Causal Abstraction: Definition, Measures and Learning

Fabio Massimo Zennaro

University of Bergen

March 31st, 2025

1. Introduction

1.1. Levels of Abstraction

Levels of Abstraction

Systems may be represented at different **levels of abstraction** (LoA) [6].

Levels of Abstraction

Systems may be represented at different **levels of abstraction** (LoA) [6].

Thermodynamics example:

Low-level / Base model:
Microscopic description $\mathbf{x}, \dot{\mathbf{x}}$.

High-level / Abstracted model:
Macroscopic description P, T, V.

Levels of Abstraction

Systems may be represented at different **levels of abstraction** (LoA) [6].

Thermodynamics example:

Low-level / Base model:
Microscopic description $\mathbf{x}, \dot{\mathbf{x}}$.

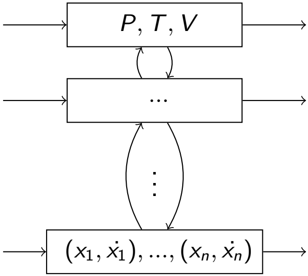
High-level / Abstracted model:
Macroscopic description P, T, V.

LoA may be inaccessible, so we may want to *shift* among LoAs.

1. We need a *mapping* between LoAs.
2. We want the mapping to be *consistent*.

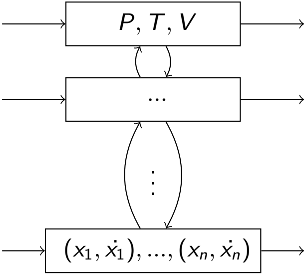
Abstraction

Abstraction (aka, *multi-level modelling* or *multi-resolution modelling*) aims at relating these levels.



Abstraction

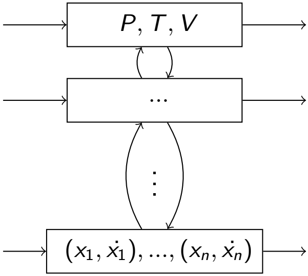
Abstraction (aka, *multi-level modelling* or *multi-resolution modelling*) aims at relating these levels.



- It combines models from *different sources*.

Abstraction

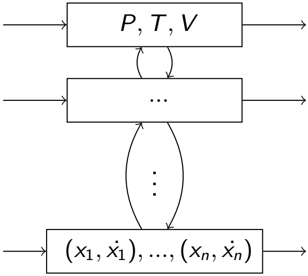
Abstraction (aka, *multi-level modelling* or *multi-resolution modelling*) aims at relating these levels.



- It combines models from *different sources*.
- It aggregates information from *different resolutions*.

Abstraction

Abstraction (aka, *multi-level modelling* or *multi-resolution modelling*) aims at relating these levels.

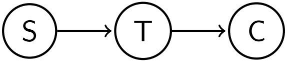


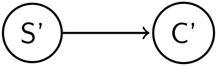
- It combines models from *different sources*.
- It aggregates information from *different resolutions*.
- It allows for *computation with minimal effort*.

Causal Abstraction

We focus on **abstraction** between **causal models**.

Lung cancer scenario example:





Causal Abstraction

We focus on **abstraction** between **causal models**.

Lung cancer scenario example:

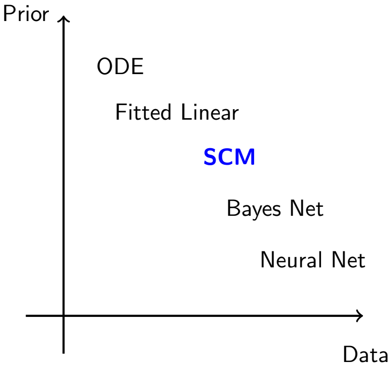
$$S \longrightarrow T \longrightarrow C \qquad\qquad S' \longrightarrow C'$$

- How do we *represent* causal systems?
- How do we *express relations* of abstraction among causal models?
- How do we *measure correctness* of causal abstraction?
- How do we *learn* LoAs?
- How do we *take advantage of* LoAs?

2. Structural Causal Models

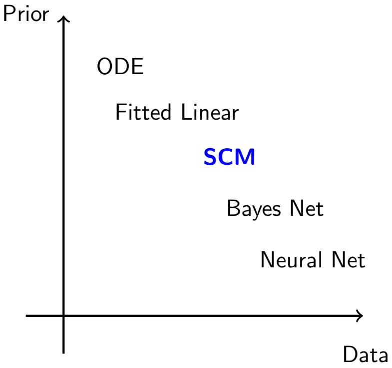
Structural Causal Modeling

Structural causal models rely on a strong prior given by *causality* [14, 15].



Structural Causal Modeling

Structural causal models rely on a strong prior given by *causality* [14, 15].

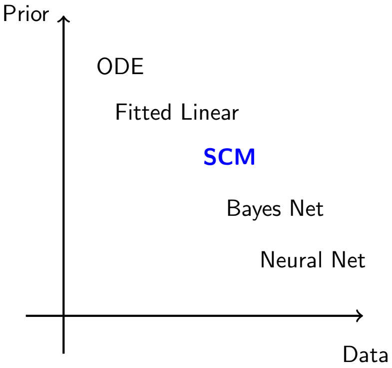


- It discriminates *correlations* and *causes*.

SCMs integrates a *graphical model* and *probabilities distributions*.

Structural Causal Modeling

Structural causal models rely on a strong prior given by *causality* [14, 15].

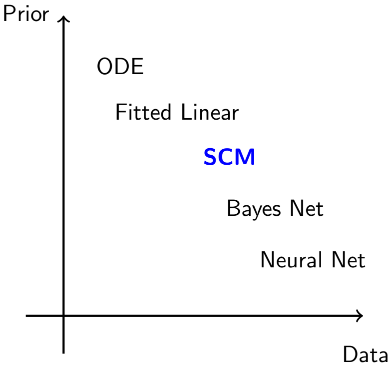


- It discriminates *correlations* and *causes*.
- It allows for reasoning about *interventions*.

SCMs integrates a *graphical model* and *probabilities distributions*.

Structural Causal Modeling

Structural causal models rely on a strong prior given by *causality* [14, 15].

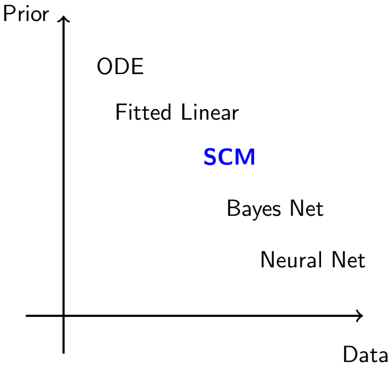


- It discriminates *correlations* and *causes*.
- It allows for reasoning about *interventions*.
- It allows for reasoning about *counterfactuals*.

SCMs integrates a *graphical model* and *probabilities distributions*.

Structural Causal Modeling

Structural causal models rely on a strong prior given by *causality* [14, 15].

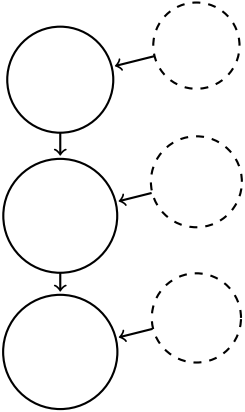


- It discriminates *correlations* and *causes*.
- It allows for reasoning about *interventions*.
- It allows for reasoning about *counterfactuals*.
- It implies a *causality ladder* of reasoning.

SCMs integrates a *graphical model* and *probabilities distributions*.

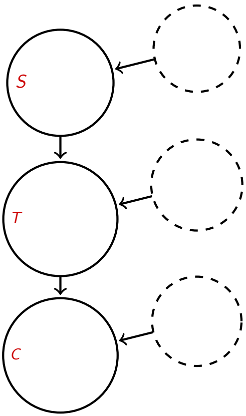
SCMs - Definition

We express a **SCM** as $\mathcal{M} = \langle \mathcal{X}, \mathcal{U}, \mathcal{F}, \mathcal{P} \rangle$ [14, 15]:



SCMs - Definition

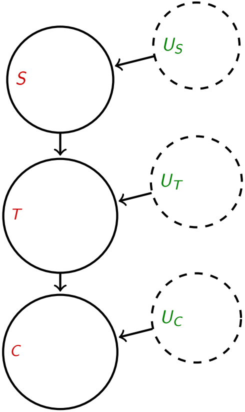
We express a **SCM** as $\mathcal{M} = \langle \mathcal{X}, \mathcal{U}, \mathcal{F}, \mathcal{P} \rangle$ [14, 15]:



- $\mathcal{X}$: set of *endogenous nodes* (S, T, C) representing variables of interest

SCMs - Definition

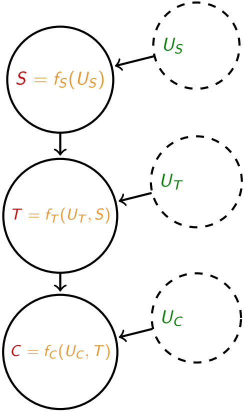
We express a **SCM** as $\mathcal{M} = \langle \mathcal{X}, \mathcal{U}, \mathcal{F}, \mathcal{P} \rangle$ [14, 15]:



- $\mathcal{X}$: set of *endogenous nodes* (S, T, C) representing variables of interest
- $\mathcal{U}$: Set of *exogenous nodes* (U_S, U_T, U_C) representing stochastic factors

SCMs - Definition

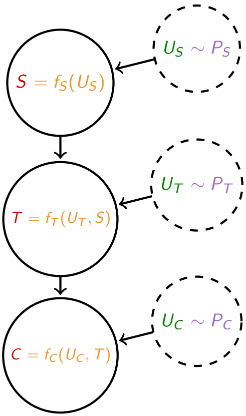
We express a **SCM** as $\mathcal{M} = \langle \mathcal{X}, \mathcal{U}, \mathcal{F}, \mathcal{P} \rangle$ [14, 15]:



- $\mathcal{X}$: set of *endogenous nodes* (S, T, C) representing variables of interest
- $\mathcal{U}$: Set of *exogenous nodes* (U_S, U_T, U_C) representing stochastic factors
- $\mathcal{F}$: Set of *structural functions* (f_S, f_T, f_C) describing the dynamics of each variable

SCMs - Definition

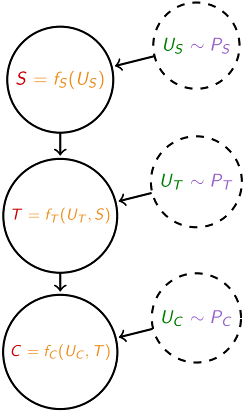
We express a **SCM** as $\mathcal{M} = \langle \mathcal{X}, \mathcal{U}, \mathcal{F}, \mathcal{P} \rangle$ [14, 15]:



- $\mathcal{X}$: set of *endogenous nodes* (S, T, C) representing variables of interest
- $\mathcal{U}$: Set of *exogenous nodes* (U_S, U_T, U_C) representing stochastic factors
- $\mathcal{F}$: Set of *structural functions* (f_S, f_T, f_C) describing the dynamics of each variable
- $\mathcal{P}$: Set of *distributions* (P_S, P_T, P_C) describing the random factors

SCMs - Definition

We express a **SCM** as $\mathcal{M} = \langle \mathcal{X}, \mathcal{U}, \mathcal{F}, \mathcal{P} \rangle$ [14, 15]:

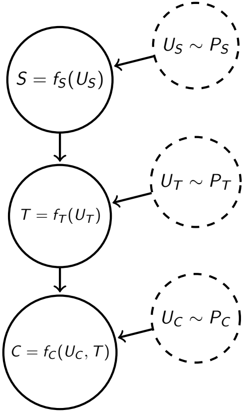


- $\mathcal{X}$: set of *endogenous nodes* (S, T, C) representing variables of interest
- $\mathcal{U}$: Set of *exogenous nodes* (U_S, U_T, U_C) representing stochastic factors
- $\mathcal{F}$: Set of *structural functions* (f_S, f_T, f_C) describing the dynamics of each variable
- $\mathcal{P}$: Set of *distributions* (P_S, P_T, P_C) describing the random factors

Every SCM $\mathcal{M}$ implies a (joint) **distribution** $P_{\mathcal{M}}$: $P_{\mathcal{M}}(S, T, C)$

SCMs - Interventions

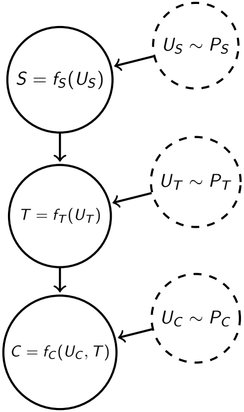
We can perform **interventions** on a causal model [14, 15]:



1.
2.

