contains a non-collider denoted as X_{l_1} on p_0. Hence, we have a subpath of p_0, *i.e.*, $p_1 :=< Y, X_1, ..., X_{l_1} >$ between Y and $X_{l_1} \in \mathbf{B}$ such that p_1 can not be blocked by $\mathbf{X}_{S^{ij} \cap S^{jk}}$. Here, we have p_1 can be blocked by the set $\mathbf{X}_{S^{ij}}$. This is because $X_{l_1} \in \mathbf{B} \subseteq \mathbf{X}_{S^j \backslash S^{ij}} \subseteq \mathbf{X}_{(S^{ij})^c}$

and $Y \perp\!\!\!\perp_G \mathsf{X}_{(S^{ij})^c} | \mathsf{X}_{S^{ij}}$. Therefore, by Lemma D.1, the set $\mathsf{A} = \mathsf{X}_{S^{ij}} \setminus \mathsf{X}_{S^{ij} \cap S^{jk}}$ contains a non-collider denoted as X_{l_2} on p_1. Repeating like this, we have either $X_1 \in \mathsf{A} \subseteq \mathsf{X}_{(S^{jk})^c}$ or $X_1 \in \mathsf{B} \subseteq \mathsf{X}_{(S^{ij})^c}$. Since X_1 is adjacent to Y, this contradicts with $Y \perp\!\!\!\perp_G \mathsf{X}_{(S^{jk})^c} | \mathsf{X}_{S^{jk}}$ or $Y \perp\!\!\!\perp_G \mathsf{X}_{(S^{ij})^c} | \mathsf{X}_{S^{ij}}$.

Further, denote $\mathsf{D} := \mathsf{X}_{(S^{ij} \cup S^{jk}) \setminus (S^{ij} \cap S^{jk})}$ and $\mathsf{F} := \mathsf{X}_{(S^i \cup S^j \cup S^k) \setminus (S^{ij} \cap S^{jk})}$, as shown in Fig. 7 (b). We have shown $Y \perp\!\!\!\perp_G \mathsf{D} | \mathsf{X}_{S^{ij} \cap S^{jk}}$ by combining the statements $Y \perp\!\!\!\perp_G \mathsf{A} | \mathsf{X}_{S^{ij} \cap S^{jk}}$ and $Y \perp\!\!\!\perp_G \mathsf{B} | \mathsf{X}_{S^{ij} \cap S^{jk}}$. Next, we will show $Y \perp\!\!\!\perp_G \mathsf{F} | \mathsf{X}_{S^{ij} \cap S^{jk}}$. This means we can construct the intersection set $S^{ik} := (S^{ij} \cap S^{jk}) \subseteq (S^i \cap S^k)$ such that $Y \perp\!\!\!\perp_G \mathsf{X}_{(S^{ik})^c} | \mathsf{X}_{S^{ik}}$, and hence proving $S^i \sim_G S^k$ by definition.

We show this by proving that any path between Y and F can be blocked by $\mathsf{X}_{S^{ij} \cap S^{jk}}$. Prove by contradiction. Suppose there is a path $p_0 := < Y, X_1, ..., X_{l_0} >$ between Y and $X_{l_0} \in \mathsf{F}$ such that p_0 can not be blocked by $\mathsf{X}_{S^{ij} \cap S^{jk}}$. We have p_0 can be blocked by $\mathsf{X}_{S^{ij}}$ or $\mathsf{X}_{S^{jk}}$. This is because we have either $X_{l_0} \in \mathsf{F} \subseteq \mathsf{X}_{(S^{ij})^c}$ or $X_{l_0} \in \mathsf{F} \subseteq \mathsf{X}_{(S^{jk})^c}$ and $Y \perp\!\!\!\perp_G \mathsf{X}_{(S^{ij})^c} | \mathsf{X}_{S^{ij}}, Y \perp\!\!\!\perp_G \mathsf{X}_{(S^{jk})^c} | \mathsf{X}_{S^{jk}}$. Without loss of generality, we consider $X_{l_0} \in \mathsf{X}_{(S^{ij})^c}$ and p_0 can be blocked by $\mathsf{X}_{S^{ij}}$. By Lemma D.1, the set $\mathsf{X}_{S^{ij} \setminus (S^{ij} \cap S^{jk})}$ contains a non-collider denoted as X_{l_1} on p_1. Hence, we have a subpath of p_0, *i.e.*, $p_1 := < Y, X_1, ..., X_{l_1} >$ between Y and $X_{l_1} \in \mathsf{X}_{S^{ij} \setminus (S^{ij} \cap S^{jk})}$ such that p_1 can not be blocked by $\mathsf{X}_{S^{ij} \cap S^{jk}}$. This contradicts with the statement $Y \perp\!\!\!\perp_G \mathsf{D} | \mathsf{X}_{S^{ij} \cap S^{jk}}$, because $\mathsf{X}_{S^{ij} \setminus (S^{ij} \cap S^{jk})} \subseteq \mathsf{X}_{(S^{ij} \cup S^{jk}) \setminus (S^{ij} \cap S^{jk})} = \mathsf{D}$.

To conclude, we have proved $\sim_G$ is reflective, symmetric, and transitive. Hence, $\sim_G$ is a legitimate equivalence relation. □

D.2. Proof of Prop. 5.4: Correctness of Alg. 2

Proposition 5.4. *For each input graph that is Markov equivalent to the ground-truth graph G, Alg. 2 can correctly recover* $\mathsf{Pow}(S)/\sim_G$.

Proof. We first show, under Asm .3.1, 3.2, all Markovian equivalent graphs have the same equivalence classes. Specifically, Markovian equivalent graphs have the same *d*-separation and *m*-separation (Pearl, 2009; Zhang, 2008). Because the equivalence relation is defined on *d*-separation and *m*-separation, they also have the same equivalence classes.

We then introduce some notions that will be used in the proof. We use the unbolded letter, *e.g.*, S^i, T^i, to denote variable sets, and the **bolded** letter, *e.g.*, $\mathsf{Pow}(S)$, R^i, to denote sets whose elements are variable sets. Recall that the equivalence class of subset S^i is denoted as $\mathbf{[}S^i\mathbf{]} := \{S^j | S^j \sim_G S^i\}$. We say a vertex X_i is Y's l-neighbour if the shortest path between Y and X_i has length l. As a special case, say X_i as the 0-neighbour of Y if there is no path between Y and X_i. Define $l_G = 0$ if $\mathsf{Neig}(Y) = \emptyset$, and $l_G = 1, 2, ..., l$ if Y has $1, 2, ..., l$-neighbours, respectively.

In the following, we will prove the correctness of Alg. 2 by induction on l_G.

Base. $l_G = 0 \Rightarrow \mathrm{Neig}(Y) = \emptyset$. Hence, any two subsets $S^i; S^j \subseteq S$ are equivalent and $\mathbf{Pow}(S) = \sim_G = \{\mathbf{[}S\mathbf{]}\} = \mathbf{recover}(G)$.

Induction hypothesis. Suppose any graph G with $l_G \leq l$ has $\mathsf{Pow}(S)/\sim_G = \mathbf{recover}(G)$.

Step. Consider G with $l_G = l + 1$.

Denote all the $2^{\deg(Y)}$ subsets of $\mathsf{Neig}(Y)$ as $T^1, T^2, ..., T^{2^{\deg(Y)}}$. We can partition the $\mathsf{Pow}(S)$ into $2^{\deg(Y)}$ sets $\mathsf{R}^1, \mathsf{R}^2, ..., \mathsf{R}^{2^{\deg(Y)}}$, with $\mathsf{Pow}(S) = \cup_{i=1}^{2^{\deg(Y)}} \mathsf{R}^i$, $\mathsf{R}^i \cap \mathsf{R}^j = \emptyset$ for $i \neq j$, and $\mathsf{R}^i := \{S^i | S^i \subseteq S, S^i \cap \mathsf{Neig}(Y) = T^i\}$. Now, consider a subset $S^i \in \mathsf{R}^i$ and another subset $S^j \in \mathsf{R}^j$, we have $S^i \not\sim_G S^j$, because $S^i \cap \mathsf{Neig}(Y) \neq S^j \cap \mathsf{Neig}(Y)$. Therefore, the equivalence classes in $\mathsf{Pow}(S)$ is the union of the equivalence classes in $\mathsf{R}^1, \mathsf{R}^2, ..., \mathsf{R}^{2^{\deg(Y)}}$. Formally:

$$\mathsf{Pow}(S)/\sim_G = \cup_{i=1}^{2^{\deg(Y)}} \mathsf{R}_i/\sim_G . \tag{32}$$

