Aligning Graphical and Functional Causal Abstractions

Willem Schooltink WILLEM.SCHOOLTINK@UIB.NO
Fabio Massimo Zennaro FABIO.ZENNARO@UIB.NO
Department of Informatics, University of Bergen, Norway

Editors: Biwei Huang and Mathias Drton

Abstract

Causal abstractions allow us to relate causal models on different levels of granularity. To ensure that the models agree on cause and effect, frameworks for causal abstractions define notions of consistency. Two distinct methods for causal abstraction are common in the literature: (i) graphical abstractions, such as Cluster DAGs, which relate models on a structural level, and (ii) functional abstractions, like α-abstractions, which relate models by maps between variables and their ranges. In this paper we will align the notions of graphical and functional consistency and show an equivalence between the class of Cluster DAGs, consistent α-abstractions with the range of abstracted variables mapped bijectively, and constructive τ-abstractions. Furthermore, we extend this alignment and the expressivity of graphical abstractions by introducing Partial Cluster DAGs. Our results provide a rigorous bridge between the functional and graphical frameworks and allow for adoption and transfer of results between them.

Keywords: Causality, Causal Abstractions, Structural Causal Models, Cluster DAG, Consistency

1. Introduction

Causality is a fundamental concept for understanding and predicting the behavior of complex systems. Uncovering the underlying causal mechanisms of a system is essential to make informed decisions and design more effective policies in critical fields such as medicine, economics, and politics. The formalism of Structural Causal Models (SCMs) (Pearl, 2009) provides us with a rigorous language to represent a causal system and reason about its behaviour not only in an observational regime ($\mathcal{L}_1$), but also under interventions ($\mathcal{L}_2$).

Whatever causal system we consider, we always need to choose at which level of resolution we want to reason. For example, we may want to reason about voting behaviours either by defining a causal model on a person-by-person basis or by considering the causal dynamics at the level of districts. The two causal models represent the same system and can be connected by a relation of abstraction: individual voting behaviours can be aggregated into district voting patterns. While most causal algorithms select a single among many possible levels of representation, switching between the levels can provide a richer understanding of a system; for instance, aggregating data at the individual and the district level may be valuable for designing successful advertisement strategies.

We can express the relation between a low-level (or base) model and a high-level (or abstracted) model through a causal abstraction map. In order to dynamically switch between models, this abstraction map must guarantee the preservation of relations of cause and effect. We propose to distinguish two main ways to express abstractions and assess their consistency in the literature. The first line of work on *graphical abstractions* is based on the grouping or clustering of low-level variables, as proposed with the Cluster DAG (CDAG) approach (Anand et al., 2023). In these frameworks, we assess *graphical consistency* in terms of the identifiability of causal queries across

© 2025 W. Schooltink & F.M. Zennaro.

the base and the abstracted model; when all relevant interventional queries are correctly identifiable in both models, the abstraction is $\mathcal{L}_2$-*consistent*. The second line of work on *functional abstractions* formalizes causal abstractions in terms of a map between variables and values in a low- and high-level model, as in the α-abstraction (Rischel, 2020) or the τ-abstraction (Rubenstein et al., 2017) framework. In these frameworks, we evaluate *functional consistency* in terms of a discrepancy between the interventional distributions implied by the abstraction; in particular, whenever this discrepancy is zero for all relevant interventions, the abstraction is $\mathcal{L}_2$-*consistent*.

As discussed in the related work, both frameworks have strong theoretical foundations and have provided the basis for relevant practical applications. To bring these frameworks and their methods together, in this paper we offer a formal bridging between graphical clustering and functional abstractions. Concretely, we work with CDAGs, which are the main graphical abstraction formalism in the literature, and with α-abstractions, which offer the most explicit representation of a functional abstraction. To define a common ground of evaluation we first align the notion of functional $\mathcal{L}_2$-consistency to graphical $\mathcal{L}_2$-consistency; then, we show that the set of $\mathcal{L}_2$-consistent CDAGs corresponds to a well-defined subset of simple α-abstractions. Next, to increase the expressivity of graphical abstractions, we introduce a natural extension of CDAGs, namely Partial CDAGs (PCDAGs) and prove that (i) PCDAGs describe a larger set of $\mathcal{L}_2$-consistent α-abstractions than CDAGs; and, (ii) under assumption of faithfulness, all bijective $\mathcal{L}_2$-consistent α-abstractions must be a PCDAG of the base model. Last, we strengthen our contribution by rigorously showing an equivalence between two functional abstraction frameworks: the α-abstraction and the constructive τ-abstraction. This allows us to extend our connection between functional and graphical causal abstraction beyond the specific α-abstraction framework. These results establish a firm connection between the different forms of abstraction proposed in the literature and their notions of consistency. From a theoretical point of view, our contributions allow for the transfer of proofs and properties between frameworks, while, practically, they suggest that PCDAGs may be a useful and grounded starting point for designing and validating consistent abstractions.

Related Work. Establishing the resolution of a SCM is a key design choice in the modeling of causal systems; the definition of variables and causal relations may be left to domain experts (Pearl, 2009), inferred from data through causal discovery or causal representation learning (Schölkopf et al., 2021), or derived from pre-existing SCMs via graphical or functional causal abstraction.

Seminal work on graphical models with clustered variables and their properties was published by Parviainen and Kaski (2017) and extended by Wahl et al. (2024); Anand et al. (2023) introduced CDAGs as a causal inference tool for partially known causal models and proved results related to causal identifiability; CDAGs have also been adopted and interpreted as abstracted models in the context of learning abstractions using neural networks (Xia and Bareinboim, 2024).

Functional causal abstraction comprise two main frameworks. An α-abstraction (Rischel, 2020; Rischel and Weichwald, 2021) defines two mappings between the variables and the values of two models; this framework has been used for learning abstractions (Zennaro et al., 2023) and relate multi-armed bandits at multiple levels of abstraction (Zennaro et al., 2024). Our work starts from the α-abstraction framework as its formulation provides a more fine-grained understanding of an abstraction. On the other hand, a τ-abstraction (Rubenstein et al., 2017; Beckers and Halpern, 2019; Massidda et al., 2022) relies only on a single function between the values of two models; this framework has also been used to explain neural networks (Geiger et al., 2021), learn abstractions (Felekis et al., 2024), derive causal models targeted at encoding a chosen dynamics (Kekić et al.,

2024), or generate surrogates of complex simulation models (Dyer et al., 2023). We connect our results to a specific form of τ-abstraction in the end of our work. Furthermore, Massidda et al. (2024) have studied a linearized version of τ-abstraction called $\mathbf{T}$-abstraction based on variable clustering; while the focus of their work was on abstraction for linear models and causal discovery, we study more generally the relation between clustering and functional abstraction.

2. Preliminaries

In this section we first provide the notions of SCM and causal hierarchy; we then review graphical abstractions through the definitions of CDAGs and graphical consistency; and we conclude with functional abstractions through the presentation of the α-framework and functional consistency.

Notation. We will denote a set of variables using bold uppercase $\mathbf{V}$, a specific variable using normal uppercase $V_i \in \mathbf{V}$, when necessary with an index subscript, and the value of that variable using lowercase v_i. We use $P(X)$ to denote a probability distribution; if necessary, we add a subscript $P_{\mathcal{M}}(X)$ to specify that the distribution is computed on model $\mathcal{M}$.

2.1. Causality

Structural Causal Models. An SCM is a graphical and functional modelling tool that allows us to encode a causal system by (i) defining relations of cause and effects among variables, and (ii) specifying the behaviour of the variables via functions and probability distributions (Pearl, 2009).

Definition 1 (Structural Causal Model) *An SCM is a 4-tuple $\mathcal{M} : \langle \mathbf{U}, \mathbf{V}, \mathcal{F}, P(\mathbf{U}) \rangle$ where:*

- $\mathbf{U}$ *is a set of exogenous variables each one with range $\mathcal{R}(U)$.*
- $\mathbf{V}$ *is a set of endogenous variables each one with range $\mathcal{R}(V)$.*
- $\mathcal{F}$ *is a collection of functions determining the value of the endogenous variables such that, for each $V \in \mathbf{V}$, there is a function $f_V(\mathbf{P}_V, \mathbf{U}_V)$ with $\mathbf{P}_V \subseteq \mathbf{V} \setminus V$, $\mathbf{U}_V \subseteq \mathbf{U}$. We say that $V_i \in \mathbf{V}$ is a* direct cause *of $V_j \in \mathbf{V}$ if and only if V_i is in the signature of the domain of the function $f_{V_j} \in \mathcal{F}$.*
- $P(\mathbf{U})$ *is the probability distribution over the exogenous variables* $\mathbf{U}$.

In the following, we will make standard assumptions about our SCMs. We will restrict our attention to *semi-Markovian* SCMs, that is, models without cyclic causal relations. It is immediate to show that such an SCM entails a Directed Acyclic Graph (DAG) $G = \langle \mathbf{V}, \mathbf{E} \rangle$ with the set of vertices $\mathbf{V}$ given by the endogenous variables and the set of edges $\mathbf{E}$ given by the collection of edges $V_i \rightarrow V_j$ if V_i is a direct cause of V_j. The set $\mathbf{P}_{V_j}$ for a function f_{V_j} represents then the graph-theoretical parents of V_j, and we will denote it as $Pa(V_j)$ from here on. Furthermore, if two variables $V_i, V_j \in \mathbf{V}$ share an exogenous parent $U \in \mathbf{U}$, we say that V_i and V_j are *confounded*, implying they are not independent even without a causal edge between them; we denote confounding with a dashed bidirectional edge $V_i \dashleftarrow\dashrightarrow V_j$ as a shorthand for $V_i \leftarrow U \rightarrow V_j$. Notice that, despite the bidirectional edge, the actual graph remains acyclic.

Finally, we will also make the assumption of *faithfulness*, that is two variables are graphically independent, or *d-separated* (Geiger et al., 1990), if and only if they are distributionally independent. It is worth pointing out that, whereas an SCM $\mathcal{M}$ entails a unique DAG G, there normally are multiple SCMs $\mathcal{M}$ with the same DAG structure G.

Importantly, causality defines a way of interacting with SCMs via interventions. We consider hard interventions defined as follows:

Definition 2 (Interventions) *Let $\mathcal{M} = \langle \mathbf{V}, \mathbf{U}, \mathcal{F}, P(\mathbf{U}) \rangle$ be an SCM and $X \in \mathbf{V}$ an endogenous variable, then an intervention $do(X = x)$ is an operation that replaces the function determining $X = f_X(Pa(X), U_X)$ with the assignment $X = x$.*

In analogy to conditioning, we will abbreviate the generic $do(X = x)$ as $do(X)$. Notice how an intervention $do(X)$ removes the direct causes of X; therefore it implies a new SCM with an underlying DAG with all the incoming edges into X removed. It is immediate to extend the definition of an intervention on a single variable $do(X)$ to multiple variables $do(\mathbf{X})$.

Causal Hierarchy. A causal query on an SCM is a causal statement that can be located onto one of the three distinct layers of the Pearl's Causal Hierarchy (PCH) (Pearl, 2009; Bareinboim et al., 2022): ($\mathcal{L}_1$) observational queries with the form $P(Y|X)$ are statistical formulas that describe the system in its natural behaviour; ($\mathcal{L}_2$) interventional queries with the form $P(Y|do(X))$ characterize the system under external manipulations; and, ($\mathcal{L}_3$), counterfactual queries study the system under hypothetical interventions that never took place. These layers are rigid, as queries on a higher layer require more information to be answered and cannot automatically be reduced to a lower layer. In order to assess the consistency of abstractions, we will focus on the first two levels of this hierarchy. We will say that a causal query on the interventional layer $\mathcal{L}_2$ is *identifiable* if it can be reduced to statistical quantities belonging to the observational layer $\mathcal{L}_1$. Under the assumption of faithfulness, *Do-calculus* is a complete theory relying only on the graphical structure G of $\mathcal{M}$ to decide whether a causal query is identifiable.

2.2. Graphical Abstraction

Cluster DAGs. The simplest way to relate two SCMs is through their underlying DAGs. CDAGs were originally introduced as graphical modelling tools allowing us to represent a system where information about the exact causal relationships among certain sets of variables were unknown (Anand et al., 2023). Nonetheless, CDAGs may also be easily used to express the process of reduction of information associated with abstraction.

Definition 3 (Cluster DAG) *Let $G = \langle \mathbf{V}, \mathbf{E} \rangle$ be a DAG and $\varphi : \mathbf{V} \to \mathbf{C}$ be a surjective function where $\mathbf{C} = \{\mathbf{C}_1, \ldots, \mathbf{C}_k\}$. The function φ induces a partition of $\mathbf{V}$. $G_{\mathbf{C}} = \langle \mathbf{C}, \mathbf{E}_{\mathbf{C}} \rangle$ is a cluster DAG (CDAG) of G if and only if the set of edges $\mathbf{E}_{\mathbf{C}}$ abides by the following rules:*

1. *An edge $\mathbf{C}_i \to \mathbf{C}_j$ is in $\mathbf{E}_{\mathbf{C}}$ if there exists a $V_i \to V_j$ in $\mathbf{E}$ such that $V_i \in \mathbf{C}_i,\ V_j \in \mathbf{C}_j, \mathbf{C}_i \neq \mathbf{C}_j$.*
2. *A bidirected (confounding) edge $\mathbf{C}_i \leftarrow\!\dashrightarrow \mathbf{C}_j$ is in $\mathbf{E}_{\mathbf{C}}$ if there exists a $V_i \leftarrow\!\dashrightarrow V_j$ in $\mathbf{E}$ such that $V_i \in \mathbf{C}_i,\ V_j \in \mathbf{C}_j, \mathbf{C}_i \neq \mathbf{C}_j$.*

Further, it is required that the graph $G_{\mathbf{C}}$ induced by φ is acyclic.