SCMs - Interventions

We can perform **interventions** on a causal model [14, 15]:

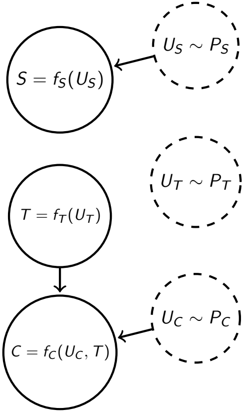


$do(T = 1)$

1.
2.

SCMs - Interventions

We can perform **interventions** on a causal model [14, 15]:

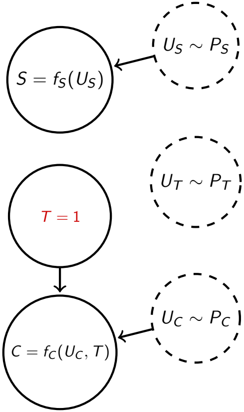


$do(T = 1)$

1. Remove incoming edges in the intervened node
2.

SCMs - Interventions

We can perform **interventions** on a causal model [14, 15]:



$do(T = 1)$

1. Remove incoming edges in the intervened node
2. Set the value of the intervened node

SCMs - Distributions

An *intervention* ι defines a new **intervened model** $\mathcal{M}_\iota$ with new distributions.

SCMs - Distributions

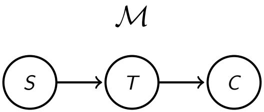
An *intervention* ι defines a new **intervened model** $\mathcal{M}_\iota$ with new distributions.

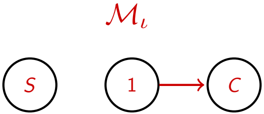
$$\mathcal{M}$$

$$S \longrightarrow T \longrightarrow C$$

SCMs - Distributions

An *intervention* ι defines a new **intervened model** $\mathcal{M}_\iota$ with new distributions.

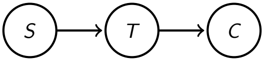




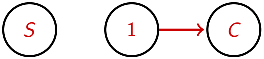
SCMs - Distributions

An *intervention* ι defines a new **intervened model** $\mathcal{M}_\iota$ with new distributions.

$\mathcal{M}$



$\mathcal{M}_\iota$

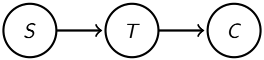


$P_{\mathcal{M}}$

SCMs - Distributions

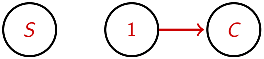
An *intervention* ι defines a new **intervened model** $\mathcal{M}_\iota$ with new distributions.

$\mathcal{M}$



$P_{\mathcal{M}}$

$\mathcal{M}_\iota$

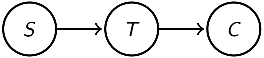


$P_{\mathcal{M}_\iota}$

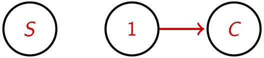
SCMs - Distributions

An *intervention* ι defines a new **intervened model** $\mathcal{M}_\iota$ with new distributions.

$\mathcal{M}$



$\mathcal{M}_\iota$



$P_{\mathcal{M}}$

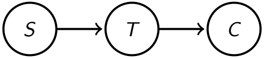
$P_{\mathcal{M}_\iota}$

$P_{\mathcal{M}}(C|S)$

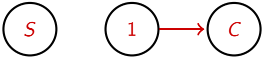
SCMs - Distributions

An *intervention* ι defines a new **intervened model** $\mathcal{M}_\iota$ with new distributions.

$\mathcal{M}$



$\mathcal{M}_\iota$



$P_\mathcal{M}$

$P_{\mathcal{M}_\iota}$

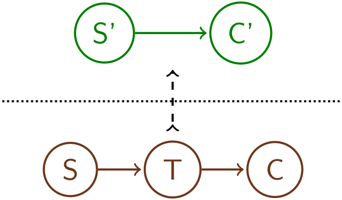
$P_\mathcal{M}(C|S)$

$P_\mathcal{M}(C|S, do(T=1)) = P_{\mathcal{M}_\iota}(C|S)$

3. Causal Abstraction

Three approaches

Lung cancer scenario example:

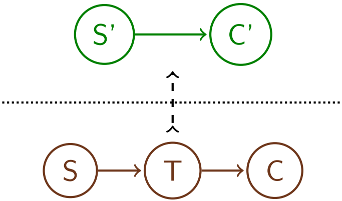


$$\mathtt{Dom}[S'] = \mathtt{Dom}[C'] = \{0, 1\}$$

$$\mathtt{Dom}[S] = \mathtt{Dom}[T] = \mathtt{Dom}[C] = \{0, 1\}$$

Three approaches

Lung cancer scenario example:



$$\mathtt{Dom}[S'] = \mathtt{Dom}[C'] = \{0, 1\}$$

$$\mathtt{Dom}[S] = \mathtt{Dom}[T] = \mathtt{Dom}[C] = \{0, 1\}$$

- The τ-**abstraction** approach [18, 1]
- The Φ-**abstraction** approach [12, 13]
- The α-**abstraction** approach [17, 16]

3.1. τ-abstraction approach

The τ-*abstraction* approach: mapping [18]

Let $\mathcal{M}$ and $\mathcal{M}'$ be two finite SCMs. An **abstraction** is a tuple

$$\langle \tau, \omega \rangle$$

where:

The τ-*abstraction* approach: mapping [18]

Let $\mathcal{M}$ and $\mathcal{M}'$ be two finite SCMs. An **abstraction** is a tuple

$$\langle \tau, \omega \rangle$$

where:

- $\tau : \mathtt{Dom}[\mathcal{X}] \to \mathtt{Dom}[\mathcal{X}']$ maps complete outputs of the low-level model to complete output of the high level model.

The τ-*abstraction* approach: mapping [18]

Let $\mathcal{M}$ and $\mathcal{M}'$ be two finite SCMs. An **abstraction** is a tuple

$$\langle \tau, \omega \rangle$$

where:

- $\tau : \mathtt{Dom}[\mathcal{X}] \to \mathtt{Dom}[\mathcal{X}']$ maps complete outputs of the low-level model to complete output of the high level model.
- $\omega : \mathcal{I} \to \mathcal{I}'$ maps low-level interventions to high-level interventions.

The τ-*abstraction* approach: mapping [18]

Given two SCMs $\mathcal{M}$ and $\mathcal{M}'$, the **transformation** τ induces a *pushforward* between distributions:

$$\tau_{\#} : P_{\mathcal{M}} \mapsto P_{\mathcal{M}'}$$

The τ-*abstraction* approach: mapping [18]

Given two SCMs $\mathcal{M}$ and $\mathcal{M}'$, the **transformation** τ induces a *pushforward* between distributions:

$$\tau_{\#} : P_{\mathcal{M}} \mapsto P_{\mathcal{M}'}$$

Under an assumption of **observational consistency**, this implies:

$$\tau_{\#}(P_{\mathcal{M}}) = P_{\mathcal{M}'}$$

The τ-*abstraction* approach: interventional consistency [18]

We want more than *observational consistency*. We want **interventional consistency**.

The τ-*abstraction* approach: interventional consistency [18]

We want more than *observational consistency*. We want **interventional consistency**.

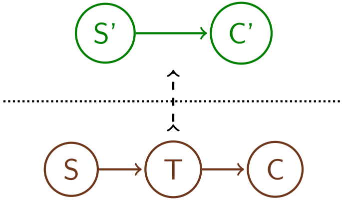
A transformation is an *exact transformation* if there exists a surjective order-preserving ω such that:

$$
\begin{tikzcd}
P_{\mathcal{M}} \arrow[r, "\tau"] \arrow[dd, "\iota"'] & \tau(P_{\mathcal{M}}) = P_{\mathcal{M}'} \arrow[d, "\omega(\iota)"] \\
 & P_{\mathcal{M}_{\omega(\iota)}} \\
P_{\mathcal{M}_\iota} \arrow[r, "\tau"'] & \tau(P_{\mathcal{M}_\iota})
\end{tikzcd}
$$

where $\tau(P_{\mathcal{M}_\iota}) = P_{\mathcal{M}_{\omega(\iota)}}, \forall \iota \in \mathcal{I}$.

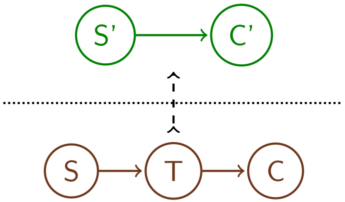
The τ-*abstraction* approach: example

Lung cancer scenario example:



The τ-*abstraction* approach: example

Lung cancer scenario example:

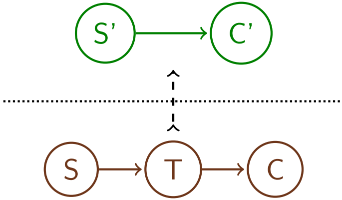


$$\tau : \mathtt{Dom}[S] \times \mathtt{Dom}[T] \times \mathtt{Dom}[C] \rightarrow \mathtt{Dom}[S'] \times \mathtt{Dom}[C']$$

$$\tau : (s, t, c) \mapsto (s, c)$$

The τ-*abstraction* approach: example

Lung cancer scenario example:



$\tau : \mathtt{Dom}[S] \times \mathtt{Dom}[T] \times \mathtt{Dom}[C] \rightarrow \mathtt{Dom}[S'] \times \mathtt{Dom}[C']$

$\tau : (s, t, c) \mapsto (s, c)$

Set of interventions: $\mathcal{I} = \{\emptyset, do(S = 0)\}$

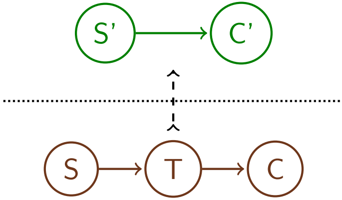
$$\omega : \begin{cases} \emptyset \mapsto \emptyset \\ do(S = 0) \mapsto do(S' = 0) \end{cases}$$

The τ-*abstraction* approach: example

Lung cancer scenario example:

$\tau : \texttt{Dom}[S] \times \texttt{Dom}[T] \times \texttt{Dom}[C] \to \texttt{Dom}[S'] \times \texttt{Dom}[C']$

$\tau : (s, t, c) \mapsto (s, c)$



Set of interventions: $\mathcal{I} = \{\emptyset, do(S = 0)\}$

$$\omega : \begin{cases} \emptyset \mapsto \emptyset \\ do(S = 0) \mapsto do(S' = 0) \end{cases}$$

Consistency condition:

$$\begin{tikzcd} P_{\mathcal{M}}(S, T, C) \arrow[r, "\tau"] \arrow[d, "\iota"'] & P_{\mathcal{M}'}(S', C') \arrow[d, "\omega(\iota)"] \\ P_{\mathcal{M}}(T, C | do(S = 0)) \arrow[r, "\tau"'] & P_{\mathcal{M}'}(C' | do(S' = 0)) \end{tikzcd}$$

3.2. Φ-abstraction approach

The Φ-*abstraction* approach: mapping [12]

An SCM $\mathcal{M}$ can be formalized as a *functor* from a syntactic category to the category of sets and Markov kernels:

$$F_{\mathcal{M}} : \mathtt{Syn}_{\mathcal{M}} \to \mathtt{FinStoch}$$

The Φ-*abstraction* approach: mapping [12]

An SCM $\mathcal{M}$ can be formalized as a *functor* from a syntactic category to the category of sets and Markov kernels:

$$F_{\mathcal{M}} : \mathtt{Syn}_{\mathcal{M}} \to \mathtt{FinStoch}$$

In this formalization, an intervention is an *endofunctor* on the syntactic category:

$$\mathrm{cut}_X : \mathtt{Syn}_{\mathcal{M}} \to \mathtt{Syn}_{\mathcal{M}}$$

The Φ-*abstraction* approach: consistency [12]

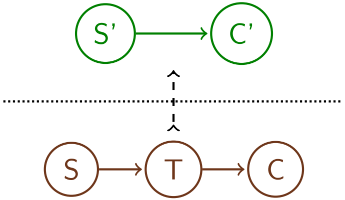
Given two SCMs $\mathcal{M}$ and $\mathcal{M}'$ with a homomorphism ϕ between their DAGs, an abstraction exists if we have a *natural transformation* between the respective functors:

$$\begin{tikzcd} \mathtt{Syn}_{\mathcal{M}} \arrow[r, "F_{\mathcal{M}}"] \arrow[d, "\Phi"'] & \mathtt{FinStoch} \arrow[d, "\mathrm{id}"] \\ \mathtt{Syn}_{\mathcal{M}'} \arrow[r, "F_{\mathcal{M}'}"'] \arrow[ur, phantom, "\Downarrow"] & \mathtt{FinStoch} \end{tikzcd}$$

Given a Φ-abstraction, the homomorphism ϕ guarantees *interventional consistency*.