A distinct virtue of the MAG constructed in Alg. 2 in line-7 is that it can represent *d*-separation and *m*-separation when selection and latent variables exist. Specifically, given any causal graph G over $\mathsf{V} = \mathsf{O} \cup \mathsf{L} \cup \mathsf{C}$, the MAG M_G over O, with C as the selection set and L as the latent set, satisfies that for any disjoint subsets $\mathsf{A}, \mathsf{B}, \mathsf{Z} \subseteq \mathsf{O}$, $\mathsf{A} \perp\!\!\!\perp_{M_G} \mathsf{B} | \mathsf{Z}$ if and only if $\mathsf{A} \perp\!\!\!\perp_G \mathsf{B} | \mathsf{Z} \cup \mathsf{C}$ (Zhang, 2008). Therefore, for the M_G over $S \setminus \mathsf{Neig}(Y)$, with S' as the selection set, $\mathsf{Neig}(Y) \setminus S'$ as the latent set, constructed in line-7, we have $S^i, S^j \subseteq S \setminus \mathsf{Neig}(Y)$ are equivalent in M_G if and only if $S^i \cup S'$, $S^j \cup S'$ are equivalent in G.

Formally, denote $\mathsf{R}^{i\prime}$ as the set attained via removing T^i[2] from each element of R^i, denote M_G^i as the MAG constructed in line-7 with T^i as the selection set, and P^i as the set attained via adding T^i to each subset in each equivalence class in

[2]Note that the T^i here equals the S^0 in the i-th loop, in line-6 of Alg. 2.

$\mathsf{R}^{i\prime}/\sim_{M^i_G}$, we have:

$$\mathsf{R}_i/\sim_G = \mathsf{P}_i. \tag{33}$$

Then, by Eq. 32 and Eq. 33, we have:

$$\mathsf{Pow}(S)/\sim_G = \cup_{i=1}^{2^{\deg(Y)}} \mathsf{P}_i. \tag{34}$$

Since $l_{M^i_G} \leq l$, by the induction hypothesis, we have $\mathsf{P}_i = \textbf{recover}(M^i_G)$. According to lines 9 and 10 of the Alg. 2, we have $\mathsf{Pow}(S)/\sim_G = \cup_{i=1}^{2^{\deg(Y)}} \textbf{recover}(M^i_G) = \textbf{recover}(G)$. □

E. Complexity analysis

We first introduce some notations and definitions that will be used in this section. We omit the subscript and denote G_S as G, d_S as d for brevity. We use the subscript $X_i, X_j \in \mathsf{X}$, $V_i, V_j \in \mathsf{V}$ to denote variables and vertices; the superscript $S', S'' \subseteq S$, $\mathsf{X}^i, \mathsf{X}^j \subseteq \mathsf{X}$, and $\mathsf{V}^i, \mathsf{V}^j \subseteq \mathsf{V}$ to denote variable and vertex subsets. For a vertex V_i, denote $\deg(V_i) := |\mathsf{Neig}(V_i)|$ as its degree. Unless otherwise specified, the causal graph in this section can be either a DAG or a Maximal Ancestral Graph (MAG). In a MAG, we use C, L to denote the selection set and the latent set, respectively. In a causal graph, we use $*-*$ to denote an edge with any possible orientation ($\rightarrow, \leftarrow$ for a DAG; $\rightarrow, \leftarrow, \leftrightarrow, -$ for a MAG).

A chunk vertex is a vertex of degree 2. Recall a chain vertex if a vertex of degree ≤ 2. A path $p := < V_1, V_2, ..., V_l >$ is a sequence of distinct vertices with V_i being adjacent to V_{i+1} for $i = 1, 2, ..., l-1$. The length of the path p is l. The path p can be *blocked* by a vertex set V^i means it can be *d*-separated by V^i when G is a DAG, and *m*-separated by V' when G is a MAG. A tree is an undirected graph in which any two vertices are connected by exactly one path. In a rooted tree, the distance of a vertex V_i to the root is the length of the path between them. The parent of a vertex V_i is the vertex connected to V_i on the path to the root. A child of a vertex V_i is a vertex of which V_i is the parent. A leaf is a vertex with no child. An internal vertex is a vertex that is not a leaf.

We represent time complexity with the following notions:

1. the Big-O notation $f(d) = O(g(d))$, which means f is bounded above by g asymptotically, *i.e.*, $\forall k > 0, \exists\, d_0, \forall d > d_0, |f(d)| \leq k g(d)$.
2. the Small-ω notation $f(d) = \omega(g(d))$, which means f dominates g asymptotically, *i.e.*, $\forall k > 0, \exists\, d_0, \forall d > d_0, f(d) > k g(d)$.
3. the Big- notation $f(d) = (g(d))$, which means f and g have asymptotically the same rank, *i.e.*, $\exists\, k_1 > 0, \exists\, k_2 > 0, \exists\, d_0, \forall d > d_0, k_1 g(d) \leq |f(d)| \leq k_2 g(d)$.
4. $f = \mathsf{P}(d)$ if f has a polynomial complexity w.r.t. d, $f = \mathsf{NP}(d)$ if the complexity is larger than any polynomial function.

E.1. Complexity of Alg. 2: Equivalence classes recovery

We first introduce the following lemma, which studies the number of leaf vertices in a tree.

Proposition E.1 (Number of leaf vertices in a tree)**.** *In a tree, denote d_L as the number of leaf vertices, $d_{>2}$ as the number of non-chain vertices. Then, we have $d_L \geq d_{>2} + 2$.*

Proof. Denote d_T as the number of all vertices, then, by the handshaking lemma, we have:

$$d_L + 2(d_T - d_L - d_{>2}) + 3d_{>2} \leq \sum_{i=1}^{d_T} \deg(V_i) = 2(d_T - 1), \tag{35}$$

which indicates $d_L \geq d_{>2} + 2$. □

Proposition E.2. *The time complexity of Alg. 2 is (N_G), hence it can be bounded by $O(N_G)$.*

Proof. Alg. 2 is a recursive algorithm, its complexity is decided by the size of the recursion tree.

Specifically, in the recursion tree of Alg. 2, the number of all vertices d_T equals to the complexity of Alg. 2, while the number of leaf vertices d_L equals to N_G. Each internal vertex in the recursion tree has at least two children because the for-loop in line 6 executes at least twice. Since each internal vertex also has a parent, its degree > 2. Then, by Lemma E.1, d_T is at most twice as d_L. Hence, the complexity of Alg. 2 is (N_G). □

E.2. Preliminary results for complexity analysis

Lemma E.3. *If $f(d) = \omega(\log(d))$, then $2^{f(d)} = \omega(d^m)$ for any constant m. In other words, $2^{f(d)} = \mathrm{NP}(d)$.*

Proof. By the definition of $f(d) = \omega(\log(d))$, $\forall k + 1 > 0, \exists d_0$ such that $\forall d > d_0 f(d) > (k + 1)(\log(d)) = k\log(d) + log(d)$. As a result, $\forall k > 0, \forall m + 1 > 0, \exists d_1 := \max\{d_0, \log(k)\}$ such that $\forall d > d_1, f(d) > m\log(d) + \log(d) > m\log(d) + \log(k)$, which is equivalent to have $2^{f(d)} > kn^m$. Thus, we have $2^{f(d)} = \omega(d^m)$ by definition. □

Claim E.4 (Chain). For any causal graph G whose skeleton is a chain, *i.e.*, $Y *\!\!-\!\!* X_d *\!\!-\!\!* X_{d-1} *\!\!-\!\!* \cdots *\!\!-\!\!* X_1$, we have $N_G = d + 1$.

Proof. We prove this claim with Alg. 2 and an induction on d.

Base. $d = 1$, $N_G = 2 = d + 1$.

Induction hypotheses. Suppose $N_G = d + 1$ holds for any chain with d vertices.