Given an SCM $\mathcal{M}$ with underlying DAG $G = \langle \mathbf{V}, \mathbf{E} \rangle$, we can see the CDAG $G_{\mathbf{C}} = \langle \mathbf{C}, \mathbf{E}_{\mathbf{C}} \rangle$ as the structure of a graphically abstracted model $\mathcal{M}'$. Importantly, a CDAG $G_{\mathbf{C}}$ does not identify a single model $\mathcal{M}'$, but the collection of SCMs sharing the same structure.

Graphical Consistency. Although the building procedure of a CDAG guarantees the preservation of basic relations of cause and effect, we still want to establish whether a causal query can be consistently identified both in the original SCM $\mathcal{M}$ and in a model $\mathcal{M}'$ with the structure given by the derived CDAG $G_{\mathbf{C}}$. Since the SCM $\mathcal{M}'$ is not specified, we can investigate identifiability only by evaluating graphical properties. We can express the relations of dependence and independence encoded graphically in a DAG through algebraic relation of equality and inequality among distributions; for instance, the structure $X \to Y$ implies, amongst others, relations such as $P(X \mid Y) \neq P(X)$ and $P(Y \mid X) = P(Y \mid do(X))$. We denote $\mathcal{G}(G)$ the set of all the algebraic constraints implied by the DAG G. We use these constraints to define graphical consistency:

Definition 4 (Graphical Consistency) *Let G be a DAG underlying $\mathcal{M}$ and $G_{\mathbf{C}}$ the CDAG underlying $\mathcal{M}'$. Let us define $\mathcal{G}(G_{\mathbf{C}}^{-1})$ be the set of all the constraints obtained by taking each constraint in $\mathcal{G}(G_{\mathbf{C}})$ and substituting each cluster $\mathbf{C}_i$ with its pre-image $\varphi^{-1}(\mathbf{C}_i)$. The two models $\mathcal{M}$ and $\mathcal{M}'$ are graphically consistent if $\mathcal{G}(G_{\mathbf{C}}^{-1}) \subseteq \mathcal{G}(G)$.*

Graphical consistency means that all the equalities and inequalities expressed in the CDAG hold in the original SCM among the clustered variables; thus, identifiable causal queries on the CDAG are identifiable also on the original SCM. However, the converse does not necessarily hold: in the original SCM there may be additional equalities and inequalities among the variables within a cluster that are not expressible in the CDAG.

Graphical consistency may be restricted only to constraints pertaining to a chosen layer of the causal hierarchy. Considering again the structure $X \to Y$, the constraint $P(X \mid Y) \neq P(X)$ belongs to $\mathcal{L}_1$, while $P(Y \mid X) = P(Y \mid do(X))$ belongs to $\mathcal{L}_2$. We denote $\mathcal{G}^{\mathcal{L}_i}(G)$ the set of all the constraints at layer $\mathcal{L}_i$ implied by the DAG G. We can express restricted forms of consistency as:

Definition 5 ((Graphical) $\mathcal{L}_i$-Consistency) *Let G be a DAG underlying $\mathcal{M}$ and $G_{\mathbf{C}}$ the CDAG underlying $\mathcal{M}'$. Let us define $\mathcal{G}^{\mathcal{L}_i}(G_{\mathbf{C}}^{-1})$ as the set of all the constraints obtained by taking each constraint in $\mathcal{G}^{\mathcal{L}_i}(G_{\mathbf{C}})$ and substituting each cluster $\mathbf{C}_i$ with its pre-image $\varphi^{-1}(\mathbf{C}_i)$. The two models $\mathcal{M}$ and $\mathcal{M}'$ are graphically consistent if $\mathcal{G}^{\mathcal{L}_i}(G_{\mathbf{C}}^{-1}) \subseteq \mathcal{G}^{\mathcal{L}_i}(G)$.*

By construction, a low-level model $\mathcal{M}$ and a high-level model $\mathcal{M}'$ with an underlying CDAG are always $\mathcal{L}_2$-consistent; therefore, for a constraint $P_{\mathcal{M}'}(Y|X) = P_{\mathcal{M}'}(Y|do(X))$ in the high-level model, the constraint $P_{\mathcal{M}}(\varphi^{-1}(Y)|\varphi^{-1}(X)) = P_{\mathcal{M}}(\varphi^{-1}(Y)|do(\varphi^{-1}(X)))$ holds in the low-level model. In general, however, equalities *across models*, such as $P_{\mathcal{M}'}(Y|X) = P_{\mathcal{M}}(\varphi^{-1}(Y)|do(\varphi^{-1}(X)))$, do not hold; however, given a CDAG, Anand et al. (2023) prove the existence of a high-level SCM $\mathcal{M}'$ with CDAG structure such that equalities across models hold (Theorem 7) and evaluations of do-calculus formulas on $\mathcal{M}'$ are equivalent to the evaluations on pre-images of the clusters in $\mathcal{M}$ (Theorem 3).

2.3. Functional Abstraction

α-Abstraction. Whereas a CDAG captures the graphical aspect of an abstraction, a functional abstraction is meant to express an abstraction in terms of a mapping between variables and values (see App. A.1 for the original definition with finite graphical models).

Definition 6 (α-abstraction) *Given two SCMs $\mathcal{M} : \langle \mathbf{V}_{\mathcal{M}}, \mathbf{U}_{\mathcal{M}}, \mathcal{F}_{\mathcal{M}}, P_{\mathcal{M}}(\mathbf{U}_{\mathcal{M}})\rangle$ and $\mathcal{M}' : \langle \mathbf{V}_{\mathcal{M}'}, \mathbf{U}_{\mathcal{M}'}, \mathcal{F}_{\mathcal{M}'}, P_{\mathcal{M}'}(\mathbf{U}_{\mathcal{M}'})\rangle$, an abstraction $\boldsymbol{\alpha} : \mathcal{M} \to \mathcal{M}'$ is a 3-tuple $\langle \mathbf{R}, \varphi, \alpha_{V'}\rangle$ where:*

1. *$\mathbf{R} \subseteq \mathbf{V}_{\mathcal{M}}$ is a subset of relevant variables in $\mathcal{M}$.*
2. *$\varphi : \mathbf{R} \to \mathbf{V}_{\mathcal{M}'}$ is a surjective map from the relevant variables to the variables in $\mathcal{M}'$.*
3. *$\alpha_{V'} : \mathcal{R}(\varphi^{-1}(V')) \to \mathcal{R}(V')$, for each $V' \in \mathbf{V}_{\mathcal{M}'}$, is a surjective function from the range of the pre-image $\varphi^{-1}(V') \subseteq \mathbf{V}_{\mathcal{M}}$ in $\mathcal{M}$ to the range of V' in $\mathcal{M}'$.*

We will define the application of $\alpha_{\mathbf{V}}$ to a distribution as the pushforward $\alpha_X[P(X)] = \alpha_{X\#}(P)(X)$.

Functional consistency. Differently from the constructive definition of a CDAG, the declarative definition of an α-abstraction does not implicitly preserve relations of cause and effect. A requirement of consistency over distributions is instead expressed in terms of commutativity of abstractions and $\mathcal{L}_i$ operations, such as conditioning ($\mathcal{L}_1$) or intervening ($\mathcal{L}_2$); we use the shorthand $P(Y|\mathcal{L}_i(X))$ for $P(Y|X)$ if $i = 1$ or $P(Y|do(X))$ if $i = 2$.

Definition 7 ((Functional) $\mathcal{L}_i$-Consistency) *Let $\boldsymbol{\alpha} : \mathcal{M} \to \mathcal{M}'$ be an abstraction. The abstraction $\boldsymbol{\alpha}$ is $\mathcal{L}_i$-consistent if, for all $\mathbf{X}, \mathbf{Y} \subseteq \mathbf{V}_{\mathcal{M}'}$, the following diagram commutes:*

$$\begin{array}{ccc} \varphi^{-1}(\mathbf{X}) & \xrightarrow{\quad\mathcal{L}_i\quad} & \varphi^{-1}(\mathbf{Y})|\mathcal{L}_i(\varphi^{-1}(\mathbf{X})) \\ \downarrow{\scriptstyle \alpha_{\mathbf{X}}} & & \downarrow{\scriptstyle \alpha_{\mathbf{Y}}} \\ \mathbf{X} & \xrightarrow{\quad\mathcal{L}_i\quad} & \mathbf{Y}|\mathcal{L}_i(\mathbf{X}) \end{array} .$$

that is:

$$P_{\mathcal{M}'}(\mathbf{Y} \mid \alpha_{\mathbf{X}}[\mathcal{L}_i(\varphi^{-1}(\mathbf{X}))]) = \alpha_{\mathbf{Y}}\left[P_{\mathcal{M}}(\varphi^{-1}(\mathbf{Y}) \mid \mathcal{L}_i(\varphi^{-1}(\mathbf{X})))\right]. \tag{1}$$

Thus, $\mathcal{L}_2$-consistency means that the result of intervening and then abstracting must be identical to that of abstracting and then intervening (Rischel, 2020). In practical applications, however, we have to deal with noisy data and the need to drop information. As a result, it may be necessary to relax the strict requirement of commutativity and introduce an error measure that quantifies how different are the distributions on the two sides of Eq.1:

Definition 8 ($\mathcal{L}_i$–Abstraction error) *Given an abstraction $\boldsymbol{\alpha} : \mathcal{M} \to \mathcal{M}'$ and a distance or divergence $D(p, q)$ between distributions p and q, the $\mathcal{L}_i$-abstraction error is computed as:*

$$e_{\mathcal{L}_i}(\boldsymbol{\alpha}) = \max_{\mathbf{X},\mathbf{Y}\subseteq\mathbf{V}_{\mathcal{M}'}} D(P_{\mathcal{M}'}(\mathbf{Y} \mid \alpha_{\mathbf{X}}[\mathcal{L}_i(\varphi^{-1}(\mathbf{X}))]), \alpha_{\mathbf{Y}}\left[P_{\mathcal{M}}(\varphi^{-1}(\mathbf{Y}) \mid \mathcal{L}_i(\varphi^{-1}(\mathbf{X}))\right]). \tag{2}$$

Notice how the abstraction error is a worst-case measure of inconsistency: considering all possible operations in the abstracted model, it returns the error corresponding to the pair $\mathbf{X}, \mathbf{Y} \subseteq \mathbf{V}_{\mathcal{M}'}$ that maximizes the discrepancy D. It is immediate to redefine functional $\mathcal{L}_i$-consistency as zero abstraction error $e_{\mathcal{L}_i}(\boldsymbol{\alpha}) = 0$.

3. Aligning Graphical and Functional Consistency

In order to relate graphical and functional abstractions, we need first of all to align the notions of graphical and functional consistency. Both $\mathcal{L}_i$-consistencies share the following implication:

Lemma 9 *$\mathcal{L}_2$-consistency implies $\mathcal{L}_1$-consistency.* [Proof in App. B.1.]

However, graphical and functional consistency are intrinsically different, as one is defined on a graphical level and the other on a functional level. Indeed, they both require different identities in order to hold, as shown in Fig.1: *graphical consistency* requires that an equality in the CDAG underlying an abstracted model $\mathcal{M}'$ hold in the original model $\mathcal{M}$; *functional consistency* implies the identity of two interventional distributions in the abstracted model $\mathcal{M}'$. We show that it is possible to align functional consistency with graphical consistency as follows:

Proposition 10 *A functional $\mathcal{L}_2$-consistent abstraction $\boldsymbol{\alpha} : \mathcal{M} \to \mathcal{M}'$ with bijective maps α_V implies graphical $\mathcal{L}_2$-consistency.*

This proposition relies on the following lemma which guarantees that a functional abstraction $\boldsymbol{\alpha} : \mathcal{M} \to \mathcal{M}'$, with all range mappings α_V bijective, preserves equalities and inequalities.

Lemma 11 (Bijective Range Maps Preserve Distribution (In)Equalities) *Let $\boldsymbol{\alpha} : \mathcal{M} \to \mathcal{M}'$ be an $\mathcal{L}_2$-consistent abstraction with $\alpha_V : \mathcal{R}\left(\varphi^{-1}(V)\right) \to \mathcal{R}\left(V\right)$ bijective for all $V \in \mathbf{V}_{\mathcal{M}'}$. Let $\mathbf{X}, \mathbf{Y}_1, \mathbf{Y}_2, \mathbf{Z}_1, \mathbf{Z}_2 \subseteq \mathbf{V}_{\mathcal{M}}$ be partitions defined by φ, then $P(\mathbf{X}|do(\mathbf{Y}_1), \mathbf{Z}_1) = P(\mathbf{X}|do(\mathbf{Y}_2), \mathbf{Z}_2)$ if and only if $\alpha_{\mathbf{X}}[P(\mathbf{X} \mid do(\mathbf{Y}_1), \mathbf{Z}_1)] = \alpha_{\mathbf{X}}[P(\mathbf{X} \mid do(\mathbf{Y}_2), \mathbf{Z}_2)]$.* [Proof in App. B.2.]

The proof of Lem.11 shows that surjectivity of α_V is sufficient to imply that inequalities in $\mathcal{M}'$ must have a corresponding inequality in $\mathcal{M}$, while bijectivity ensures also that all equalities in $\mathcal{M}'$ have a corresponding equality in $\mathcal{M}$.