The Φ-*abstraction* approach: example

Lung cancer scenario example:

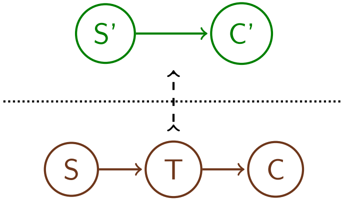


The Φ-*abstraction* approach: example

Lung cancer scenario example:

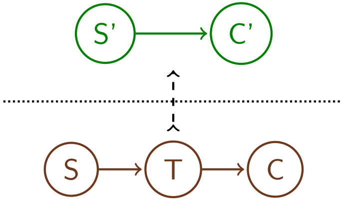
$$\mathtt{Syn}_{\mathcal{M}} : \bullet_S \longrightarrow \bullet_T \longrightarrow \bullet_C$$

$$\mathtt{Syn}_{\mathcal{M}'} : \bullet_{S'} \longrightarrow \bullet_{C'}$$



The Φ-*abstraction* approach: example

Lung cancer scenario example:



$\mathtt{Syn}_{\mathcal{M}} : \bullet_S \longrightarrow \bullet_T \longrightarrow \bullet_C$

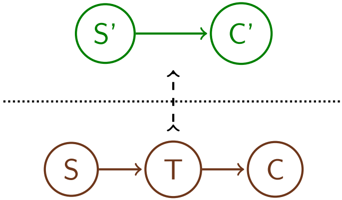
$\mathtt{Syn}_{\mathcal{M}'} : \bullet_{S'} \longrightarrow \bullet_{C'}$

$$F_{\mathcal{M}} : \begin{cases} \bullet \mapsto \{0,1\} \\ \longrightarrow \mapsto \begin{bmatrix} \cdot & \cdot \\ \cdot & \cdot \end{bmatrix} \end{cases}$$

$$F_{\mathcal{M}'} : \begin{cases} \bullet \mapsto \{0,1\} \\ \longrightarrow \mapsto \begin{bmatrix} \cdot & \cdot \\ \cdot & \cdot \end{bmatrix} \end{cases}$$

The Φ-*abstraction* approach: example

Lung cancer scenario example:



$\mathtt{Syn}_{\mathcal{M}} : \bullet_S \longrightarrow \bullet_T \longrightarrow \bullet_C$

$\mathtt{Syn}_{\mathcal{M}'} : \bullet_{S'} \longrightarrow \bullet_{C'}$

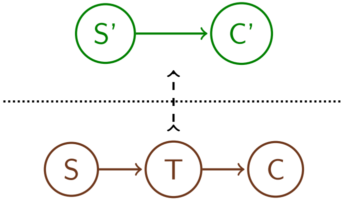
$$F_{\mathcal{M}} : \begin{cases} \bullet \mapsto \{0,1\} \\ \longrightarrow \mapsto \begin{bmatrix} \cdot & \cdot \\ \cdot & \cdot \end{bmatrix} \end{cases}$$

$$F_{\mathcal{M}'} : \begin{cases} \bullet \mapsto \{0,1\} \\ \longrightarrow \mapsto \begin{bmatrix} \cdot & \cdot \\ \cdot & \cdot \end{bmatrix} \end{cases}$$

$$\Phi : \bullet_S \mapsto \bullet_{S'}, \bullet_T \mapsto \bullet_{S'}, \bullet_C \mapsto \bullet_{C'}$$

The Φ-*abstraction* approach: example

Lung cancer scenario example:



$$\mathtt{Syn}_{\mathcal{M}} : \bullet_S \longrightarrow \bullet_T \longrightarrow \bullet_C$$

$$\mathtt{Syn}_{\mathcal{M}'} : \bullet_{S'} \longrightarrow \bullet_{C'}$$

$$F_{\mathcal{M}} : \begin{cases} \bullet \mapsto \{0,1\} \\ \longrightarrow \mapsto \begin{bmatrix} \cdot & \cdot \\ \cdot & \cdot \end{bmatrix} \end{cases}$$

$$F_{\mathcal{M}'} : \begin{cases} \bullet \mapsto \{0,1\} \\ \longrightarrow \mapsto \begin{bmatrix} \cdot & \cdot \\ \cdot & \cdot \end{bmatrix} \end{cases}$$

$$\Phi : \bullet_S \mapsto \bullet_{S'}, \bullet_T \mapsto \bullet_{S'}, \bullet_C \mapsto \bullet_{C'}$$

A natural transformation is a *collection of maps* in `FinStoch`.

3.3. α-abstraction approach

The α-*abstraction* approach: mapping [17]

Let $\mathcal{M}$ and $\mathcal{M}'$ be two finite SCMs with finite domains. An **abstraction** is a tuple

$$\langle R, a, \alpha \rangle$$

where:

The *α-abstraction* approach: mapping [17]

Let $\mathcal{M}$ and $\mathcal{M}'$ be two finite SCMs with finite domains. An **abstraction** is a tuple

$$\langle R, a, \alpha \rangle$$

where:

- $R \subseteq \mathcal{X}_{\mathcal{M}}$ is a subset of *relevant nodes* among the endogenous nodes of $\mathcal{M}$.

The α-*abstraction* approach: mapping [17]

Let $\mathcal{M}$ and $\mathcal{M}'$ be two finite SCMs with finite domains. An **abstraction** is a tuple

$$\langle R, a, \alpha \rangle$$

where:

- $R \subseteq \mathcal{X}_{\mathcal{M}}$ is a subset of *relevant nodes* among the endogenous nodes of $\mathcal{M}$.
- $a : R \rightarrow \mathcal{X}_{\mathcal{M}'}$ is a *surjective function* mapping a low-level node in $\mathcal{M}$ to a high-level node in $\mathcal{M}'$.

The α-*abstraction* approach: mapping [17]

Let $\mathcal{M}$ and $\mathcal{M}'$ be two finite SCMs with finite domains. An **abstraction** is a tuple

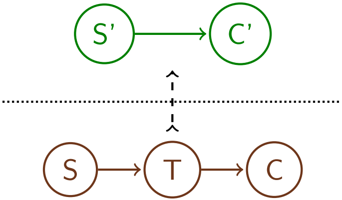
$$\langle R, a, \alpha \rangle$$

where:

- $R \subseteq \mathcal{X}_{\mathcal{M}}$ is a subset of *relevant nodes* among the endogenous nodes of $\mathcal{M}$.
- $a : R \to \mathcal{X}_{\mathcal{M}'}$ is a *surjective function* mapping a low-level node in $\mathcal{M}$ to a high-level node in $\mathcal{M}'$.
- α is a *collection of surjective functions*, one for each high-level node X', defined as $\alpha_{X'} : \mathtt{Dom}[a^{-1}(X')] \to \mathtt{Dom}[X']$.
  α'_X maps an output of the low-level nodes sent onto X' by a onto an output of X'.

The α-*abstraction* approach: example (I)

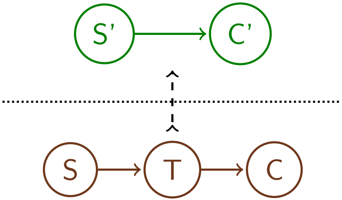
Lung cancer scenario example:



The α-*abstraction* approach: example (I)

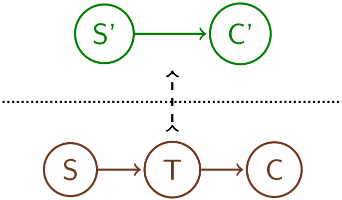
Lung cancer scenario example:

$$R = \{S, C\} \subseteq \mathcal{X}_{\mathcal{M}}$$



The α-*abstraction* approach: example (I)

Lung cancer scenario example:



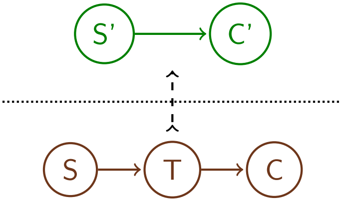
$$R = \{S, C\} \subseteq \mathcal{X}_{\mathcal{M}}$$

$$a : R \rightarrow \mathcal{X}_{\mathcal{M}'}$$

$$a : \begin{cases} S \mapsto S' \\ C \mapsto C' \end{cases}$$

The α-*abstraction* approach: example (I)

Lung cancer scenario example:



$$R = \{S, C\} \subseteq \mathcal{X}_{\mathcal{M}}$$

$$a : R \to \mathcal{X}_{\mathcal{M}'}$$

$$a : \begin{cases} S \mapsto S' \\ C \mapsto C' \end{cases}$$

$$\alpha : \begin{cases} \alpha_{S'} : \{0,1\} \to \{0,1\} \\ \alpha_{S'} : s \mapsto s \\ \alpha_{C'} : \{0,1\} \to \{0,1\} \\ \alpha_{C'} : c \mapsto c \end{cases}$$

The α-*abstraction* approach: abstraction error

We want an abstraction to guarantee *interventional consistency*.

The α-*abstraction* approach: abstraction error

We want an abstraction to guarantee *interventional consistency*.

.

$$S' \xrightarrow[P_{\mathcal{M}'_{\iota'}}(C'|do(S'))]{\nu} C'$$

The α-*abstraction* approach: abstraction error

We want an abstraction to guarantee *interventional consistency*.

$$\begin{tikzcd}
S \arrow[d, "\alpha_{S'}"'] & \\
S' \arrow[r, "\nu", "P_{\mathcal{M}'_{\iota'}}(C'|do(S'))"'] & C'
\end{tikzcd}$$

.

The α-*abstraction* approach: abstraction error

We want an abstraction to guarantee *interventional consistency*.

$$
\begin{tikzcd}
S \arrow[r, "\mu", "P_{\mathcal{M}_{\iota}}(C|do(S))"'] \arrow[d, "\alpha_{S'}"'] & C \\
S' \arrow[r, "\nu", "P_{\mathcal{M'}_{\iota'}}(C'|do(S'))"'] & C'
\end{tikzcd}
$$

.

The α-*abstraction* approach: abstraction error

We want an abstraction to guarantee *interventional consistency*.

$$\begin{tikzcd} S \arrow[r, "\mu", "P_{\mathcal{M}_\iota}(C|do(S))"'] \arrow[d, "\alpha_{S'}"'] & C \arrow[d, "\alpha_{C'}"] \\ S' \arrow[r, "\nu", "P_{\mathcal{M}'_{\iota'}}(C'|do(S'))"'] & C' \end{tikzcd}$$

.

The α-*abstraction* approach: abstraction error

We want an abstraction to guarantee *interventional consistency*.

$$\begin{tikzcd} S \arrow[r, "\mu", "P_{\mathcal{M}_\iota}(C|do(S))"'] \arrow[d, "\alpha_{S'}"'] & C \arrow[d, "\alpha_{C'}"] \\ S' \arrow[r, "\nu", "P_{\mathcal{M}'_{\iota'}}(C'|do(S'))"'] & C' \end{tikzcd}$$

- Ideally, mechanisms and abstractions *commute*.

The α-*abstraction* approach: abstraction error

We want an abstraction to guarantee *interventional consistency*.

$$\begin{tikzcd}
S \arrow[r, "\mu", "P_{\mathcal{M}_\iota}(C|do(S))"'] \arrow[d, "\alpha_{S'}"'] & C \arrow[d, "\alpha_{C'}"] \\
S' \arrow[r, "\nu", "P_{\mathcal{M'}_{\iota'}}(C'|do(S'))"'] & C'
\end{tikzcd}$$

- Ideally, mechanisms and abstractions *commute*.
- Otherwise, we compute an abstraction error as the *worst-case discrepancy* over all possible interventions:

$$E_{\alpha}(S', C') = \max_{\iota} D(\alpha_{C'} \cdot \mu, \nu \cdot \alpha_{S'})$$

The α-*abstraction* approach: abstraction error

In general, an abstraction may imply multiple *causal mechanism diagrams*:

The α-*abstraction* approach: abstraction error

In general, an abstraction may imply multiple *causal mechanism diagrams*:

$$\begin{tikzcd} S \arrow[r, "\mu"] \arrow[d, "\alpha_{S'}"'] & T \arrow[d, "\alpha_{T'}"] \\ S' \arrow[r, "\nu"] & T' \end{tikzcd}$$

The α-*abstraction* approach: abstraction error

In general, an abstraction may imply multiple *causal mechanism diagrams*:

$$\begin{tikzcd} S \arrow[r, "\mu"] \arrow[d, "\alpha_{S'}"'] & T \arrow[d, "\alpha_{T'}"] \\ S' \arrow[r, "\nu"] & T' \end{tikzcd} \qquad \begin{tikzcd} T \arrow[r, "\mu'"] \arrow[d, "\alpha_{T'}"'] & C \arrow[d, "\alpha_{C'}"] \\ T' \arrow[r, "\nu'"] & C' \end{tikzcd}$$