Step. For a chain with $d + 1$ vertices. We consider the case when X_{d+1} is a collider (similarly a non-collider). With $\{X_{d+1}\}$ as the selection set, the induced MAG is $Y *\!\!-\!\!* X_d *\!\!-\!\!* X_{d-1} *\!\!-\!\!* \cdots *\!\!-\!\!* X_1$, which is a chain with d vertices and has $d + 1$ equivalence classes by the induction hypotheses. With $\emptyset$ as the selection set, the induced MAG is $Y \;\; X_d *\!\!-\!\!* X_{d-1} *\!\!-\!\!* \cdots *\!\!-\!\!* X_1$ and has 1 equivalence class. Therefore, we have $N_G = d + 1 + 1 = d + 2$. □

Claim E.5 (Circle). For any causal graph G whose skeleton is a circle, *i.e.*, $Y *\!\!-\!\!* X_d *\!\!-\!\!* X_{d-1} *\!\!-\!\!* \cdots *\!\!-\!\!* X_1$ and $Y *\!\!-\!\!* X_1$, we have $N_G = (d^2 + d + 2)/2 = \quad (d^2)$.

Proof. We prove the claim with Alg. 2 and Claim E.4. Denote a circle with d vertices as G_d.

Consider the case when X_d is a collider (similarly a non-collider). With $\{X_{d+1}\}$ as the selection set, the induced MAG is $Y *\!\!-\!\!* X_d *\!\!-\!\!* \cdots *\!\!-\!\!* X_1$ and $Y *\!\!-\!\!* X_1$, *i.e.*, a circle with $d - 1$ vertices. With $\emptyset$ as the selection set, the induced MAG is $Y *\!\!-\!\!* X_1 *\!\!-\!\!* \cdots *\!\!-\!\!* X_{d-1}$, *i.e.*, a chain with $d - 1$ vertices. Hence, we have $N_{G_d} = d + N_{G_{d-1}}$, which means $\{N_{G_d}\}_d$ is an arithmetic sequence and $N_{G_d} = \quad (d^2)$. □

Lemma E.6 (Adding/deleting an edge)**.** *For any causal graph G, adding an edge does not decrease N_G, deleting an edge does not increase N_G.*

Proof. For a causal graph G_0, add an edge in it and call the resulting graph G_1 (which can also be viewed as deleting an edge in G_1 and getting a graph G_0). We prove $N_{G_0} \leq N_{G_1}$ by showing $\forall\, S', S'', S' \not\sim_{G_0} S'' \Rightarrow S' \not\sim_{G_1} S''$.

Prove by contradiction. Suppose there are S', S'' such that $S' \not\sim_{G_0} S''$ and $S' \sim_{G_1} S''$. By $S' \sim_{G_1} S''$, we have $\exists\, S_\cap \subseteq_{G_1} S' \cap S''$ such that $Y \perp\!\!\!\perp_{G_1} \mathbf{X}_{S_1^c} | \mathbf{X}_{S_\setminus}$. Because adding an edge does not change the vertex sets, we have $S_\cap \subseteq_{G_0} S' \cap S''$. Because $S' \not\sim_{G_0} S''$, we have $Y \not\perp\!\!\!\perp_{G_0} \mathbf{X}_{S_1^c} | \mathbf{X}_{S_\setminus}$. In other words, there is a path p in G_0 between Y and $\mathbf{X}_{S_1^c}$ such that p can not be blocked by $\mathbf{X}_{S_\setminus}$.

In the following, we show that in G_1 the path p can not be blocked by $\mathbf{X}_{S_\setminus}$, neither; which contradicts with $Y \perp\!\!\!\perp_{G_1} \mathbf{X}_{S_1^c} | \mathbf{X}_{S_\setminus}$. Specifically, p can not be blocked by $\mathbf{X}_{S_\setminus}$ in G_0 means $\mathbf{X}_{S_\setminus}$ does not contain any non-collider on p, and $\mathbf{X}_{S_\setminus}$ contains every collider (or its descendants) on p in G_0. Because in G_1, p is still a path between Y and $\mathbf{X}_{S_1^c}$, and any collider X_i on p in G_0 is still a collider on p in G_1. Any vertex in $\mathrm{De}(X_i)$, where X_i is a collider on p in G_0, is still a descendant of the collider on p in G_1. Any non-collider on p in G_0 is still a non-collider on p in G_1. We have the path p can not be blocked by $\mathbf{X}_{S_\setminus}$ in G_1, neither. □

Lemma E.7 (Melting property)**.** *For a causal graph G over $\mathbf{X} \cup Y$. Consider three three disjoint non-empty vertex sets* C*,* L*, and* O $:=$ X$\setminus$(L $\cup$ C)*. Let M_G be the MAG constructed*[3] *over* O*, with* C *as the selection set,* L *as the latent set. Then, we have $N_G > N_{M_G}$.*

Proof. Recall that Alg. 2 traverses over every $S' \subseteq \mathrm{Neig}(Y)$ and constructs $2^{\deg(Y)}$ MAGs, and N_G is the summation of the number of equivalence classes in the $2^{\deg(Y)}$ MAGs.

[3]We say that the M_G is constructed from G via "melting" vertices in $\mathbf{C}$ and $\mathbf{L}$.

Now, modified Alg. 2 in the following way. For element $S' \subseteq \text{Neig}(Y)$, if S' matches $< \mathbf{C}, \mathbf{L} >$[4], construct a MAG and recover the equivalence classes in it; Otherwise, ignore S' and continue. In this regard, the modified algorithm recovers the equivalence classes in M_G. Since parts of the $2^{\deg(Y)}$ MAGs are ignored, we have $N_G > N_{M_G}$. □

Claim E.8 (Complexity of tree). For any causal graph G whose skeleton is a tree with d_L leaves, $N_G = \omega(c^{d_L})$ for some $1 < c < 2$.

Proof. We first prove the following claim. Suppose the skeleton of G is a tree with d_L leaves, every internal vertex of the tree is a non-chain vertex, then $N_G = \omega(c^{d_L})$ for some $1 < c < 2$.

Recall that an inducing path p with respect to $< \mathbf{C}, \mathbf{L} >$ between V_1 and V_2 is a path where every non-endpoint vertex on p is either in $\mathbf{L}$ or a collider, and every collider on p is an ancestor[5] of either V_1, V_2, or a member of $\mathbf{C}$. Two vertices in the MAG are adjacent if there is an inducing path between them with respect to $< \mathbf{C}, \mathbf{L} >$.

1. To show $N_G = \omega(c^{d_L})$, we can use Lemma E.7 and show $\exists$ sets $\mathbf{O}, \mathbf{C}, \mathbf{L}$ such that:

(a) there is an inducing path w.r.t. $< \mathbf{C}, \mathbf{L} >$ between Y and every vertex in $\mathbf{O}$, and

(b) $|\mathbf{O}| = \quad (d_L)$.

2. Put vertices in G into different layers according to their distances from Y

Algorithm 6 Rules to adjust the sets.

1: **if** V_L is Y **then**
2: O.add($X_{i_1}, ..., X_{i_r}$).
3: **else if** $V_L \in$ O **then**
4: **if** V_L is a complete collider **then**
5: O.remove(V_L), C.add(V_L).
6: O.add($X_{i_1}, ..., X_{i_r}$).
7: **else if** V_L is a complete non-collider **then**
8: O.remove(V_L), L.add(V_L).
9: O.add($X_{i_1}, ..., X_{i_r}$).
10: **else**
11: **if** r = 2 **then**
12: suppose E_{i_1} has a tail on V_L, E_{i_2} has an arrowhead on V_L.
13: O.add(X_{i_2}).
14: **if** E_{i_1} has a tail on X_{i_1} **then**
15: O.remove(V_L), L.add(V_L).
16: O.add(X_{i_1}).
17: **else**
18: keep V_L in O.
19: C.add(X_{i_1}).
20: **end if**
21: **else**
22: suppose E_{i_1} has a tail on V_L.
23: O.remove(V_L), L.add(V_L).
24: **if** E_{i_1} has a tail on X_{i_1} **then**
25: O.add($X_{i_1}, ..., X_{i_r}$).
26: **else**
27: C.add(X_{i_1}).
28: O.add($X_{i_2}, ..., X_{i_r}$)
29: **end if**
30: **end if**
31: **end if**
32: **else**
33: **if** V_L is a complete collider **then**
34: keep V_L in C.
35: O.add($X_{i_1}, ..., X_{i_r}$).
36: **else**
37: suppose E_{i_1} has a tail on V_L.
38: C.remove(V_L), L.add(V_L).
39: **if** E_{i_1} has a tail on X_{i_1} **then**
40: O.add($X_{i_1}, ..., X_{i_r}$).
41: **else**
42: C.add(X_{i_1}).
43: O.add($X_{i_2}, ..., X_{i_r}$).
44: **end if**
45: **end if**
46: **end if**

When $r \geq 3$ (line 21 to line 30), because V_L is neither a complete collider nor a complete non-collider, one edge of $E_{i_1}, ..., E_{i_r}$ has a tail on V_L. Without loss of generality, we suppose this edge is E_{i_1}. Then, similarly to the scenario where r = 2, we put V_L into L, X_{i_1} into O if E_{i_1} is $V_L - X_{i_1}$ and into C if $V_L \to X_{i_1}$. This makes sure the existence of inducing paths between Y and vertices newly added to O (**2.** (a) holds). In addition, the number of leaves increases by $r - 1$, and $|\mathsf{O}|$ increases by at least $r - 2$, which indicates **2.** (b) holds.

iii) Rule-3 (line 33 to line 45). Firstly note that if V_L is not in O

Back to T, we have at least $d_2 - 8d_1$ vertices having the following properties: **i)** They are chunk vertices in T, and **ii)** both of their two neighbors are chunk vertices in T. Call such vertices pipe vertices.

