

# **Causal Inference with Deep Learning and Generative Models**

## **UAI 2025 Tutorial**

**Murat Kocaoglu**

Purdue University

(@Johns Hopkins University  
from Fall 2025)

**Md. Musfiqur Rahman**

Purdue University

# Causal Inference with Deep Learning and Generative Models

## *Outline*

- Background
  - Causal Inference Basics
  - Neural Network Basics
- A Taxonomy of Deep Learning Approaches for Causal Inference
  - Function Modeling (a.k.a. Curve Fitting)
  - Feature Extraction
  - Generative Causal Inference

# Disclaimer!

- This tutorial is incomplete!
  - There are too many papers to cover in one tutorial on deep learning for causal inference.
  - We will highlight some of the main ideas. Omission not reflection of a paper's importance, but our ignorance!
- This tutorial is brief!
  - We may be imprecise at times, and not fully rigorous for brevity.
- This tutorial is biased!
  - We will focus on SCM view even though a paper is using PO.

# What this tutorial is NOT

We do not cover the following:

- Causal discovery. We will assume graph is given.
- Causal analysis of deep networks. We will focus on the other way around: how deep learning can be leveraged for causal inference.



# What this tutorial is

- Three main ideas repeatedly used by several papers.
- Cover and interpret key papers that represent these main ideas.
- Explore their strengths and weaknesses (e.g., assumptions).

# Causal Inference with Deep Learning and Generative Models

## *Outline*

- Background
  - Causal Inference Basics
  - Neural Network Basics
- A Taxonomy of Deep Learning Approaches for Causal Inference
  - Function Modeling (a.k.a. Curve Fitting)
  - Feature Extraction
  - Generative Causal Inference

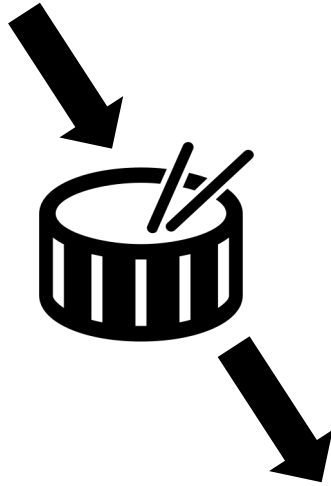
# Causal Inference

# Sun-Eating Beasts and How to Chase Them





# Sun-Eating Beasts and How to Chase Them



# Modeling Probabilistic Causation

$X$  : Percentage of population w/ access to clean water

$Y$  : Child mortality

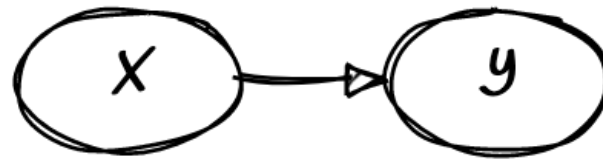
X	Y
22	165
97	15
85	33
100	3
51	154
....	....

<http://data.un.org>

# Modeling Probabilistic Causation

$X$  : Percentage of population w/ access to clean water

$Y$  : Child mortality



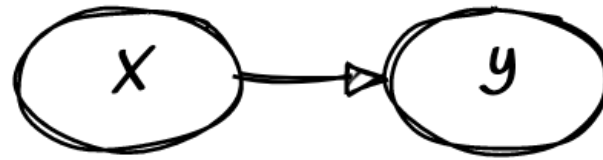
X	Y
22	165
97	15
85	33
100	3
51	154
....	....

<http://data.un.org>

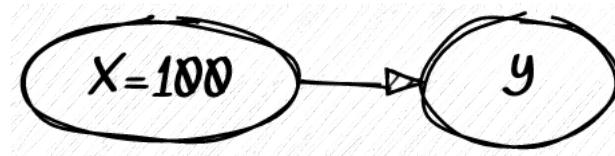
# Modeling Probabilistic Causation

$X$  : Percentage of population w/ access to clean water

$Y$  : Child mortality



Magic wand to  
intervene/do:



Y
3
1
5
....

X	Y
22	165
97	15
85	33
100	3
51	154
....	....

<http://data.un.org>



# Modeling Probabilistic Causation

$X$  : Percentage of population w/ access to clean water

$Y$  : Child mortality

$X$  is said to cause  $Y$



intervening on  $X$

$$Y = f(X, E)$$

changes

the distribution of  $Y$

Magic wand to  
intervene/do:

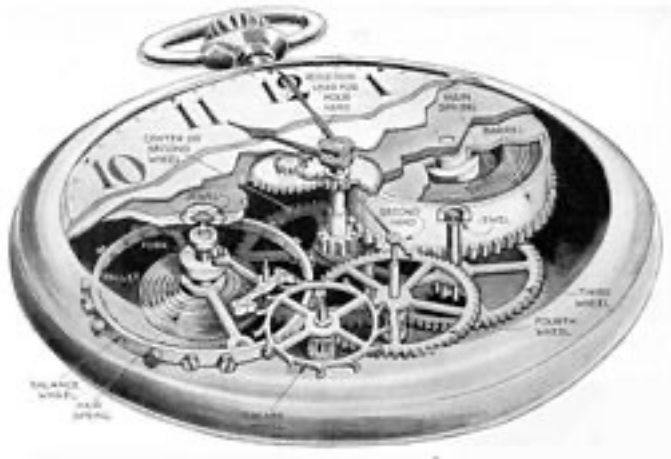


$Y$
3
1
5
....

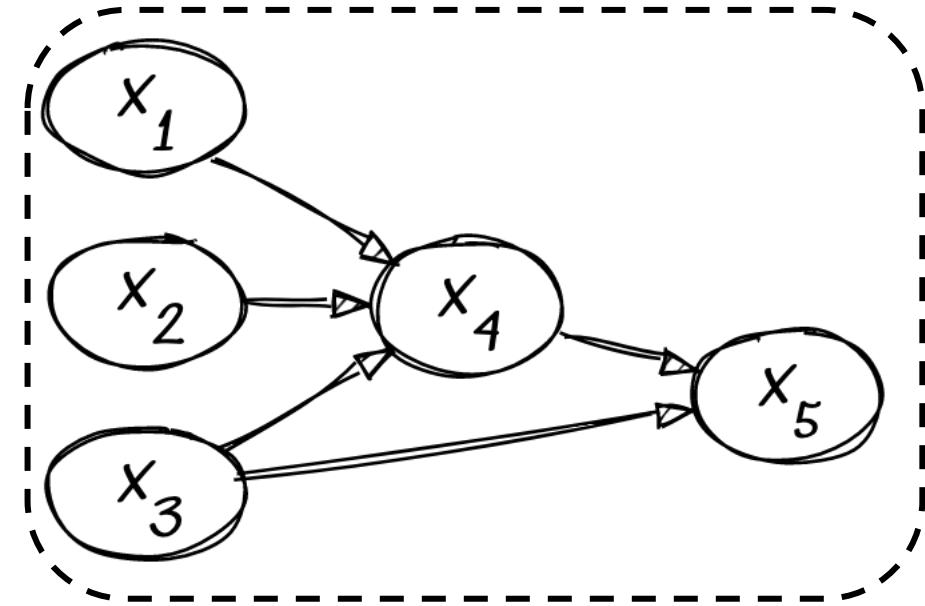
$X$	$Y$
22	165
97	15
85	33
100	3
51	154
....	....

<http://data.un.org>

# Causal Modeling



Causal Graph



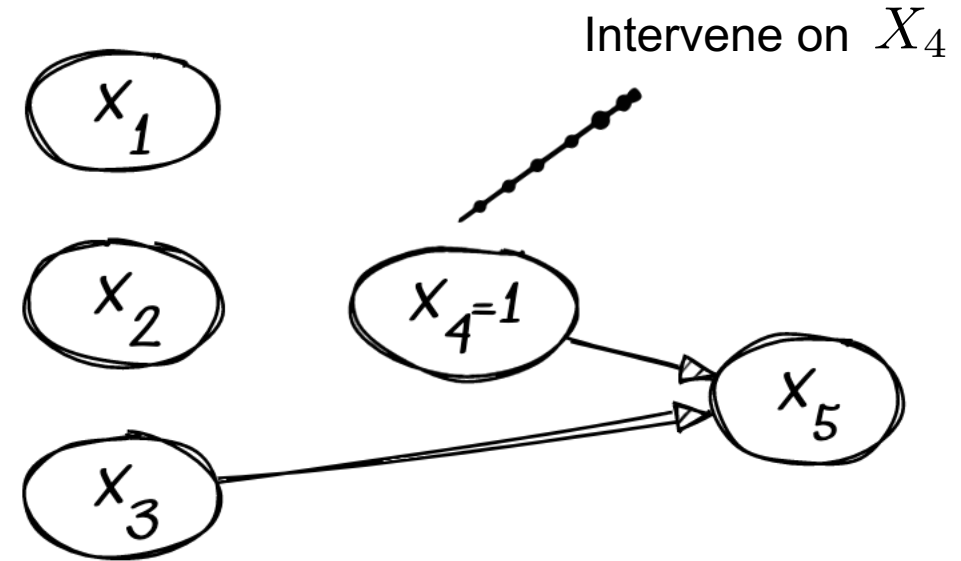
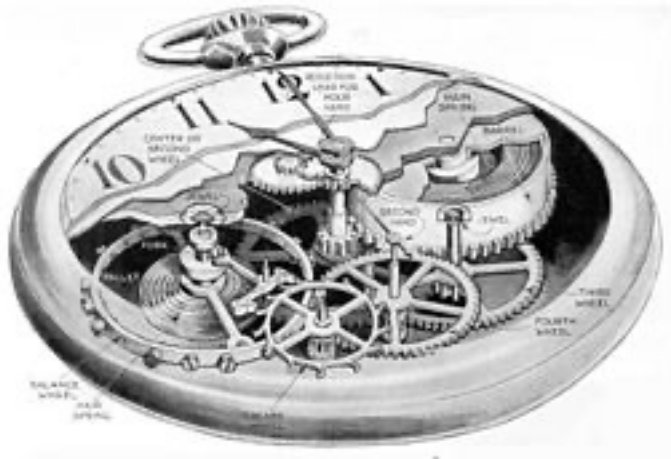
**Vertices:** Random variables

**Edges** : Causal relations

$$X_i = f_i(Pa_i, E_i)$$

$Pa_i$ : Set of parents of  $X_i$  in the causal graph

# Causal Modeling

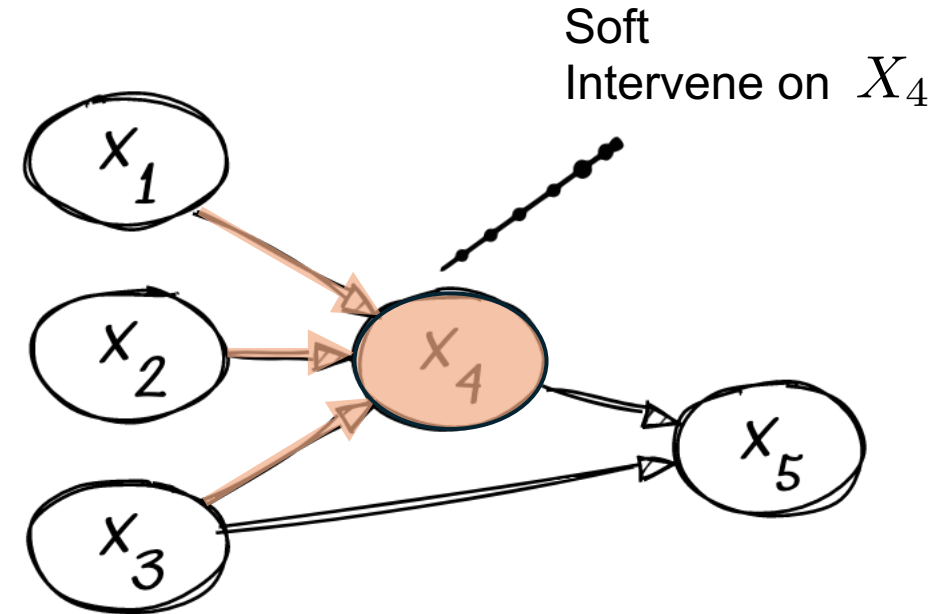
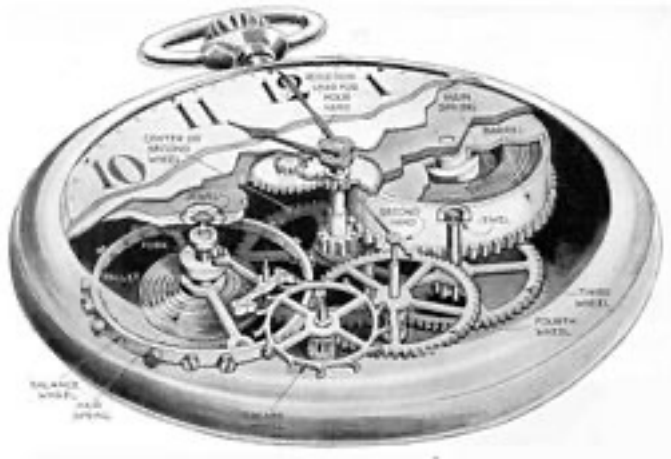


**Vertices:** Random variables  
**Edges** : Causal relations

$$X_i = f_i(Pa_i, E_i)$$

$Pa_i$ : Set of parents of  $X_i$  in the causal graph

# Causal Modeling



**Vertices:** Random variables  
**Edges** : Causal relations

$$X_i = f_i(Pa_i, E_i)$$

$Pa_i$ : Set of parents of  $X_i$  in the causal graph

# Inferring Causation

Does going to college have any causal effect on income at 30?

Went to College	Income at 30 > 50k
0	0
0	1
0	0
1	1
1	1
....	....

# Inferring Causation

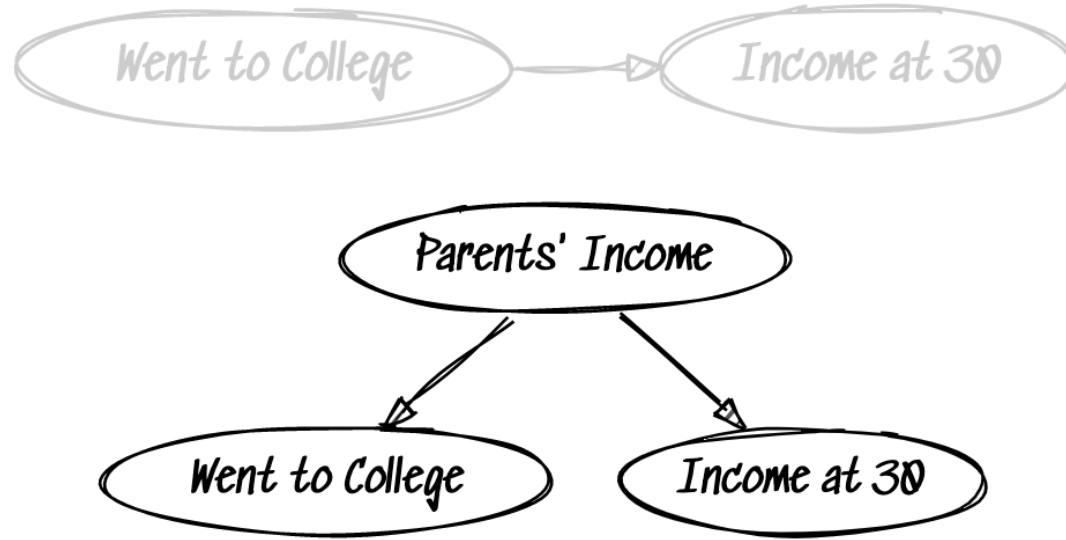
Does going to college have any causal effect on income at 30?