The α-*abstraction* approach: abstraction error

In general, an abstraction may imply multiple *causal mechanism diagrams*:

$$\begin{tikzcd} S \arrow[r, "\mu"] \arrow[d, "\alpha_{S'}"'] & T \arrow[d, "\alpha_{T'}"] \\ S' \arrow[r, "\nu"] & T' \end{tikzcd} \qquad \begin{tikzcd} T \arrow[r, "\mu'"] \arrow[d, "\alpha_{T'}"'] & C \arrow[d, "\alpha_{C'}"] \\ T' \arrow[r, "\nu'"] & C' \end{tikzcd}$$

$$\begin{tikzcd} S \arrow[r, "\mu' \circ \mu"] \arrow[d, "\alpha_{S'}"'] & C \arrow[d, "\alpha_{C'}"] \\ S' \arrow[r, "\nu' \circ \nu"] & C' \end{tikzcd}$$

The α-*abstraction* approach: abstraction error

In general, an abstraction may imply multiple *causal mechanism diagrams*:

$$\begin{tikzcd} S \arrow[r, "\mu"] \arrow[d, "\alpha_{S'}"'] & T \arrow[d, "\alpha_{T'}"] \\ S' \arrow[r, "\nu"] & T' \end{tikzcd} \qquad \begin{tikzcd} T \arrow[r, "\mu'"] \arrow[d, "\alpha_{T'}"'] & C \arrow[d, "\alpha_{C'}"] \\ T' \arrow[r, "\nu'"] & C' \end{tikzcd}$$

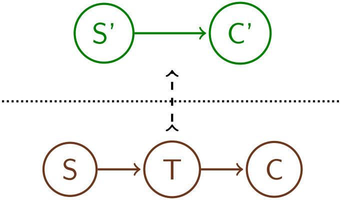
$$\begin{tikzcd} S \arrow[r, "\mu' \circ \mu"] \arrow[d, "\alpha_{S'}"'] & C \arrow[d, "\alpha_{C'}"] \\ S' \arrow[r, "\nu' \circ \nu"] & C' \end{tikzcd}$$

A **(global) abstraction error** [17] $e(\boldsymbol{\alpha})$ is the maximum abstraction error over all diagrams.

$$e(\boldsymbol{\alpha}) = \sup_{\mathbf{X}', \mathbf{Y}' \subseteq \mathcal{X}'} E_{\boldsymbol{\alpha}}(\mathbf{X}', \mathbf{Y}')$$

The α-*abstraction* approach: example (II)

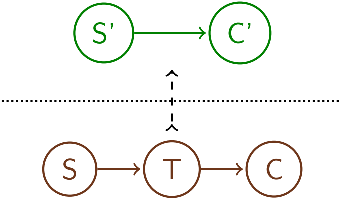
Lung cancer scenario example:



The α-*abstraction* approach: example (II)

Lung cancer scenario example:

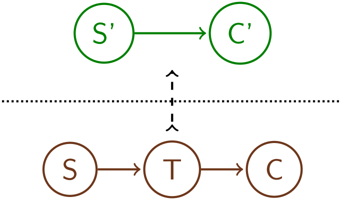
Assuming no commutativity



$$\begin{tikzcd} \mathtt{Dom}[S] \arrow[r, "\mu_C"] \arrow[d, "\alpha_{S'}"'] & \mathtt{Dom}[C] \arrow[d, "\alpha_{C'}"] \\ \mathtt{Dom}[S'] \arrow[r, "\nu_{C'}"'] & \mathtt{Dom}[C'] \end{tikzcd}$$

The α-*abstraction* approach: example (II)

Lung cancer scenario example:



Assuming no commutativity

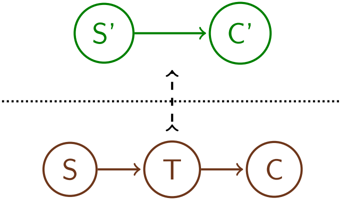
$$\begin{array}{ccc} \mathtt{Dom}[S] & \xrightarrow{\mu_C} & \mathtt{Dom}[C] \\ \downarrow{\scriptstyle \alpha_{S'}} & & \downarrow{\scriptstyle \alpha_{C'}} \\ \mathtt{Dom}[S'] & \xrightarrow[\nu_{C'}]{} & \mathtt{Dom}[C'] \end{array}$$

I can compute *abstraction error*:

$E(\boldsymbol{\alpha}, S', C') = D_{JSD}(\alpha_{C'} \circ \mu_C, \nu_{C'} \circ \alpha_{S'})$

The α-*abstraction* approach: example (II)

Lung cancer scenario example:



Assuming no commutativity

$$\begin{tikzcd} \mathtt{Dom}[S] \arrow[r, "\mu_C"] \arrow[d, "\alpha_{S'}"'] & \mathtt{Dom}[C] \arrow[d, "\alpha_{C'}"] \\ \mathtt{Dom}[S'] \arrow[r, "\nu_{C'}"'] & \mathtt{Dom}[C'] \end{tikzcd}$$

I can compute *abstraction error*:
$E(\boldsymbol{\alpha}, S', C') = D_{JSD}(\alpha_{C'} \circ \mu_C, \nu_{C'} \circ \alpha_{S'})$

Since there are not other subsets this is also the *overall abstraction error*:
$e(\boldsymbol{\alpha}) = E(\boldsymbol{\alpha}, S', C')$

Summary of approaches

- **τ-abstraction approach:** works at the *distributional* level.
- **Φ-abstraction approach:** works at the *structural* level.
- **α-abstraction approach:** works at the *distributional*/*structural* level.

Aligning Approaches [19]

Can we *relate* τ-abstraction and α-abstraction?

- $\times$ Different definition of *abstraction*
- $\times$ Different definition of *consistency*

Aligning Approaches [19]

Can we *relate* τ-abstraction and α-abstraction?

- $\times$ Different definition of *abstraction*
- $\times$ Different definition of *consistency*

It is possible to relate *τ-abstraction*, *α-abstraction* and *cluster DAGs*!

Aligning Approaches [19]

Can we *relate* τ-abstraction and α-abstraction?

- × Different definition of *abstraction*
- × Different definition of *consistency*

It is possible to relate *τ-abstraction*, *α-abstraction* and *cluster DAGs*!

α-abstraction is **equivalent** *to constructive τ-abstraction* (under the existence of an exogenous context giving rise to endogenous setting).

4. Measuring Abstraction Error

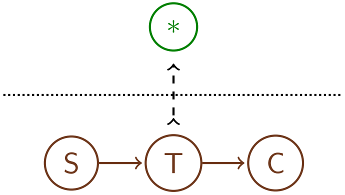
Reference

Quantifying Consistency and Information Loss for Causal Abstraction Learning

Fabio Massimo Zennaro[1] , **Paolo Turrini**[1] and **Theodoros Damoulas**[1]
[1]University of Warwick,Coventry, United Kingdom
{fabio.zennaro, p.turrini, t.damoulas}@warwick.ac.uk,

Measuring Abstraction Error [22]

In the *α-abstraction* framework, does **abstraction error** tell us the whole story about abstraction?



Let $\mathcal{M}'$ be the trivial singleton model.

Then, $e_\alpha = 0$.

We want other *quantitative measures* for an abstraction.

Generalizing Abstraction Error [22]

The abstraction error can be expressed more generally as:

$$E_{\boldsymbol{\alpha}}(\mathbf{X}', \mathbf{Y}') = \operatorname*{agg}_{x' \in \mathbf{X}'} D(p, q)$$

$$e(\boldsymbol{\alpha}) = \operatorname*{agg}_{(\mathbf{X}', \mathbf{Y}') \in \mathcal{J}} E_{\boldsymbol{\alpha}}(\mathbf{X}', \mathbf{Y}')$$

parametrized by aggregation functions, distances, intervention sets, pseudo-inverse, and paths.

$$\begin{tikzcd}
\mathtt{Dom}[S] \arrow[r, "\mu"] \arrow[d, bend right, "\alpha_{S'}"'] & \mathtt{Dom}[T] \arrow[d, bend right, "\alpha_{T'}"'] \\
\mathtt{Dom}[S'] \arrow[r, "\nu"] \arrow[u, bend right, "\alpha^{+}_{S'}"'] & \mathtt{Dom}[T'] \arrow[u, bend right, "\alpha^{+}_{T'}"']
\end{tikzcd}$$

Parameters for a Generalized Abstraction Error [22]

- Aggregation functions:
  - Which *guarantees* do we want?
  - How do we *weight* errors?

Parameters for a Generalized Abstraction Error [22]

- Aggregation functions:
  - Which *guarantees* do we want?
  - How do we *weight* errors?
- Distances:
  - What *metric* do we use on the statistical manifold?
  - Which *properties* does each measure entail?

Parameters for a Generalized Abstraction Error [22]

- Aggregation functions:
  - Which *guarantees* do we want?
  - How do we *weight* errors?
- Distances:
  - What *metric* do we use on the statistical manifold?
  - Which *properties* does each measure entail?
- Intervention sets:
  - Which interventions are *non-redundant*?
  - Which interventions are *relevant*?

Parameters for a Generalized Abstraction Error [22]

- Aggregation functions:
  - Which *guarantees* do we want?
  - How do we *weight* errors?
- Distances:
  - What *metric* do we use on the statistical manifold?
  - Which *properties* does each measure entail?
- Intervention sets:
  - Which interventions are *non-redundant*?
  - Which interventions are *relevant*?
- Pseudo-inverse:
  - How should be an *inverse* defined at all?

Paths: new error measures [22]

If we consider different *paths*, we derive *new error measures*:

Interventional consistency (IC)

$$
\begin{tikzcd}
\mathtt{Dom}[S] \arrow[r, "\mu"] \arrow[d, bend right, "\alpha_{S'}"'] & \mathtt{Dom}[T] \arrow[d, bend right, "\alpha_{T'}"'] \\
\mathtt{Dom}[S'] \arrow[r, "\nu"] \arrow[u, bend right, "\alpha^{+}_{S'}"'] & \mathtt{Dom}[T'] \arrow[u, bend right, "\alpha^{+}_{T'}"']
\end{tikzcd}
$$

Consistency projected on the abstracted model.

Paths: new error measures [22]

If we consider different *paths*, we derive *new error measures*:

Interventional consistency (IC)

$$
\begin{tikzcd}
\mathrm{Dom}[S] \arrow[r, "\mu"] \arrow[d, bend right, "\alpha_{S'}"'] & \mathrm{Dom}[T] \arrow[d, bend right, "\alpha_{T'}"'] \\
\mathrm{Dom}[S'] \arrow[r, "\nu"] \arrow[u, bend right, "\alpha^{+}_{S'}"'] & \mathrm{Dom}[T'] \arrow[u, bend right, "\alpha^{+}_{T'}"']
\end{tikzcd}
$$

Consistency projected on the abstracted model.

Interventional information loss (IIL)

$$
\begin{tikzcd}
\mathrm{Dom}[S] \arrow[r, "\mu"] \arrow[d, bend right, "\alpha_{S'}"'] & \mathrm{Dom}[T] \arrow[d, bend right, "\alpha_{T'}"'] \\
\mathrm{Dom}[S'] \arrow[r, "\nu"] \arrow[u, bend right, "\alpha^{+}_{S'}"'] & \mathrm{Dom}[T'] \arrow[u, bend right, "\alpha^{+}_{T'}"']
\end{tikzcd}
$$

Loss in abstracting and reconstructing.

Paths: new error measures [22]

Interventional superresolution information loss (ISIL)

$$\begin{tikzcd}
\mathtt{Dom}[S] \arrow[r, "\mu"] \arrow[d, bend right, "\alpha_{S'}"'] & \mathtt{Dom}[T] \arrow[d, bend right, "\alpha_{T'}"'] \\
\mathtt{Dom}[S'] \arrow[r, "\nu"] \arrow[u, bend right, "\alpha^{+}_{S'}"'] & \mathtt{Dom}[T'] \arrow[u, bend right, "\alpha^{+}_{T'}"']
\end{tikzcd}$$

Loss in reconstructing and abstracting.

Paths: new error measures [22]

Interventional superresolution information loss (ISIL)

$$
\begin{tikzcd}
\mathtt{Dom}[S] \arrow[r, "\mu"] \arrow[d, bend right, "\alpha_{S'}"'] & \mathtt{Dom}[T] \arrow[d, bend right, "\alpha_{T'}"'] \\
\mathtt{Dom}[S'] \arrow[r, "\nu"] \arrow[u, bend right, "\alpha^+_{S'}"'] & \mathtt{Dom}[T'] \arrow[u, bend right, "\alpha^+_{T'}"']
\end{tikzcd}
$$

Loss in reconstructing and abstracting.