4. Aligning Graphical and Functional Abstraction

After relating the notions of consistency, we now consider how graphical clustering and functional abstraction themselves are related. To unify the notation, we will use $V_i \in \mathbf{V}$ for variables in the base model $\mathcal{M}$ and $\mathbf{C}_i \in \mathbf{C}$ for variables in the abstracted model $\mathcal{M}'$, both in case of a graphical or functional abstraction. Now, as by Def.3, CDAGs have two important limitations: (i) they require every variable in $\mathcal{M}$ to belong to a cluster; and (ii) they specify an aggregation over variables in $\mathcal{M}$, but not over their values. Correspondingly, an aligned α-abstraction must: (i) consider all variables as relevant, $\mathbf{R} = \mathbf{V}$; and, (ii) have *bijective range mappings*, as in Lem.11. Under these two conditions, we now show a correspondence between bijective $\mathcal{L}_2$-consistent α-abstractions with $\mathbf{R} = \mathbf{V}$ and the set of CDAGs. Note that for applications the restrictions of the CDAG may impede the use of it for real world abstractions. We will illustrate this using an example in the next section.

In the direction α-abstraction $\Rightarrow$ CDAG, we prove that a bijective $\mathcal{L}_2$-consistent α-abstraction with $\mathbf{R} = \mathbf{V}$ entails a unique CDAG. To do so, we show that an $\mathcal{L}_2$-consistent α-abstraction defines a structure which complies with the two constructive rules of a CDAG in Def.3. Rule 1 is satisfied by showing that the mapping of variables $\varphi : \mathbf{V} \to \mathbf{C}$ of an $\mathcal{L}_2$-consistent α-abstraction uniquely determines the causal edges of $\mathcal{M}'$ as a CDAG would:

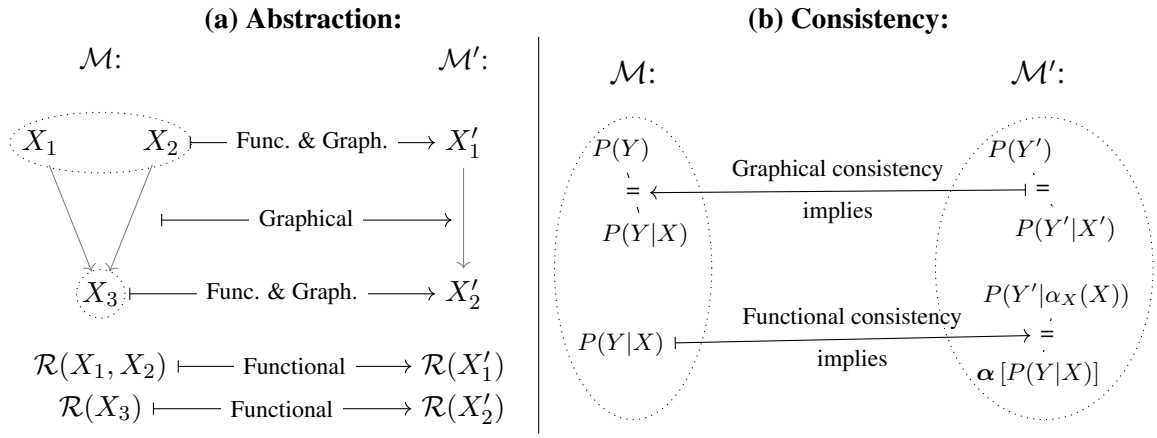

Figure 1: Visualization of abstractions **(a)** and consistency **(b)**. Figure **(a)** illustrates the overlap and the differences between graphical and functional abstractions: both involve mapping and clustering variables; however, graphical abstractions preserve edges, while functional abstractions exploit the more detailed models to map variable ranges. Figure **(b)**: given an SCM $\mathcal{M}$ and an abstraction $\mathcal{M}'$, graphical consistency implies preservation of equality relations between distributions under abstraction, whereas functional consistency guarantees that abstraction and $\mathcal{L}_i$ operations (in this case conditioning) commute, yielding the same resulting distributions.

Lemma 12 ($\mathcal{L}_2$-Consistency Uniquely Determines Adjacencies) *Given an $\mathcal{L}_2$-consistent bijective abstraction $\boldsymbol{\alpha} : \mathcal{M} \to \mathcal{M}'$, adjacencies between variables in $\mathcal{M}'$ are uniquely determined by the map $\varphi : \mathbf{V} \to \mathbf{C}$ and comply with Rule 1 of Def.3 (CDAG).* [Proof in App. B.3.]

The proof of Lem.12 shows that surjectivity alone guarantees that a causal edge in the abstraction implies a compatible causal edge in the base model. Conversely, for any bijective $\mathcal{L}_2$-consistent α-abstraction, if there exists an adjacency in the base model, there must exists a corresponding adjacency in the abstraction.

Next we prove that Rule 2 is also satisfied by showing that confounding edges are similarly uniquely defined by an $\mathcal{L}_2$-consistent α-abstraction in accordance with the CDAG definition:

Lemma 13 ($\mathcal{L}_2$-Consistency Uniquely Determines Confounding Edges) *Given an $\mathcal{L}_2$-consistent bijective abstraction $\boldsymbol{\alpha} : \mathcal{M} \to \mathcal{M}'$, confounding edges between variables in $\mathcal{M}'$ are uniquely determined by the map $\varphi : \mathbf{V} \to \mathbf{C}$ and comply with Rule 2 of Def.3 (CDAG).* [Proof in App. B.4.]

Determining adjacencies according to Rule 1 and confounding according to Rule 2 immediately implies a unique CDAG structure:

Theorem 14 (α-Abstraction $\Rightarrow$ CDAG) *Given any bijective $\mathcal{L}_2$-consistent α-abstraction $\boldsymbol{\alpha} : \mathcal{M} \to \mathcal{M}'$ with $\mathbf{R} = \mathbf{V}$, the DAG of $\mathcal{M}'$ is a CDAG of the graph of $\mathcal{M}$.* [Proof in App. B.5.]

This theorem uncovers a clear connection between graphical and functional abstraction. Despite the fact that the definition of α-abstraction makes *no* reference at all to the graphical structure of

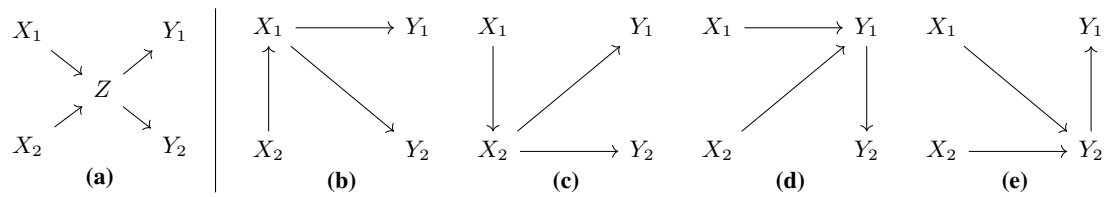

Figure 2: Given the DAG representing an SCM **(a)**, there are 4 ways of clustering such that X_1, X_2, Y_1, and Y_2 are in separate clusters : $\{\{X_1, Z\}, \{X_2\}, \{Y_1\}, \{Y_2\}\}$ **(b)**, $\{\{X_1\}, \{X_2, Z\}, \{Y_1\}, \{Y_2\}\}$ **(c)**, $\{\{X_1\}, \{X_2\}, \{Y_1, Z\}, \{Y_2\}\}$ **(d)**, $\{\{X_1\}, \{X_2\}, \{Y_1\}, \{Y_2, Z\}\}$ **(e)**. Observe that CDAGs **(b)** and **(c)** lose the ability to intervene on X_1 and X_2 independently, while **(d)** and **(e)** retain the ability to intervene independently on X_1 and X_2, but make Y_1 a direct cause of Y_2 or vice versa.

an abstracted model, this theorem shows that for a bijective α-abstraction, if we want to guarantee $\mathcal{L}_2$-consistency, there is a *unique* admissible graphical structure; and this structure is exactly the one derived using the CDAG construction rules.

In the direction CDAG $\Rightarrow$ α-abstraction, we can show that for every CDAG there is an immediate bijective $\mathcal{L}_2$-consistent α-abstraction such that the underlying structure is that CDAG.

Theorem 15 (CDAG $\Rightarrow$ α-Abstraction) *Given a CDAG there exists an equivalent $\mathcal{L}_2$-consistent α-abstraction $\boldsymbol{\alpha} : \mathcal{M} \to \mathcal{M}'$ with $\mathbf{R} = \mathbf{V}$, φ given by the clustering, and all maps $\alpha_{\mathbf{C}}$ bijective.* [Proof in App. B.6.]

Thm.14 and Thm.15 establish a correspondence between the set of bijective $\mathcal{L}_2$-consistent α-abstraction and the set of CDAGs. However, notice there might still exist non-bijective $\mathcal{L}_2$-consistent α-abstraction or even non-$\mathcal{L}_2$-consistent α-abstraction with an underlying CDAG. For intuition about the bijectivity assumption, consider Example A.2 in the appendix.

5. Extending Graphical Abstractions: Partial CDAGs

Graphical abstractions in the form of CDAGs are bound to account in clustering for all variables ($\mathbf{R} = \mathbf{V}$). This restriction limits the expressivity and usability of graphical clustering as illustrated in the following example.

Example 1 Let $\mathcal{M} : \langle \mathbf{U}, \mathbf{V}, \mathcal{F}, P(\mathbf{U}) \rangle$ be an SCM modeling the effects of smoking (X_1) and air pollution (X_2) on lung cancer (Y_1) and shortness of breath (Y_2), through the mediating variable of tar deposits (Z). We assume X_1 and X_2 to be independent, and Y_1 and Y_2 to be independent given Z, as shown in the DAG of Fig.2 **(a)**. Assume we are interested in the effects of X_1, X_2 (smoking and air quality) on Y_1, Y_2 (lung cancer and shortness of breath), and we have no way of measuring the tar deposits Z. If we want to keep the possibility of intervening on X_1 or X_2 independently, and the ability of predicting Y_1 or Y_2 independently, any CDAG abstracting away Z as in Fig.2 will sacrifice one of these possibilities: either our ability to intervene (subfigures **(b)**, **(c)**) or to independently predict (subfigures **(d)**, **(e)**). The structure of the CDAGs does not seem to align to the description of two causes and two effects because of the need to include Z in one of the clusters. Ideally, we want to create an abstraction removing Z that preserves our ability to intervene on X_1, X_2, predictability of Y_1, Y_2, and any confounding effect introduced by Z.

To overcome the limitation described in the example, we now extend the expressivity of graphical models by defining Partial CDAGs:

Definition 16 (Partial CDAG) *Let $G = \langle \mathbf{V}, \mathbf{E} \rangle$ be a DAG and $\varphi : \mathbf{V} \to \mathbf{C}$ be a partial surjective function where $\mathbf{C} = \{\mathbf{C}_1, \dots, \mathbf{C}_k\}$. Let $\mathbf{Q} \subseteq \mathbf{V}$ be the remainder set of vertices that are not mapped by φ. $\mathbf{C} \cup \mathbf{Q}$ forms a partition of the vertices $\mathbf{V}$. $G_{\mathbf{C}} = \langle \mathbf{C}, \mathbf{E}_{\mathbf{C}} \rangle$ is a partial CDAG (PCDAG) of G if and only if the set of edges $\mathbf{E}_{\mathbf{C}}$ abides by the following rules:*

1. *An edge $\mathbf{C}_i \to \mathbf{C}_j$ is in $\mathbf{E}_{\mathbf{C}}$ if there exists some path $V_i \to \cdots \to V_j$ with zero or more intermediate variables in $\mathbf{V}$, such that $V_i \in \mathbf{C}_i,\ V_j \in \mathbf{C}_j,\ \ \mathbf{C}_i \neq \mathbf{C}_j$, and all intermediate variables are in $\mathbf{Q}$.*

2A. *A bidirected (confounding) edge $\mathbf{C}_i \leftarrow\!\!-\!\!\rightarrow \mathbf{C}_j$ is in $\mathbf{E}_{\mathbf{C}}$ if there exists some $V_i \leftarrow \ldots \leftarrow\!\!-\!\!\rightarrow \ldots \to V_j$ with zero or more intermediate variables on either side in $\mathbf{V}$, such that $V_i \in \mathbf{C}_i,\ V_j \in \mathbf{C}_j,\ \ \mathbf{C}_i \neq \mathbf{C}_j$, and all intermediate variables are in $\mathbf{Q}$.*

2B. *A bidirected (confounding) edge $\mathbf{C}_i \leftarrow\!\!-\!\!\rightarrow \mathbf{C}_j$ is in $\mathbf{E}_{\mathbf{C}}$ if there exists some $Q \in \mathbf{Q}$ with paths $V_i \leftarrow \cdots \leftarrow Q \to \cdots \to V_j$ and $V_i \in \mathbf{C}_i,\ V_j \in \mathbf{C}_j,\ \mathbf{C}_i \neq \mathbf{C}_j$, such that all intermediate vertices in the paths $Q \to \cdots \to V_i$ and $Q \to \cdots \to V_j$ are in $\mathbf{Q}$.*

Further, it is required that the graph $G_{\mathbf{C}}$ is induced by φ acyclic.

The first two rules follow intuitively as extensions of CDAGs: Rule 1 ensures that directed paths and adjacencies are maintained, while Rule 2A ensures preservation of confounding edges in PCDAGs. However, by allowing for partial clustering more confounding edges might be introduced; therefore, Rule 2B is required to capture confounding edges introduced when shared parents are dropped. To understand the role of 2B see the following example.

Example 2 We continue with the SCM of Example 1, but we now define a partial clustering $\mathbf{C} = \{\{X_1\}, \{X_2\}, \{Y_1\}, \{Y_2\}\}$ with remainder set $\mathbf{Q} = \{Z\}$. By applying only Rule 1 and Rule 2A we would obtain the second model in Fig.3. However, notice that in the original model it holds that $Y_1 \not\perp\!\!\!\perp Y_2 \mid X_1, X_2$ due to Z. In the proposed PCDAG this dependence does not hold. In order to guarantee that $\{Y_1\} \not\perp\!\!\!\perp \{Y_2\} \mid \{X_1\}, \{X_2\}$, a confounding arrow $\{Y_1\} \leftarrow\!\!-\!\!\rightarrow \{Y_2\}$ must be introduced as by Rule 2B. The last DAG of Fig.3 shows the PCDAG abstraction.