Went to College	Income at 30 > 50k
0	0
0	1
0	0
1	1
1	1
....	....

# Inferring Causation

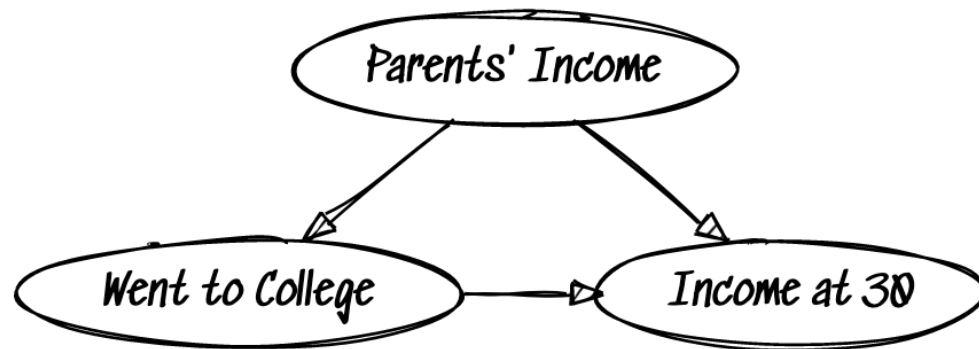
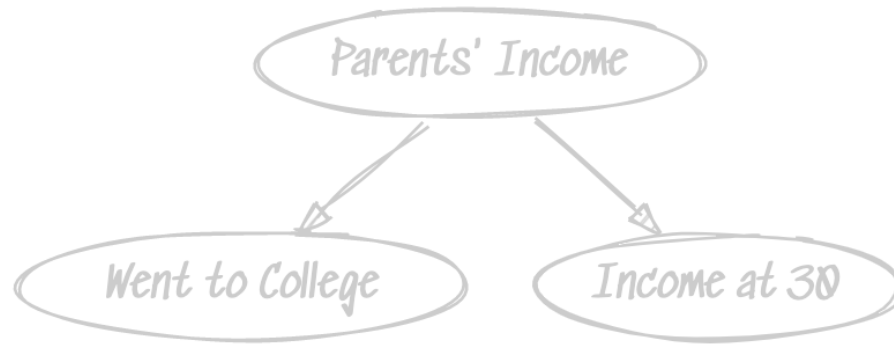
Does going to college have any causal effect on income at 30?



Went to College	Income at 30 > 50k	Parents' Income
0	0	0
0	1	1
0	0	0
1	1	1
1	1	1
....	....	....

# Inferring Causation

Does going to college have any causal effect on income at 30?

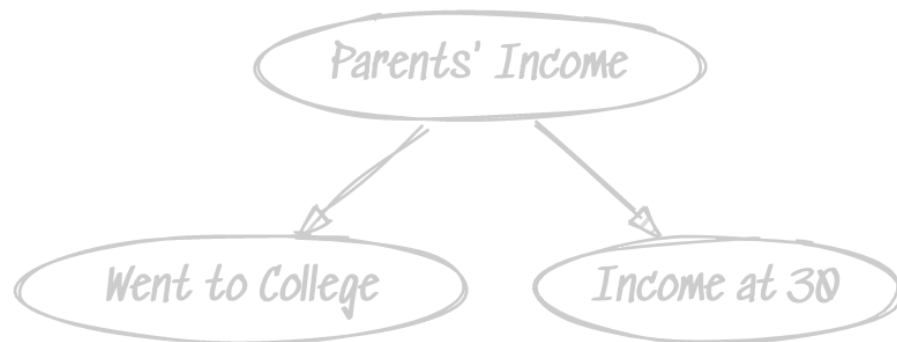


Went to College	Income at 30 > 50k	Parents' Income
0	0	0
0	1	1
0	0	0
1	1	1
1	1	1
....	....	....



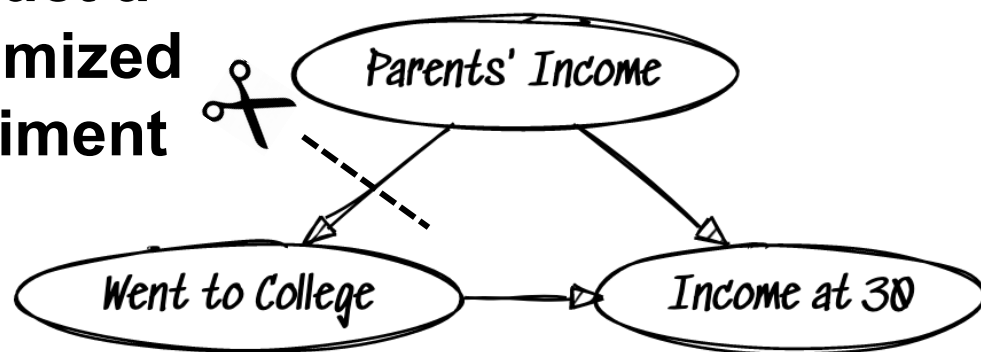
# Inferring Causation

Does going to college have any causal effect on income at 30?



Went to College	Income at 30 > 50k	Parents' Income
0	0	0
0	1	1
0	0	0
1	1	1
1	1	1
....	....	....

**Conduct a  
Randomized  
Experiment**



# Inferring Causation

## Conduct intervention (RCT)

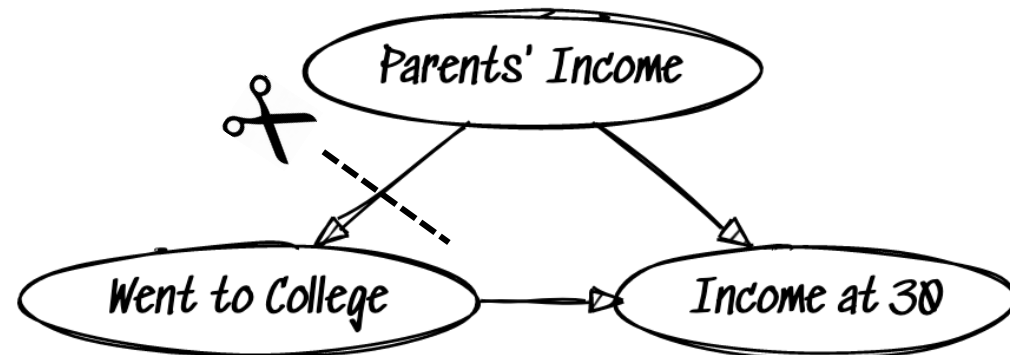
- Force half the people to go

Went to College	Income at 30 > 50k
1	1
1	1
1	0
....	....

- Force other half to NOT go

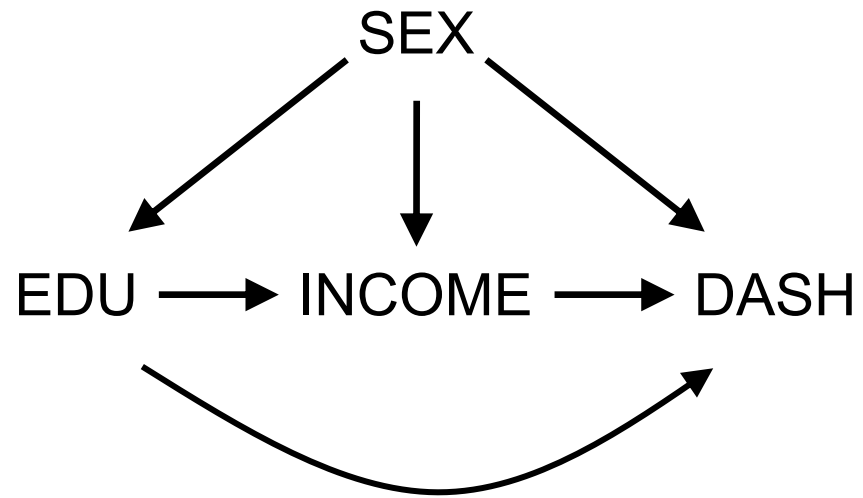
Went to College	Income at 30 > 50k
0	0
0	1
0	0
....	....

- Compare income of both populations



# From Observational Data

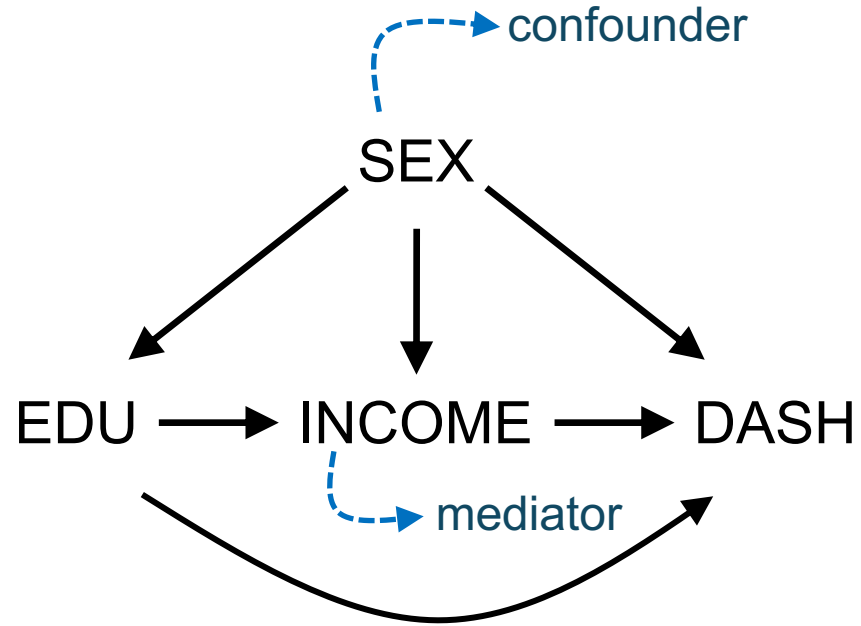
*To adjust or not to adjust?*



DASH: Dietary Approach to Stop Hypertension

# From Observational Data

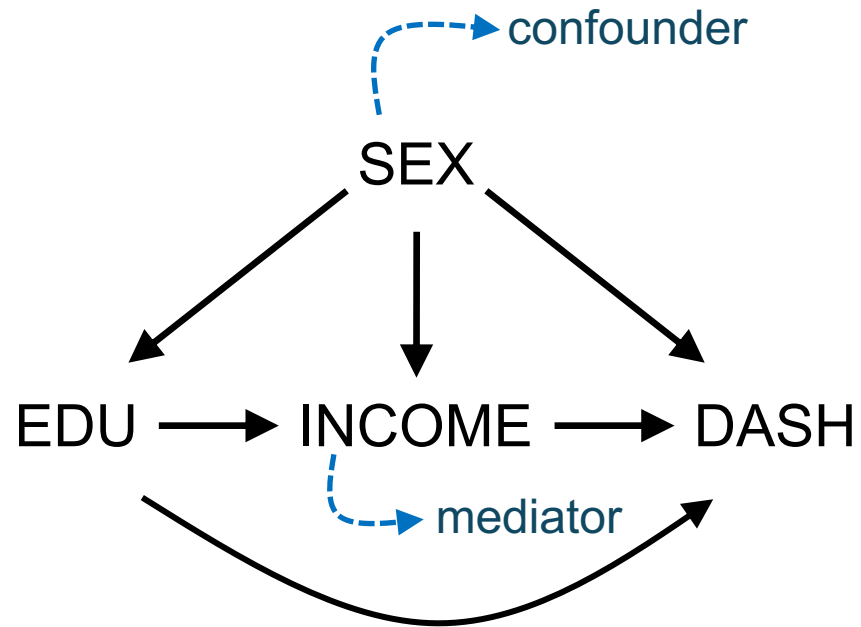
*To adjust or not to adjust?*



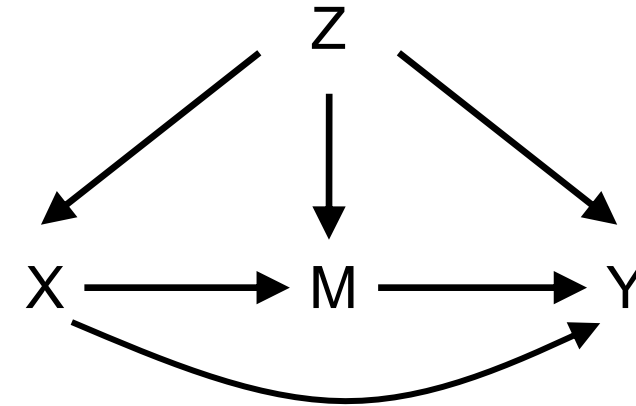
DASH: Dietary Approach to Stop Hypertension

# From Observational Data

## *To adjust or not to adjust?*



DASH: Dietary Approach to Stop Hypertension



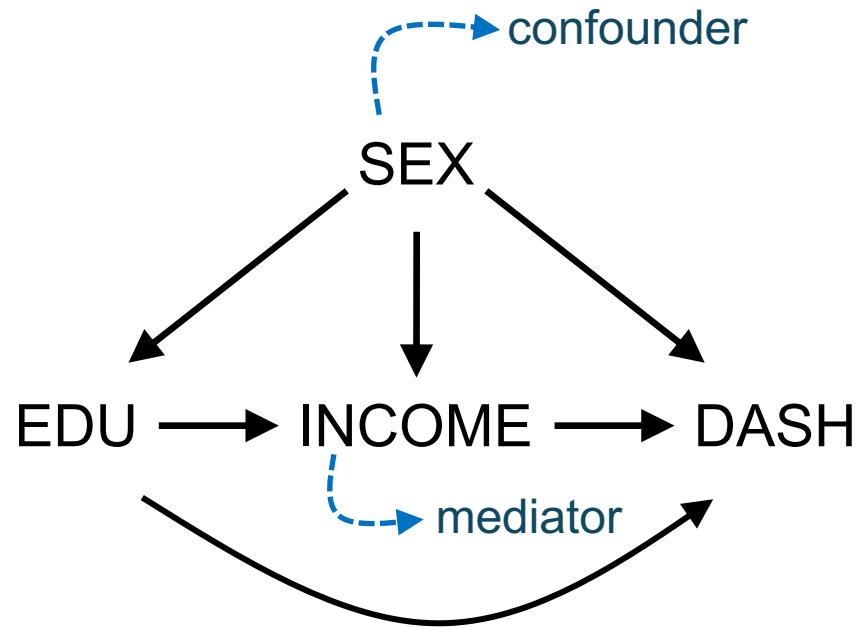
Positivity  
Consistency  
Ignorability

$$Y(0), Y(1) \perp\!\!\!\perp X \mid Z$$

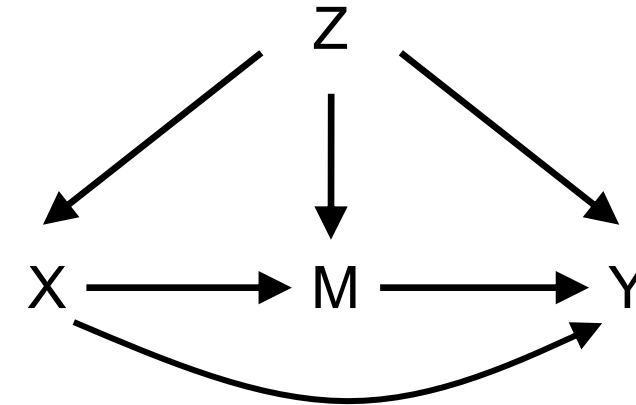
$$\mathbb{E}[Y(1) - Y(0)] = \mathbb{E}[\mathbb{E}[Y|X = 1, Z]] - \mathbb{E}[\mathbb{E}[Y|X = 0, Z]]$$

# From Observational Data

## *To adjust or not to adjust?*



DASH: Dietary Approach to Stop Hypertension



Positivity  
Consistency  
Ignorability

$Y(0), Y(1) \perp\!\!\!\perp X | Z$

$$\mathbb{E}[Y(1) - Y(0)] = \mathbb{E}[\mathbb{E}[Y | X = 1, Z]] - \mathbb{E}[\mathbb{E}[Y | X = 0, Z]]$$

X	Z	Y
0	..	..
0	..	..
0	..	..
1	..	..
1	..	..