Now we discuss the number of leaves in G's maximum leaf spanning tree.

Suppose T is a maximum leaf spanning tree. Then, there are at least $d_2 - 8d_1$ pipe vertices in T. Let U be a pipe vertex in T and consider any edge $U *\!\!-\!\!* V$ from U that is not in T. We show V must be a leaf in T. Prove by contradiction. Suppose V is not a leaf in T, then add the edge $U *\!\!-\!\!* V$ into T and delete one of the other edges incident to U to break the cycle (so the result is still a spanning tree). This makes one of U's neighbors in T a leaf, which indicates the number of leaves increases. Because T is a maximum spanning tree, this is a contradiction and V is a leaf in T. To conclude, for every pipe vertex U, every edge incident to U except the two in T goes to a leaf in T.

Each pipe vertex in T is a non-chain vertex in G, so we have at least $d_2 - 8d_1$ pipe vertices in T having deg ≥ 3 in G. As a result, there are at least $d_2 - 8d_1$ edges incident to pipe vertices to the d_1 leaves in T.

Next, we prove $d_2 - 8d_1 \leq d_1$, which indicates $d_1 \geq \frac{1}{9}d_2$ and together with Lemma E.1 indicates $d_1 = \quad (d)$. Prove by contradiction. Suppose $d_2 - 8d_1 > d_1$. Then at least one leaf in T, say V, is connected to two pipe vertices in T, say U_1, U_2. Then, add the edges $V *\!\!-\!\!* U_1$, $V *\!\!-\!\!* U_2$ to T and delete one of the incident edges of U_1, U_2 each, we lose one leaf vertex V in T, however, obtain two more (one of U_1's neighbors and one of U_2's neighbors), which contradict with T being a maximum leaf spanning tree. □

Lemma E.10. *For a connected causal graph G, if $d_{>2} = \omega(\log(d))$, then $N_G = \mathsf{NP}(d)$.*

Proof. We prove the lemma by showing that any G with $d_{>2} = \omega(\log(d))$ has a maximum leaf spanning tree with $\omega(d_{>2})$ leaves. In the regard, by Lemma E.8, the maximum leaf spanning tree has at least $\omega(2^{\log(d)}) = \mathsf{NP}(d)$ equivalence classes. Then, we can delete the edges in G until G becomes its maximum leaf spanning tree and use Lemma E.6 to show $N_G = \mathsf{NP}(d)$.

We first construct a lower bound graph $\underline{G}$ of G such that $N_{\underline{G}} \leq N_G$, by keeping all vertices of deg$=1$, deg>2, and removing all the chunk vertices which have deg$=2$. Specifically, use Lemma E.7 and iteratively melt the chunk vertex X_i, whose two neighbors denoted as A_i, B_i, with the following rules: **i)** If $\deg(A_i) = 1$, or $\deg(B_i) = 1$, or $\deg(A_i) = \deg(B_i) = 2$, or $\deg(A_i) = \deg(B_i) = 3$, put X_i into L if it is a non-collider, C if otherwise. **ii)** Otherwise, one of $\deg(A_i), \deg(B_i)$ is 2, the other one is 3. Without loss of generality, suppose $\deg(A_i) = 2, \deg(B_i) = 3$. If A_i is adjacent to B_i and A_i, X_i, B_i form a cycle, then delete the edge $A_i *\!\!-\!\!* X_i$. Otherwise, melt the vertex X_i (put it into L if it is a non-collider, C if otherwise).

Next, we show that the $\underline{G}$ is a connected graph with at least $d_{>2}$ vertices, such that every vertex in $\underline{G}$ is either of deg$=1$ or a non-chain vertex. This is because all vertices of deg$=1$ in G are still of deg$=1$ in $\underline{G}$, all non-chain vertices in G are still of deg≥ 3 in $\underline{G}$.

Hence, $\underline{G}$ has a maximum leaf spanning tree $\underline{T}$ with at least $\quad (d_{>2})$ leaves, according to Lemma E.9. By Claim E.8, we have $N_{\underline{T}} = \omega(c^{d_{>2}})$ for some $1 < c < 2$. By Lemma E.7, we have $N_G > N_{\underline{G}} > N_{\underline{T}}$, which together with $d_{>2} = \omega(\log(d))$ and Lemma E.3 indicates $N_G = \mathsf{NP}(d)$. □

Lemma E.11. *Consider two vertex sets $\mathbf{V}^i, \mathbf{V}^j$, and a path p between two vertices V_1, V_l. If p can be blocked by $\mathbf{V}^i$, but can not be blocked by $\mathbf{V}^i \cup \mathbf{V}^j$, then, we have $V_1 \not\perp\!\!\!\perp_G \mathbf{V}^j | \mathbf{V}^i$.*

Proof. To prove the lemma, we construct a path p_1 between V_1 and a vertex in $\mathbf{V}^j$ such that p_1 can not be blocked by $\mathbf{V}^i$.

We first prove the following properties **i)**-**iv)**:

i) p must contain a collider. Prove by contradiction. Suppose all vertices on p are non-colliders. Then, since $\mathbf{V}^i$ can block p, we have $\mathbf{V}^j$ contains at least one non-collider on p. Hence, the union set $\mathbf{V}^i \cup \mathbf{V}^j$ also contains a non-collider on p. Therefore, p can be blocked by $\mathbf{V}^i \cup \mathbf{V}^j$, which is a contradiction.

ii) In a similar way, we can prove that $\mathbf{V}^i$ and $\mathbf{V}^i \cup \mathbf{V}^j$ do not contain any non-collider on p.

iii) Since p can be blocked by $\mathbf{V}^i$ and **ii)**, we have: $\exists$ a collider on p such that the collider and its descendants are all in $\mathbf{V}^i$.

iv) For any collider V_c on p, if V_c and any vertex in $\mathsf{De}(V_c)$ are all not in $\mathbf{V}^i$, then, either V_c or a vertex in $\mathsf{De}(V_c)$ is in $\mathbf{V}^j$. This is because if otherwise, V_i and all vertices in $\mathsf{De}(V_i)$ are not in $\mathbf{V}^i \cup \mathbf{V}^j$. Therefore, the path p can be blocked by

$\mathbf{V}^i \cup \mathbf{V}^j$, which is a contradiction.

We then construct the path p_1 in the following way:

Denote those colliders on p such that themselves and their descendants are all not in $\mathbf{V}^i$ as (in the order of their distance to V_1) as $\{V_{c_1}, V_{c_2}, ..., V_{c_l}\}$.

Now, consider the subpath p' of p with $p' :=< V_1, V_2, ..., V_{c_1-1}, V_{c_1} >$. We have the following analyses: **i)** By the definition of V_{c_1}, among $V_1, V_2, ..., V_{c_1-1}$, all colliders and their descendants are in $\mathbf{V}^i$. **ii)** Among $V_1, V_2, ..., V_{c_1-1}$, all non-colliders are not in $\mathbf{V}^i$. **iii)** Either V_{c_1} or a vertex in $\mathbf{De}(V_{c_1})$ is in $\mathbf{V}^j$.