PCDAGs allow for dropping variables while keeping their confounding effects; this offers more versatility as, in the case of the example above, it allows us to preserve independent predictability (over Y_1, Y_2) *and* independent interventions (over X_1, X_2).

Similarly to CDAGs, we now prove some key graphical properties of PCDAGs related to the preservation of adjancencies and directed paths. First of all, in PCDAGs we require a more flexible notion of adjacency to account for dropped variables:

Definition 17 (Mediated Adjacency) *Let $\mathcal{M}$ be an SCM with a PCDAG defined by clusters $\mathbf{C}$ and remainder set $\mathbf{Q}$. Given two variables $V_i, V_j \in \mathbf{V}$, there exists a mediated adjacency $V_i \rightsquigarrow V_j$ if and only if there exists a directed path $V_i \to \cdots \to V_j$ such that all intermediate variables are in $\mathbf{Q}$.*

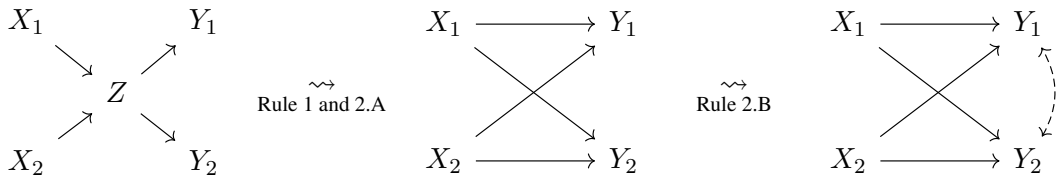

Figure 3: Given an SCM represented by the left DAG, there exists an abstraction that preserves marginal independence $X_1 \perp\!\!\!\perp X_2$, and independent predictability of Y_1 and Y_2 given by the PCDAG. Notice how the application of rules 1 and 2A preserve causal and confounding edges from the original model, whereas 2B introduces confounding edges preserving confounding effects of removed fork structures.

This concept of mediated adjacency is equivalent to $\mathbf{T}$-direct path in linear abstraction (Massidda et al., 2024). PCDAGs preserve mediated adjacencies by construction:

Lemma 18 (PCDAGs Preserve Mediated Adjancencies) *Let $\mathcal{M}$ be an SCM with a PCDAG defined by clusters $\mathbf{C}$ and remainder set $\mathbf{Q}$. Let $\mathbf{C}_i, \mathbf{C}_j \in \mathbf{C},\ \mathbf{C}_i \neq \mathbf{C}_j$ and $V_i \in \mathbf{C}_i,\ V_j \in \mathbf{C}_j$, then a mediated adjacency between V_i and V_j exists if and only if there exists an adjacency between $\mathbf{C}_i$ and $\mathbf{C}_j$.* [Proof in App. B.7.]

Preservation of mediated adjacencies immediately implies preservation of directed paths, as any directed path can be decomposed into a series of consecutive adjacencies:

Lemma 19 (PCDAGs Preserve Directed Paths) *Let $\mathcal{M}$ be an SCM with a PCDAG defined by clusters $\mathbf{C}$ and remainder set $\mathbf{Q}$. Let $\mathbf{C}_i, \mathbf{C}_j \in \mathbf{C},\ \mathbf{C}_i \neq \mathbf{C}_j$ and $V_i \in \mathbf{C}_i,\ V_j \in \mathbf{C}_j$, then a directed path $\mathbf{C}_i \to \cdots \to \mathbf{C}_j$ exists if there exists a directed path $V_i \to \cdots \to V_j$.* [Proof in App. B.8.]

We thus align ourselves with the results of Anand et al. (2023) by showing that PCDAGs preserve adjacencies and directed paths. This in turn allows us to discuss the causal consistency of PCDAGs and show that a PCDAG is compatible with functional $\mathcal{L}_2$-consistency:

Theorem 20 ($\mathcal{L}_2$-consistency of PCDAGs) *Given an SCM $\mathcal{M}$ and a PCDAG $\mathcal{M}'$ of $\mathcal{M}$, there exists a set of mechanisms $\mathcal{F}_{\mathcal{M}'}$ such that $\mathcal{M}'$ is $\mathcal{L}_2$-consistent with $\mathcal{M}$.* [Proof in App. B.9.]

As PCDAGs can be compatible with functional $\mathcal{L}_2$-consistency, Prop.10 states that PCDAGs are necessarily graphically $\mathcal{L}_2$-consistent.

6. Re-Aligning Graphical and Functional Abstraction

We now extend our previous results to the alignment of graphical abstraction in the form of PCDAGs and functional abstractions. Differently from a CDAG, a PCDAG has only one limitation compared to α-abstractions: (i) it specifies an aggregation over variables in $\mathcal{M}$, but not over the values of the variables. Therefore, a corresponding α-abstraction only needs to (i) have *bijective range mappings*. With this condition, we now show a correspondance between bijective $\mathcal{L}_2$-consistent α-abstraction (without necessarily $\mathbf{R} = \mathbf{V}$) and the set of PCDAGs.

In the direction α-abstraction $\Rightarrow$ PCDAG, we prove that a bijective $\mathcal{L}_2$-consistent α-abstraction entails a unique PCDAG. As before, we show that a bijective $\mathcal{L}_2$-consistent α-abstraction defines a structure that complies with the three constructive rules of a PCDAG in Def.16. Rule 1 is satisfied by showing that the mapping of variables $\varphi : \mathbf{V} \to \mathbf{C}$ of an $\mathcal{L}_2$-consistent α-abstraction uniquely determines all the causal edges of $\mathcal{M}'$ as a PCDAG would:

Lemma 21 ($\mathcal{L}_2$-Consistency Uniquely Determines Adjacencies) *Given a bijective $\mathcal{L}_2$-consistent abstraction $\boldsymbol{\alpha} : \mathcal{M} \to \mathcal{M}'$, adjacencies between variables in $\mathcal{M}'$ are uniquely determined by the map $\varphi : \mathbf{V} \to \mathbf{C}$ and comply with Rule 1 of Def.16 (PCDAG).* [Proof in App. B.10.]

Next we prove that confounding edges of a bijective $\mathcal{L}_2$-consistent abstraction are necessarily equal to those given by Rule 2A and 2B of a PCDAG in Def.16:

Lemma 22 ($\mathcal{L}_2$-Consistency Uniquely Determines Confounding Edges) *Given a bijective $\mathcal{L}_2$-consistent abstraction $\boldsymbol{\alpha} : \mathcal{M} \to \mathcal{M}'$, confounding edges between variables in $\mathcal{M}'$ are uniquely determined by the map $\varphi : \mathbf{V} \to \mathbf{C}$ and comply with Rule 2A and 2B of Def.16 (PCDAG).* [Proof in App. B.11.]

Determining mediated adjacencies according to Rule 1 and confounding according to Rule 2A and 2B immediately implies a unique PCDAG structure:

Theorem 23 (α-Abstraction $\Rightarrow$ PCDAG) *Given any bijective $\mathcal{L}_2$-consistent α-abstraction $\boldsymbol{\alpha} : \mathcal{M} \to \mathcal{M}'$, the DAG of $\mathcal{M}'$ is a PCDAG of the graph of $\mathcal{M}$.* [Proof in App. B.12.]

And, mirroring the results of the CDAG, in the direction PCDAG $\Rightarrow$ $\mathcal{L}_2$-consistent α-abstraction we get that for all PCDAGs we can construct an equivalent $\mathcal{L}_2$-consistent α-abstraction by taking the range maps $\alpha_{\mathbf{C}}$ to be bijective.

Theorem 24 (PCDAG $\Rightarrow$ α-Abstraction) *Given a PCDAG there exists an equivalent $\mathcal{L}_2$-consistent α-abstraction $\boldsymbol{\alpha} : \mathcal{M} \to \mathcal{M}'$ with $\mathbf{R} \subseteq \mathbf{V}$, φ given by the clustering, and all maps $\alpha_{\mathbf{C}}$ bijective.* [Proof in App. B.13.]

Furthermore, by Prop.10, the PCDAG is necessarily graphically $\mathcal{L}_2$-consistent. Thus, if we want a bijective $\mathcal{L}_2$-consistent abstraction, we can start by applying the rules of the PCDAG. Moreover, by the proof of Lem.12 we get that the PCDAG can also be useful for surjective $\mathcal{L}_2$-consistent abstractions, albeit that not all edges in the PCDAG may be necessary. In other words, PCDAGs can also describe the structure of surjective $\mathcal{L}_2$-consistent abstractions, but may lose faithfulness.

7. Relation to other functional abstractions

Another alternative functional abstraction approach is given by τ-ω framework (Rubenstein et al., 2017); see App. A.3 for a definition of a τ-ω abstraction. Here we show that a particular well-behaved form of τ-ω abstraction called constructive τ-abstraction (Beckers and Halpern (2019)) is equivalent to an α-abstraction. This equivalence immediately allows us to extend the connection between PCDAGs and bijective constructive τ-abstractions.

Corollary 25 (Equivalence α-abstraction and Constructive τ-abstraction) *The α-abstraction is equivalent to the constructive τ-abstraction, if for all settings $\mathbf{v} \in \mathcal{R}(\mathbf{V})$ there exists a $\mathbf{u} \in \mathcal{R}(\mathbf{U})$ giving rise to $\mathbf{v}$.* [Proof in App. B.14.]

8. Conclusion

In this paper we have shown how graphical and functional abstraction are related by showing the connection between CDAGs/PCDAGs and bijective $\mathcal{L}_2$-consistent α-abstractions. This alignment highlights the dual graphical and functional nature of abstractions (similar to SCMs) and allows us to take advantage both of the constructive definitions of graphical abstractions (which can provide strong consistency guarantees by construction) and the declarative definitions of functional abstractions (which establish explicit maps between the models and the data generated by the models). Furthermore, alignment of graphical and functional abstraction suggests that any functional abstraction learning algorithm aimed at learning new simplified models with the requirement of $\mathcal{L}_2$-consistency can rely on the algorithmic procedure in the definition of the PCDAG to learn the structure of an abstracted model. Future work will consider further theoretical study of the relations between the abstraction frameworks, in particular considering implicit and explicit restrictions and their implications; this would allow for a more immediate transfer of results and methods across frameworks. Finally, our results may be exploited to improve abstraction learning algorithm by taking advantage of graphical and functional aspects.

Acknowledgments

We want to recognize the constructive feedback and insightful suggestions from the anonymous reviewers, which have contributed to improving the quality of this paper.

References

Tara V Anand, Adele H Ribeiro, Jin Tian, and Elias Bareinboim. Causal effect identification in cluster dags. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 12172–12179, 2023.

Elias Bareinboim, Juan D. Correa, Duligur Ibeling, and Thomas Icard. *On Pearl's Hierarchy and the Foundations of Causal Inference*, page 507–556. Association for Computing Machinery, New York, NY, USA, 1 edition, 2022. ISBN 9781450395861. URL https://doi.org/10.1145/3501714.3501743.

Sander Beckers and Joseph Y Halpern. Abstracting causal models. In *Proceedings of the aaai conference on artificial intelligence*, volume 33, pages 2678–2685, 2019.

Joel Dyer, Nicholas Bishop, Yorgos Felekis, Fabio Massimo Zennaro, Anisoara Calinescu, Theodoros Damoulas, and Michael Wooldridge. Interventionally consistent surrogates for agent-based simulators. *arXiv preprint arXiv:2312.11158*, 2023.

Yorgos Felekis, Fabio Massimo Zennaro, Nicola Branchini, and Theodoros Damoulas. Causal optimal transport of abstractions. In *Causal Learning and Reasoning*, pages 462–498. PMLR, 2024.

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. Causal abstractions of neural networks. *Advances in Neural Information Processing Systems*, 34:9574–9586, 2021.

Dan Geiger, Thomas Verma, and Judea Pearl. d-separation: From theorems to algorithms. In *Machine intelligence and pattern recognition*, volume 10, pages 139–148. Elsevier, 1990.

Armin Kekić, Bernhard Schölkopf, and Michel Besserve. Targeted reduction of causal models. In *Uncertainty in Artificial Intelligence*, pages 1953–1980. PMLR, 2024.

Riccardo Massidda, Atticus Geiger, Thomas Icard, and Davide Bacciu. Causal abstraction with soft interventions. *arXiv preprint arXiv:2211.12270*, 2022.

Riccardo Massidda, Sara Magliacane, and Davide Bacciu. Learning causal abstractions of linear structural causal models. In *Uncertainty in Artificial Intelligence*, pages 2486–2515. PMLR, 2024.

Pekka Parviainen and Samuel Kaski. Learning structures of bayesian networks for variable groups. *International Journal of Approximate Reasoning*, 88:110–127, 2017. ISSN 0888-613X. doi: https://doi.org/10.1016/j.ijar.2017.05.006. URL https://www.sciencedirect.com/science/article/pii/S0888613X17303134.

Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, USA, 2nd edition, 2009. ISBN 052189560X.

Eigil F. Rischel and Sebastian Weichwald. Compositional abstraction error and a category of causal models. In Cassio de Campos and Marloes H. Maathuis, editors, *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pages 1013–1023. PMLR, 27–30 Jul 2021. URL https://proceedings.mlr.press/v161/rischel21a.html.

Eigil Fjeldgren Rischel. The category theory of causal models. *Master's thesis, University of Copenhagen*, 2020.

Paul K Rubenstein, Sebastian Weichwald, Stephan Bongers, Joris M Mooij, Dominik Janzing, Moritz Grosse-Wentrup, and Bernhard Schölkopf. Causal consistency of structural equation models. *33rd Conference on Uncertainty in Artificial Intelligence 2017*, 2017.

Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021. doi: 10.1109/JPROC.2021.3058954.

Jonas Wahl, Urmi Ninad, and Jakob Runge. Foundations of causal discovery on groups of variables. *Journal of Causal Inference*, 12, 07 2024. doi: 10.1515/jci-2023-0041.

Kevin Xia and Elias Bareinboim. Neural causal abstractions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 20585–20595, 2024.

Fabio Massimo Zennaro, Máté Drávucz, Geanina Apachitei, W. Dhammika Widanage, and Theodoros Damoulas. Jointly learning consistent causal abstractions over multiple interventional distributions. In *2nd Conference on Causal Learning and Reasoning*, 2023. URL https://openreview.net/forum?id=RNs7aMS6zDq.

Fabio Massimo Zennaro, Nicholas George Bishop, Joel Dyer, Yorgos Felekis, Ani Calinescu, Michael J Wooldridge, and Theodoros Damoulas. Causally abstracted multi-armed bandits. In *The 40th Conference on Uncertainty in Artificial Intelligence*, 2024.

Appendix A. Supplements

A.1. Finite Graphical Model and α-abstraction

This supplement defines the α-abstraction as introduced by Rischel (2020); Rischel and Weichwald (2021). First we must introduce the Finite Graphical Model: a formalism for causal models, different from the SCM.

Definition 26 (Finite Graphical Model) (Rischel (2020)) *A finite graphical model $\mathcal{M}$ contains the following:*

1. *A Directed Acyclic Graph (DAG) $G = \langle \mathbf{V}, \mathbf{E} \rangle$.*
2. *For each vertex $V \in \mathbf{V}$ a finite set of values it can take, denoted by $\mathcal{M}[V]$.*
3. *For each vertex $V \in \mathbf{V}$ a stochastic matrix $\mathcal{M}[\psi_V]$ giving the probability distributions of V for all values its parents, $Pa(V)$, can take.*

The Finite Graphical Model is the basis for the definition of the α-abstraction. The α-abstraction defines the maps and properties that describe an abstraction from a low level Finite Graphical Model to a high level one.

Definition 27 (α-abstraction) (Rischel (2020); Rischel and Weichwald (2021)) *An abstraction of finite graphical models $\alpha : \mathcal{M} \to \mathcal{M}'$ consists of the following:*

1. *A subset $\mathbf{R} \subseteq \mathbf{V}_{\mathcal{M}}$ of relevant variables.*
2. *A surjective map $\varphi : \mathbf{R} \to \mathbf{V}_{\mathcal{M}'}$. Mapping all relevant variables to the variables of the abstraction.*
3. *For each $V \in \mathbf{V}_{\mathcal{M}'}$, a surjective function $\mathcal{R}(\varphi^{-1}(V)) \to \mathcal{R}(V)$. Mapping the range of the pre-image $\varphi^{-1}(V)$ in $\mathcal{M}$ to the range of V in $\mathcal{M}'$.*

A.2. Example - Surjectivity vs. CDAG

Example 3 (Surjectivity does not imply a CDAG) Here we will construct an example showing that only surjectivity is not a sufficient assumption to ensure that a $\mathcal{L}_2$-consistent α-abstraction agrees with the structure of a CDAG, motivating the bijectivity assumption used throughout the paper. Note however, that bijectivity itself is a sufficient but not minimal assumption, as will become clear in this example.

First consider the SCM $\mathcal{M}$ with variables $\mathbf{V}_{\mathcal{M}} = \{X, Z, Y\}$, $\mathcal{R}(X) = \{1, 2\}$, $\mathcal{R}(Z) = \{1, 3, 5, 7, 9, \dots\}$, and $f_Y \in \mathcal{F}_{\mathcal{M}} = X \times Z$, giving rise to the DAG:

$$\begin{tikzcd} X \arrow[dr] & \\ & Y \\ Z \arrow[ur] & \end{tikzcd}$$

Secondly, consider the SCM $\mathcal{M}'$ with variables $\mathbf{V}_{\mathcal{M}'} = \{X', Z', Y'\}$, $\mathcal{R}(X') = \{1, 2\}$, $\mathcal{R}(Z') = \{1, 3, 5, 7, 9, \dots\}$, and $f_{Y'} \in \mathcal{F}_{\mathcal{M}'} = X \times Z \mod 2$. It follows immediately that Y' is dependent only on the value of X as for all $z \in \mathcal{R}(Z)$, $X \times z \mod 2$ equals 1 if X is 1, and 0 if X is 2.

There is a natural abstraction $\boldsymbol{\alpha} : \mathcal{M} \to \mathcal{M}'$ given by the maps:

$$\begin{aligned} \varphi &:= \{X \mapsto X', Z \mapsto Z', Y \mapsto Y'\} \\ \alpha_{X'} &:= \mathcal{R}(X) \overset{identity}{\longmapsto} \mathcal{R}(X') \\ \alpha_{Z'} &:= \mathcal{R}(Z) \overset{identity}{\longmapsto} \mathcal{R}(Z') \\ \alpha_{Y'} &:= \begin{cases} Y \mapsto 0 & Y \text{ is even} \\ Y \mapsto 1 & Y \text{ is odd} \end{cases} \end{aligned}$$

One can verify that this abstraction is functionally $\mathcal{L}_2$ consistent. However, through the surjective mapping $\alpha_{Y'} : Y \to Y'$ the dependence of Y' on Z' is removed. Essentially what we have done here is summing out the effect of Z over the possible values of X.

Now if one is to construct the CDAG of $\mathcal{M}$ given by φ we get:

$$\begin{tikzcd} X \arrow[rr, dotted, mapsto] \arrow[dr] & & X' \arrow[dr] & \\ & Y \arrow[rr, dotted, mapsto] & & Y' \\ Z \arrow[ur] \arrow[rr, dotted, mapsto] & & Z' \arrow[ur] & \end{tikzcd}$$

with all edges preserved according to the rules of the CDAG.

However, we have shown that the dependency of Y' on Z' no longer exists, and therefore the edge $Z' \to Y'$ is not faithful. This illustrates that the CDAG is not necessarily compatible with surjective $\mathcal{L}_2$-consistent α-abstractions.

A.3. Definition Constructive τ abstraction

Definition 28 *Let $Rst(\mathbf{V}, x) = \{v \in \mathcal{R}(V) | x$ is the restriction of v to $X\}$ and let $\tau : \mathcal{R}(\mathbf{V}_{\mathcal{M}}) \to \mathbf{C}_{\mathcal{M}'}$ be given, then define $\omega_\tau(do(V_i = v_i)) = do(\mathbf{C}_i = \mathbf{c}_i)$ if $\mathbf{C}_i \in \mathbf{C}$, $\mathbf{c}_i \in \mathcal{R}(\mathbf{C}_i)$ and $\tau(Rst(V_i, v_i)) = Rst(\mathbf{C}_i, \mathbf{c}_i)$. (Beckers and Halpern, 2019)*

Definition 29 (Constructive τ abstraction (Beckers and Halpern, 2019)**)** *Let $\mathcal{M}$, and $\mathcal{M}'$ be SCMs, then $\mathcal{M}'$ is a constructive τ-abstraction of $\mathcal{M}$ if there are:*

- *a surjective function* $\tau : \mathcal{R}(\mathbf{V}_{\mathcal{M}}) \rightarrow \mathcal{R}(\mathbf{C}_{\mathcal{M}'})$*, such that there exists a partition* $P = \{Z_1, \ldots, Z_{n+1}\}$ *of* $\mathbf{V}$ *with* $Z_1, \ldots, Z_n$ *non-empty and mappings* $\tau_i : \mathcal{R}(Z_i) \rightarrow \mathcal{R}(\mathbf{C}_i)$ *for* $i = 1, \ldots, n$ *such that* $\tau = (\tau_1, \ldots, \tau_n)$*.*
- *a surjective function* $\tau_{\mathbf{U}} : \mathcal{R}(\mathbf{U}_{\mathcal{M}}) \rightarrow \mathcal{R}(\mathbf{U}_{\mathcal{M}'})$ *mapping the exogenous range of* $\mathbf{U}_{\mathcal{M}}$ *to the exogenous range of* $\mathbf{U}_{\mathcal{M}'}$*, compatible with* τ*.*
- *the intervention set* $\mathcal{I}_{\mathcal{M}'} = \omega_\tau(\mathcal{I}_{\mathcal{M}})$*, such that all* $\mathcal{I}_{\mathcal{M}'}$ *contains all possible interventions in* $\mathcal{M}'$*.*

$\tau_{\mathbf{U}} : \mathcal{R}(\mathbf{U}_{\mathcal{M}}) \rightarrow \mathcal{R}(\mathbf{U}_{\mathcal{M}'})$ is compatible with $\tau : \mathcal{R}(\mathbf{V}_{\mathcal{M}}) \rightarrow \mathcal{R}(\mathbf{V}_{\mathcal{M}'})$ if for all $do(V = v) \in \mathcal{I}_{\mathcal{M}}$ and $\mathbf{u} \in \mathcal{R}(\mathbf{U}_{\mathcal{M}})$, $\tau(\mathcal{M}(\mathbf{u}, do(V = v))) = \mathcal{M}'(\tau_{\mathbf{U}}(\mathbf{u}), \omega(do(V = v)))$. (Beckers and Halpern, 2019)

Appendix B. Proofs

B.1. Proof of lemma 9

Lemma 9 *$\mathcal{L}_2$-consistency implies $\mathcal{L}_1$-consistency.*
Proof Let $\boldsymbol{\alpha} : \mathcal{M} \rightarrow \mathcal{M}'$ be an $\mathcal{L}_2$-consistent abstraction, and $\mathbf{X}, \mathbf{Y} \subseteq \mathbf{V}_{\mathcal{M}'}$ with $\mathbf{x} \in \mathcal{R}(\mathbf{X}), \mathbf{y} \in \mathcal{R}(\mathbf{Y})$.

We will show that by the null intervention $\mathcal{L}_2$-consistency implies consistency over marginal and joint probabilities. We continue by showing that consistency over marginal and joint probabilities guarantee consistency for conditional distributions. Therefore, we conclude that $\mathcal{L}_2$-consistency implies $\mathcal{L}_1$ consistency. Important for this proof is surjectivity of all range maps $\alpha_V(v)$ to ensure the pre-image $\alpha_V^{-1}(v)$ exists for all $v \in \mathcal{R}(V)$.

First, by definition of $\mathcal{L}_2$-consistency the following two equations hold:

$$P(\varphi^{-1}(\mathbf{X}) = \alpha_{\mathbf{X}}^{-1}(\mathbf{x}) | do(\emptyset)) \overset{\mathcal{L}_2}{=} P(\mathbf{X} = \mathbf{x} | do(\emptyset)),$$

$$P(\varphi^{-1}(\mathbf{Y}) = \alpha_{\mathbf{Y}}^{-1}(\mathbf{y}), \varphi^{-1}(\mathbf{X}) = \alpha_{\mathbf{X}}^{-1}(\mathbf{x}) | do(\emptyset)) \overset{\mathcal{L}_2}{=} P(\mathbf{Y} = \mathbf{y}, \mathbf{X} = \mathbf{x} | do(\emptyset)).$$

And, by definition of the null intervention:

$$P(\varphi^{-1}(\mathbf{X}) = \alpha_{\mathbf{X}}^{-1}(\mathbf{x})) \overset{\mathcal{L}_2}{=} P(\mathbf{X} = \mathbf{x}), \text{ and}$$

$$P(\varphi^{-1}(\mathbf{Y}) = \alpha_{\mathbf{Y}}^{-1}(\mathbf{y}), \varphi^{-1}(\mathbf{X}) = \alpha_{\mathbf{X}}^{-1}(\mathbf{x})) \overset{\mathcal{L}_2}{=} P(\mathbf{Y} = \mathbf{y}, \mathbf{X} = \mathbf{x}).$$

It follows from these equations that $\mathcal{L}_2$-consistency implies:

$$\frac{P(\varphi^{-1}(\mathbf{Y}) = \alpha_{\mathbf{Y}}^{-1}(\mathbf{y}), \varphi^{-1}(\mathbf{X}) = \alpha_{\mathbf{X}}^{-1}(\mathbf{x}))}{P(\varphi^{-1}(\mathbf{X}) = \alpha_{\mathbf{X}}^{-1}(\mathbf{x}))} \overset{\mathcal{L}_2}{=} \frac{P(\mathbf{Y} = \mathbf{y}, \mathbf{X} = \mathbf{x})}{P(\mathbf{X} = \mathbf{x})}$$

Finally, by applying Bayes' Theorem the ratios can be substituted by conditional distributions:

$$P\left(\varphi^{-1}(\mathbf{Y}) = \alpha_{\mathbf{Y}}^{-1}(\mathbf{y}) | \varphi^{-1}(\mathbf{X}) = \alpha_{\mathbf{X}}^{-1}(\mathbf{x})\right) \overset{\mathcal{L}_2}{=} P\left(\mathbf{Y} = \mathbf{y} | \mathbf{X} = \mathbf{x}\right)$$

Note that the last equation immediately implies $\mathcal{L}_1$-consistency. Therefore, any $\mathcal{L}_2$-consistent abstraction $\boldsymbol{\alpha} : \mathcal{M} \rightarrow \mathcal{M}'$ is necessarily $\mathcal{L}_1$-consistent.