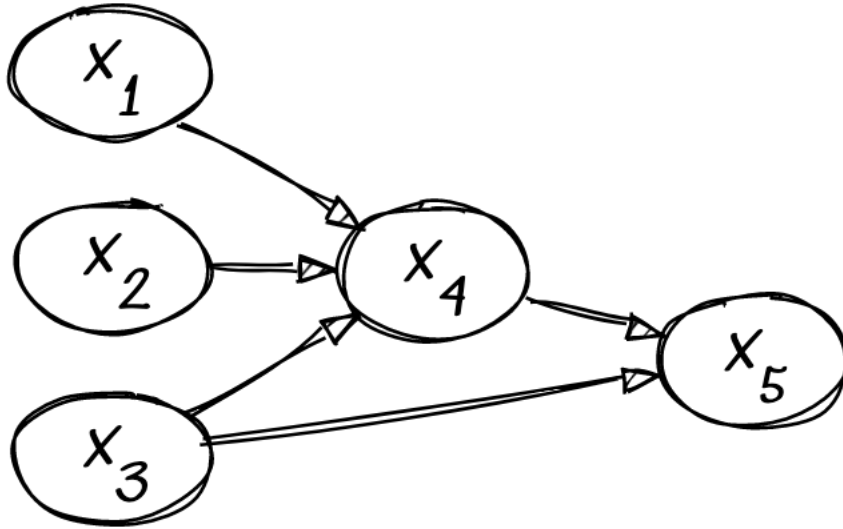
Regress Y on Z → Average w/ Z (unconditional) ★

Regress Y on Z → Average w/ Z (unconditional) ■

# Causal Inference

- It is possible to estimate interventional distribution from observational data **and** the causal graph.
- **Do-calculus rules** are sufficient to convert an identifiable interventional query into a function of the observed distribution, **if it is possible**.
- Sound and complete **ID**entification algorithms are given by (Shpitser & Pearl) and (Tian & Pearl).

# Causal Graphs Imply Dependency Models



Induced Conditional  
Independences (CI):

$$X_1 \perp\!\!\!\perp X_2$$

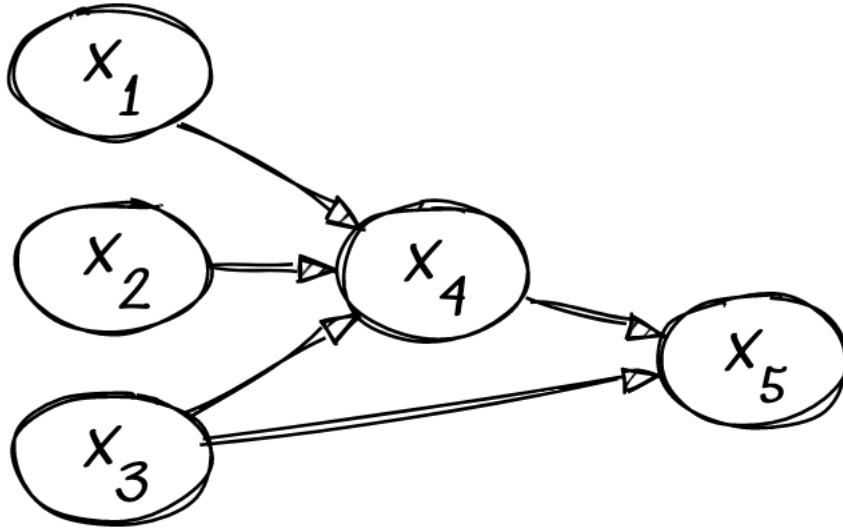
$$X_1 \not\perp\!\!\!\perp X_2 \mid X_4$$

$$X_1 \not\perp\!\!\!\perp X_3 \mid X_5$$

$$X_1 \perp\!\!\!\perp X_5 \mid X_4, X_3$$



# Causal Graphs Imply Dependency Models



Induced Conditional  
Independences (CI):

$$X_1 \perp\!\!\!\perp X_2$$

$$X_1 \not\perp\!\!\!\perp X_2 \mid X_4$$

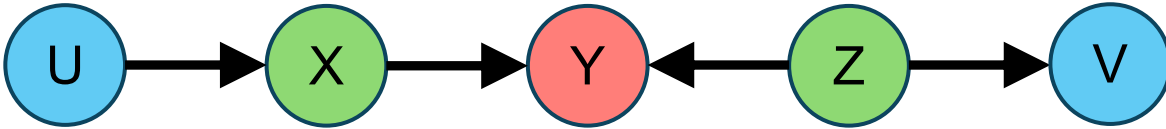
$$X_1 \not\perp\!\!\!\perp X_3 \mid X_5$$

$$X_1 \perp\!\!\!\perp X_5 \mid X_4, X_3$$

*Causal DAG is a Bayes Net for the induced distribution!*

# d-separation

Causal Graph

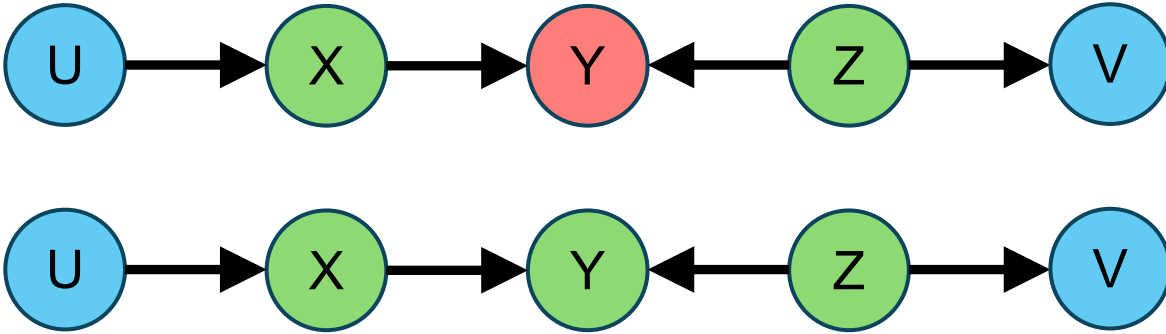


Induced CI

$$U \perp\!\!\!\perp V \mid \emptyset$$

# d-separation

Causal Graph



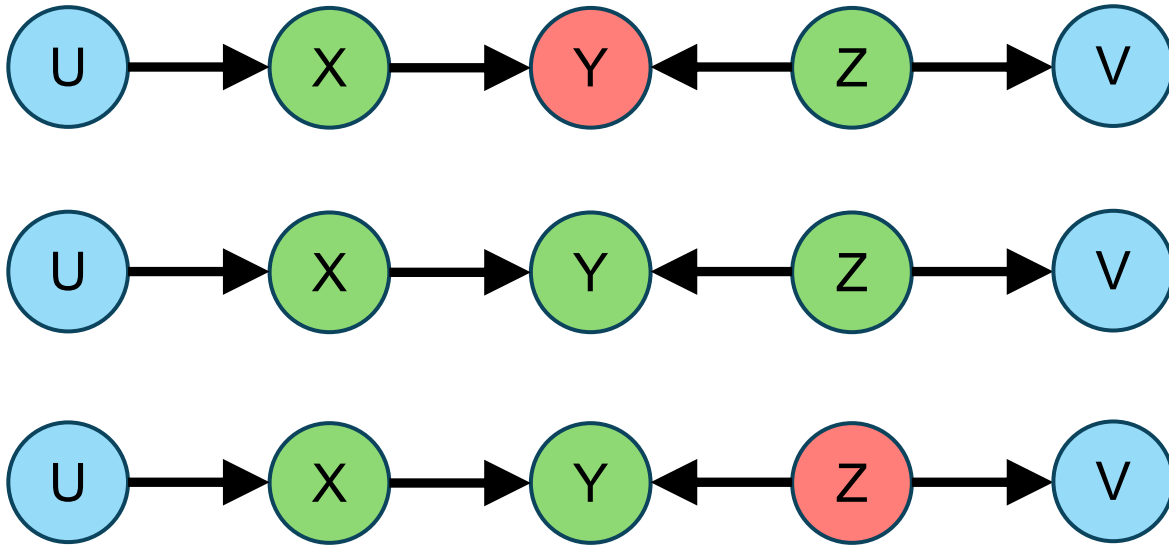
Induced CI

$$U \perp\!\!\!\perp V \mid \emptyset$$

$$U \not\perp\!\!\!\perp V \mid Y$$

# d-separation

Causal Graph



Induced CI

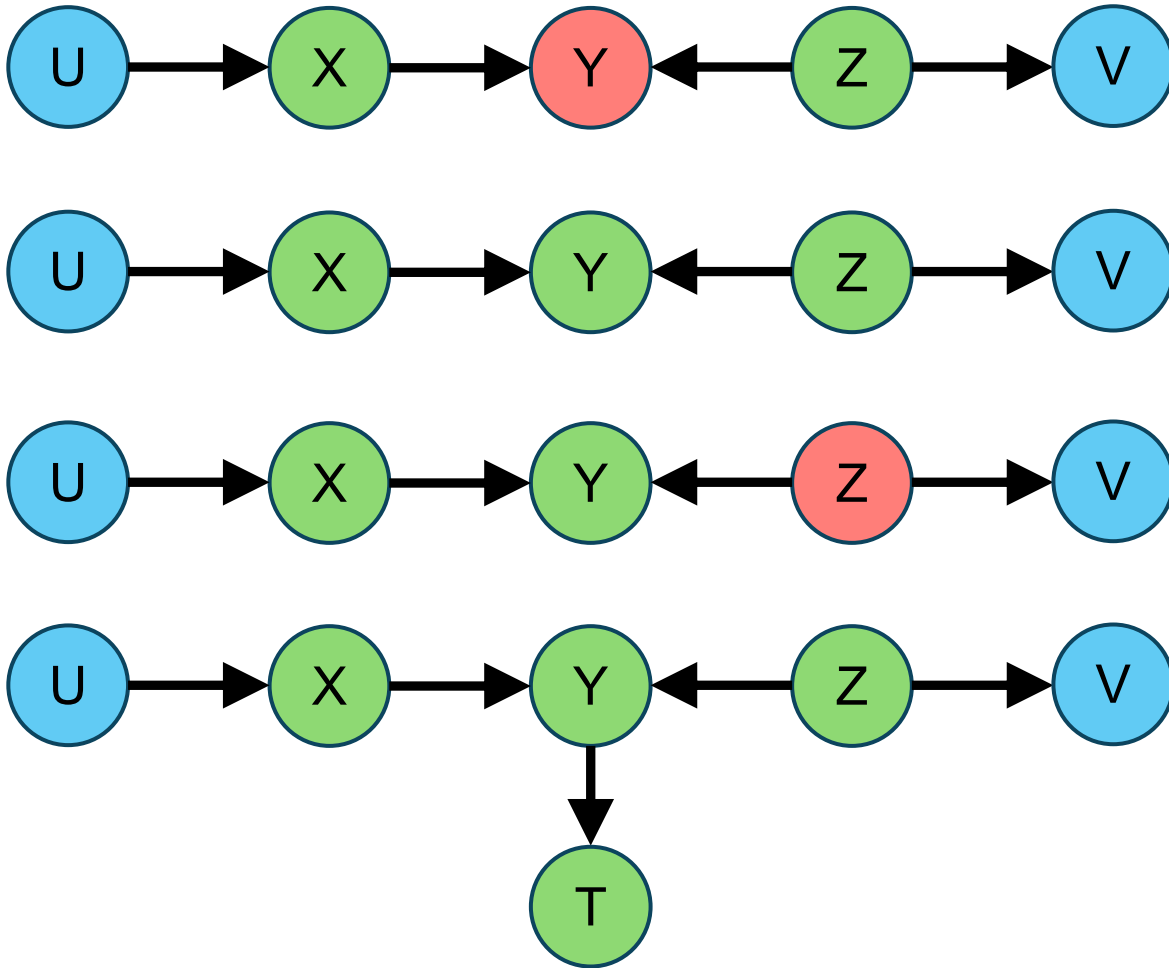
$$U \perp\!\!\!\perp V \mid \emptyset$$

$$U \not\perp\!\!\!\perp V \mid Y$$

$$U \perp\!\!\!\perp V \mid Y, Z$$

# d-separation

Causal Graph



Induced CI

$$U \perp\!\!\!\perp V \mid \emptyset$$

$$U \not\perp\!\!\!\perp V \mid Y$$

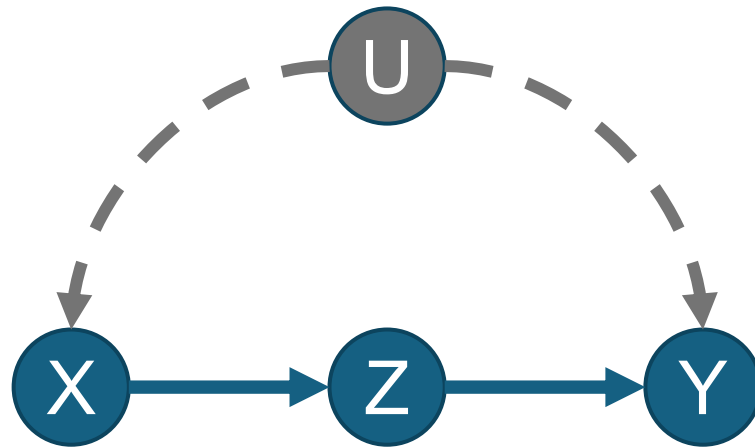
$$U \perp\!\!\!\perp V \mid Y, Z$$

$$U \not\perp\!\!\!\perp V \mid T$$

# Do-Calculus

## Example 1

- Inferring about interventions from observational data.



$$p(z|do(x))$$

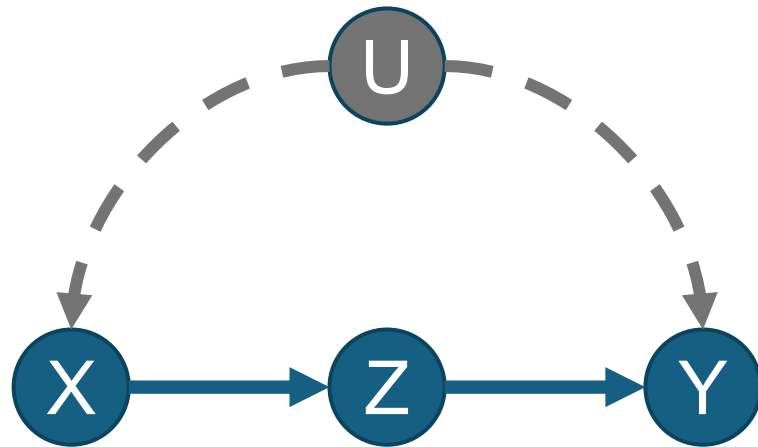
**Conditioning** on X changes posterior of U  
**Intervention** does not

When are the two identical?