If it is $V_{c_1} \in \mathbf{V}^j$, then we have the path $p_1 =< V_1, V_2, ..., V_{c_1} >$ between V_1 and a vertex in $\mathbf{V}^j$ satisfying that p_1 can not be blocked by $\mathbf{V}^i$, because of **i)** and **ii)**; If it is a vertex in $\mathbf{De}(\mathbf{V}^j)$ that is in $\mathbf{V}^j$, then, we have the path $p_1 =< V_1, V_2, ..., V_{c_1} \to \cdots \to V_j)$ for $V_j \in \mathbf{De}(V_{c_1} >$, between V_1 and a vertex in $\mathbf{V}^j$ satisfying that p_1 can not be blocked by $\mathbf{V}^i$, because of **i)**, **ii)**, and the definition of V_{c_1}. □

Lemma E.12 (Merging property)**.** *For any causal graph G where Y is adjacent to a vertex X_0, and vertices in $\mathbf{V}\backslash\{Y, X_0\}$ are adjacent to at most one vertex in $\{Y, X_0\}$, merge*[7] *Y, X_0 into a new vertex Υ and denote the resulted graph as $\tilde{G}$. Then, we have $N_{\tilde{G}} + 1 \leq N \leq 2N_{\tilde{G}}$.*

Proof. During the proof, we omit the subscript and denote N_G as N, $N_{\tilde{G}}$ as $\tilde{N}$, $\mathbf{X}_{S_i}$ as $\mathbf{X}^i$ for brevity.

Proof of the right side. We show for any $\mathbf{T} \subseteq \{X_0\}$, $\mathbf{X}^i, \mathbf{X}^j \subseteq \mathbf{V}\backslash\{Y, X_0\}$, if $\mathbf{X}^i \sim_{\tilde{G}} \mathbf{X}^j$, then $\mathbf{X}^i \cup \mathbf{T} \sim_G \mathbf{X}^j \cup \mathbf{T}$. In this regard, for any subset $\mathbf{T}$, there are at most $\tilde{N}$ equivalent classes in G, thus $N \leq 2^{|\mathbf{T}|}\tilde{N} = 2\tilde{N}$.

1. By $\mathbf{X}^i \sim_{\tilde{G}} \mathbf{X}^j$, we have $\exists\, \mathbf{X}^{ij} \subseteq \mathbf{X}^i \cap \mathbf{X}^j$ such that $\Upsilon \perp\!\!\!\perp_{\tilde{G}} \mathbf{X}^{(ij)^c} | \mathbf{X}^{ij}$. In other word, we have $\{Y, X_0\} \perp\!\!\!\perp_G \mathbf{X}^{(ij)^c} | \mathbf{X}^{ij}$.

2. When $\mathbf{T} = \emptyset$, by **1.**, we have $Y \perp\!\!\!\perp_G \mathbf{X}^{(ij)^c} | \mathbf{X}^{ij}$ and thus $\mathbf{X}^i \sim_G \mathbf{X}^j$ holds.

3. When $\mathbf{T} = \{X_0\}$, by **1.**, we have: **i)** any path in G between $\mathbf{X}^{(ij)^c}$ and Y can be blocked by $\mathbf{X}^{ij}$, and **ii)** any path in G between $\mathbf{X}^{(ij)^c}$ and $\mathbf{T}$ can be blocked $\mathbf{X}^{ij}$. Next, we show any path in G between $\mathbf{X}^{(ij)^c}$ and Y can also be blocked by $\mathbf{X}^{ij} \cup \mathbf{T}$, which indicates $\mathbf{X}^i \cup \mathbf{T} \sim_G \mathbf{X}^j \cup \mathbf{T}$.

Prove by contradiction. Suppose there is a path between $\mathbf{X}^{(ij)^c}$ and Y that can be blocked by $\mathbf{X}^{ij}$ and can not be blocked by $\mathbf{X}^{ij} \cup \mathbf{T}$. By Lemma. E.11, we can construct a path between $\mathbf{X}^{(ij)^c}$ and $\mathbf{T}$ such that it can not be blocked by $\mathbf{X}^{ij}$, which contradicts with **3. ii)**.

Proof of the left side. We first prove the in-equation under the case when X_0 is a complete collider[8]. With $\{X_0\}$ as the selection set, the induced MAG is $\tilde{G}$, since there is at least one equivalent class when not conditioning on X_0, we have $\tilde{N} + 1 \leq N$ by Alg. 2 and Lemma E.7.

For the cases when X_0 is a complete non-collider, or X_0 is a partial collider and $\exists\, X_i \in \mathbf{De}(X_0)$ such that X_i is incident to a tail-tail[9] edge, we can prove $\tilde{N} + 1 \leq N$ in a similar way.

Next, we discuss the case where X_0 is a partial collider and $\forall X_i \in \mathbf{De}(X_0)$, X_i is not incident to a tail-tail edge. We first show the following properties **1.** and **2.**.

1. In G, for two vertex sets $\mathbf{X}^i, \mathbf{X}^j$, if $X_0 \in \mathbf{X}^i$ and $X_0 \notin \mathbf{X}^j$, then $\mathbf{X}^i \not\sim_G \mathbf{X}^j$. This is because Y is adjacent to X_0 in G.

Further, we show for two vertex sets $\mathbf{X}^i, \mathbf{X}^j$, if $\mathbf{X}^i \cap \mathbf{De}(X_0) \neq \emptyset$ and $\mathbf{X}^j \cap \mathbf{De}(X_0) = \emptyset$, then $\mathbf{X}^i \not\sim_G \mathbf{X}^j$. This is proved as follows. For $X_i \in \mathbf{X}^i \cap \mathbf{De}(X_0)$, there is a path $p :=< Y * \to X_0 \to \cdots \to X_i >$ from Y to X_i. Since $\mathbf{X}^j \cap \mathbf{De}(X_0) = \emptyset$, $\forall\, \mathbf{X}^{ij} \subseteq \mathbf{X}^i \cap \mathbf{X}^j$, we have $\mathbf{X}^{ij} \cap \mathbf{De}(X_0) = \emptyset$ and $X_i \in \mathbf{X}^{(ij)^c}$. As a result, $\forall\, \mathbf{X}^{ij} \subseteq \mathbf{X}^i \cap \mathbf{X}^j$, there is a path p between Y and $\mathbf{X}^{(ij)^c}$ such that p can not be blocked by $\mathbf{X}^{ij}$, *i.e.*, $\mathbf{X}^i \not\sim_G \mathbf{X}^j$.

In $\tilde{G}$, similarly, for two vertex sets $\mathbf{X}^i, \mathbf{X}^j$, if $\mathbf{X}^i \cap \mathbf{De}(X_0) \neq \emptyset$ and $\mathbf{V}_j \cap \mathbf{De}(X_0) = \emptyset$, then $\mathbf{X}_i \not\sim_{\tilde{G}} \mathbf{X}^j$.

[7] The merging operation means contradicting the edge $(Y; X_0)$ and merging $Y; X_0$ into a new vertex $\tilde{Y}$. Edges incident to $\tilde{Y}$ in $\tilde{G}$ are edges incident to either Y or X_0 in G, their orientations on the $\tilde{Y}$ side can be randomly assigned, and do not influence N_G, while orientations on the other side keep the same as in G.

[8] X_0 is called a complete (non-)collider if it is a (non-)collider on any path $p :=< X_i; X_0; Y >$ with $X_i \in \mathbf{Neig}_G(X_0)\backslash\{Y_0\}$.

[9] A tail-tail edge is an edge $*-*$ with orientations at both sides being tails, *i.e.*, $-$. According to the definition of a tail-tail edge (Zhang, 2008), a vertex is incident to a tail-tail edge means it is an ancestor of a member of the selection set.

2. In G, by **1.**, divide subsets of $\mathbf{X}$ into those that contain X_0 and those that do not contain X_0. Denote the number of equivalent classes in them as N_1, N_2, respectively, we have $N = N_1 + N_2$. Further, divide those subsets that do not contain X_0 into those that have an intersection with $\mathbf{De}(X_0)$ and those that have no intersection with $\mathbf{De}(X_0)$. Denote the number of equivalent classes in them as N_3, N_4, respectively. We have $N_2 = N_3 + N_4$ and thus $N = N_1 + N_3 + N_4$.

Similarly, in $\bar{G}$, divide subsets of $\mathbf{X}\backslash\{X_0\}$ into those that have an intersection with $\mathbf{De}(X_0)$ and those that have no intersection with $\mathbf{De}(X_0)$. Denote the number of equivalent classes in them as $\bar{N}_1, \bar{N}_2$, respectively. We have $\bar{N} = \bar{N}_1 + \bar{N}_2$.

It is straightforward to have $N_4 \geq 1$. In the following, we will show $N_1 \geq \bar{N}_2, N_3 = \bar{N}_1$ and thus $N \geq \bar{N} + 1$.