Interventional superresolution consistency (ISC)

$$
\begin{tikzcd}
\mathtt{Dom}[S] \arrow[r, "\mu"] \arrow[d, bend right, "\alpha_{S'}"'] & \mathtt{Dom}[T] \arrow[d, bend right, "\alpha_{T'}"'] \\
\mathtt{Dom}[S'] \arrow[r, "\nu"] \arrow[u, bend right, "\alpha^+_{S'}"'] & \mathtt{Dom}[T'] \arrow[u, bend right, "\alpha^+_{T'}"']
\end{tikzcd}
$$

Consistency projected on the base model.

Some properties of these new error measures [22]

For all the measures above (IC,IIL,ISIL,ISC) with supremum aggregation:

- *Non-monotonicty*: not given that $e(\boldsymbol{\beta\alpha}) \geq e(\boldsymbol{\alpha})$
- *Triangle inequality*: $e(\boldsymbol{\beta\alpha}) \leq e(\boldsymbol{\alpha}) + e(\boldsymbol{\beta})$
- *Ordering:* IIL $\geq$ IC, IIL $\geq$ ISC, IC $\geq$ ISIL, ISC $\geq$ ISIL
- *Finiteness condition*: error is finite if a is order-preserving
- *Different minima:* IC, IIL, ISC, ISIL may disagree on minima

Aside: Causal Emergence [9, 8]

A *Markov chain transition matrix* can be encoded in an *SCM*.

Aside: Causal Emergence [9, 8]

A *Markov chain transition matrix* can be encoded in an *SCM*.

- We can immediately subsume *abstraction between MCs* into *abstraction between SCMs*

Aside: Causal Emergence [9, 8]

A *Markov chain transition matrix* can be encoded in an *SCM*.

- We can immediately subsume *abstraction between MCs* into *abstraction between SCMs*
- We can apply **effective information** to measure causal abstraction

Aside: Causal Emergence [9, 8]

A *Markov chain transition matrix* can be encoded in an *SCM*.

- We can immediately subsume *abstraction between MCs* into *abstraction between SCMs*
- We can apply **effective information** to measure causal abstraction

How are IC and EI related?

What can EI tell us about causal abstraction?

5. Learning Abstractions

Reference

Jointly Learning Consistent Causal Abstractions Over Multiple Interventional Distributions

Fabio Massimo Zennaro FABIO.ZENNARO@WARWICK.AC.UK
Máté Drávucz MATE.DRAVUCZ@WARWICK.AC.UK
Geanina Apachitei GEANINA.APACHITEI@WARWICK.AC.UK
W. Dhammika Widanage DHAMMIKA.WIDANALAGE@WARWICK.AC.UK
Theodoros Damoulas T.DAMOULAS@WARWICK.AC.UK
Dept. of Computer Science & Dept. of Statistics & WMG, University of Warwick, Coventry, CV4 7AL, UK
The Faraday Institution, Harwell Science and Innovation, Campus, Quad One, Didcot, UK

Learning Abstractions [21]

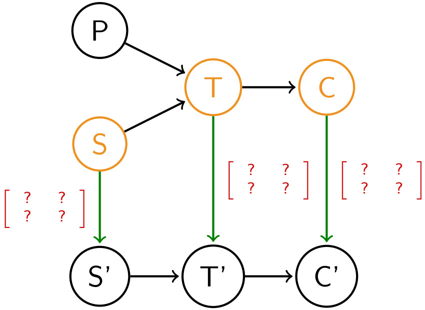
If I am only given the SCMs, can we **learn** an abstraction?

Learning Abstractions [21]

If I am only given the SCMs, can we **learn** an abstraction?

Starting point: Given a partially define *abstraction* $\boldsymbol{\alpha}$ in terms of $\langle R, a \rangle$ can I learn α_i as:

$$\min_{\boldsymbol{\alpha}} \; e(\boldsymbol{\alpha})$$



Challenges [21]

(i) *Multiple related problems*

$$\alpha_{S'} = \begin{bmatrix} ? & ? \\ ? & ? \end{bmatrix}, \alpha_{T'} = \begin{bmatrix} ? & ? \\ ? & ? \end{bmatrix}, \alpha_{C'} = \begin{bmatrix} ? & ? \\ ? & ? \end{bmatrix}$$

$$\begin{tikzcd} S \arrow[r, "\mu"] \arrow[d, "\alpha_{S'}"'] & T \arrow[d, "\alpha_{T'}"] \\ S' \arrow[r, "\nu"] & T' \end{tikzcd} \qquad \begin{tikzcd} T \arrow[r, "\mu'"] \arrow[d, "\alpha_{T'}"'] & C \arrow[d, "\alpha_{C'}"] \\ T' \arrow[r, "\nu'"] & C' \end{tikzcd}$$

$$\begin{tikzcd} S \arrow[r, "\mu' \circ \mu"] \arrow[d, "\alpha_{S'}"'] & C \arrow[d, "\alpha_{C'}"] \\ S' \arrow[r, "\nu' \circ \nu"] & C' \end{tikzcd}$$

Challenges [21]

(i) *Multiple related problems*

(ii) *Combinatorial optimization*

$$\alpha_{S'} = \begin{bmatrix} ? & ? \\ ? & ? \end{bmatrix}, \alpha_{T'} = \begin{bmatrix} ? & ? \\ ? & ? \end{bmatrix}, \alpha_{C'} = \begin{bmatrix} ? & ? \\ ? & ? \end{bmatrix}$$

$$\begin{tikzcd} S \arrow[r, "\mu"] \arrow[d, "\alpha_{S'}"'] & T \arrow[d, "\alpha_{T'}"] \\ S' \arrow[r, "\nu"] & T' \end{tikzcd} \qquad \begin{tikzcd} T \arrow[r, "\mu'"] \arrow[d, "\alpha_{T'}"'] & C \arrow[d, "\alpha_{C'}"] \\ T' \arrow[r, "\nu'"] & C' \end{tikzcd}$$

$$\begin{tikzcd} S \arrow[r, "\mu' \circ \mu"] \arrow[d, "\alpha_{S'}"'] & C \arrow[d, "\alpha_{C'}"] \\ S' \arrow[r, "\nu' \circ \nu"] & C' \end{tikzcd}$$

Challenges [21]

(i) *Multiple related problems*

(ii) *Combinatorial optimization*

(iii) *Surjectivity constraints*

$$\alpha_{S'} = \begin{bmatrix} ? & ? \\ ? & ? \end{bmatrix}, \alpha_{T'} = \begin{bmatrix} ? & ? \\ ? & ? \end{bmatrix}, \alpha_{C'} = \begin{bmatrix} ? & ? \\ ? & ? \end{bmatrix}$$

$$\begin{tikzcd} S \arrow[r, "\mu"] \arrow[d, "\alpha_{S'}"'] & T \arrow[d, "\alpha_{T'}"] \\ S' \arrow[r, "\nu"] & T' \end{tikzcd} \qquad \begin{tikzcd} T \arrow[r, "\mu'"] \arrow[d, "\alpha_{T'}"'] & C \arrow[d, "\alpha_{C'}"] \\ T' \arrow[r, "\nu'"] & C' \end{tikzcd}$$

$$\begin{tikzcd} S \arrow[r, "\mu' \circ \mu"] \arrow[d, "\alpha_{S'}"'] & C \arrow[d, "\alpha_{C'}"] \\ S' \arrow[r, "\nu' \circ \nu"] & C' \end{tikzcd}$$

Challenges [21]

$$\alpha_{S'} = \begin{bmatrix} ? & ? \\ ? & ? \end{bmatrix}, \alpha_{T'} = \begin{bmatrix} ? & ? \\ ? & ? \end{bmatrix}, \alpha_{C'} = \begin{bmatrix} ? & ? \\ ? & ? \end{bmatrix}$$

(i) *Multiple related problems*

(ii) *Combinatorial optimization*

(iii) *Surjectivity constraints*

$$\begin{tikzcd} S \arrow[r, "\mu"] \arrow[d, "\alpha_{S'}"'] & T \arrow[d, "\alpha_{T'}"] \\ S' \arrow[r, "\nu"] & T' \end{tikzcd} \qquad \begin{tikzcd} T \arrow[r, "\mu'"] \arrow[d, "\alpha_{T'}"'] & C \arrow[d, "\alpha_{C'}"] \\ T' \arrow[r, "\nu'"] & C' \end{tikzcd}$$

Baselines: parallel or sequential approaches.

$$\begin{tikzcd} S \arrow[r, "\mu' \circ \mu"] \arrow[d, "\alpha_{S'}"'] & C \arrow[d, "\alpha_{C'}"] \\ S' \arrow[r, "\nu' \circ \nu"] & C' \end{tikzcd}$$

Relaxation and parametrization [21]

We address *(ii) combinatorial optimization* by *relaxing* and *parametrizing* all α_i.

$$\min_{\boldsymbol{\alpha}(\mathbf{W})} \; e(\boldsymbol{\alpha}(\mathbf{W}))$$

$\alpha_{S'}, \alpha_{T'}, \alpha_{C'} \in \mathbb{R}^{2\times 2}$

$$\begin{bmatrix} 0.7 & 1.2 \\ -0.2 & 3.3 \end{bmatrix}$$

Relaxation and parametrization [21]

We address *(ii) combinatorial optimization* by *relaxing* and *parametrizing* all α_i.

$$\min_{\boldsymbol{\alpha}(\mathbf{W})} \; e(\boldsymbol{\alpha}(\mathbf{W}))$$

$$\alpha_{S'}, \alpha_{T'}, \alpha_{C'} \in \mathbb{R}^{2\times 2}$$

$$\begin{bmatrix} 0.7 & 1.2 \\ -0.2 & 3.3 \end{bmatrix}$$

We add *tempering* $t(W) = \frac{e^{\frac{W_i j}{T}}}{\sum_i e^{\frac{W_i j}{T}}}$ along the matrix columns to binarize them.

$$\mathcal{L}_1 : \min_{\boldsymbol{\alpha}(\mathbf{W})} \; e(\boldsymbol{\alpha}(t(\mathbf{W})))$$

$$\alpha_{S'}, \alpha_{T'}, \alpha_{C'} \in [0,1]^{2\times 2}$$

$$t\left(\begin{bmatrix} 0.7 & 1.2 \\ -0.2 & 3.3 \end{bmatrix}\right) = \begin{bmatrix} 0.99 & 0.02 \\ 0.01 & 0.98 \end{bmatrix}$$

Enforcing surjectivity [21]

We address *(iii) surjective constraints* through a *penalty function*:

$$\mathcal{L}_2 : \min_{\mathbf{W}} \sum_{\mathbf{W}} \sum_{i} \left(1 - \max_{j} t(W)_{ij} \right)$$

$$\alpha_{S'}, \alpha_{T'}, \alpha_{C'} \in [0,1]^{2\times 2}$$

$$\begin{bmatrix} 0.99 & 0.02 \\ 0.01 & 0.98 \end{bmatrix} \overset{\mathcal{L}_2}{\rightsquigarrow}$$

$$(1-0.99)+(1-0.98)$$

Solution by gradient descent [21]

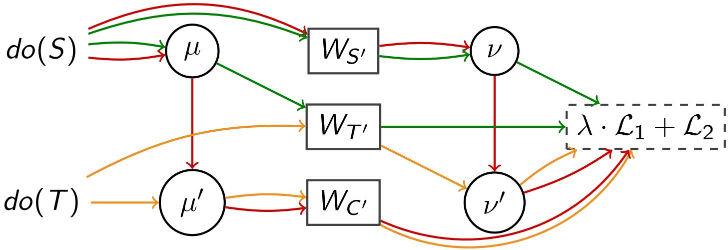
We address *(i) multiple related problems* by *jointly* solving all the problems via *gradient descent*:

$$\begin{tikzcd} S \arrow[r, "\mu"] \arrow[d, "\alpha_{S'}"'] & T \arrow[d, "\alpha_{T'}"] \\ S' \arrow[r, "\nu"] & T' \end{tikzcd} \qquad \begin{tikzcd} T \arrow[r, "\mu'"] \arrow[d, "\alpha_{T'}"'] & C \arrow[d, "\alpha_{C'}"] \\ T' \arrow[r, "\nu'"] & C' \end{tikzcd} \qquad \begin{tikzcd} S \arrow[r, "\mu' \circ \mu"] \arrow[d, "\alpha_{S'}"'] & C \arrow[d, "\alpha_{C'}"] \\ S' \arrow[r, "\nu' \circ \nu"] & C' \end{tikzcd}$$

Solution by gradient descent [21]

We address *(i) multiple related problems* by *jointly* solving all the problems via *gradient descent*:

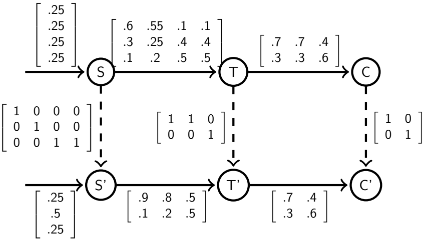
$$\begin{tikzcd} S \arrow[r, "\mu"] \arrow[d, "\alpha_{S'}"'] & T \arrow[d, "\alpha_{T'}"] \\ S' \arrow[r, "\nu"] & T' \end{tikzcd} \qquad \begin{tikzcd} T \arrow[r, "\mu'"] \arrow[d, "\alpha_{T'}"'] & C \arrow[d, "\alpha_{C'}"] \\ T' \arrow[r, "\nu'"] & C' \end{tikzcd} \qquad \begin{tikzcd} S \arrow[r, "\mu' \circ \mu"] \arrow[d, "\alpha_{S'}"'] & C \arrow[d, "\alpha_{C'}"] \\ S' \arrow[r, "\nu' \circ \nu"] & C' \end{tikzcd}$$

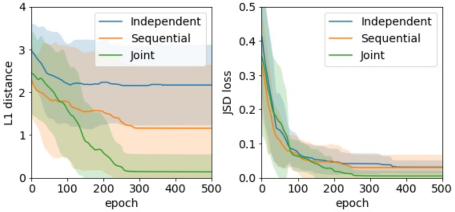


Synthetic Experiments [21]

We evaluated our learning method:

- On multiple synthetic models;
- Against *independent* and *sequential* approach;
- Monitoring *loss functions*, *L1-dist from ground truth*, *wall-clock time*.