# Do-Calculus

## Example 1

- Inferring about interventions from observational data.



$$p(z|do(x))$$

**Conditioning** on X changes posterior of U  
**Intervention** does not

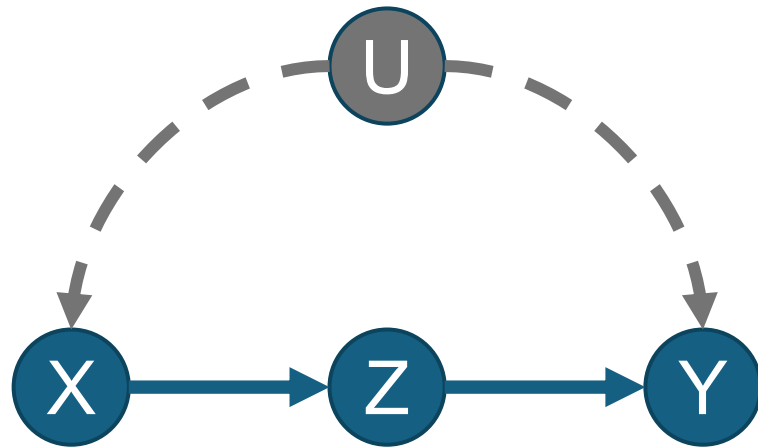
When are the two identical?

If there is **no active *backdoor* path**  
from intervened variable

# Do-Calculus

## Example 1

- Inferring about interventions from observational data.



$$p(z|do(x))$$

**Conditioning** on X changes posterior of U  
**Intervention** does not

When are the two identical?

If there is **no active *backdoor* path**  
from intervened variable

$$(X \perp\!\!\!\perp Z)_{G_{\underline{X}}}$$



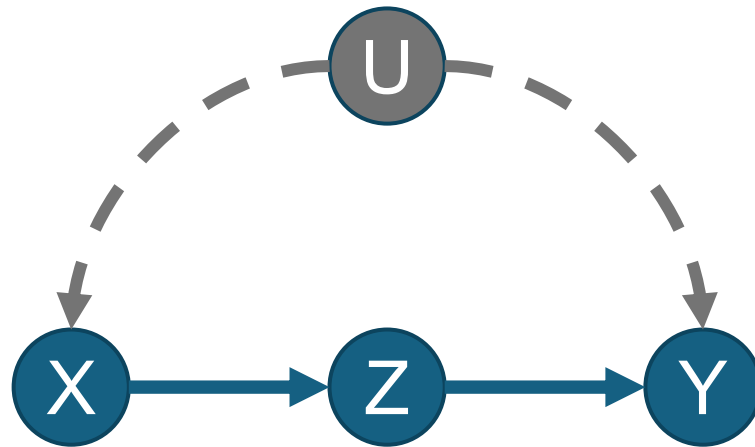
$$p(z|do(x)) = p(z|x)$$



# Do-Calculus

## Example 2

- Inferring about interventions from observational data.



$$p(x|do(z))$$

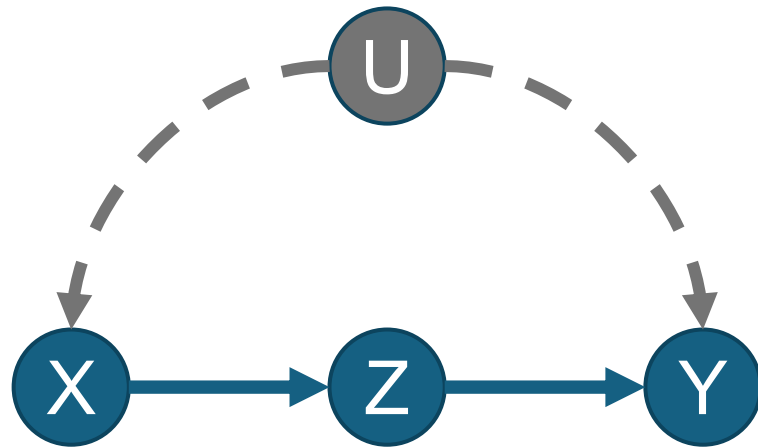
When does an **intervention** have no effect?

If the only connections are **through backdoor paths**.

# Do-Calculus

## Example 2

- Inferring about interventions from observational data.



$$p(x|do(z))$$

When does an **intervention** have no effect?

If the only connections are **through backdoor paths**.

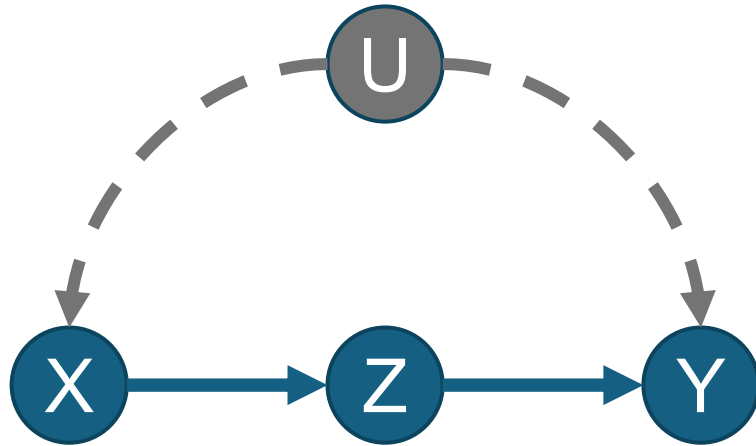
$$(X \perp\!\!\!\perp Z)_{G_{\overline{Z}}}$$

$$\Downarrow$$

$$p(x|do(z)) = p(x)$$

# Do-Calculus

## Example 3



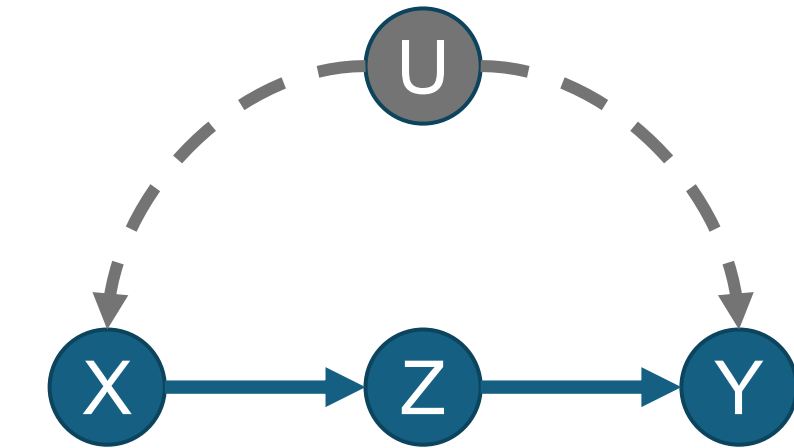
$$p(y|do(z))$$

$$p(y|do(z)) \neq p(y)$$

$$p(y|do(z)) \neq p(y|z)$$

# Do-Calculus

## Example 3



$$p(y|do(z))$$

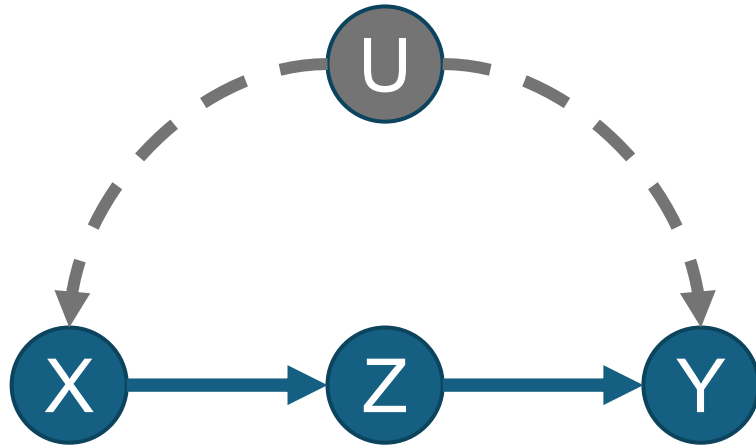
$$p(y|do(z)) \neq p(y)$$

$$p(y|do(z)) \neq p(y|z)$$

Block the path to Y by **conditioning on X!**  
How?

# Do-Calculus

## Example 3



$$p(y|do(z)) \neq p(y)$$

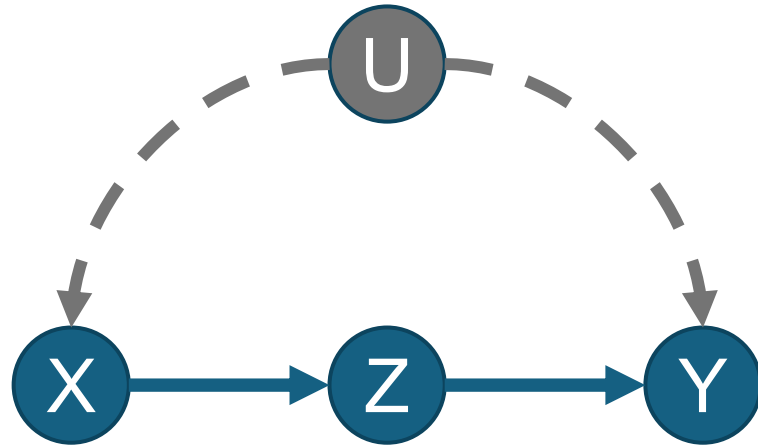
$$p(y|do(z)) \neq p(y|z)$$

Block the path to Y by **conditioning on X!**  
How?

$$p(y|do(z)) = \sum_x p(y|x, do(z))p(x|do(z))$$

# Do-Calculus

## Example 3



$$p(y|do(z)) \neq p(y)$$

$$p(y|do(z)) \neq p(y|z)$$

Block the path to Y by **conditioning on X!**  
How?

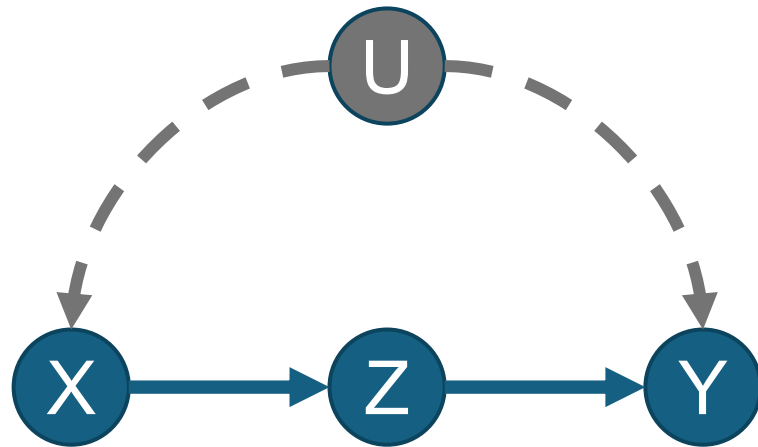
$$p(y|do(z)) = \sum_x p(y|x, do(z))p(x|do(z))$$

$$p(y|x, do(z)) = p(y|x, z)$$

$$p(x|do(z)) = p(x)$$

# Do-Calculus

## Example 3



$$p(y|do(z)) \neq p(y)$$

$$p(y|do(z)) \neq p(y|z)$$

Block the path to Y by **conditioning on X!**  
How?

$$p(y|do(z)) = \sum_x p(y|x, do(z))p(x|do(z))$$

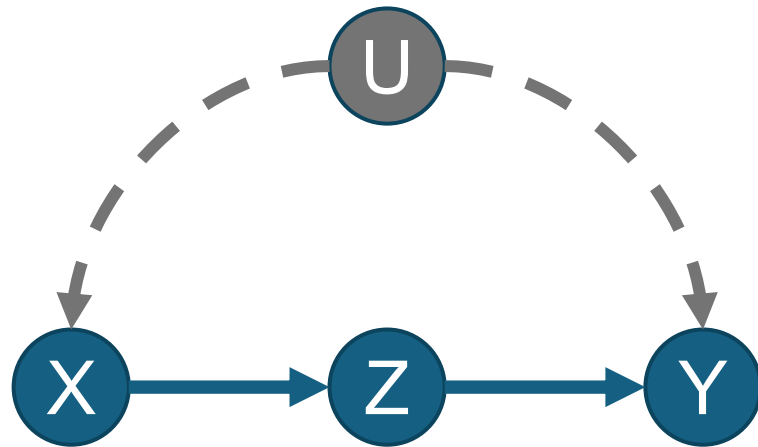
$$p(y|x, do(z)) = p(y|x, z)$$

$$p(x|do(z)) = p(x)$$

$$\Rightarrow p(y|do(z)) = \sum_x p(y|x, z)p(x)$$

# Do-Calculus

## Example 3



$$p(y|do(z)) \neq p(y)$$

$$p(y|do(z)) \neq p(y|z)$$

Block the path to Y by **conditioning on X!**  
How?

$$p(y|do(z)) = \sum_x p(y|x, do(z))p(x|do(z))$$

$$p(y|x, do(z)) = p(y|x, z)$$

$$p(x|do(z)) = p(x)$$

$$\Rightarrow p(y|do(z)) = \sum_x p(y|x, z)p(x)$$

This is called  
**Backdoor Adjustment**



# Rules of Do-Calculus [Pearl'95]

*Rule 1 (insertion/deletion of observations):*

$$\text{pr}(y|\check{x}, z, w) = \text{pr}(y|\check{x}, w) \quad \text{if } (Y \perp\!\!\!\perp Z | X, W)_{G_{\check{X}}}.$$

*Rule 2 (action/observation exchange):*

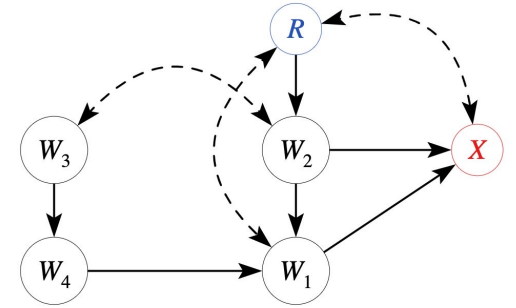
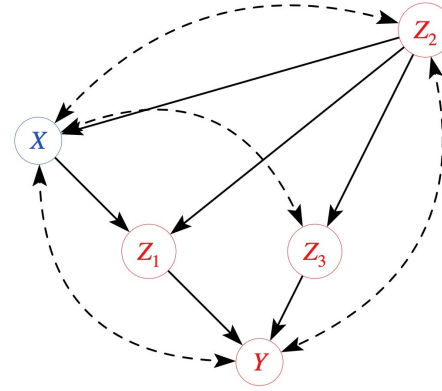
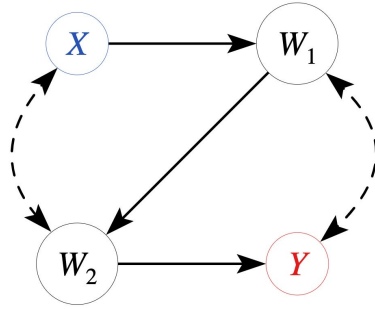
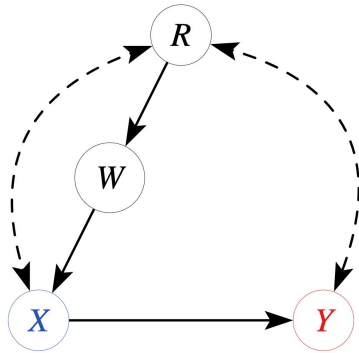
$$\text{pr}(y|\check{x}, \check{z}, w) = \text{pr}(y|\check{x}, z, w) \quad \text{if } (Y \perp\!\!\!\perp Z | X, W)_{G_{\check{X}\check{Z}}}.$$

*Rule 3 (insertion/deletion of actions):*

$$\text{pr}(y|\check{x}, \check{z}, w) = \text{pr}(y|\check{x}, w) \quad \text{if } (Y \perp\!\!\!\perp Z | X, W)_{G_{\check{X}, \overline{\check{X}(w)}}},$$