Claim. 1. For two subsets of vertex $\mathbf{X}^i, \mathbf{X}^j$ such that $\mathbf{X}^i \cap (X_0 \cup \mathbf{De}(X_0)) = \emptyset$ and $\mathbf{X}^j \cap (X_0 \cup \mathbf{De}(X_0)) = \emptyset$, then $\mathbf{X}^i \sim_{\tilde{G}} \mathbf{X}^j \Leftrightarrow \mathbf{X}^i \cup X_0 \sim_G \mathbf{X}^j \cup X_0$, which indicates $N_1 \geq \bar{N}_2$.

Proof of **Claim. 1**. $\Rightarrow$ can be proved similarly as the proof of the right side.

$\Leftarrow$ Suppose $\mathbf{X}^i \cup X_0 \sim_G \mathbf{X}^j \cup X_0$, we will show $\mathbf{X}^i \sim_{\tilde{G}} \mathbf{X}^j$, given the fact that $\mathbf{X}^i, \mathbf{X}^j$ do not contain X_0 nor its descendants, and any member of $\{X_0\} \cup \mathbf{De}(X_0)$ is not incident to a tail-tail edge.

1. For $X_i \in \mathbf{De}(X_0)$ and $X_j \notin \mathbf{De}(X_0)$, since $X_i - X_j$ and $X_i \to X_j$ is not allowed, it must be $X_i \leftarrow * X_j$. Similarly, we have $X_0 \to X_i$ and $X_0 \leftarrow * X_j$.

2. By $\mathbf{X}^i \cup X_0 \sim_G \mathbf{X}^j \cup X_0$, we have $\exists\, \mathbf{X}^{ij} \subseteq (\mathbf{X}^i \cup X_0) \cap (\mathbf{X}^j \cup X_0)$ such that $Y \perp\!\!\!\perp_G (\mathbf{X}^i \cup \mathbf{X}^j \cup X_0) \backslash \mathbf{X}^{ij} | \mathbf{X}^{ij}$. Since Y is adjacent to X_0, $\mathbf{X}^{ij}$ must contain X_0. That is, $\exists\, \mathbf{X}^{ij} \subseteq \mathbf{X}^i \cap \mathbf{X}^j$ such that $Y \perp\!\!\!\perp_G (\mathbf{X}^i \cup \mathbf{X}^j \cup X_0) \backslash (\mathbf{X}^{ij} \cup X_0) | \mathbf{X}^{ij} \cup X_0$, which is equivalent to $Y \perp\!\!\!\perp_G \mathbf{X}^{(ij)^c} | \mathbf{X}^{ij} \cup X_0$.

3. We first show $Y \perp\!\!\!\perp_G \mathbf{X}^{(ij)^c} | \mathbf{X}^{ij}$, which is equivalent to showing any path between Y and $\mathbf{X}^{(ij)^c}$ can be blocked by $\mathbf{X}^{ij}$. Prove by contradiction. Suppose there is a path $p_0 := < X_{k_1}, X_{k_2}, ..., Y >$ between Y and $\mathbf{X}^{(ij)^c}$ that can be blocked by $\mathbf{X}^{ij} \cup X_0$, and can not be blocked by $\mathbf{X}^{ij}$. By Lemma. D.1, the set $\{X_0\}$ contains a non-collider on p_0, which means X_0 is a non-collider on p_0.

Then, X_0 is incident to at least one tail on p_0, since $X_0 \leftarrow * X_j$ for $X_j \notin \mathbf{De}(X_0)$, p_0 must contain a member of $\mathbf{De}(X_0)$ and $p_0 =< X_{k_1}, X_{k_2}, ..., X_0 \to X_i, ..., Y >$ for $X_i \in \mathbf{De}(X_0)$. Since Y is not a member of $\mathbf{De}(X_0)$, there is $X_0 \to \cdots \to X_i \leftarrow * X_j$ for $X_i \in \mathbf{De}(X_0)$ and $X_j \notin \mathbf{De}(X_0)$ on p_0. As a result, p_0 contains a collider that itself nor its descendants are in $\mathbf{X}^{ij}$. This means p_0 can be blocked by $\mathbf{X}^{ij}$ and thus a contradiction.

4. We then show $X_0 \perp\!\!\!\perp_G \mathbf{X}^{(ij)^c} | \mathbf{X}^{ij}$, which together with **3.** means $\{Y, X_0\} \perp\!\!\!\perp_G \mathbf{X}^{(ij)^c} | \mathbf{X}^{ij}$ and thus $\mathbf{X}^i \sim_{\tilde{G}} \mathbf{X}^j$. We show this by proving any path between $\mathbf{X}^{(ij)^c}$ and X_0 can be blocked by $\mathbf{X}^{ij}$. Prove by contradiction. Suppose there is a path $p_1 := < X_{k_1}, X_{k_2}, ..., X_{k_l}, X_0 >$ that can not be blocked by $\mathbf{X}^{ij}$.

Then, if $X_{k_l} \notin \mathbf{De}(X_0)$, we have a path $p_2 := < X_{k_1}, ..., X_{k_l} * \to X_0 \leftarrow * Y >$ such that p_2 can not be blocked by $\mathbf{X}^{ij} \cup X_0$, which contradicts with $Y \perp\!\!\!\perp_G \mathbf{X}^{(ij)^c} | \mathbf{X}^{ij} \cup X_0$. Otherwise $X_{k_l} \in \mathbf{De}(X_0)$, there is $X_{k_1}, ..., X_{k_l} \leftarrow X_0 \leftarrow * Y >$. Since $X_{k_1} \in \mathbf{X}^{(ij)^c}$ and thus $X_{k_1} \notin \mathbf{De}(X_0)$, there is

path $p_2 := < X_{k_1}, X_{k_2}, ..., X_{k_l}, X_0 \leftarrow * Y >$ constructed from p_1. If X_0 is a non-collider, since $\mathbf{X}^{ij}$ does not contain X_0, p_2 can not be blocked by $\mathbf{X}^{ij}$. Otherwise X_0 is a collider, since $\mathbf{X}^{ij}$ contains a member of $\mathbf{De}(X_0)$, p_2 can not be blocked by $\mathbf{X}^{ij}$ neither. These results contradict with $Y \perp\!\!\!\perp_G \mathbf{X}^{(ij)^c} | \mathbf{X}^{ij}$. Hence, we have $X_0 \perp\!\!\!\perp_G \mathbf{X}^{(ij)^c} | \mathbf{X}^{ij}$.

To conclude, **1.** and **2.** mean $\Leftarrow$ is true. **Claim.1** indicates $N_1 \geq \mathcal{N}_2$, **Claim.2** indicates $N_3 = \mathcal{N}_1$, and thus $N \geq \mathcal{N} + 1$. □

Corollary E.13 (Merging property for multiple vertices)**.** *For any causal graph G where Y is adjacent to a connected vertex sets $\mathbf{X}_0$, and vertices in $\mathbf{V} \backslash (\mathbf{X}_0 \cup Y)$ is adjacent to at most one vertex in $\mathbf{X}_0 \cup Y$. Merge $Y, \mathbf{X}_0$ into a new vertex $\mathcal{Y}$ and call the resulted graph $\tilde{G}$. Then, $\mathcal{N} + |\mathbf{X}_0| \leq N \leq 2^{|\mathbf{X}_0|}\mathcal{N}$.*

Proof. Proof of the right side is the same as Lemma. E.12.

Proof of the left side. Since $\mathbf{X}_0$ is a connected set and Y is adjacent to $\mathbf{X}_0$, we can delete edges among $\mathbf{X}_0 \cup Y$ until the subgraph over $\mathbf{X}_0 \cup Y$ becomes a spanning tree over $\mathbf{X}_0 \cup Y$ with Y as the root vertex. Then, iteratively merge vertices and use Lemma. E.12, Lemma E.6, we have $N \geq \mathcal{N} + |\mathbf{X}_0|$. □

E.3. Details of Prop. 5.5: Complexity

In this section, we discuss the complexity of searching N_G equivalence classes. We show that compared to the exponential cost $O(2^{d_S})$ of exhaustive search, our search strategy enjoys a polynomial cost $\mathrm{P}(d_S)$ when G_S is mainly composited of chain vertices. Our analysis mainly uses the results in Lemma E.10 and Lemma E.12. The idea is briefed as follows:

Lemma E.10 shows that any G_S with $d_{>2} = \omega(\log(d))$ has $N_G = \mathrm{NP}(d_S)$. Hence, we need to look at cases when $d_{>2} = O(\log(d))$. For these cases, Lemma E.12 shows that the non-chain vertices in G_S do not influence the rank of N_G. This is because we can iteratively merge the non-chain vertices into Y and have N_G being squeezed within $N_{\tilde{G}} \sim 2^{d_{>2}} N_{\tilde{G}}$. Since $2^{d_{>2}} = \mathrm{P}(d_S)$, we have $N_G = \mathrm{P}(d_S)$ if and only if $N_{\tilde{G}} = \mathrm{P}(d_S)$. Therefore, the rank of N_G when $d_{>2} = O(\log(d_S))$ is decided by $N_{\tilde{G}}$, in other words, by the chain vertices with deg≤ 2 in G_S.