Real-World Experiments [21]

We want to model the stage of **coating** in lithium-ion battery manufacturing:

$$\text{Mass Loading} = \text{f(input)}$$

Experiments are costly, so we want to integrate data[1] collected by two groups running similar (but not identical) experiments:

LRCS (France)

Collection of few statistics in each a few stages of battery manufacturing [2].

WMG (UK)

Collection of detailed space- and time-dependent measurements during coating.

[1] https://chemistry-europe.onlinelibrary.wiley.com/doi/full/10.1002/batt.201900135
https://github.com/mattdravucz/jointly-learning-causal-abstraction/

Real-World Experiments [21]

We evaluated our learning method:

- Performing abstraction of data from base to abstracted (WMG $\rightarrow$ LRCS);
- Evaluating change in performance using aggregated data when predicting *out-of-sample* (k).

	Training set	**Test Set**	**MSE**
(a)	LRCS[$CG \neq k$]	LRCS[$CG = k$]	1.86 ± 1.75
(b)	LRCS[$CG \neq k$] + WMG	LRCS[$CG = k$]	0.22 ± 0.26
(c)	LRCS[$CG \neq k$] + WMG[$CG \neq k$]	LRCS[$CG = k$] + WMG[$CG = k$]	1.22 ± 0.95

Further Learning Approaches

A number of approaches consider learning abstractions using different assumptions and methods:

Further Learning Approaches

A number of approaches consider learning abstractions using different assumptions and methods:

- [5] learn *optimal transport* maps between multiple pairs of interventional distributions.

Further Learning Approaches

A number of approaches consider learning abstractions using different assumptions and methods:

- [5] learn *optimal transport* maps between multiple pairs of interventional distributions.
- [7] learn abstractions between *neural networks* and *interpretable models*.

Further Learning Approaches

A number of approaches consider learning abstractions using different assumptions and methods:

- [5] learn *optimal transport* maps between multiple pairs of interventional distributions.
- [7] learn abstractions between *neural networks* and *interpretable models*.
- [11] learn abstractions in the *linear regime*.

Further Learning Approaches

A number of approaches consider learning abstractions using different assumptions and methods:

- [5] learn *optimal transport* maps between multiple pairs of interventional distributions.
- [7] learn abstractions between *neural networks* and *interpretable models*.
- [11] learn abstractions in the *linear regime*.
- [10] learn abstractions centered around a *target variable*.

Further Learning Approaches

A number of approaches consider learning abstractions using different assumptions and methods:

- [5] learn *optimal transport* maps between multiple pairs of interventional distributions.
- [7] learn abstractions between *neural networks* and *interpretable models*.
- [11] learn abstractions in the *linear regime*.
- [10] learn abstractions centered around a *target variable*.
- [4] learn abstractions of *agent-based models*.

Further Learning Approaches

A number of approaches consider learning abstractions using different assumptions and methods:

- [5] learn *optimal transport* maps between multiple pairs of interventional distributions.
- [7] learn abstractions between *neural networks* and *interpretable models*.
- [11] learn abstractions in the *linear regime*.
- [10] learn abstractions centered around a *target variable*.
- [4] learn abstractions of *agent-based models*.
- [3] learns abstractions solving an *optimization problem* on the Stiefel manifold.

6. Learning with Abstractions

Reference

Causally Abstracted Multi-armed Bandits

Fabio Massimo Zennaro[1] **Nicholas Bishop**[2] **Joel Dyer**[2] **Yorgos Felekis**[3] **Anisoara Calinescu**[2]
Michael Wooldridge[2] **Theodoros Damoulas**[3]

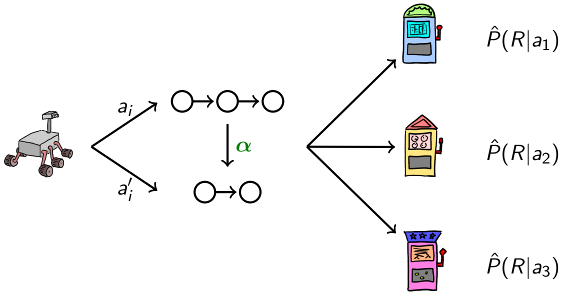
[1]University of Bergen
[2]University of Oxford
[3]University of Warwick

Causally abstracted multi-armed bandits (CAMABs) [20]

In a CAMAB, an agent has **multiple causal models**.

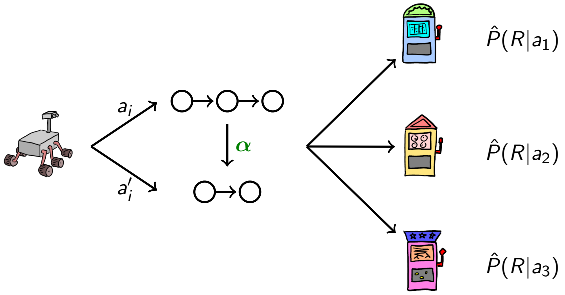
Causally abstracted multi-armed bandits (CAMABs) [20]

In a CAMAB, an agent has **multiple causal models**.



Causally abstracted multi-armed bandits (CAMABs) [20]

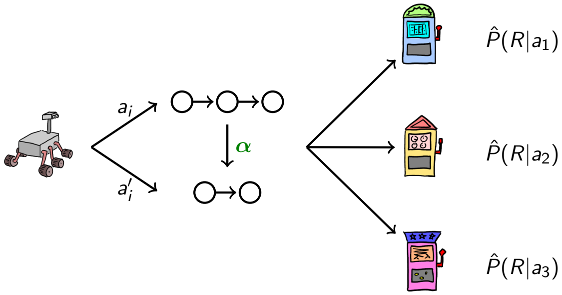
In a CAMAB, an agent has **multiple causal models**.



✓ A CAMAB capture a setting where *multiple actors* tackle the same problem at different levels of abstraction.

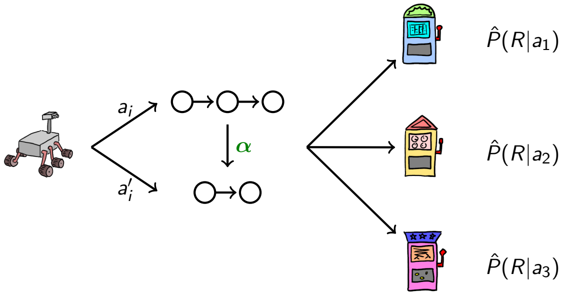
Causally abstracted multi-armed bandits (CAMABs) [20]

How do we take advantage of α?



Causally abstracted multi-armed bandits (CAMABs) [20]

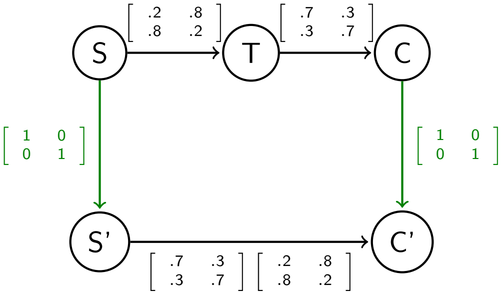
How do we take advantage of α?



We will consider some approaches inspired by *reinforcement learning*.

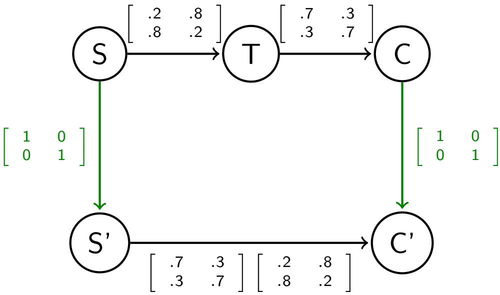
CAMAB - Transporting Optimal Action [20]

Let us consider a *CAMAB* made up by two CMABs $\mathcal{M}$ and $\mathcal{M}'$:



CAMAB - Transporting Optimal Action [20]

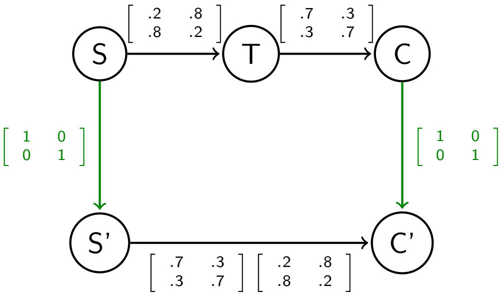
Let us consider a *CAMAB* made up by two CMABs $\mathcal{M}$ and $\mathcal{M}'$:



Let us assume:

CAMAB - Transporting Optimal Action [20]

Let us consider a *CAMAB* made up by two CMABs $\mathcal{M}$ and $\mathcal{M}'$:

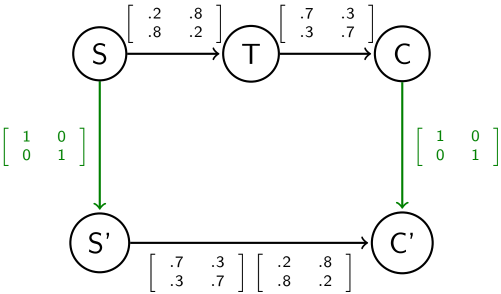


Let us assume:

- An abstraction α with zero error;

CAMAB - Transporting Optimal Action [20]

Let us consider a *CAMAB* made up by two CMABs $\mathcal{M}$ and $\mathcal{M}'$:

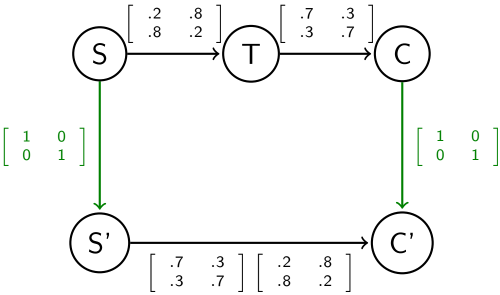


Let us assume:

- An abstraction α with zero error;
- An *optimal action* a^* in $\mathcal{M}$.

CAMAB - Transporting Optimal Action [20]

Let us consider a *CAMAB* made up by two CMABs $\mathcal{M}$ and $\mathcal{M}'$:



Let us assume:

- An abstraction α with zero error;
- An *optimal action* a^* in $\mathcal{M}$.

Does it hold that: $a'^* = \alpha(a^*)$?

CAMAB - Transporting Optimal Action [20]

It does **NOT**:

$$
\begin{tikzcd}
S \arrow[r, "\begin{bmatrix} .2 & .8 \\ .8 & .2 \end{bmatrix}"] \arrow[dd, "\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}"'] & T \arrow[r, "\begin{bmatrix} .7 & .3 \\ .3 & .7 \end{bmatrix}"] & C \arrow[dd, "\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}"] \\
& & \\
S' \arrow[rr, "\begin{bmatrix} .7 & .3 \\ .3 & .7 \end{bmatrix}\begin{bmatrix} .2 & .8 \\ .8 & .2 \end{bmatrix}"'] & & C'
\end{tikzcd}
$$

CAMAB - Transporting Optimal Action [20]

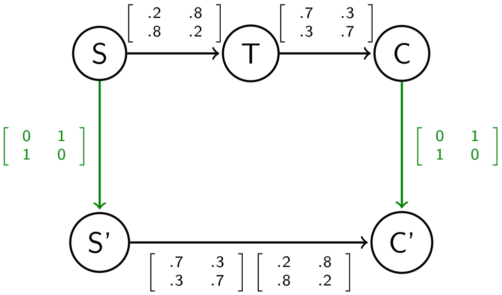
It does **NOT**:

$$
\begin{tikzcd}[column sep=huge, row sep=huge]
S \arrow[r, "\begin{bmatrix} .2 & .8 \\ .8 & .2 \end{bmatrix}"] \arrow[d, swap, "\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}"] & T \arrow[r, "\begin{bmatrix} .7 & .3 \\ .3 & .7 \end{bmatrix}"] & C \arrow[d, "\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}"] \\
S' \arrow[rr, swap, "\begin{bmatrix} .7 & .3 \\ .3 & .7 \end{bmatrix} \begin{bmatrix} .2 & .8 \\ .8 & .2 \end{bmatrix}"] & & C'
\end{tikzcd}
$$

Optimality may not be preserved:

CAMAB - Transporting Optimal Action [20]

It does **NOT**:



Optimality may not be preserved:

- If actions and outcomes are *consistently* flipped.