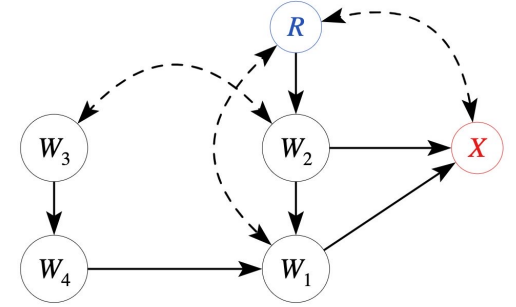
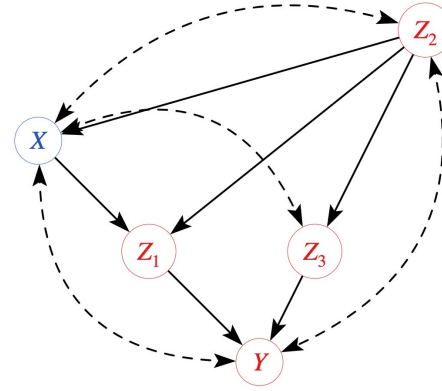
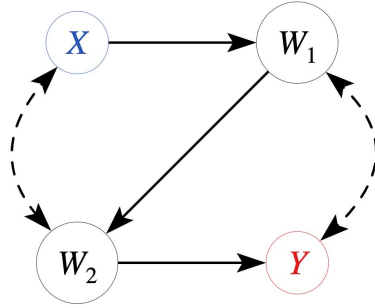
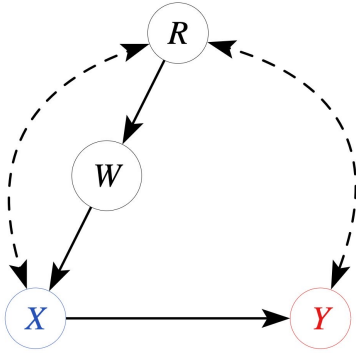
# From Observational Data

*To adjust or not to adjust?*



# From Observational Data

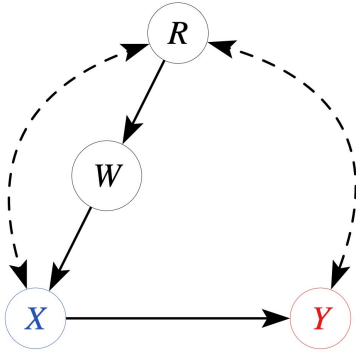
*To adjust or not to adjust?*



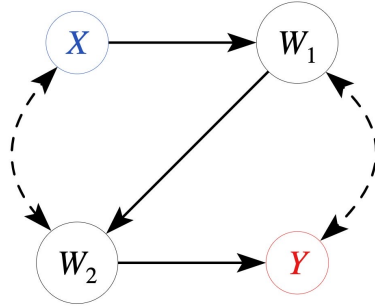
$$p_x(y) = \frac{\sum_r p(x, y|r, w)p(r)}{\sum_r p(x|r, w)p(r)}$$

# From Observational Data

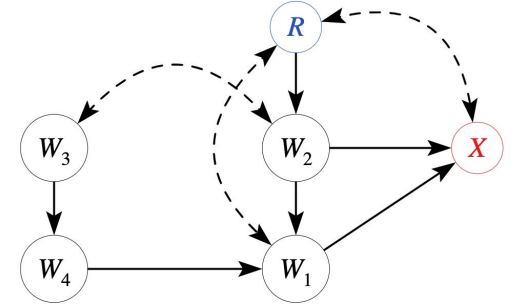
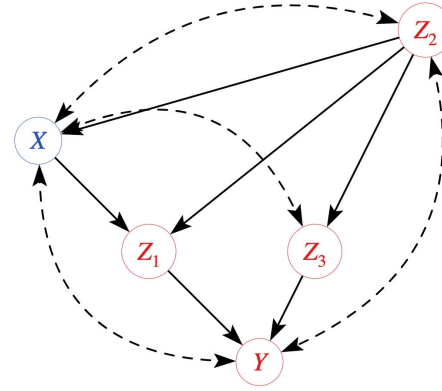
*To adjust or not to adjust?*



$$p_x(y) = \frac{\sum_r p(x, y|r, w)p(r)}{\sum_r p(x|r, w)p(r)}$$

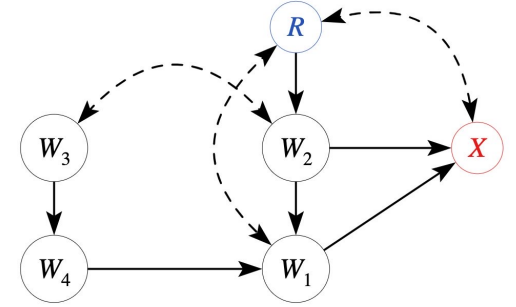
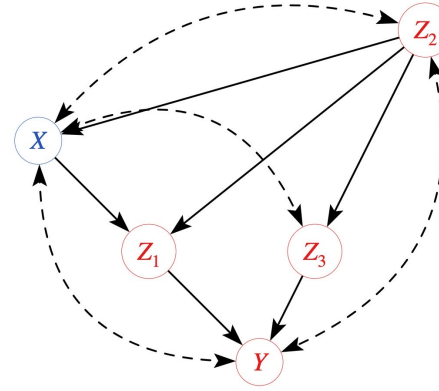
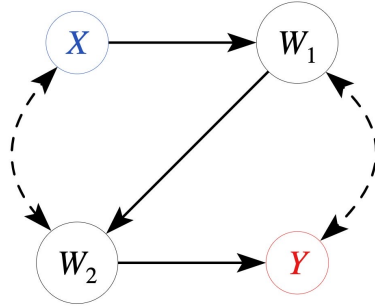
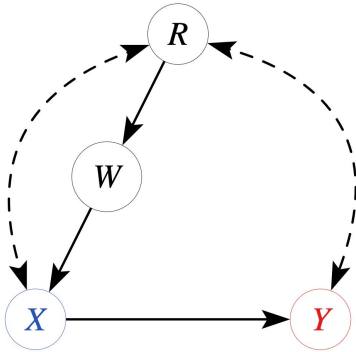


$$p_x(y) = \sum_{w_1, w_2} p(y|w_1, w_2, x)p(w_1|x) \sum_{x'} p(w_2|w_1, x')p(x')$$



# From Observational Data

## *To adjust or not to adjust?*



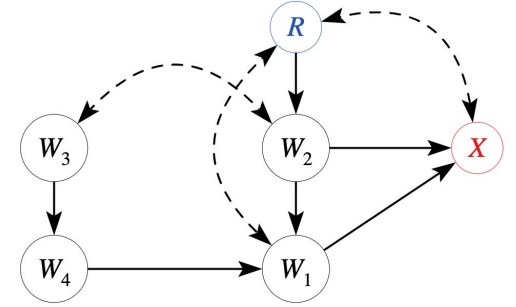
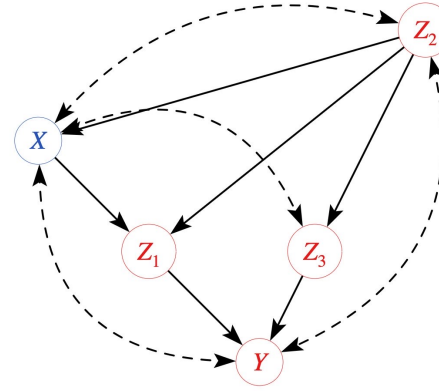
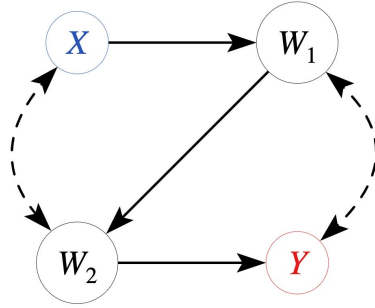
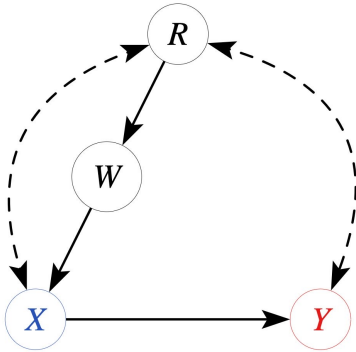
$$p_x(y) = \frac{\sum_r p(x, y|r, w)p(r)}{\sum_r p(x|r, w)p(r)}$$

$$p_x(y) = \sum_{w_1, w_2} p(y|w_1, w_2, x)p(w_1|x) \sum_{x'} p(w_2|w_1, x')p(x')$$

$$p_x(z_1, z_2, z_3, y) = p(z_3|z_2)p(z_1|x, z_2) \frac{\sum_{x'} p(y, z_3|x', z_1, z_2)p(x', z_2)}{\sum_{x'} p(z_3|x', z_1, z_2)p(x', z_2)} p(z_2)$$

# From Observational Data

## *To adjust or not to adjust?*



$$p_x(y) = \frac{\sum_r p(x, y|r, w)p(r)}{\sum_r p(x|r, w)p(r)}$$

$$p_x(y) = \sum_{w_1, w_2} p(y|w_1, w_2, x)p(w_1|x) \sum_{x'} p(w_2|w_1, x')p(x')$$

$$p_x(z_1, z_2, z_3, y) = p(z_3|z_2)p(z_1|x, z_2) \frac{\sum_{x'} p(y, z_3|x', z_1, z_2)p(x', z_2)}{\sum_{x'} p(z_3|x', z_1, z_2)p(x', z_2)} p(z_2)$$

$$p_r(x) = \sum_{w_1, w_2, w_3, w_4} \frac{\sum_{r'} p(w_1|r', w_2, w_3, w_4)p(r'|w_3, w_4)}{\sum_{r'} p(w_1, x|r', w_2, w_3, w_4)p(r'|w_3, w_4)} p(w_4|w_3)p(w_2, w_3|r) \sum_{r'} p(w_1|r', w_2, w_3, w_4)p(r'|w_3, w_4)$$

**No Estimation w/out Identification!**

# Different Frameworks

## Pearl's SCMs

1. **Assume a set of** (unknown) **structural equations** interpreted as assignment operators.
2. **Define intervention** on these equations.
3. Understand the formula that transforms observational distribution to interventional distribution (**estimand**).

## Rubin-Neyman PO

1. Start by **defining counterfactual variables**.
2. **Assume (conditional) independence** between counterfactual variables.
3. Develop **estimators** that give causal effect under these independence assumptions.

# Causal Inference with Deep Learning and Generative Models

## *Outline*

- Background
  - Causal Inference Basics
  - Neural Network Basics
- A Taxonomy of Deep Learning Approaches for Causal Inference
  - Function Modeling (a.k.a. Curve Fitting)
  - Feature Extraction
  - Generative Causal Inference



# Neural Networks

# Muffin vs. Chihuahua

Learn a mapping that outputs class labels.

Given the image, decide if

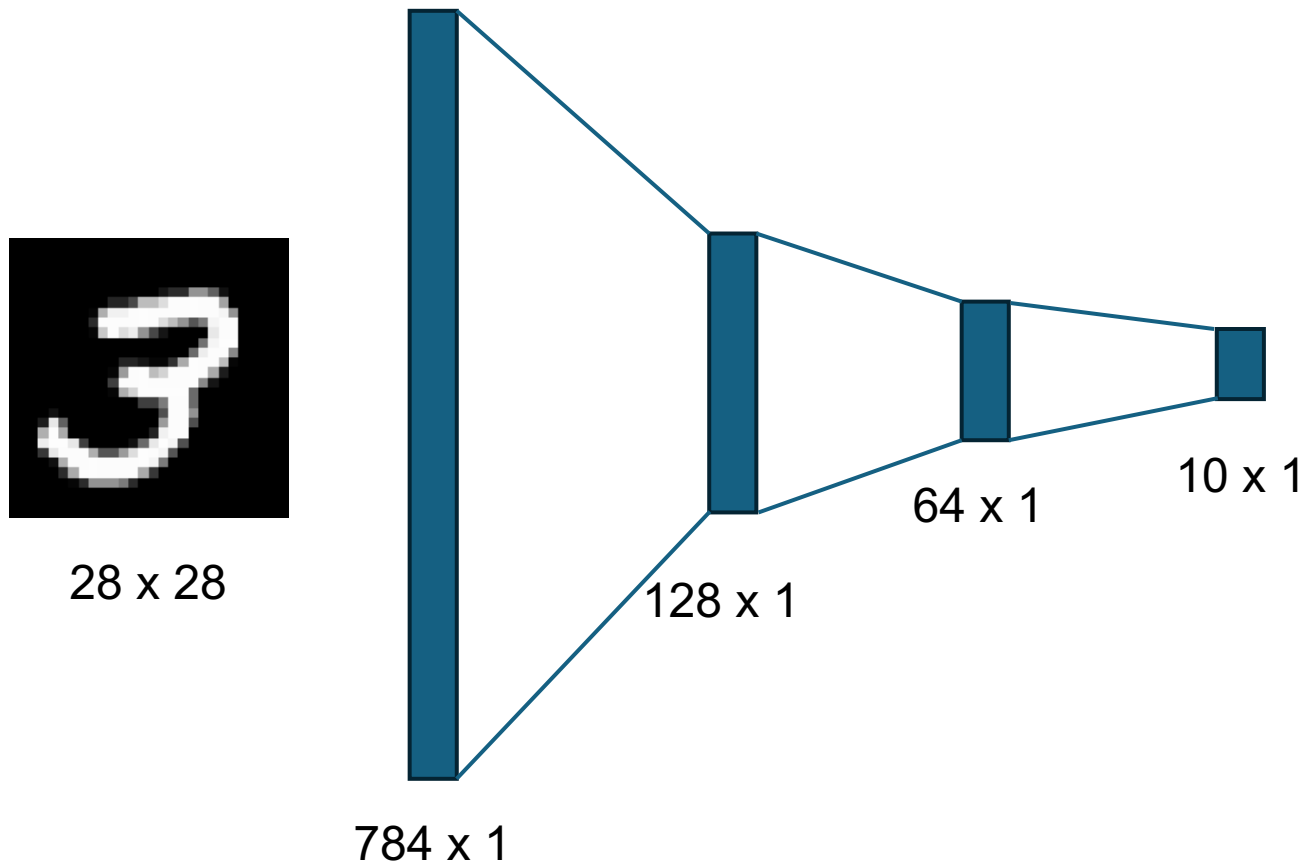
0 - Muffin

1 - Chihuahua



# Multi Layer Perceptron for Image Classification

A sample architecture:

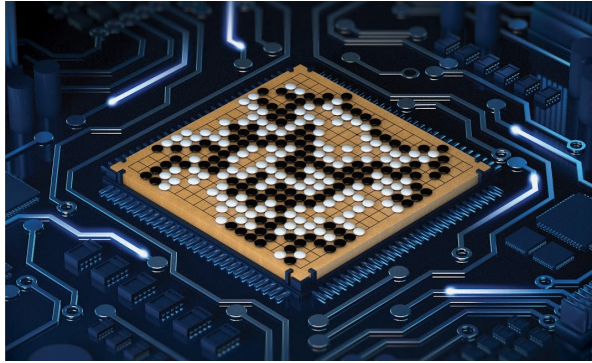


```
class MNISTClassifier(nn.Module):
    def __init__(self):
        super().__init__()
        # Input: 28x28 = 784
        # Hidden layers: 128, 64
        # Output: 10 classes
        self.flatten = nn.Flatten()
        self.fc1 = nn.Linear(784, 128)
        self.fc2 = nn.Linear(128, 64)
        self.fc3 = nn.Linear(64, 10)

        # He initialization for ReLU
        nn.init.kaiming_normal_(self.fc1.weight)
        nn.init.kaiming_normal_(self.fc2.weight)
        nn.init.kaiming_normal_(self.fc3.weight)

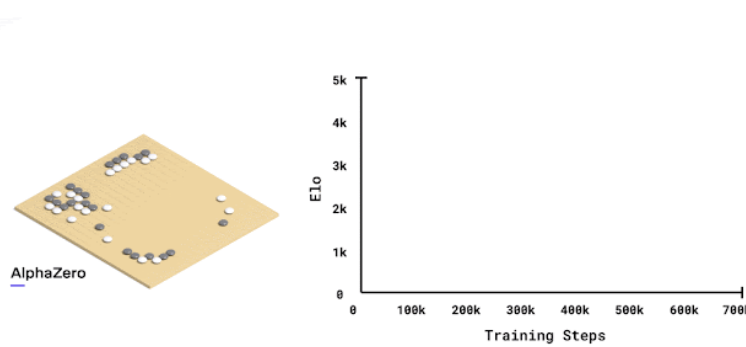
    def forward(self, x):
        x = self.flatten(x)
        x = torch.relu(self.fc1(x))
        x = torch.relu(self.fc2(x))
        x = self.fc3(x)
        return x
```