Intuitively, when the chain vertices compose different chains that do not intersect each other, by Claim E.4, the $N_{\tilde{G}}$ is the product of the chains' lengths. Hence, the more "intensive" the chain vertices distribute, the smaller $N_{\tilde{G}}$ and thus N_G will be. In the following, we will provide a formal metric F_G to measure the intensity of chain vertices.

We first define the following structures on G_S. For brevity, we omit the subscript and denote G_S as G, d_S as d, respectively.

1. A path is a sequence of distinct vertices $< V_1, V_2, ..., V_l >$ where $V_i, V_{i+1} (i = 1, 2, ..., l-1)$ are adjacent in G. The length of the path is l. Define the distance between a vertex and Y and the length of the shortest path between them.

2. A chain is a path where every vertex on it has deg ≤ 2 in G. The head of a chain is the vertex in it that is closest to Y. A maximal chain is a chain that can not be made longer by adding new vertices.

3. For a maximal chain c, let $c \in \mathbf{Ch}'(Y)$ if there is no other maximal chain in the shortest path between the head of c and Y. For two maximal chains c_1, c_2, let $c_2 \in \mathbf{De}'(c_1)$ if there is a path between a vertex in c_2 and Y that contains c_1.

4. For a maximal chain $c \in \mathbf{Ch}'(Y)$, define a set of operations $\mathrm{opt}_c(G)$ on G. Specifically, if $\mathbf{De}'(c) = \emptyset$, define $\mathrm{opt}_c(G) := \{\text{remove } c\}$; Otherwise, define $\mathrm{opt}_c(G) := \{\text{remove } \mathbf{X}_c^{1:i} | i = 1, 2, ..., l\} \cup \{\text{replace } c \text{ with an edge}\}$, with l the number of vertices on c, X_c^i the i-th one (in the order of the distance to Y), and $\mathbf{X}_c^{1:i} := \{X_c^1, X_c^2, ..., X_c^i\}$.

5. For a maximal chain with l vertices, define cost$(c) := l + 1$ if c has one head vertex, $(l^2 + l + 2)/2$ if c has two head vertices (that is both sides of c have equal distance to Y).

Proposition 5.5 (Complexity)**.** Let F_G be an recursive metric defined over maximal chains in G:

$$F_G := \sum_{\substack{c \in \mathbf{Ch}_G^0(Y) \\ \mathbf{De}^0(c) = \emptyset}} \mathrm{cost}(c) \sum_{\mathrm{opt} \in \bigtimes_{c2 \mathbf{Ch}_G^0(Y)} \mathrm{opt}_c} F_{\mathrm{opt}(G)}.$$

Then, $N_G = \mathrm{P}(d)$ if and only if $d_{>2} = O(\log(d))$ and $F_G = \mathrm{P}(d)$.

Proof. To prove the proposition, we will show **i)** if $d_{>2} = \omega(\log(d))$, then $N_G = \mathsf{NP}(d)$; **ii)** if $d_{>2} = O(\log(d))$, then $N_G = \mathsf{P}(d) \Leftrightarrow F_G = \mathsf{P}(d)$.

Specifically, **ii)** means if $d_{>2} = O(\log(d))$ and $F_G = \mathsf{P}(d)$, then $N_G = \mathsf{P}(d)$, which shows $\Leftarrow$ of the proposition. **i)** means if $N_G = \mathsf{P}(d)$, then $d_{>2} = O(\log(d))$, together with **ii)**, it means if $N_G = \mathsf{P}(d)$, then $d_{>2} = O(\log(d))$ and $F_G = \mathsf{P}(d)$, which shows $\Rightarrow$ of the proposition.

The proof of **i)** is at Lemma. E.10. The proof of **ii)** is as follows:

Claim. 1. $F_G \leq N_G \leq 2^{d_{>2}} F_G$.

Proof of **Claim. 1**. Modify Alg. 2 in the following way:

i) Add before line-2: if $\mathsf{Neig}(Y)$ contains non-chain vertices, then merge them into Y until $\mathsf{Neig}(Y)$ only contains non-chain vertices. Call the resulting graph as $\tilde{G}$. **ii)** Replace the G with $\tilde{G}$, G' with $\tilde{G}'$, and N_G with $N_{\tilde{G}}$, in lines 4-12. Call the modified algorithm as $N_{\tilde{G}} = \mathrm{count}'(G)$.

Next, we first show $F_G = N_{\tilde{G}}$, then prove **Claim. 1** via the $\mathrm{count}'(G)$ algorithm.

After merging non-chain vertices around Y, in $\tilde{G}$, Y's neighbors are the head vertices of the maximal chains in $\mathsf{Ch}'(Y)$. Recursively conduct the count′ algorithm on $\tilde{G}$ and its induced MAGs $\tilde{\mathcal{M}}_G$, until all vertices in all maximal chains in $\mathsf{Ch}'(Y)$ have been traversed.

For maximal chains without descendants, since they are disjoint with the other maximal chains, we have $N_{\tilde{G}} = (\prod_{c \in \mathbf{Ch}^0_G(Y); \mathbf{De}^0_G(c)=\emptyset} \mathrm{cost}(c)) N_{\mathrm{opt}_1(\tilde{G})}$, where $\mathrm{cost}(c) = l + 1$ if c has one head vertex (see Claim E.4) and $(l^2 + l + 2)/2$ if c has two head vertices (see Claim E.5), and $\mathrm{opt}_1 := \prod_{c \in \mathbf{Ch}^0_G(Y); \mathbf{De}^0_G(c)=\emptyset} \mathrm{cost}(c)\mathrm{opt}_c$, and $\mathrm{opt}_1(\tilde{G})$ is the causal graph after removing all maximal chains without descendants.

In $\mathrm{opt}_1(\tilde{G})$, denote the remained maximal chains in $\mathsf{Ch}'(Y)$ as $\{c_i\}_{i=1:r}$ and the vertices on them as $\{X^i_1, ..., X^i_{l_i}\}_{i=1:r}$. To obtain $N_{\mathrm{opt}_1(\tilde{G})}$, for each c_i, similarly as the analysis of the Claim E.4, we need to consider the following $l_i + 1$ situations: X^i_1 is blocked[10]; X^i_1 is open, X^i_2 is blocked; ...; $X^i_1, ..., X^i_{l_i-1}$ are blocked, $X^i_{l_i}$ is open; and $X^i_1, ..., X^i_{t_i}$ are open. Because vertices in the r maximal chains are disjoint, we in total need to consider $\prod_{i=1}^{r} l_i + 1$ situations, and $N_{\mathrm{opt}_1(\tilde{G})} = \sum_{j=1}^{\prod_{i=1}^{r} l_i + 1} N_{\mathrm{opt}_1(\tilde{G})^j_j}$, where $\mathrm{opt}_1(\tilde{G})'_j$ denotes the induced subgraph from $\mathrm{opt}_1(\tilde{G})$ in the j-th situation.

Note that each subgraph $\mathrm{opt}_1(\tilde{G})'_j$ corresponds to an operation in $\prod_{c \in \mathbf{Ch}^0_G(Y)} \mathrm{opt}_c$ on $\tilde{G}$, we have $N_{\mathrm{opt}_1(\tilde{G})} = \sum_{\mathrm{opt} \in \prod_{c \in \mathbf{Ch}^0_G(Y)} \mathrm{opt}_c} N_{\mathrm{opt}(\tilde{G})}$. Hence, $N_{\tilde{G}} = \prod_{c \in \mathbf{Ch}^0_G(Y); \mathbf{De}^0_G(c)=\emptyset} \mathrm{cost}(c) \sum_{\mathrm{opt} \in \prod_{c \in \mathbf{Ch}^0_G(Y)} \mathrm{opt}_c} N_{\mathrm{opt}(\tilde{G})}$ and $N_{\tilde{G}} = F_G$.