■

B.2. Proof of lemma 11

Lemma 11 (Bijective Range Maps Preserve Distribution (In)Equalities) *Let $\boldsymbol{\alpha} : \mathcal{M} \to \mathcal{M}'$ be an $\mathcal{L}_2$-consistent abstraction with $\alpha_V : \mathcal{R}\left(\varphi^{-1}(V)\right) \to \mathcal{R}(V)$ bijective for all $V \in \mathbf{V}_{\mathcal{M}'}$. Let $\mathbf{X}, \mathbf{Y}_1, \mathbf{Y}_2, \mathbf{Z}_1, \mathbf{Z}_2 \subseteq \mathbf{V}_{\mathcal{M}}$ be partitions defined by φ, then $P(\mathbf{X}|do(\mathbf{Y}_1), \mathbf{Z}_1) = P(\mathbf{X}|do(\mathbf{Y}_2), \mathbf{Z}_2)$ if and only if $\alpha_{\mathbf{X}}[P(\mathbf{X} \mid do(\mathbf{Y}_1), \mathbf{Z}_1)] = \alpha_{\mathbf{X}}[P(\mathbf{X} \mid do(\mathbf{Y}_2), \mathbf{Z}_2)]$*

Proof Let $\boldsymbol{\alpha} : \mathcal{M} \to \mathcal{M}'$ be a $\mathcal{L}_2$-consistent abstraction with $\alpha_V : \mathcal{R}\left(\varphi^{-1}(V)\right) \to \mathcal{R}(V)$ bijective for all $V \in \mathbf{V}_{\mathcal{M}'}$. Let $\mathbf{X}, \mathbf{Y}_1, \mathbf{Y}_2, \mathbf{Z}_1, \mathbf{Z}_2 \subseteq \mathbf{V}_{\mathcal{M}}$, such that $\mathbf{X}, \mathbf{Y}_1, \mathbf{Y}_2, \mathbf{Z}_1, \mathbf{Z}_2$ are pre-images of φ.

This proof will consist of two parts, first we show that surjectivity of the range map gives us that

$$\alpha_{\mathbf{X}}[P(\mathbf{X} \mid do(\mathbf{Y}_1), \mathbf{Z}_1)] \neq \alpha_{\mathbf{X}}[P(\mathbf{X} \mid do(\mathbf{Y}_2), \mathbf{Z}_2)]$$
$$\text{implies}$$
$$P(\mathbf{X} \mid do(\mathbf{Y}_1), \mathbf{Z}_1) \neq P(\mathbf{X} \mid do(\mathbf{Y}_2), \mathbf{Z}_2).$$

Secondly, we show that bijectivity gives us the inverse:

$$P(\mathbf{X} \mid do(\mathbf{Y}_1), \mathbf{Z}_1) \neq P(\mathbf{X} \mid do(\mathbf{Y}_2), \mathbf{Z}_2)$$
$$\text{implies}$$
$$\alpha_{\mathbf{X}}[P(\mathbf{X} \mid do(\mathbf{Y}_1), \mathbf{Z}_1)] \neq \alpha_{\mathbf{X}}[P(\mathbf{X} \mid do(\mathbf{Y}_2), \mathbf{Z}_2)],$$

completing the proof.

Part 1. First consider the case where

$$\alpha_{\mathbf{X}}[P(\mathbf{X} \mid do(\mathbf{Y}_1), \mathbf{Z}_1)] \neq \alpha_{\mathbf{X}}[P(\mathbf{X} \mid do(\mathbf{Y}_2), \mathbf{Z}_2)].$$

Surjectivity ensures that the distributions have pre-images in $\alpha_{\mathbf{X}}$. Given a surjective function g if $g(a) \neq g(b)$ then $a \neq b$, and since pre-images exist for both distributions, these must also not be equal. So,

$$\alpha_{\mathbf{X}}[P(\mathbf{X} \mid do(\mathbf{Y}_1), \mathbf{Z}_1)] \neq \alpha_{\mathbf{X}}[P(\mathbf{X} \mid do(\mathbf{Y}_2), \mathbf{Z}_2)]$$
$$\text{implies}$$
$$P(\mathbf{X} \mid do(\mathbf{Y}_1), \mathbf{Z}_1) \neq P(\mathbf{X} \mid do(\mathbf{Y}_2), \mathbf{Z}_2).$$

Part 2. Secondly, consider the case where

$$P(\mathbf{X} \mid do(\mathbf{Y}_1), \mathbf{Z}_1) \neq P(\mathbf{X} \mid do(\mathbf{Y}_2), \mathbf{Z}_2).$$

Bijectivity of $\alpha_{\mathbf{X}}$ ensures there exists an inverse to $\alpha_{\mathbf{X}}$ such that

$$\alpha_{\mathbf{X}}^{-1}[\alpha_{\mathbf{X}}[P(\mathbf{X} \mid do(\mathbf{Y}_1), \mathbf{Z}_1)]] = P(\mathbf{X} \mid do(\mathbf{Y}_1), \mathbf{Z}_1)$$
$$\text{and}$$
$$\alpha_{\mathbf{X}}^{-1}[\alpha_{\mathbf{X}}[P(\mathbf{X} \mid do(\mathbf{Y}_2), \mathbf{Z}_2)]] = P(\mathbf{X} \mid do(\mathbf{Y}_2), \mathbf{Z}_2).$$

Again, consider that given a function g if $g(a) \neq g(b)$ then $a \neq b$. So since

$$P(\mathbf{X} \mid do(\mathbf{Y}_1), \mathbf{Z}_1) \neq P(\mathbf{X} \mid do(\mathbf{Y}_2), \mathbf{Z}_2)$$
$$\text{it must hold that}$$
$$\alpha_{\mathbf{X}}[P(\mathbf{X} \mid do(\mathbf{Y}_1), \mathbf{Z}_1)] \neq \alpha_{\mathbf{X}}[P(\mathbf{X} \mid do(\mathbf{Y}_2), \mathbf{Z}_2)].$$

Conclusion. We have illustrated that, given a bijective $\mathcal{L}_2$-consistent abstraction $\boldsymbol{\alpha}$,

$$P(\mathbf{X} \mid do(\mathbf{Y}_1), \mathbf{Z}_1) \neq P(\mathbf{X} \mid do(\mathbf{Y}_2), \mathbf{Z}_2)$$
$$\text{if and only if}$$
$$\alpha_{\mathbf{X}}[P(\mathbf{X} \mid do(\mathbf{Y}_1), \mathbf{Z}_1)] \neq \alpha_{\mathbf{X}}[P(\mathbf{X} \mid do(\mathbf{Y}_2), \mathbf{Z}_2)],$$

and therefore also

$$P(\mathbf{X} \mid do(\mathbf{Y}_1), \mathbf{Z}_1) = P(\mathbf{X} \mid do(\mathbf{Y}_2), \mathbf{Z}_2)$$
$$\text{if and only if}$$
$$\alpha_{\mathbf{X}}[P(\mathbf{X} \mid do(\mathbf{Y}_1), \mathbf{Z}_1)] = \alpha_{\mathbf{X}}[P(\mathbf{X} \mid do(\mathbf{Y}_2), \mathbf{Z}_2)].$$

■

B.3. Proof of lemma 12

Lemma 12 ($\mathcal{L}_2$-Consistency Uniquely Determines Adjacencies) *Given an $\mathcal{L}_2$-consistent bijective abstraction $\boldsymbol{\alpha} : \mathcal{M} \to \mathcal{M}'$, adjacencies between variables in $\mathcal{M}'$ are uniquely determined by the map $\varphi : \mathbf{V} \to \mathbf{C}$ and comply with Rule 1 of Def.3 (CDAG).*

Proof The proof follows from the generalized proof of Thm.12. Specifically, Rule 1 of Def.16 (PCDAG) is equivalent to Rule 1 of Def.3 (CDAG) for any abstraction with $\mathbf{R} = \mathbf{V}_{\mathcal{M}}$. ■

B.4. Proof of lemma 13

Lemma 13 ($\mathcal{L}_2$-Consistency Uniquely Determines Confounding Edges) *Given an $\mathcal{L}_2$-consistent bijective abstraction $\boldsymbol{\alpha} : \mathcal{M} \to \mathcal{M}'$, confounding edges between variables in $\mathcal{M}'$ are uniquely determined by the map $\varphi : \mathbf{V} \to \mathbf{C}$ and comply with Rule 2 of Def.3 (CDAG).*

Proof The proof follows from the generalized proof of Thm.22. Specifically, Rule 2A of Def.16 (PCDAG) is equivalent to Rule 2 of Def.3 (CDAG) for any abstraction with $\mathbf{R} = \mathbf{V}_{\mathcal{M}}$, and any edges produced by rule 2B of the PCDAG require a non-empty remainder set. ■

B.5. Proof of theorem 14

Theorem 14 (α-Abstraction $\Rightarrow$ CDAG) *Given any bijective $\mathcal{L}_2$-consistent α-abstraction $\boldsymbol{\alpha} : \mathcal{M} \to \mathcal{M}'$ with $\mathbf{R} = \mathbf{V}$, the DAG of $\mathcal{M}'$ is a CDAG of the graph of $\mathcal{M}$.*

Proof This proof follows immediately from the results of Lem.12 and Lem.13. Together the lemmas state that both all directed and all confounding edges of a bijective $\mathcal{L}_2$-consistent abstraction with $\mathbf{R} = \mathbf{V}_{\mathcal{M}}$ are necessarily equal to those given by the constructive rules of the CDAG. ■

B.6. Proof of theorem 15

Theorem 15 (CDAG $\Rightarrow$ α-Abstraction) *Given a CDAG there exists an equivalent $\mathcal{L}_2$-consistent α-abstraction $\boldsymbol{\alpha} : \mathcal{M} \to \mathcal{M}'$ with $\mathbf{R} = \mathbf{V}$, φ given by the clustering, and all maps $\alpha_{\mathbf{C}}$ bijective.*
Proof The proof follows immediately from the result of Thm.20, showing there exists an $\mathcal{L}_2$-consistent α-abstraction for each PCDAG by taking bijective range maps $\alpha_{\mathbf{C}}$ and the set of mechanisms $\mathcal{F}_{\mathcal{M}'}$ to be the composite mechanisms as implied by φ. Note that a CDAG is by deinition also a PCDAG, so the result transfers. ■

B.7. Proof of lemma 18

Lemma 18 (PCDAGs Preserve Mediated Adjancencies) *Let $\mathcal{M}$ be an SCM with a PCDAG defined by clusters $\mathbf{C}$ and remainder set $\mathbf{Q}$. Let $\mathbf{C}_i, \mathbf{C}_j \in \mathbf{C},\ \mathbf{C}_i \neq \mathbf{C}_j$ and $V_i \in \mathbf{C}_i,\ V_j \in \mathbf{C}_j$, then a mediated adjacency between V_i and V_j exists if and only if there exists an adjacency between $\mathbf{C}_i$ and $\mathbf{C}_j$.*
Proof It follows directly from Def.16.1 (Partial Cluster DAG) that an mediated adjacency between $V_i \in \mathbf{C}_i$ and $V_j \in \mathbf{C}_j$ creates an edge between $\mathbf{C}_i$ and $\mathbf{C}_j$. It follows that, given two clusters $\mathbf{C}_i, \mathbf{C}_j \in \mathbf{C}$, if there exists an mediated adjacency between two variables $V_i \in \mathbf{C}_i, V_j \in \mathbf{C}_j$ in $\mathcal{M}$ there exists an adjacency between $\mathbf{C}_i$ and $\mathbf{C}_j$ in $\mathcal{M}'$.

Furthermore, this is the only rule creating a directed edge, and therefore solely defines adjacencies in a PCDAG. It follows that, given two clusters $\mathbf{C}_i, \mathbf{C}_j \in \mathbf{C}$, if there does not exists some combination $V_i \in \mathbf{C}_i, V_j \in \mathbf{C}_j$ in $\mathcal{M}$ with an mediated adjacency between V_i and V_j then there is no adjacency between $\mathbf{C}_i$ and $\mathbf{C}_j$ in $\mathcal{M}'$.

Thus showing that for each mediated adjacency in $\mathcal{M}$ there exists a corresponding adjacency in $\mathcal{M}'$, and for all adjacencies in $\mathcal{M}'$ there exists at least one corresponding mediated adjacency in $\mathcal{M}$. ■

B.8. Proof of lemma 19

Lemma 19 (PCDAGs Preserve Directed Paths) *Let $\mathcal{M}$ be an SCM with a PCDAG defined by clusters $\mathbf{C}$ and remainder set $\mathbf{Q}$. Let $\mathbf{C}_i, \mathbf{C}_j \in \mathbf{C},\ \mathbf{C}_i \neq \mathbf{C}_j$ and $V_i \in \mathbf{C}_i,\ V_j \in \mathbf{C}_j$, then a directed path $\mathbf{C}_i \to \cdots \to \mathbf{C}_j$ exists if there exists a directed path $V_i \to \cdots \to V_j$.*
Proof First, by Def.16.1 (Partial Cluster DAG) adjacencies in the PCDAG have the same direction as the mediated adjacencies in the original DAG. Secondly, by Lem.18 adjacencies in the PCDAG are consistent with mediated adjacencies in $\mathcal{M}$. Note that adjacency are directed paths of length 1, so if follows immediately that such directed paths are preserved.