CAMAB - Transporting Optimal Action [20]

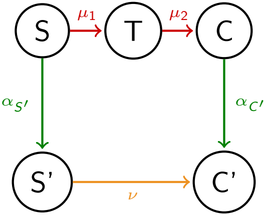
It does **NOT**:

$$
\begin{tikzcd}
S \arrow[r, "\begin{bmatrix} .2 & .8 \\ .8 & .2 \end{bmatrix}"] \arrow[d, "\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}"'] & T \arrow[r, "\begin{bmatrix} .7 & .3 \\ .3 & .7 \end{bmatrix}"] & C \arrow[d, "\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}"] \\
S' \arrow[rr, "\begin{bmatrix} .7 & .3 \\ .3 & .7 \end{bmatrix}\begin{bmatrix} .2 & .8 \\ .8 & .2 \end{bmatrix}"'] & & C'
\end{tikzcd}
$$

Optimality may not be preserved:

- If actions and outcomes are *consistently* flipped.
- (If the domains of the outcomes are different).

CAMAB - Reward Discrepancy [20]



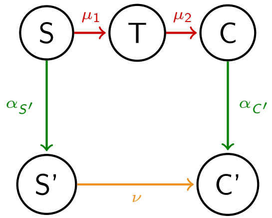
CAMAB - Reward Discrepancy [20]

$$\begin{tikzcd} S \arrow[r, "\mu_1"] \arrow[d, "\alpha_{S'}"'] & T \arrow[r, "\mu_2"] & C \arrow[d, "\alpha_{C'}"] \\ S' \arrow[rr, "\nu"'] & & C' \end{tikzcd}$$

If we want to study CAMABs *abstraction error* is not enough:

$$e(\boldsymbol{\alpha}) = \sup_{\mathbf{X}',\mathbf{Y}' \subseteq \mathcal{X}'} \max_{\iota} D(\alpha_{C'} \cdot \mu_2 \cdot \mu_1, \nu \cdot \alpha_{S'})$$

CAMAB - Reward Discrepancy [20]



If we want to study CAMABs *abstraction error* is not enough:

$$e(\boldsymbol{\alpha}) = \sup_{\mathbf{X}', \mathbf{Y}' \subseteq \mathcal{X}'} \max_{\iota} D(\alpha_{C'} \cdot \mu_2 \cdot \mu_1, \nu \cdot \alpha_{S'})$$

We want to consider also **reward discrepancy**:

$$s(\boldsymbol{\alpha}) = \sup_{\mathbf{X}', \mathbf{Y}' \subseteq \mathcal{X}'} \max_{\iota} D(\mu_2 \cdot \mu_1, \nu \cdot \alpha_{S'})$$

CAMAB - Reward Discrepancy [20]

$$\begin{tikzcd}
S \arrow[r, "\mu_1"] \arrow[d, "\alpha_{S'}"'] & T \arrow[r, "\mu_2"] & C \arrow[d, "\alpha_{C'}"] \\
S' \arrow[rr, "\nu"'] & & C'
\end{tikzcd}$$

If we want to study CAMABs *abstraction error* is not enough:

$$e(\boldsymbol{\alpha}) = \sup_{\mathbf{X}',\mathbf{Y}' \subseteq \mathcal{X}'} \max_{\iota} D(\alpha_{C'} \cdot \mu_2 \cdot \mu_1, \nu \cdot \alpha_{S'})$$

We want to consider also **reward discrepancy**:

$$s(\boldsymbol{\alpha}) = \sup_{\mathbf{X}',\mathbf{Y}' \subseteq \mathcal{X}'} \max_{\iota} D(\mu_2 \cdot \mu_1, \nu \cdot \alpha_{S'})$$

(Assuming same dimension of the domains of C and C')

CAMAB - Triangular Inequality [20]

Abstraction error:

$$e(\boldsymbol{\alpha}) = \sup_{\mathbf{X}', \mathbf{Y}' \subseteq \mathcal{X}'} \max_{\iota} D(\alpha_{C'} \cdot \mu_2 \cdot \mu_1, \nu \cdot \alpha_{S'})$$

CAMAB - Triangular Inequality [20]

Abstraction error:

$$e(\boldsymbol{\alpha}) = \sup_{\mathbf{X}',\mathbf{Y}' \subseteq \mathcal{X}'} \max_{\iota} D(\alpha_{C'} \cdot \mu_2 \cdot \mu_1, \nu \cdot \alpha_{S'})$$

Reward discrepancy:

$$s(\boldsymbol{\alpha}) = \sup_{\mathbf{X}',\mathbf{Y}' \subseteq \mathcal{X}'} \max_{\iota} D(\mu_2 \cdot \mu_1, \nu \cdot \alpha_{S'})$$

CAMAB - Triangular Inequality [20]

Abstraction error:

$$e(\boldsymbol{\alpha}) = \sup_{\mathbf{X}',\mathbf{Y}' \subseteq \mathcal{X}'} \max_{\iota} D(\alpha_{C'} \cdot \mu_2 \cdot \mu_1, \nu \cdot \alpha_{S'})$$

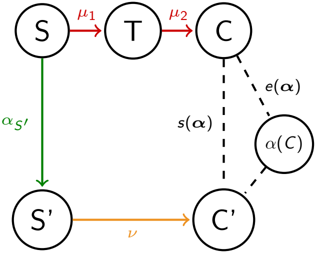
Reward discrepancy:

$$s(\boldsymbol{\alpha}) = \sup_{\mathbf{X}',\mathbf{Y}' \subseteq \mathcal{X}'} \max_{\iota} D(\mu_2 \cdot \mu_1, \nu \cdot \alpha_{S'})$$

This immediately gives us a **triangular inequality**:

$$|\mu_{a'} - \mu_{\boldsymbol{\alpha}(a)}| \leq e(\boldsymbol{\alpha}) + s(\boldsymbol{\alpha})$$

CAMAB - Triangular Inequality [20]



Abstraction error:

$$e(\boldsymbol{\alpha}) = \sup_{\mathbf{X}', \mathbf{Y}' \subseteq \mathcal{X}'} \max_{\iota} D(\alpha_{C'} \cdot \mu_2 \cdot \mu_1, \nu \cdot \alpha_{S'})$$

Reward discrepancy:

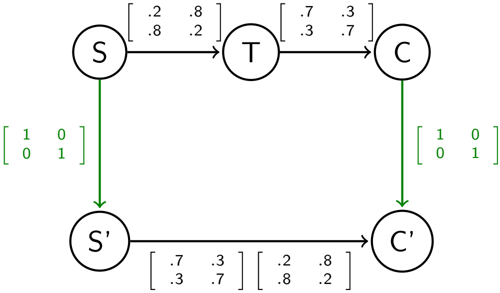
$$s(\boldsymbol{\alpha}) = \sup_{\mathbf{X}', \mathbf{Y}' \subseteq \mathcal{X}'} \max_{\iota} D(\mu_2 \cdot \mu_1, \nu \cdot \alpha_{S'})$$

This immediately gives us a **triangular inequality**:

$$|\mu_{a'} - \mu_{\boldsymbol{\alpha}(a)}| \leq e(\boldsymbol{\alpha}) + s(\boldsymbol{\alpha})$$

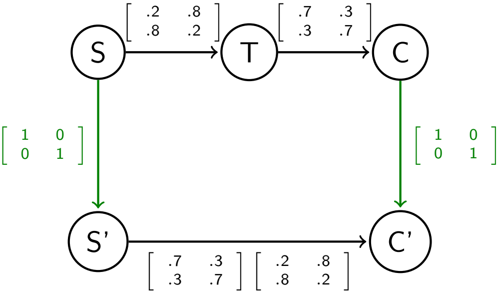
CAMAB - Transporting Actions [20]

Let us consider a *CAMAB* made up by two CMABs $\mathcal{M}$ and $\mathcal{M}'$:



CAMAB - Transporting Actions [20]

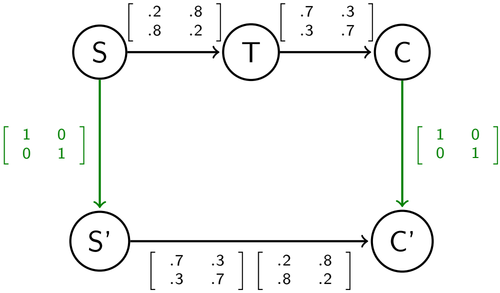
Let us consider a *CAMAB* made up by two CMABs $\mathcal{M}$ and $\mathcal{M}'$:



Let us assume:

CAMAB - Transporting Actions [20]

Let us consider a *CAMAB* made up by two CMABs $\mathcal{M}$ and $\mathcal{M}'$:

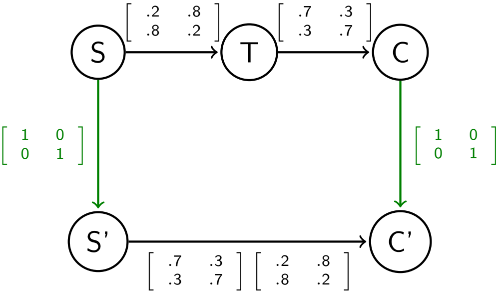


Let us assume:

- We have the collection of all the action $a^{(t)}$ taken on $\mathcal{M}$.

CAMAB - Transporting Actions [20]

Let us consider a *CAMAB* made up by two CMABs $\mathcal{M}$ and $\mathcal{M}'$:

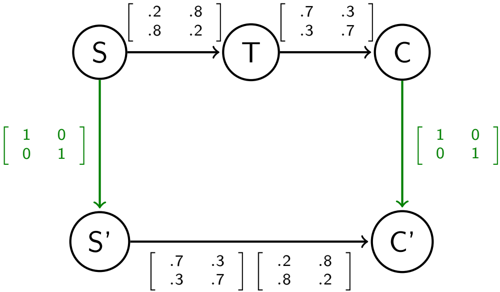


Let us assume:

- We have the collection of all the action $a^{(t)}$ taken on $\mathcal{M}$.
- We have *optimality preservation*.

CAMAB - Transporting Actions [20]

Let us consider a *CAMAB* made up by two CMABs $\mathcal{M}$ and $\mathcal{M}'$:



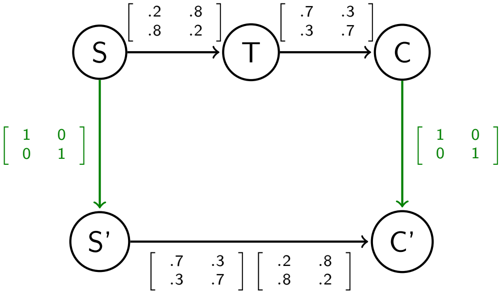
Let us assume:

- We have the collection of all the action $a^{(t)}$ taken on $\mathcal{M}$.
- We have *optimality preservation*.

Can I earn anything by *imitation*, that is playing: $a'^{(t)} = \alpha(a^{(t)})$?

CAMAB - Transporting Actions [20]

Let us consider a *CAMAB* made up by two CMABs $\mathcal{M}$ and $\mathcal{M}'$:



Let us assume:

- We have the collection of all the action $a^{(t)}$ taken on $\mathcal{M}$.
- We have *optimality preservation*.

Can I earn anything by *imitation*, that is playing: $a'^{(t)} = \alpha(a^{(t)})$?
If so, when?