# Modern Success Stories of Neural Nets



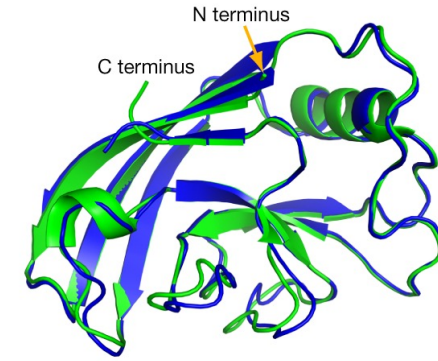
AlphaGo, 2015

*Deep RL + Search  
Self-play*



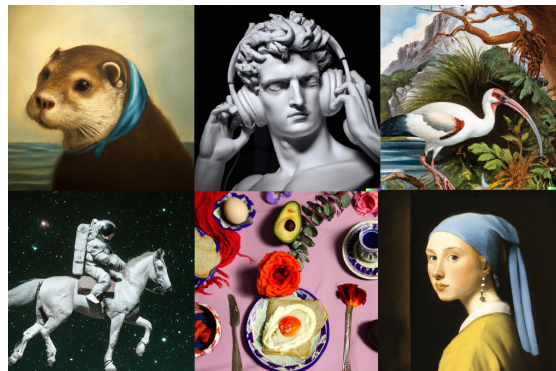
AlphaZero, 2017

*Only self-play  
No access to  
opening books, endgames.*



AlphaFold, 2020

*Uses Attention*



DALL-E, 2021

*Transformers, Diffusion*



ChatGPT, 2022

*Transformers  
Supervised Learning, RLHF  
Search*



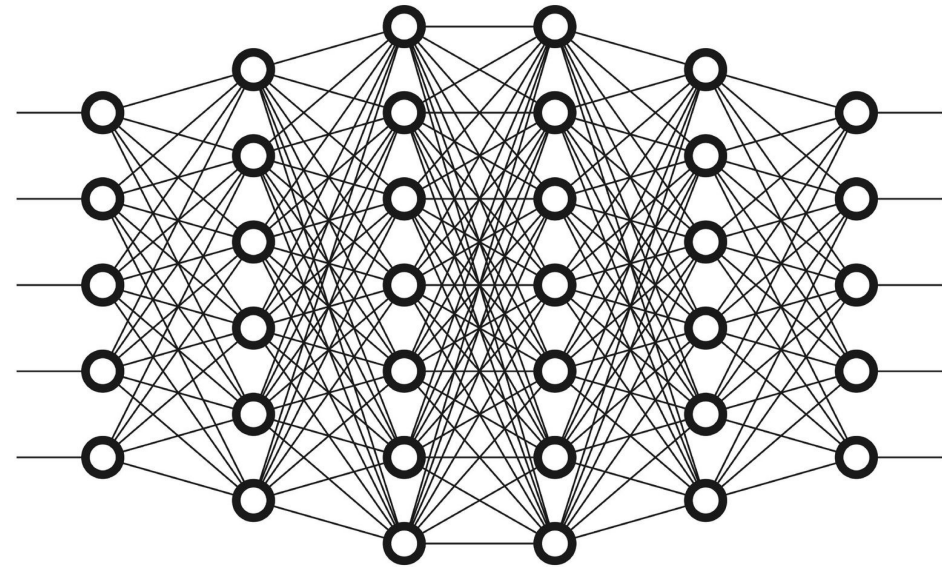
SORA, 2024

*Diffusion*

# Neural Network Basics

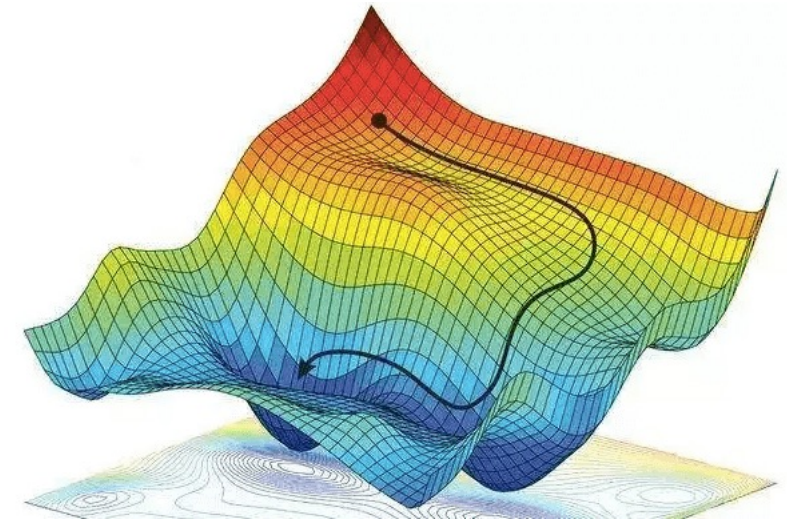
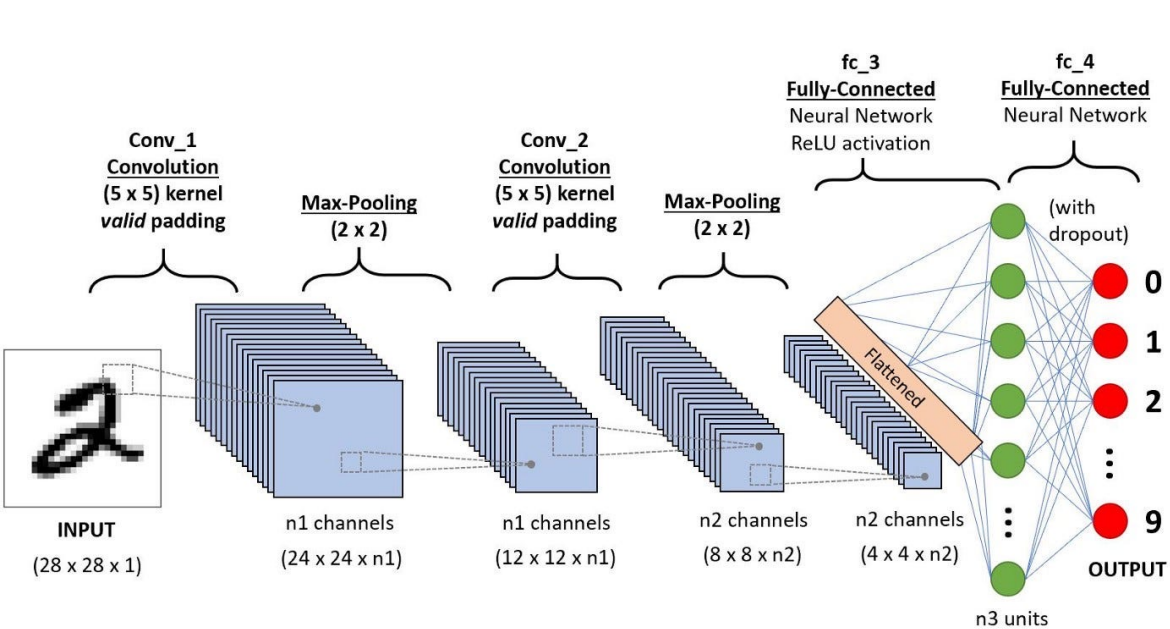
## *Deep Neural Networks*

- Efficiently computable
- Easy to train
  - Differentiable



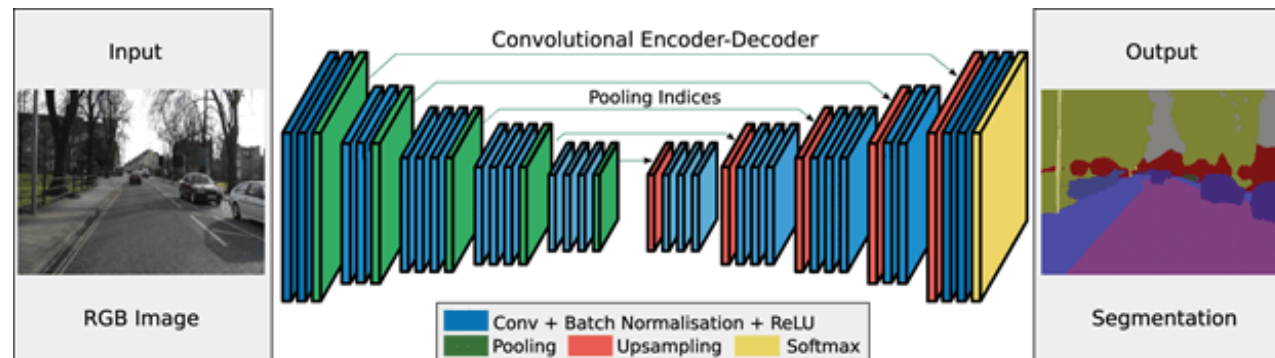


# Neural Net Architectures Fit Complex Functions to Data aka Curve-fitting



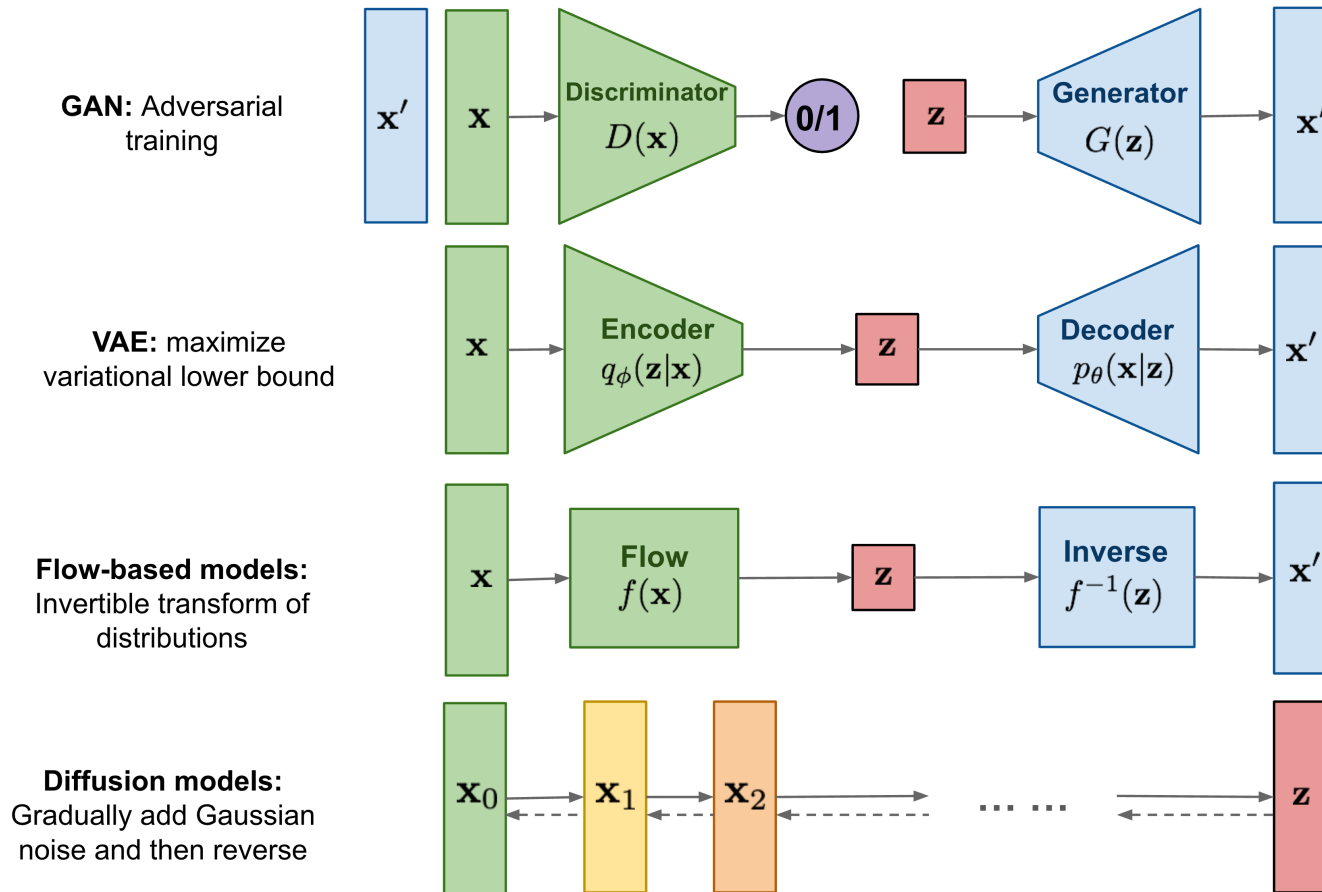
<https://easyai.tech/en/ai-definition/gradient-descent/>

<https://medium.com/@arieljumba/deep-learning-task-image-classification-with-convolutional-neural-networks-cnns-ddd061b6e84b>



Vijay Badrinarayanan et. al 2017 "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation"

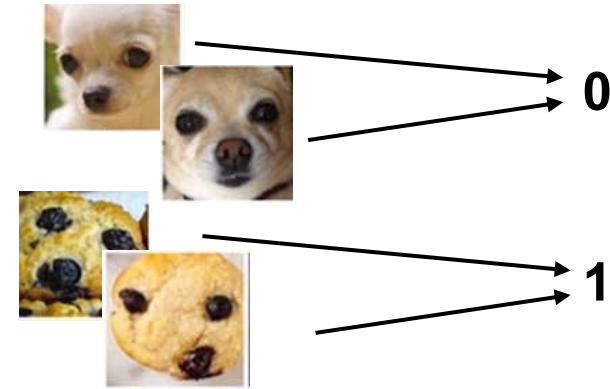
# Modern NNs Can Learn to Sample from Complex Data Distributions



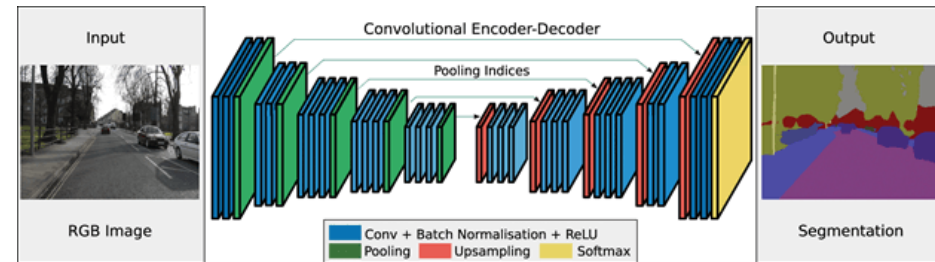
Karras et al. A Style-Based Generator Architecture for Generative Adversarial Networks CVPR 2019.

# What can we wish from NNs for causality?

- Fit really complicated functions.