Now consider directed paths of length greater than 1. Let $V_i \in \mathbf{C}_i,\ V_j \in \mathbf{C}_j$ with $\mathbf{C}_i, \mathbf{C}_j \in \mathbf{C}$ and let there be a directed path $V_i \to \cdots \to V_j$. Any directed path can be split up into a sequence of mediated adjacencies. Take V_i and the first vertex V_x along the path such that $V_x \notin \mathbf{R}$, then either (i) there exists an edge $V_i \to V_x$ or (ii) there exist a path $V_i \to \cdots \to V_x$ such that any intermediate vertices are in $\mathbf{R}$. Therefore, there exists a mediated adjacency between V_i and V_x, by Lem.18 this implies $\mathbf{C}_i \to \mathbf{C}_x,\ V_x \in \mathbf{C}_x$ if $\mathbf{C}_x \neq \mathbf{C}_i$. By repeated application of this process starting from V_x, until we reach V_j, all generated edges between clusters can be composed into the directed path $\mathbf{C}_i \to \cdots \to \mathbf{C}_j$. ■

B.9. Proof of theorem 20

Theorem 20 ($\mathcal{L}_2$-consistency of PCDAGs) *Given an SCM $\mathcal{M}$ and a PCDAG $\mathcal{M}'$ of $\mathcal{M}$, there exists a set of mechanisms $\mathcal{F}_{\mathcal{M}'}$ such that $\mathcal{M}'$ is $\mathcal{L}_2$-consistent with $\mathcal{M}$.*

Proof For this proof we will show that a given two models $\mathcal{M}, \mathcal{M}'$ where $\mathcal{M}'$ is PCDAG of $\mathcal{M}$, there exists a configuration of $\mathcal{M}'$ such that $\mathcal{M}'$ is $\mathcal{L}_2$ consistent with $\mathcal{M}$. We will define the functions for $\mathcal{M}'$ according to the clustering, and show that these functions correspond to the structure of the PCDAG. After we show that cluster variables together with these functions give us $\mathcal{L}_2$ consistency.

First, let $\mathcal{M} : \langle \mathbf{U_V}, \mathbf{V}, \mathcal{F}_\mathbf{V}, P_\mathbf{V}(\mathbf{U_V}) \rangle$ and $\mathcal{M}' : \langle \mathbf{U_C}, \mathbf{C}, \mathcal{F}_\mathbf{C}, P_\mathbf{C}(\mathbf{U_C}) \rangle$ be two SCMs such that the DAG of $\mathcal{M}'$ is a PCDAG of $\mathcal{M}$ and $\mathbf{C}$ the corresponding clusters.

We choose $\mathbf{U_C} = \mathbf{U_V}$ and $P_\mathbf{C}(\mathbf{U_C}) = P_\mathbf{V}(\mathbf{U_V})$, and let the variables $\mathbf{C}_i \in \mathbf{C}$ be given by the clustering such that

$$\mathbf{C}_i = \begin{pmatrix} V_1^i \\ \vdots \\ V_n^i \end{pmatrix},$$

with n as the number of variables in cluster $\mathbf{C}_i$ and $V_j^i \in \mathbf{C}_i$ as the j-th variable in the cluster $\mathbf{C}_i$. Now construct the functions $f_{\mathbf{C}_i} \in \mathcal{F}_\mathbf{C}$ again following the clustering:

$$f_{\mathbf{C}_i} = \begin{pmatrix} f_{V_1^i}(Pa(V_1^i), \mathbf{U}_{V_1^i}) \\ \vdots \\ f_{V_n^i}(Pa(V_n^i), \mathbf{U}_{V_n^i}) \end{pmatrix},$$

with $f_{V_j^i} \in \mathcal{F}_\mathbf{V}$ the function for V_j^i in $\mathcal{M}$, and $\mathbf{U}_{V_j^i}$ the exogenous parents of V_j^i.

This definition of $f_{\mathbf{C}_i}$ leaves two potential issues:

1. There can exist a function $f_{V_j^i}$ part of $f_{\mathbf{C}_i}$ which relies on a variable mapped to $\mathbf{Q}$. By Def.16 of the PCDAG there exists no edges to or from $\mathbf{Q}$, therefore such reliance must somehow be removed.

2. There can exist a function $f_{V_j^i}$ part of $f_{\mathbf{C}_i}$ which relies on a variable V_k^i, which is also mapped to $\mathbf{C}_i$. Acyclicity requires there exist no edge $\mathbf{C}_i \to \mathbf{C}_i$, therefore such reliance must also be removed.

First we describe how we can rewrite functions in the case of 1.: take $f_{V_j^i}(Pa(V_j^i), \mathbf{U}_{V_j^i})$ part of $f_{\mathbf{C}_i}$ with $Pa(V_j^i) \cap \mathbf{Q} \neq \emptyset$. Let $\mathbf{P} = Pa(V_j^i) \cap \mathbf{Q}$ be the set of parents of V_j^i that are mapped to $\mathbf{Q}$. All variables $P \in \mathbf{P}$ can be substituted with their functions instead. These functions will in turn depend on a set of parents and exogenous parents. By acyclicity recursively applying this technique all functions will have $Pa(P) \cap \mathbf{Q} = \emptyset$ in a finite number of applications. When $Pa(P) \cap \mathbf{Q} = \emptyset$ all parents of P are part of some cluster, or there are no endogenous parents left. Note that this gets reliance on exogenous parents of all Ps encountered.

For case 2. we follow a similar approach: take $f_{V_j^i}(Pa(V_j^i), \mathbf{U}_{V_j^i})$ part of $f_{\mathbf{C}_i}$ with some subset $\mathbf{P} \subseteq Pa(V_j^i) \cap \mathbf{C}_i \neq \emptyset$. Applying the same substitution technique as before gives a function that only relies on variables outside of the cluster $\mathbf{C}_i$.

These functions agree with the structure given by Def.16 of the PCDAG as causal edges $\mathbf{C}_i \to \mathbf{C}_j$ exists if and only if there exists some $V_i \in \mathbf{C}_i$ and $V_j \in \mathbf{C}_j$ such that there is a mediated adjacency $V_i \rightsquigarrow V_j$. Furthermore, recall that $\mathbf{C}_i \leftarrow\!\!\dashrightarrow \mathbf{C}_j$ is a shorthand for $\mathbf{C}_i \leftarrow U \to \mathbf{C}_j$. Given the definition of the functions $\mathcal{F}_{\mathbf{C}}$ such confounding edges occur either (i) if there exists $V_i \in \mathbf{C}_i$ and $V_j \in \mathbf{C}_j$ such that $V_i \leftarrow U \to V_j$, or (ii) $V_i \in \mathbf{C}_i$, $V_j \in \mathbf{C}_j$, and $Q \in \mathbf{Q}$ such that

$$\begin{array}{ccccc} V_i & \leftarrow & Q & \rightarrow & V_j \\ & & \uparrow & & \\ & & U & & \end{array}.$$

Note that these cases exactly correspond to Def.16 (Partial Cluster DAG). This shows that the definition of $\mathcal{F}_{\mathbf{C}}$ is compatible with the structure defined by the PCDAG.

This leaves to show that such definitions give rise to $\mathcal{L}_2$-consistency. For this, first note that by clustering there is a bijection between the ranges of a cluster $\mathbf{C}_i \in \mathbf{C}$ and the variables that make up the cluster $\mathbf{V}_i \subseteq \mathbf{V}$. Additionally, the function $f_{\mathbf{C}_i}$ is simply a combination of the functions of $\mathbf{V}_i$. So since $\mathbf{U}_{\mathbf{C}} = \mathbf{U}_{\mathbf{V}}$ and $P_{\mathbf{C}}(\mathbf{U}_{\mathbf{C}}) = P_{\mathbf{V}}(\mathbf{U}_{\mathbf{V}})$ it follows immediately that

$$P_{\mathbf{C}}(\mathbf{C}_i \mid do(\mathbf{C}_j), \mathbf{C}_k) = P_{\mathbf{V}}(\mathbf{V}_i \mid do(\mathbf{V}_j), \mathbf{V}_k).$$

Therefore, it is immediate that there exists a model $\mathcal{M}'$ which is a compatible PCDAG of $\mathcal{M}$ that is $\mathcal{L}_2$-consistent with $\mathcal{M}$. ■

B.10. Proof of lemma 21

Lemma 21 ($\mathcal{L}_2$-Consistency Uniquely Determines Adjacencies) *Given a bijective $\mathcal{L}_2$-consistent abstraction $\boldsymbol{\alpha} : \mathcal{M} \to \mathcal{M}'$, adjacencies between variables in $\mathcal{M}'$ are uniquely determined by the map $\varphi : \mathbf{V} \to \mathbf{C}$ and comply with Rule 1 of Def.16 (PCDAG).*

Proof We will show that adjacencies in an abstracted model is uniquely determined by the map of the endogenous variables $\varphi : \mathbf{V} \to \mathbf{C}$. To get to this, we show that adjacencies $\mathbf{C}_i \to \mathbf{C}_j$ for any $\mathbf{C}_i, \mathbf{C}_j \in \mathbf{C}$ exists if, and only if, there exist an mediated adjacency between the corresponding pre-images of φ .

Let $\mathbf{C}$ be a (partial) clustering of $\mathbf{V}$. Given two variables $V_i, V_j \in \mathbf{V}$ such that $V_i \in \mathbf{C}_i$, $V_j \in \mathbf{C}_j$ and $\mathbf{C}_i, \mathbf{C}_j \in \mathbf{C}$, with $\mathbf{C}_i \neq \mathbf{C}_j$.

First, consider the case where there exists an mediated adjacency $V_i \rightsquigarrow V_j$. By definition $P(V_j) \neq P(V_j \mid do(V_i))$, and by $\mathcal{L}_2$-consistency $P(\mathbf{C}_j) \neq P(\mathbf{C}_j \mid do(\mathbf{C}_i))$, so there must exist a path $\mathbf{C}_i \to \mathbf{C}_j$. Furthermore, by definition of the mediated adjacency there does not exist some $\mathbf{Z} \in \mathbf{C}$ that blocks the effect of $do(V_i)$ such that

$$P(V_j \mid do(V_i),\ do(\varphi^{-1}(\mathbf{Z}))) = P(V_j \mid do(\varphi^{-1}(\mathbf{Z}))).$$

Lem.11 illustrates that

$$\begin{gathered} P(V_j \mid do(V_i),\ do(\varphi^{-1}(\mathbf{Z}))) = P(V_j \mid do(\varphi^{-1}(\mathbf{Z}))) \\ \text{implies} \\ P(\mathbf{C}_j \mid do(\mathbf{C}_i),\ do(\mathbf{Z})) = P(\mathbf{C}_j \mid do(\mathbf{Z})). \end{gathered}$$

In other words, there exists no intervention in $\mathcal{M}'$ that blocks the path $\mathbf{C}_i \to \mathbf{C}_j$, and since all interventions are allowed, it must be that $\mathbf{C}_i \to \mathbf{C}_j$ is an adjacency.

Second, consider the inverse: there exists no mediated adjacency $V_i \rightsquigarrow V_j$. Then we have that there exists some $\mathbf{Z} \subseteq \mathbf{C}$, $V_i \notin \mathbf{Z}$ such that

$$P(V_j \mid do(V_i),\ do(\varphi^{-1}(\mathbf{Z})) = P(V_j \mid do(\varphi^{-1}(\mathbf{Z}))).$$

Interventional consistency requires that

$$P(\mathbf{C}_j \mid do(\mathbf{C}_i),\ do(\mathbf{Z})) = P(\mathbf{C}_j \mid do(\mathbf{Z})).$$

So, if there exists a path $V_i \to V_j$ that can be blocked by an intervention $do(\varphi^{-1}(\mathbf{Z}))$ with $\mathbf{Z} \subseteq \mathbf{C}$, $\mathbf{Z} \neq V_j$, then the path between $\mathbf{C}_i \to \mathbf{C}_j$ can also be blocked by $do(\mathbf{Z})$. Consequently, we have that if there does not exists an mediated adjacency $V_i \rightsquigarrow V_j$ then there is no adjacency $\mathbf{C}_i \to \mathbf{C}_j$.

Therefore, for any given abstraction $\boldsymbol{\alpha} : \mathcal{M} \to \mathcal{M}'$ with $\varphi : \mathbf{V} \to \mathbf{C}$ adjacencies in $\mathcal{M}'$ are uniquely given by interventional consistency of $\boldsymbol{\alpha}$. Moreover, we note that that the preservation of mediated adjacencies exactly complies to rule 1. of Def.16 (PCDAG). ■

B.11. Proof of lemma 22

Lemma 22 ($\mathcal{L}_2$-Consistency Uniquely Determines Confounding Edges) *Given a bijective $\mathcal{L}_2$-consistent abstraction $\boldsymbol{\alpha} : \mathcal{M} \to \mathcal{M}'$, confounding edges between variables in $\mathcal{M}'$ are uniquely determined by the map $\varphi : \mathbf{V} \to \mathbf{C}$ and comply with Rule 2A and 2B of Def.16 (PCDAG).*

Proof The proof for this Lemma is split into two parts: first we show that $\mathcal{L}_2$ distributions can be used to identify confounding edges between non adjacent variables. Secondly, we show the same is possible for adjacent variables, but the proof requires a different approach.

Non Adjacent Confounding Edges. For this part we will show that given two variables $X, Y \in \mathbf{V}$ of a model $\mathcal{M}$ with X and Y not adjacent there exists some $\mathbf{Z} \subseteq \mathbf{V} \setminus \{X, Y\}$ such that

$$P(Y \mid X, do(\mathbf{Z})) = P(Y \mid do(\mathbf{Z}))$$

if and only if there does not exist a confounding edge $X \leftarrow\!\!-\!\!\rightarrow Y$. First we show that non existence of a confounding edge $X \leftarrow\!\!-\!\!\rightarrow Y$ implies

$$P(Y \mid X, do(\mathbf{Z})) = P(Y \mid do(\mathbf{Z})).$$

After, we show that $X \leftarrow\!\!-\!\!\rightarrow Y$ implies

$$P(Y \mid X, do(\mathbf{Z})) \neq P(Y \mid do(\mathbf{Z})),$$

completing this part of the proof.