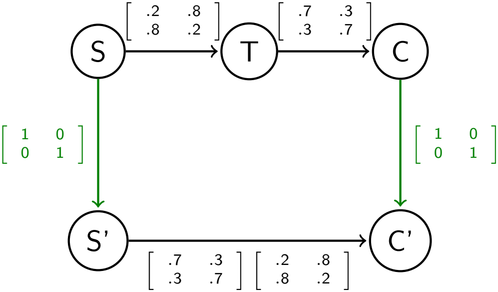
CAMAB - Transporting Actions [20]

Let us refine our assumptions further:

$$
\begin{tikzcd}
S \arrow[r, "\begin{bmatrix} .2 & .8 \\ .8 & .2 \end{bmatrix}"] \arrow[dd, "\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}"'] & T \arrow[r, "\begin{bmatrix} .7 & .3 \\ .3 & .7 \end{bmatrix}"] & C \arrow[dd, "\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}"] \\
 & & \\
S' \arrow[rr, "\begin{bmatrix} .7 & .3 \\ .3 & .7 \end{bmatrix} \begin{bmatrix} .2 & .8 \\ .8 & .2 \end{bmatrix}"'] & & C'
\end{tikzcd}
$$

CAMAB - Transporting Actions [20]

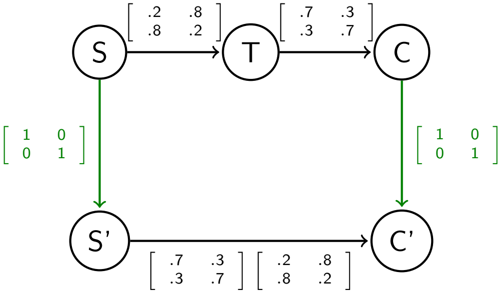
Let us refine our assumptions further:



Let us assume:

CAMAB - Transporting Actions [20]

Let us refine our assumptions further:

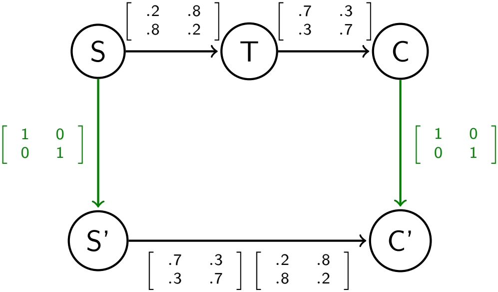


Let us assume:

- We have run the *UCB* algorithm on $\mathcal{M}$ for T steps.

CAMAB - Transporting Actions [20]

Let us refine our assumptions further:



Let us assume:

- We have run the *UCB* algorithm on $\mathcal{M}$ for T steps.

When is it that the *imitation* algorithm on $\mathcal{M}'$ performs better than *UCB* on $\mathcal{M}'$?

CAMAB - Transporting Actions [20]

The *imitation* protocol has a lower regret bound than UCB if:

$$\underbrace{3\sum_{a'\in\mathcal{A}'}\Delta(a')\left[1-\mathcal{K}(a')\right]}_{\text{fixed cost with possible oversampling arms}} \quad + 16\log T \underbrace{\sum_{a'\in\mathcal{A}'}\left[\frac{\Delta(a')}{\Delta(a')^2} - \sum_{a\in\mathcal{A}|\alpha(a)=a'}\frac{\Delta(a')}{\Delta(a)^2}\right]}_{\text{variable cost driven by the base model}} \geq 0$$

CAMAB - Transporting Actions [20]

The *imitation* protocol has a lower regret bound than UCB if:

$$\underbrace{3\sum_{a'\in\mathcal{A}'}\Delta(a')\left[1-\mathcal{K}(a')\right]}_{\text{fixed cost with possible oversampling arms}} \quad +16\log T\underbrace{\sum_{a'\in\mathcal{A}'}\left[\frac{\Delta(a')}{\Delta(a')^2}-\sum_{a\in\mathcal{A}|\alpha(a)=a'}\frac{\Delta(a')}{\Delta(a)^2}\right]}_{\text{variable cost driven by the base model}}\geq 0$$

- $\mathcal{K}(a')$ gives us the *number of base actions* a mapping to a'.
- $\Delta(a)$ is the *optimality gap* for action a.

CAMAB - Transporting Actions [20]

The *imitation* protocol has a lower regret bound than UCB if:

$$\underbrace{3 \sum_{a' \in \mathcal{A}'} \Delta(a') \left[1 - \mathcal{K}(a')\right]}_{\text{fixed cost with possible oversampling arms}} + 16 \log T \underbrace{\sum_{a' \in \mathcal{A}'} \left[\frac{\Delta(a')}{\Delta(a')^2} - \sum_{a \in \mathcal{A} | \alpha(a) = a'} \frac{\Delta(a')}{\Delta(a)^2} \right]}_{\text{variable cost driven by the base model}} \geq 0$$

- $\mathcal{K}(a')$ gives us the *number of base actions* a mapping to a'.
- $\Delta(a)$ is the *optimality gap* for action a.
- Bound derived from results on UCB:

CAMAB - Transporting Actions [20]

The *imitation* protocol has a lower regret bound than UCB if:

$$\underbrace{3 \sum_{a' \in \mathcal{A}'} \Delta(a') \left[1 - \mathcal{K}(a')\right]}_{\text{fixed cost with possible oversampling arms}} \quad + 16 \log T \underbrace{\sum_{a' \in \mathcal{A}'} \left[\frac{\Delta(a')}{\Delta(a')^2} - \sum_{a \in \mathcal{A} | \alpha(a) = a'} \frac{\Delta(a')}{\Delta(a)^2} \right]}_{\text{variable cost driven by the base model}} \geq 0$$

- $\mathcal{K}(a')$ gives us the *number of base actions* a mapping to a'.
- $\Delta(a)$ is the *optimality gap* for action a.
- Bound derived from results on UCB:
  - Fixed cost of sampling all actions;

CAMAB - Transporting Actions [20]

The *imitation* protocol has a lower regret bound than UCB if:

$$\underbrace{3 \sum_{a' \in \mathcal{A}'} \Delta(a') \left[1 - \mathcal{K}(a')\right]}_{\text{fixed cost with possible oversampling arms}} + \underbrace{16 \log T \sum_{a' \in \mathcal{A}'} \left[\frac{\Delta(a')}{\Delta(a')^2} - \sum_{a \in \mathcal{A} | \alpha(a) = a'} \frac{\Delta(a')}{\Delta(a)^2} \right]}_{\text{variable cost driven by the base model}} \geq 0$$

- $\mathcal{K}(a')$ gives us the *number of base actions* a mapping to a'.
- $\Delta(a)$ is the *optimality gap* for action a.
- Bound derived from results on UCB:
  - Fixed cost of sampling all actions;
    - If many actions a are mapped to the same a' you will oversample a'

CAMAB - Transporting Actions [20]

The *imitation* protocol has a lower regret bound than UCB if:

$$\underbrace{3 \sum_{a' \in \mathcal{A}'} \Delta(a') \left[1 - \mathcal{K}(a')\right]}_{\text{fixed cost with possible oversampling arms}} \quad + \underbrace{16 \log T \sum_{a' \in \mathcal{A}'} \left[\frac{\Delta(a')}{\Delta(a')^2} - \sum_{a \in \mathcal{A} | \alpha(a) = a'} \frac{\Delta(a')}{\Delta(a)^2} \right]}_{\text{variable cost driven by the base model}} \geq 0$$

- $\mathcal{K}(a')$ gives us the *number of base actions* a mapping to a'.
- $\Delta(a)$ is the *optimality gap* for action a.
- Bound derived from results on UCB:
    - Fixed cost of sampling all actions;
        - If many actions a are mapped to the same a' you will oversample a'
    - Variables cost to achieve a level of confidence;

CAMAB - Transporting Actions [20]

The *imitation* protocol has a lower regret bound than UCB if:

$$\underbrace{3 \sum_{a' \in \mathcal{A}'} \Delta(a') \left[1 - \mathcal{K}(a')\right]}_{\text{fixed cost with possible oversampling arms}} \quad + 16 \log T \underbrace{\sum_{a' \in \mathcal{A}'} \left[\frac{\Delta(a')}{\Delta(a')^2} - \sum_{a \in \mathcal{A} | \alpha(a) = a'} \frac{\Delta(a')}{\Delta(a)^2} \right]}_{\text{variable cost driven by the base model}} \geq 0$$

- $\mathcal{K}(a')$ gives us the *number of base actions* a mapping to a'.
- $\Delta(a)$ is the *optimality gap* for action a.
- Bound derived from results on UCB:
  - Fixed cost of sampling all actions;
    - If many actions a are mapped to the same a' you will oversample a'
  - Variables cost to achieve a level of confidence;
    - If action a has big optimality gap, it will make the corresponding action a' oversampled.

CAMAB - Transporting Actions [20]

The *imitation* protocol has a lower regret bound than UCB if:

$$\underbrace{3 \sum_{a' \in \mathcal{A}'} \Delta(a') \left[1 - \mathcal{K}(a')\right]}_{\text{fixed cost with possible oversampling arms}} \quad \underbrace{+16 \log T \sum_{a' \in \mathcal{A}'} \left[\frac{\Delta(a')}{\Delta(a')^2} - \sum_{a \in \mathcal{A} | \alpha(a) = a'} \frac{\Delta(a')}{\Delta(a)^2} \right]}_{\text{variable cost driven by the base model}} \geq 0$$

- $\mathcal{K}(a')$ gives us the *number of base actions* a mapping to a'.
- $\Delta(a)$ is the *optimality gap* for action a.
- Bound derived from results on UCB:
  - Fixed cost of sampling all actions;
    - If many actions a are mapped to the same a' you will oversample a'
  - Variables cost to achieve a level of confidence;
    - If action a has big optimality gap, it will make the corresponding action a' oversampled.
- Ideally, optimal action a^* and a number of actions with small gap $\Delta(a)$ maps to the optimal a'^*

7. Conclusion

Conclusions

Large space for conceptual and practical development of **causal abstraction frameworks**:

- *Foundations* of the framemorks
- *Characterization* of these frameworks
- *Algorithmic and empirical* development

More about abstraction:
https://github.com/FMZennaro/CausalAbstraction/

CAR Workshop 2025

UAI 2025 will host a workshop on **causal abstraction** and **causal representation learning**!

`https://sites.google.com/view/car-25/`

Join us in *Rio de Janeiro* in July!

Thanks!

Thank you for your attention!

References I

[1] Sander Beckers and Joseph Y Halpern. Abstracting causal models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 2678–2685, 2019.

[2] Ricardo Pinto Cunha, Teo Lombardo, Emiliano N Primo, and Alejandro A Franco. Artificial intelligence investigation of nmc cathode manufacturing parameters interdependencies. *Batteries & Supercaps*, 3(1):60–67, 2020.

[3] Gabriele D'Acunto, Fabio Massimo Zennaro, Yorgos Felekis, and Paolo Di Lorenzo. Causal abstraction learning based on the semantic embedding principle. *arXiv preprint arXiv:2502.00407*, 2025.

[4] 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.

References II

[5] Yorgos Felekis, Fabio Massimo Zennaro, Nicola Branchini, and Theodoros Damoulas. Causal optimal transport of abstractions. *arXiv preprint arXiv:2312.08107*, 2023.

[6] Luciano Floridi. The method of levels of abstraction. *Minds and machines*, 18:303–329, 2008.

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

[8] Erik P Hoel. When the map is better than the territory. *Entropy*, 19(5):188, 2017.

[9] Erik P Hoel, Larissa Albantakis, and Giulio Tononi. Quantifying causal emergence shows that macro can beat micro. *Proceedings of the National Academy of Sciences*, 110(49):19790–19795, 2013.

[10] Armin Kekić, Bernhard Schölkopf, and Michel Besserve. Targeted reduction of causal models. *arXiv preprint arXiv:2311.18639*, 2023.

References III

[11] Riccardo Massidda, Sara Magliacane, and Davide Bacciu. Learning causal abstractions of linear structural causal models. *arXiv preprint arXiv:2406.00394*, 2024.

[12] Jun Otsuka and Hayato Saigo. On the equivalence of causal models: A category-theoretic approach. *arXiv preprint arXiv:2201.06981*, 2022.

[13] Jun Otsuka and Hayato Saigo. The process theory of causality: an overview. 2022.

[14] Judea Pearl. *Causality*. Cambridge University Press, 2009.

[15] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: Foundations and learning algorithms*. MIT Press, 2017.

[16] Eigil F Rischel and Sebastian Weichwald. Compositional abstraction error and a category of causal models. *arXiv preprint arXiv:2103.15758*, 2021.

References IV

[17] Eigil Fjeldgren Rischel. The category theory of causal models. 2020.

[18] Paul K Rubenstein, Sebastian Weichwald, Stephan Bongers, Joris M Mooij, Dominik Janzing, Moritz Grosse-Wentrup, and Bernhard Schölkopf. Causal consistency of structural equation models. In *33rd Conference on Uncertainty in Artificial Intelligence (UAI 2017)*, pages 808–817. Curran Associates, Inc., 2017.

[19] Willem Schooltink and Fabio Massimo Zennaro. Aligning graphical and functional causal abstractions. *arXiv preprint arXiv:2412.17080*, 2024.

[20] 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.

References V

[21] 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.

[22] Fabio Massimo Zennaro, Paolo Turrini, and Theo Damoulas. Quantifying consistency and information loss for causal abstraction learning. In *Proceedings of the Thrity-Second International Conference on International Joint Conferences on Artificial Intelligence*, 2023.