- Retrieve semantically meaningful latent features.



Vijay Badrinarayanan et. al 2017 "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation"

- Fit really complicated distributions.



Karras et al. A Style-Based Generator Architecture for Generative Adversarial Networks CVPR 2019.

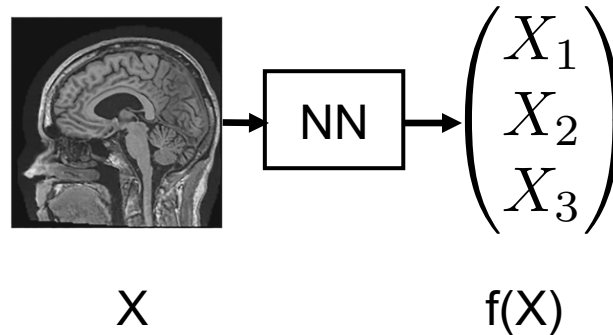


# A Taxonomy of Deep Learning Approaches for Causal Inference

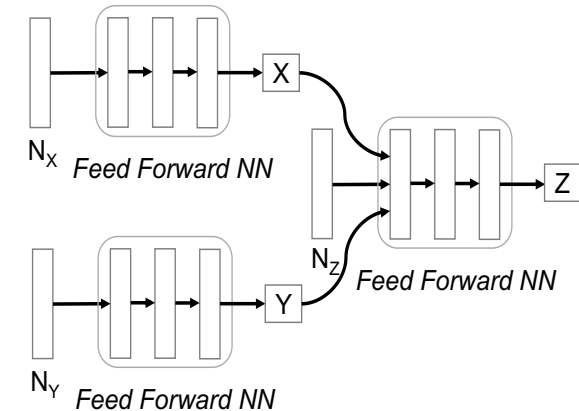
## Function Modeling

$$f : \mathbb{R}^k \rightarrow [0, 1]$$

## Feature Extraction



## Generative Modeling



# Causal Inference with Deep Learning and Generative Models

## *Outline*

- Background
  - Causal Inference Basics
  - Neural Network Basics
- A Taxonomy of Deep Learning Approaches for Causal Inference
  - Function Modeling (a.k.a. Curve Fitting)
  - Feature Extraction
  - Generative Causal Inference

# A Taxonomy of Deep Learning Approaches for Causal Inference

Function Modeling

$$f : \mathbb{R}^k \rightarrow [0, 1]$$

Feature Extraction

Generative Modeling

# Function Modeling (a.k.a. Curve Fitting)

- We can talk about many functions that help evaluate causal effect.
- We can use neural networks to model them.
- The differentiable loss allows us to “nudge” the learnt function to have certain properties that we desire.

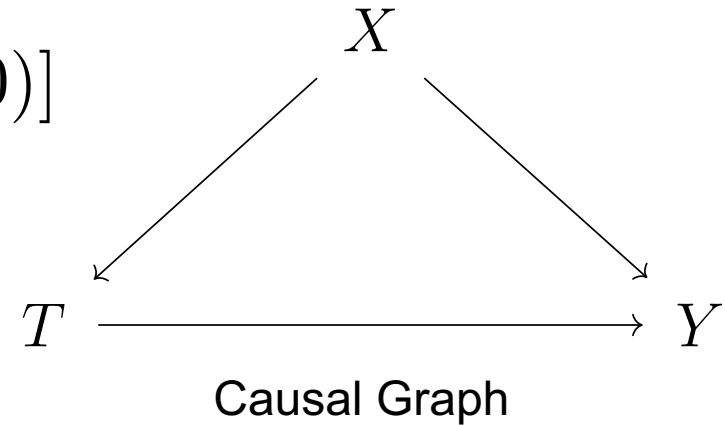
# Average Treatment Effect

- Goal is to estimate

$$ATE = \mathbb{E}[Y|do(T = 1)] - \mathbb{E}[Y|do(T = 0)]$$

$Y_1$

$Y_0$

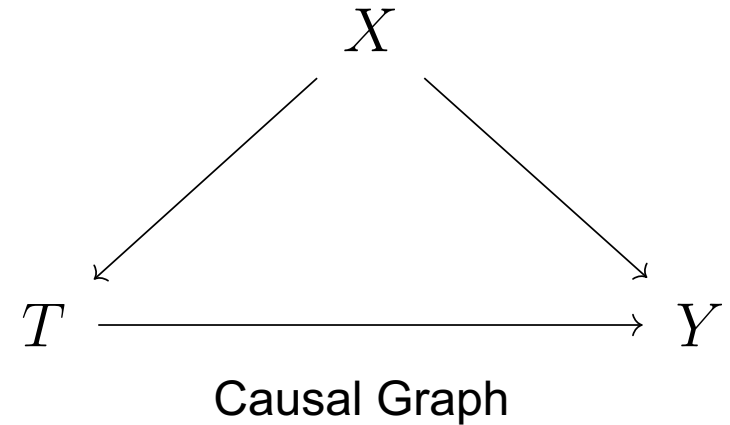


# Direct Modeling of Counterfactuals in PO

## *Learning Representations for Counterfactual Inference*

Johansson, Shalit, Sontag

- $Y_0(x)$  vs.  $Y_1(x)$  are counterfactual variables.

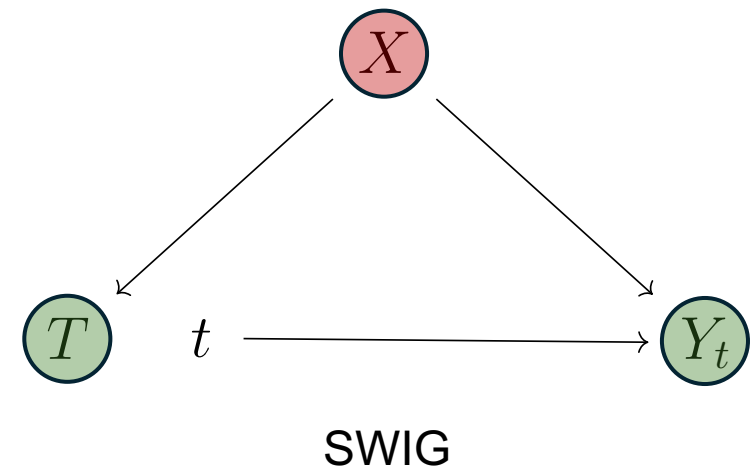
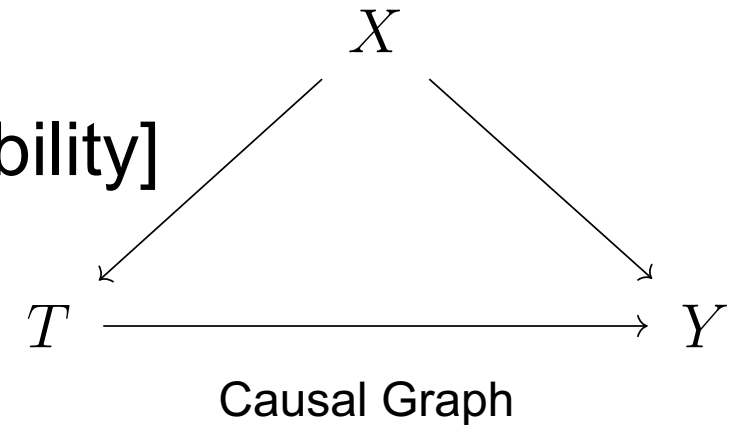


# Direct Modeling of Counterfactuals in PO

## *Learning Representations for Counterfactual Inference*

Johansson, Shalit, Sontag

- $Y_0(x)$  vs.  $Y_1(x)$  are counterfactual variables.
- $Y_t$  is independent from  $T$  given  $X$ . [ignorability]

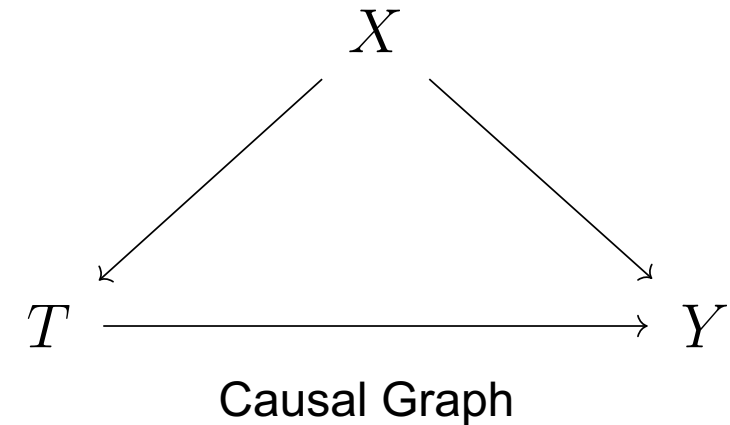


# Direct Modeling of Counterfactuals in PO

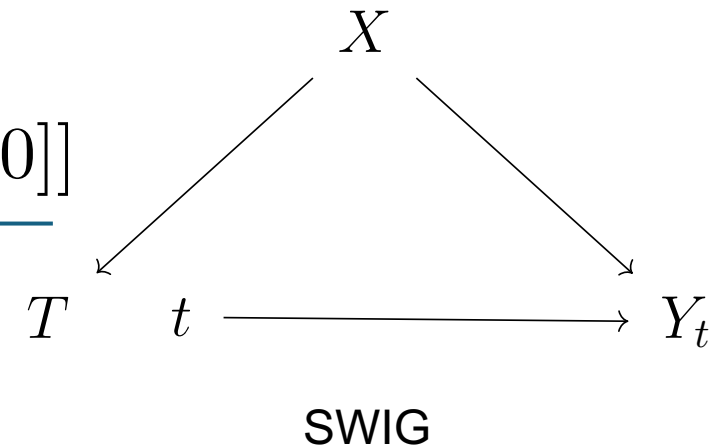
## *Learning Representations for Counterfactual Inference*

Johansson, Shalit, Sontag

- $Y_0(x)$  vs.  $Y_1(x)$  are counterfactual variables.
- $Y_t$  is independent from  $T$  given  $X$ .
- Adjustment with  $X$  is sufficient.



$$\begin{aligned} \text{ATE} &= \mathbb{E}[Y | do(T = 1)] - \mathbb{E}[Y | do(T = 0)] \\ &= \underline{\mathbb{E}[\mathbb{E}[Y | X, T = 1]]} - \underline{\mathbb{E}[\mathbb{E}[Y | X, T = 0]]} \end{aligned}$$





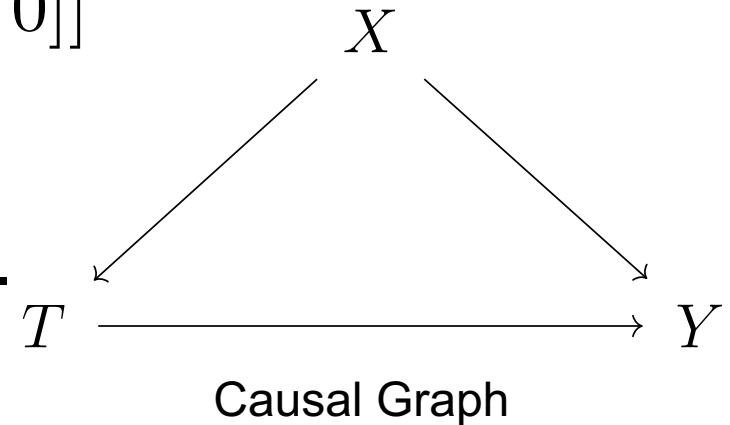
# Direct Modeling of Counterfactuals in PO

## *Learning Representations for Counterfactual Inference*

Johansson, Shalit, Sontag

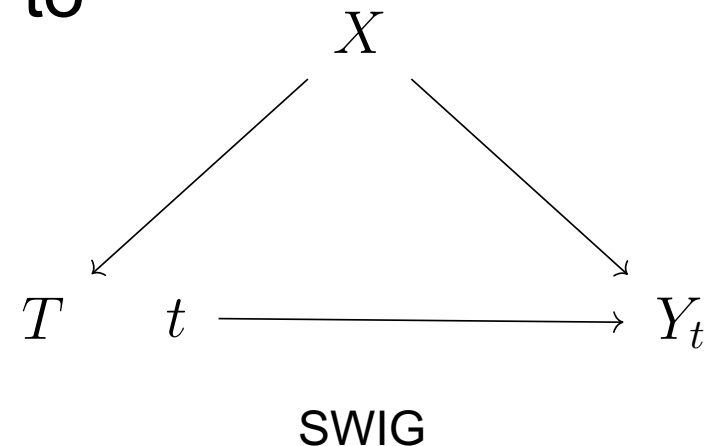
$$\text{ATE} = \mathbb{E}[\mathbb{E}[Y|X, T = 1]] - \mathbb{E}[\mathbb{E}[Y|X, T = 0]]$$

This may be biased if overlap is not strong.

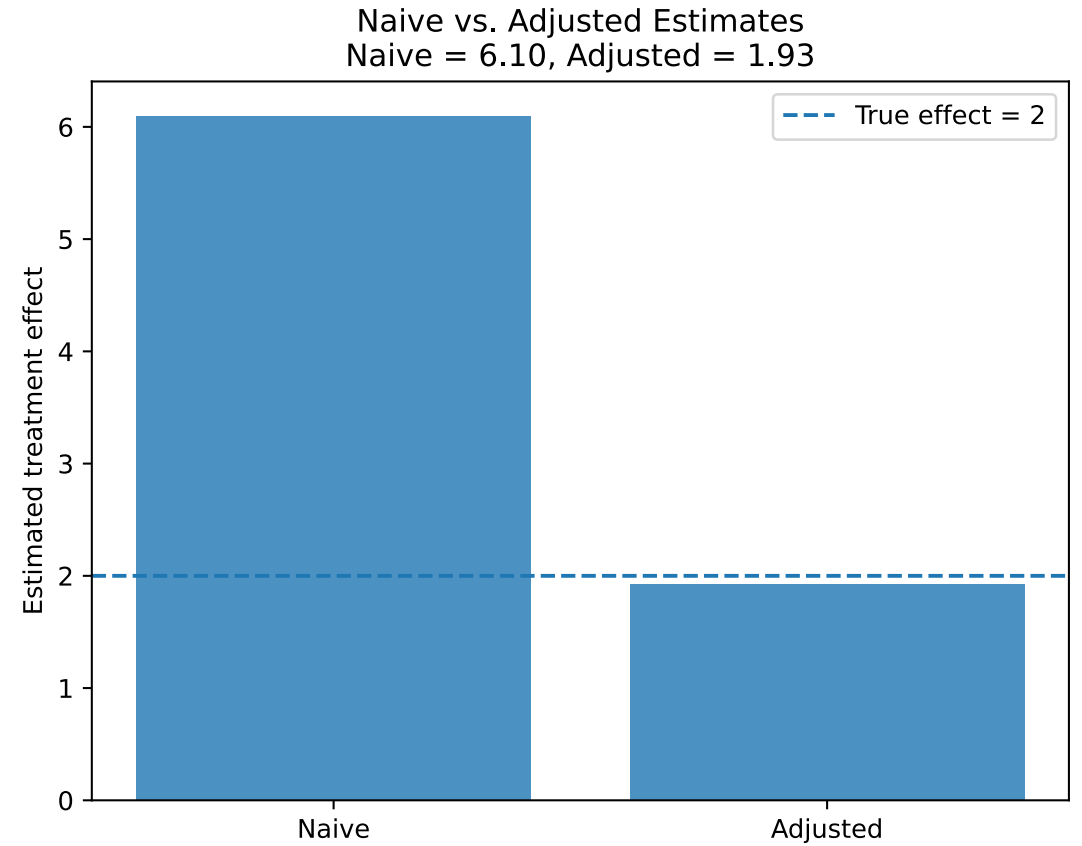
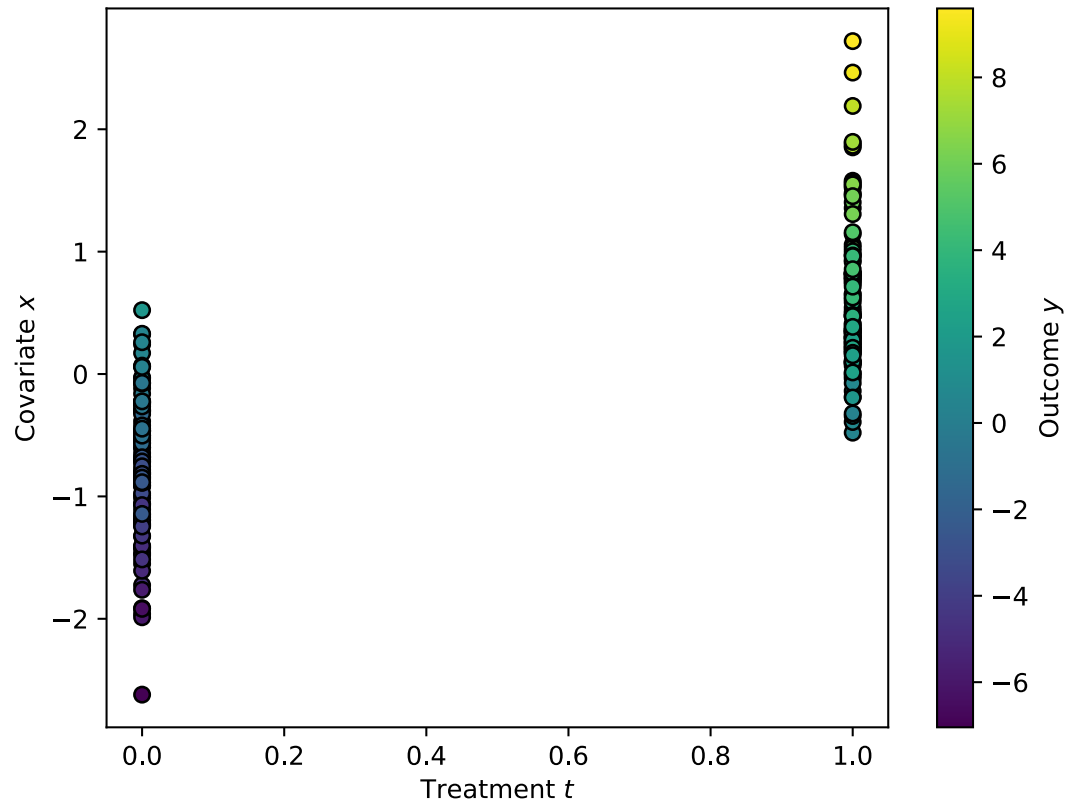