During the recursive execution of the $\mathrm{count}'(G)$ to obtain $N_{\tilde{G}}$, there are at most $d_{>2}$ non-chain vertices merged into Y. As a result, by Lemma. E.12, we have $N_G \leq 2^{d_{>2}} N_{\tilde{G}}$. The number of non-chain vertices merged into Y is at least 0, so we also have $N_{\tilde{G}} \leq N_G$.

To conclude, we have $F_G \leq N_G \leq 2^{d_{>2}} F_G$, which means **Claim. 1.** and hence the proposition is true. □

Remark E.14. If the skeleton of G is a tree, for two maximal chains c_1, c_2, define $c_2 \in \mathsf{Ch}'(c_1)$ if c_1 contains the first non-chain vertex in the path from the head of c_2 to Y. The F_G degenerates to:

$$F_G = \prod_{\tilde{c} \in \mathbf{Ch}^0(Y)} f(\tilde{c}),$$

with $f(c) := \mathrm{cost}(c) + \prod_{\tilde{c} \in \mathbf{Ch}^0(c)} f(\tilde{c})$, $\mathrm{cost}(c) = \mathrm{len}(c) + \mathbb{1}(\mathsf{Ch}'(c) = \emptyset)$.

[10] A vertex of deg=2 is blocked if it is a non-collider and is put in the selection set, or it is a collider and is put in the latent set. A vertex is open if it is not blocked

F. Experiment

F.1. Implementation details

All codes are implemented with PyTorch 1.10 and run on an Intel Xeon E5-2699A v4@2.40GHz CPU.

Baselines.

1. Vanilla. $\mathbb{E}[Y|\boldsymbol{x}]$ is implemented by the same neural network as f_{S^0}.
2. ICP (`https://github.com/juangamella/icp`). The level of the test procedure is set to 0.05. The estimator is implemented by the same neural network as f_{S^0}.
3. IC (`https://github.com/mrojascarulla/causal_transfer_learning`). The level of the test procedure is set to 0.05. Levene test is used. The estimator is implemented by the same neural network as f_{S^0}.
4. DRO (`https://github.com/duchi-lab/certifiable-distributional-robustness`). The γ is set to 2. The estimator is implemented by the same neural network as f_{S^0}.
5. Surgery estimator. Since there is no official implementation available, we implement it based on our method. Specifically, we pick $2 \sim 3$ validation environments from $\mathcal{E}_{\text{tr}}$ and use the validation loss to select S^*.
6. IRM (`https://github.com/facebookresearch/InvariantRiskMinimization`). The best ϕ is chosen by comparing the validation loss of $\text{reg} = 0, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}$.
7. HRM (`https://github.com/LhSthu/HRM`). The cluster number is set to the number of deployment environments. σ and λ are both set to 0.1. The overall threshold for subset selection is set to 0.25.
8. IB-IRM (`https://github.com/ahujak/IB-IRM`). The λ_{ib} is set to 0.1, the λ_{irm} is set to 0.75. The estimator is implemented by the same neural network as f_{S^0}.
9. Anchor regression (`https://github.com/rothenhaeusler/anchor-regression`). The γ is set to 1.5.

Synthetic study. The neural networks to implement f_{S^0} and h are two-layers MLPs. We use a sigmoid activation function in the hidden layer to add non-linearity. We use the Adam optimizer. The learning rate is set to 0.02, and epochs are set to 10000 with an early stop.

Alzheimer's disease diagnosis. The neural networks to implement f_{S^0} and h are the same as the synthetic study. We use the SGD optimizer. For the estimation of f_{S^0}, the epochs are set to 5000, the learning rate is set to 0.25 in the first 4000 epochs, and decreased to 0.1 in the last 1000 epochs. For the estimation of $\mathcal{L}$, the epochs are set to 12000 with an early stop, the learning rate is set to 0.4.

Gene function prediction. The neural networks to implement f_{S^0} and h are the same as the synthetic study. We use the SGD optimizer. The epochs are set to 10000. For the estimation of f_{S^0}, the learning rate is set to 0.01. For the estimation of $\mathcal{L}$, the learning rate is set to 0.05.

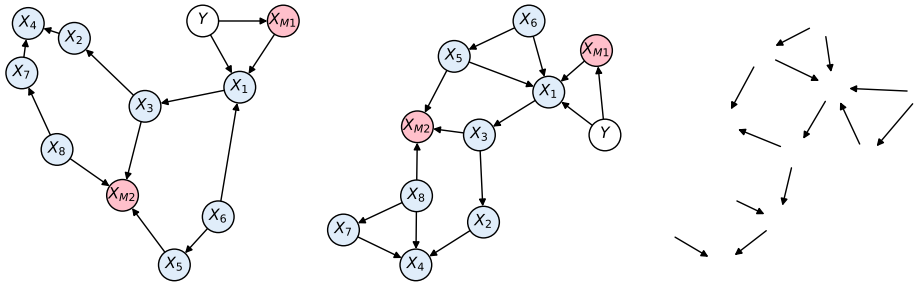


Figure 8: The synthetic causal graphs for complexity analysis. Stable and mutable variables are respectively marked blue and red. We have $d_{>2} = 1; 2; 5; 6$ in (a), (b), (c), (d), respectively. The sparse graphs (a), (b) are generated by deleting edges from Fig. 4 (a). The dense graphs (c), (d) are generated by adding edges to Fig. 4 (a).

Table 4: Indices for brain region partition.

Abbreviation	Brain region	AAL index (Tzourio-Mazoyer et al., 2002)
FSL	Frontal superior lobe	2101,2102,2111,2112,2601,2602
FML	Frontal middle lobe	2201,2202,2211,2212,2611,2612
FIL	Frontal inferior lobe	2301,2302,2311,2312,2321,2322
TSL	Temporal superior lobe	8111,8112
TML	Temporal medial lobe	8201,8202
TIL	Temporal inferior lobe	8301,8302
TP	Temporal pole	8121,8122,8211,8212
PSL	Parietal superior lobe	6101,6102
PIL	Parietal inferior lobe	6201,6202
OSL	Occipital superior lobe	5101,5102
OML	Occipital middle lobe	5201,5202
OIL	Occipital inferior lobe	5301,5302
CA	Cingulum anterior	4001,4002
CM	Cingulum middle	4011,4012
CP	Cingulum posterior	4021,4022
INS	Insula	3001,3002
AMY	Amygdala	4201,4202
CAU	Caudate	7001,7002
HP	Hippocampus	4101,4102
PAL	Pallidum	7021,7022
PUT	Putamen	7011,7012
THA	Thalamus	7101,7102

F.2. Extra results

Table 5: Max. MSE evaluation on synthetic and IMPC datasets. The first column notes the methods we compare. The second column represents the max. MSE over deployment environments. Data for Syn-a,b,c,d are respectively generated by the causal graphs (a) ,(b), (c), and (d) shown in Fig. 8. The best results are **boldfaced**.

Method	max. MSE (↓)				
	Syn-a	Syn-b	Syn-c	Syn-d	IMPC
Vanilla	$15.946_{\pm 2:7}$	$3.033_{\pm 2:7}$	$5.613_{\pm 3:5}$	$1.814_{\pm 0:4}$	$1.227_{\pm 0:1}$
ICP (Peters et al., 2016)	$1.777_{\pm 0:6}$	$1.629_{\pm 0:6}$	$1.631_{\pm 0:6}$	$1.097_{\pm 0:1}$	$1.291_{\pm 0:3}$
IC (Rojas-Carulla et al., 2018)	$5.580_{\pm 0:3}$	$1.631_{\pm 0:4}$	$2.322_{\pm 0:7}$	$1.665_{\pm 0:3}$	$1.253_{\pm 0:2}$
DRO (Sinha et al., 2018)	$4.511_{\pm 1:8}$	$1.628_{\pm 0:4}$	$2.311_{\pm 0:7}$	$1.827_{\pm 0:4}$	$1.196_{\pm 0:1}$
Surgery (Subbaswamy et al., 2019)	$1.325_{\pm 0:0}$	$1.086_{\pm 0:0}$	$1.005_{\pm 0:1}$	$1.190_{\pm 0:2}$	$1.071_{\pm 0:1}$
IRM (Arjovsky et al., 2019)	$6.328_{\pm 2:3}$	$1.439_{\pm 0:2}$			

Figure 9: Detailed results for Fig. 5. (a)g 0 G