First, assume X and Y do not share a confounding edge $X \leftarrow\!\!-\!\!\rightarrow Y$. Non-adjacency gives that there exists some $\mathbf{Z} \subset \mathbf{V} \setminus \{X, Y\}$ blocking all paths between X and Y that have at least one intermediate vertex in $\mathbf{V}$. A path without a vertex in $\mathbf{V}$ would imply $X \leftarrow\!\!-\!\!\rightarrow Y$, which by assumption do not exist. So, there exists some $\mathbf{Z} \subset \mathbf{V} \setminus \{X, Y\}$ such that $Y \perp\!\!\!\perp X \mid do(\mathbf{Z})$. Or, in terms of distributions:

$$P(Y \mid X, do(\mathbf{Z})) = P(Y \mid do(\mathbf{Z}))$$

for all non-adjacent X, Y if there does not exist a confounding edge $X \leftarrow\!\!-\!\!\rightarrow Y$.

Secondly, we show the inverse: assume there exists a confounding edge $X \leftarrow\!\dashrightarrow Y$. By assumption of $X \leftarrow\!\dashrightarrow Y$ there exists a path between X and Y that cannot be blocked by any intervention. So there cannot exist a $\mathbf{Z} \in \mathbf{V} \setminus \{X, Y\}$ such that $Y \perp\!\!\!\perp X \mid do(\mathbf{Z})$. This is expressed in terms of distributions as

$$P(Y \mid X, do(\mathbf{Z})) \neq P(Y \mid do(\mathbf{Z}))$$

for all $\mathbf{Z}$, if there is a $X \leftarrow\!\dashrightarrow Y$.

We conclude that all confounding edges between two non-adjacent variables X and Y can be identified using $\mathcal{L}_2$ distributions.

Adjacent Confounding Edges. Whereas one can identify confounding effects by blocking causal paths when considering non-adjacent variables, this is not possible in the case where $X \to Y$. The following part illustrates that $\mathcal{L}_2$ distributions can still be used to identify this confounding effect through inequalities.

Given some $X, Y \in \mathbf{V}$ of $\mathcal{M}$, such that $X \to Y$. we will show that the absence of a confounder $X \leftarrow\!\dashrightarrow Y$ implies

$$P(Y \mid X, do(\mathbf{Z})) = P(Y \mid do(X), do(\mathbf{Z})).$$

Secondly, we show that the presence of a confounder $X \leftarrow\!\dashrightarrow Y$ implies

$$P(Y \mid X, do(\mathbf{Z})) \neq P(Y \mid do(X), do(\mathbf{Z})).$$

First, recall that any indirect path between X and Y can be blocked by some $\mathbf{Z} \in \mathbf{V} \setminus \{X, Y\}$. It follows that given such a $\mathbf{Z}$ we have

$$P(Y \mid X, do(\mathbf{Z})) = P(Y \mid do(X), do(\mathbf{Z})).$$

Now let's consider the following diagram:

$$X \xrightarrow{\overset{U}{\leftarrow\!\dashrightarrow}} Y.$$

Recall that $X \leftarrow\!\dashrightarrow Y$ is an shorthand for writing $X \leftarrow U \to Y$. Note also that an arrow $X \to Y$ implies $P(Y) \neq P(Y \mid X)$ and $P(X) \neq P(X \mid Y)$, assuming non-canceling paths. Now consider what happens when conditioning on X by taking $P(Y \mid X)$: there exists an arrow $U \to X$, so $P(U \mid X) \neq P(U)$.

Alternatively, consider $P(Y \mid do(X))$ would break the edge $U \to X$, so $P(U) = P(U \mid do(X))$. Furthermore the distributions $P(Y \mid X)$ and $P(Y \mid do(X))$ are determined by a function $f_Y(U, U_Y, X)$ with U_Y some optional exogenous variable affecting Y. Therefore, given X and a distribution $P(U_Y)$ the generated distributions depend only on the distribution $P(U)$, and since

$$P(U \mid do(X)) = P(U) \neq P(U \mid X)$$

it follows that

$$P(Y \mid X) \neq P(Y \mid do(X)).$$

In the case the model has more variables than X and Y any other path can be blocked by some $\mathbf{Z}$, similar to the first part.

In conclusion, given two adjacent variables X, Y such that $X \to Y$, there exists a confounding arrow $X \leftarrow\!\dashrightarrow Y$ if, and only if,

$$P(Y \mid X, do(\mathbf{Z})) \neq P(Y \mid do(X), do(\mathbf{Z})).$$

Preservation of Confounding Edges. Given a bijective $\mathcal{L}_2$-consistent α-abstraction $\boldsymbol{\alpha} : \mathcal{M} \to \mathcal{M}'$, Lem.11 states that all (in)equalities among distributions in $\mathcal{M}'$ have a corresponding (in)equality in $\mathcal{M}$. As confounding edges can be determined uniquely by these inequalities, if $\mathbf{R} = \mathbf{V}_{\mathcal{M}}$ then a confounding edge $\mathbf{V}_i \to \mathbf{V}_j$ in $\mathcal{M}$ implies a corresponding confounding edge $\mathbf{C}_i \leftarrow\!\dashrightarrow \mathbf{C}_j$ in $\mathcal{M}'$, in accordance to rule 2A of Def.16 (PCDAG).

Creation of Confounding edges. Additionally, note that if $\mathbf{R} \subset \mathbf{V}_{\mathcal{M}}$ such that the remainder set in non-empty: $\mathbf{Q} \neq \emptyset$ all $Z \in \mathbf{Q}$ are not considered in the preservation of (in)equalities of Lem.11, and therefore can essentially be considered exogenous variables. This implies that if there exists two clusters $\mathbf{C}_i, \mathbf{C}_j \in \mathbf{C}_{\mathcal{M}'}$ some $Z \in \mathbf{Q}$ and $\mathbf{V}_i \in \mathbf{C}_i$, $\mathbf{V}_j \in \mathbf{C}_j$, such that $\mathbf{V}_i \leftsquigarrow Z \rightsquigarrow \mathbf{V}_j$ then there must exists a confounding arrow $\mathbf{C}_i \leftarrow\!\dashrightarrow \mathbf{C}_j$, in accordance to rule 2B of Def.16.

Finally, as all other distributional (in)equalities are preserved over the abstraction, and all confounding edges can be identified by these (in)equalities there cannot be any other confounding edges in $\mathcal{M}'$. Therefore, the confounding edges of a bijective $\mathcal{L}_2$-consistent abstractions are exactly described by the constructive rules of the PCDAG.

■

B.12. Proof of theorem 24

Theorem 24 (PCDAGs describe all bijective $\mathcal{L}_2$-consistent α-abstractions) *Given a bijective $\mathcal{L}_2$-consistent α-abstraction $\boldsymbol{\alpha} : \mathcal{M} \to \mathcal{M}'$, $\mathcal{M}'$ is a permissible Partial Cluster DAG of $\mathcal{M}$.*

Proof This proof follows immediately from the results of Lem.21 and Lem.22. Together the lemmas state that both all directed and all confounding edges of a bijective $\mathcal{L}_2$-consistent abstraction are necessarily equal to those given by the constructive rules of the PCDAG. ■

B.13. Proof of theorem 23

Theorem 24 (PCDAG $\Rightarrow$ α-Abstraction) *Given a PCDAG there exists an equivalent $\mathcal{L}_2$-consistent α-abstraction $\boldsymbol{\alpha} : \mathcal{M} \to \mathcal{M}'$ with $\mathbf{R} \subseteq \mathbf{V}$, φ given by the clustering, and all maps $\alpha_{\mathbf{C}}$ bijective.*

Proof The proof follows immediately from the result of Thm.20, showing there exists an $\mathcal{L}_2$-consistent α-abstraction for each PCDAG by taking bijective range maps $\alpha_{\mathbf{C}}$ and the set of mechanisms $\mathcal{F}_{\mathcal{M}'}$ to be the composite mechanisms as implied by φ. ■

B.14. Proof of Corollary 25

Corollary 25 (Equivalence α-abstraction and Constructive τ-abstraction) *The α-abstraction is equivalent to the constructive τ-abstraction, if for all settings $\mathbf{v} \in \mathcal{R}(\mathbf{V})$ there exists a $\mathbf{u} \in \mathcal{R}(\mathbf{U})$ giving rise to $\mathbf{v}$.*

Proof In this proof we will first outline the similarities of the two abstractions, by unpacking the definitions. After that we will show that the frameworks do not as neatly align on mapping the exogenous variables. In short, the problem arises from the fact that the α-abstraction is not defined over SCMs but Finite Graphical Models.

Similarities α-abstraction and constructive τ-abstraction. This part will be rather straightforward, as we will see that all maps that make up the abstractions are equivalent by definition.

Let $\boldsymbol{\alpha} : \mathcal{M} \to \mathcal{M}'$ be an α-abstraction following definition 6.

1. First note that the map $\varphi : \mathbf{R} \to \mathbf{V}_{\mathcal{M}'}$ of $\boldsymbol{\alpha}$ is surjective. It follows immediately that φ induces a partition $\mathbf{P}$ over $\mathbf{R}$. Consequently, $\mathbf{P}$ together with the set of irrelevant variables forms a partition over $\mathbf{V}_{\mathcal{M}}$, as required by the constructive τ-abstraction.

2. Secondly, given an abstract variable $X' \in \mathbf{V}_{\mathcal{M}'}$ and its pre-image $\mathbf{X} = \varphi^{-1}(X')$ the range mapping $\mathcal{R}(\mathbf{X}) \to \mathcal{R}(X')$ is surjective by definition. This is immediately equivalent to the subjectivity requirement imposed on range mappings in the constructive τ-abstraction.

3. thirdly, the constructive τ abstraction requires that the set of allowed interventions at the high level is equal to the set of all high level interventions. In other words, there must exist a surjective map from the set of low level interventions to the set of high level interventions. Given an α-abstraction $\boldsymbol{\alpha} : \mathcal{M} \to \mathcal{M}'$ the intervention map is implicitly given by $P(\varphi^{-1}(Y) \mid do(\varphi^{-1}(X))) \to P(Y \mid do(X))$, for all $X, Y \in \mathbf{V}_{\mathcal{M}'}$. It follows from surjectivity of φ and the range maps $\mathcal{R}(\varphi^{-1}(X)) \to \mathcal{R}(X)$ that the intervention map is also surjective.

Finite Graphical Model as a minimal SCM. Finally, a constructive τ-abstraction is by definition a τ-abstraction, which requires that there exists a surjective map from range of the exogenous variables of the low level model to that of the high level model. Here is the problem, as the exogenous variables are not explicitly defined for the Finite Graphical Model. However, one can construct a minimal range for the exogenous variables required to generate the distributions defined by the finite graphical model:

Let $\mathcal{M} : \langle \mathbf{V}, \mathbf{U}, \mathcal{F}, P(\mathbf{U}) \rangle$ be an SCM. Given a setting of the exogenous variables $\mathbf{u}$ there exists exactly one compatible setting of the endogenous variables $\mathbf{v}$, by the fact that all $f \in \mathcal{F}$ are deterministic. This means there is a surjective function from the range of $\mathbf{U}$ to the range of $\mathbf{V}$. Consequently, we can define a partition of equivalence classes on the range of $\mathbf{U}$, such that for each equivalence class all settings of $\mathbf{U}$ in that class will map to the same setting of $\mathbf{V}$. Therefore, the minimal range of $\mathbf{U}$ is such that there exists exactly one value for each equivalence class. It follows immediately that there exists a bijective map from the minimal range of $\mathbf{U}$ to the range of $\mathbf{V}$.

let $\boldsymbol{\alpha} : \mathcal{M} \to \mathcal{M}'$ be an alpha abstraction, by definition we have $\varphi : \mathcal{R}(\mathbf{V}_{\mathcal{M}}) \to \mathcal{R}(\mathbf{V}_{\mathcal{M}'})$. Let $\mathbf{U}^*_{\mathcal{M}}$ and $\mathbf{U}^*_{\mathcal{M}'}$ be minimal exogenous variables for $\mathcal{M}$ and $\mathcal{M}'$ respectively. There is a bijection between $\mathbf{V}_{\mathcal{M}}$ and $\mathbf{U}^*_{\mathcal{M}}$ and φ surjective, therefore there exists a $\varphi_U : \mathbf{U}^*_{\mathcal{M}} \to \mathbf{U}^*_{\mathcal{M}'}$ that is surjective.

The τ-abstraction requires a surjective map $\tau_U : U_{\mathcal{M}} \to U_{\mathcal{M}'}$, which is not necessary for the α-abstraction. However, we have shown that the α-abstraction does guarantee a surjective map $\mathbf{U}^*_{\mathcal{M}} \to \mathbf{U}^*_{\mathcal{M}'}$. Diagrammatically this can be described as:

$$
\begin{tikzcd}
\mathbf{U}^*_{\mathcal{M}} \arrow[rr, "{[sur]}", "\varphi_U"'] \arrow[ddr, bend right, "{[bij]}"'] & & \mathbf{U}^*_{\mathcal{M}'} \arrow[ddl, bend left, "{[bij]}"] \\
& \mathbf{U}_{\mathcal{M}} \arrow[ul, "{[sur]}"] \arrow[r, "{[sur]}", "\tau_U"'] \arrow[d, "\mathcal{F}_{\mathcal{M}}"', "{[sur]}"] \arrow[dl, bend right, "\mathcal{F}_{\mathcal{M}}"'] & \mathbf{U}_{\mathcal{M}'} \arrow[ur, "{[sur]}"'] \arrow[d, "{[sur]}"', "\mathcal{F}_{\mathcal{M}'}"] \arrow[dr, bend left, "\mathcal{F}_{\mathcal{M}'}"] \\
& \mathbf{V}_{\mathcal{M}} \arrow[r, "{[sur]}", "\varphi,\ \tau"'] & \mathbf{V}_{\mathcal{M}'}
\end{tikzcd}
$$

The inner square represents the constructive τ-abstraction, and the outer square represents the α-abstraction.

Conclusion. We have illustrated the equivalence of the α-abstraction and the constructive τ-abstraction. With the one caveat being that the constructive τ-abstraction is more restrictive on the mapping of exogenous variables. ■