High-dimensional confounder may be hard to condition on.

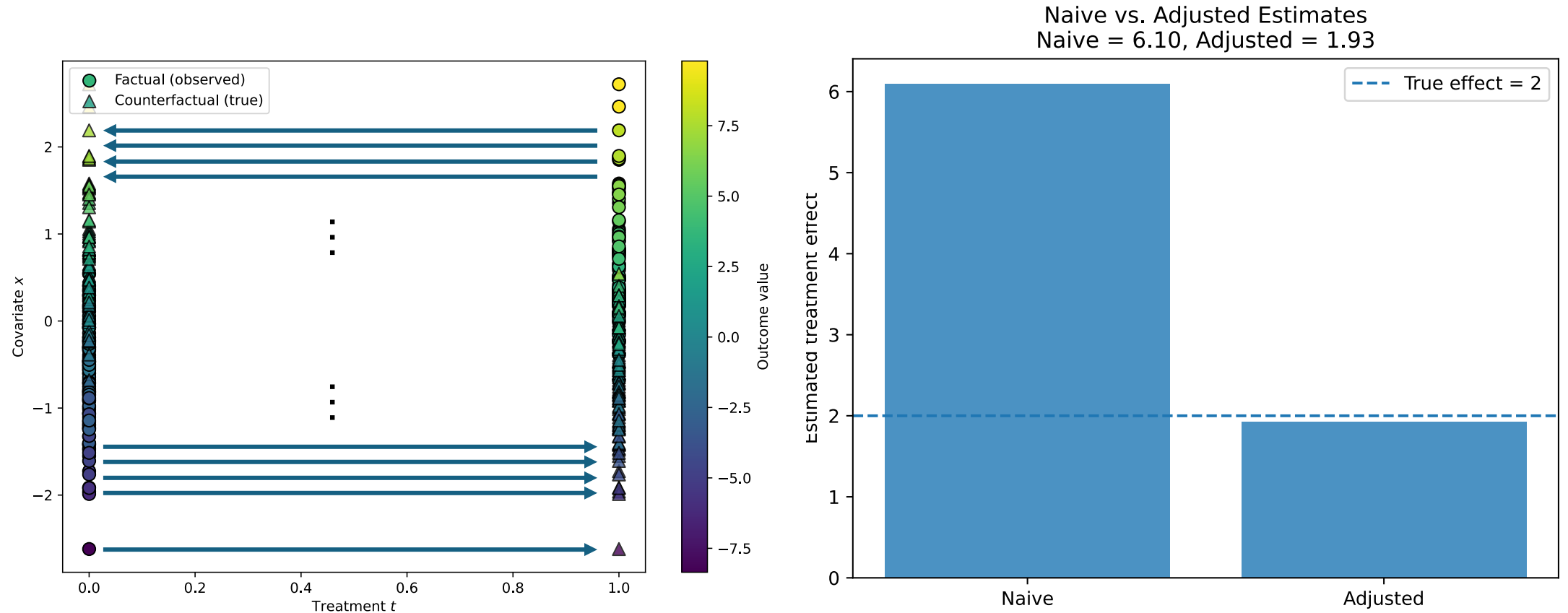
Idea: Learn a *representation* of  $X$  that can be used for adjustment.



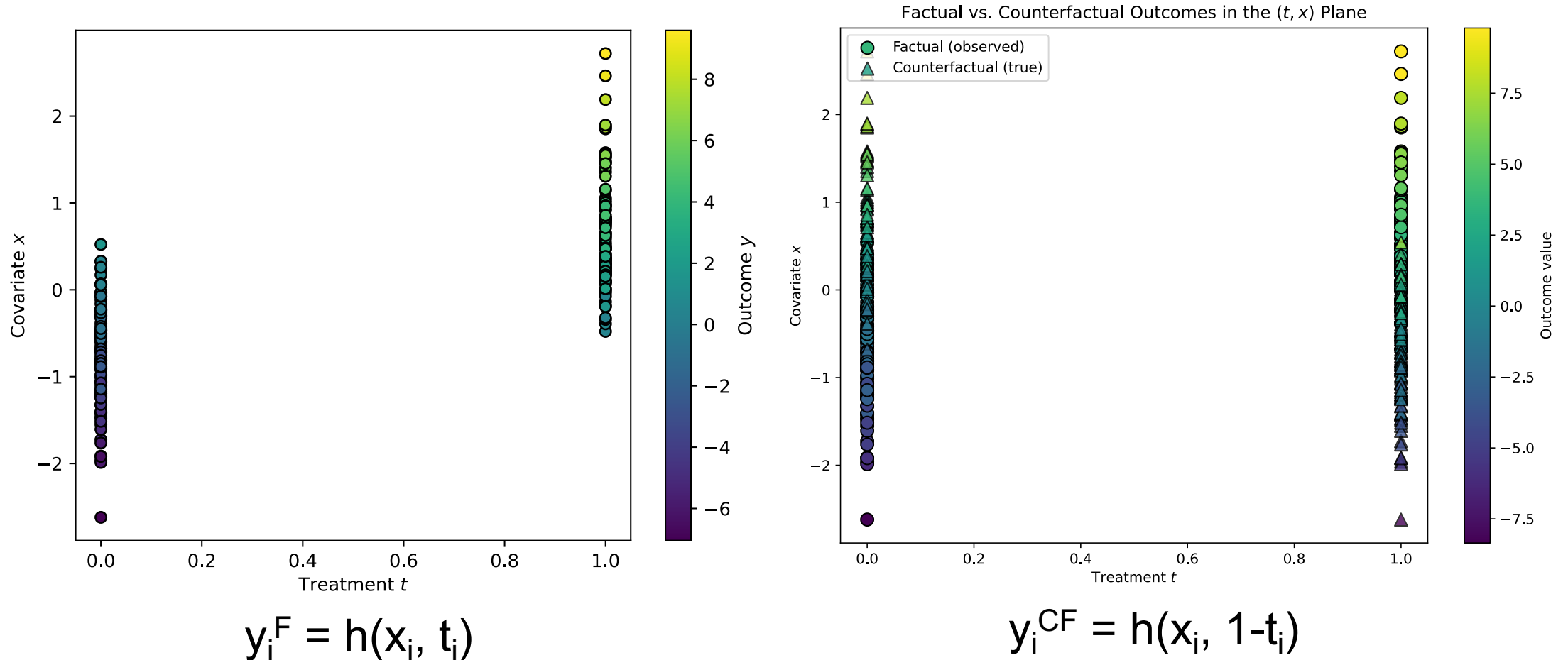
# Function Modeling Perspective for Causal Inference



# Function Modeling Perspective for Causal Inference

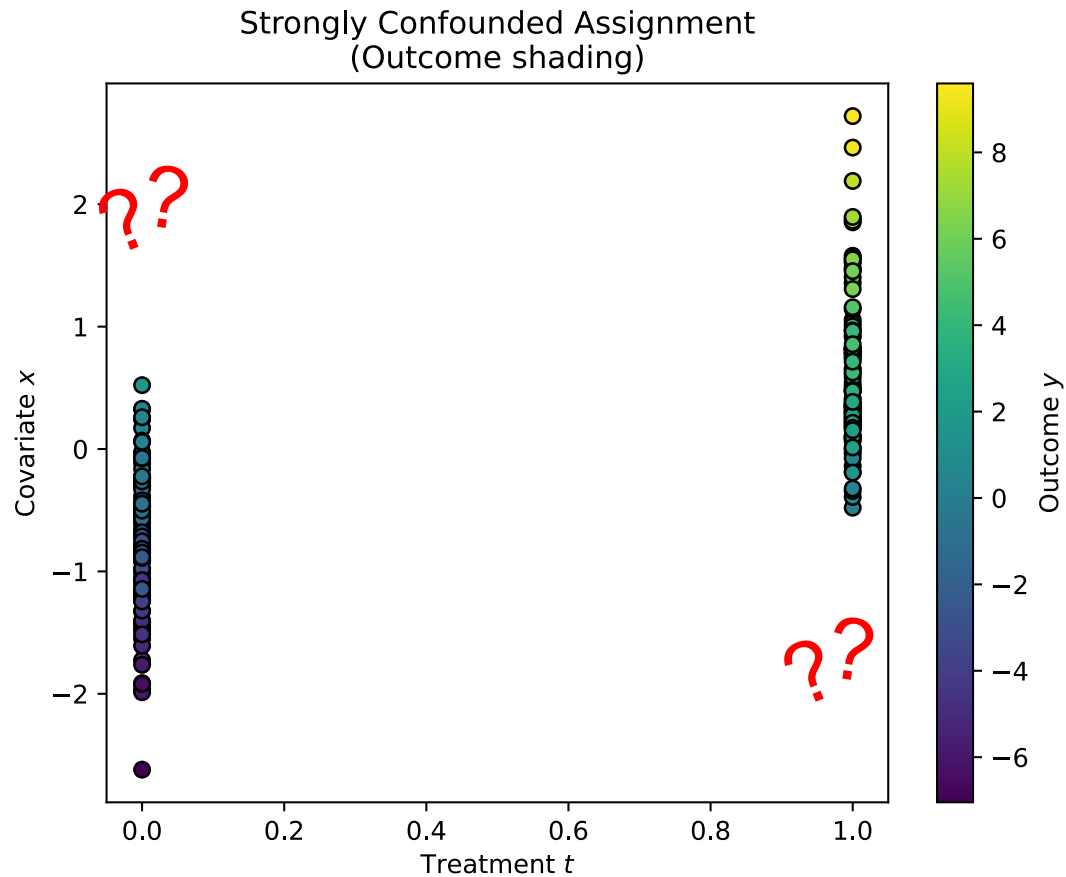


# Function Modeling Perspective for Causal Inference

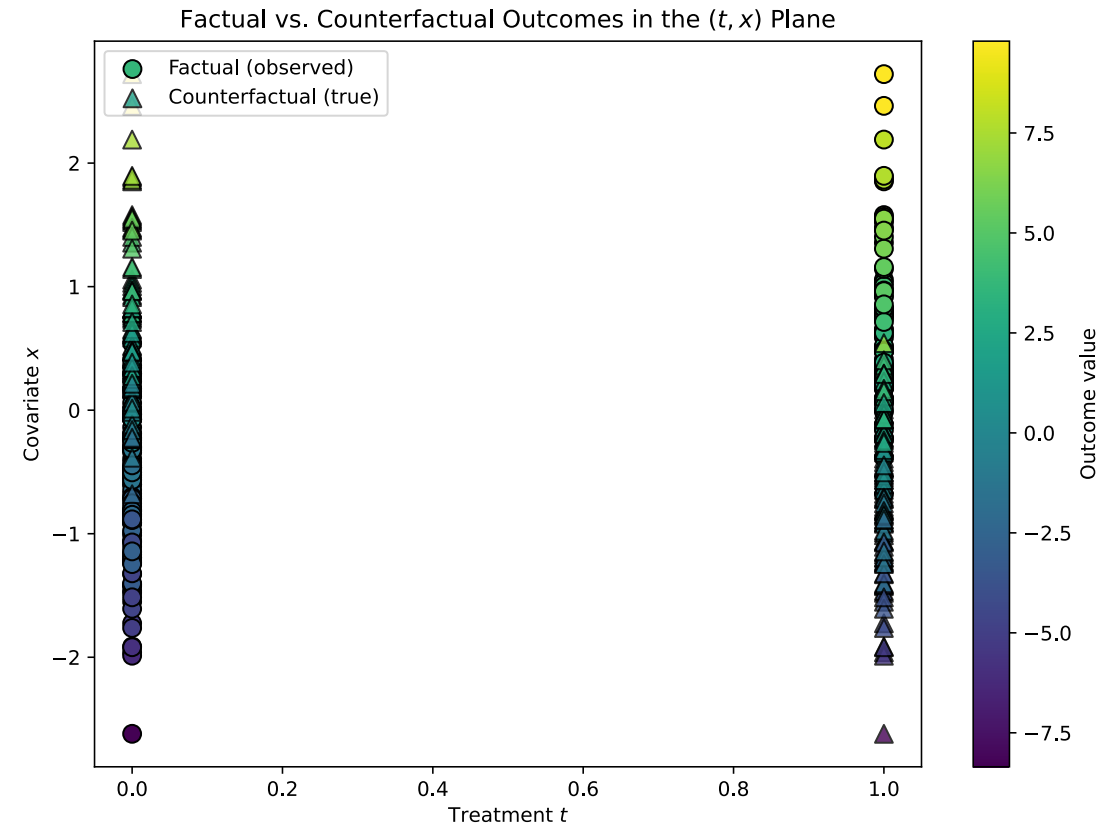


If we can model function  $h$ , we can obtain counterfactual pairs and use it for ITE, ATE.

# Function Modeling Perspective for Causal Inference



$$y_i^F = h(x_i, t_i)$$



$$y_i^{CF} = h(x_i, 1-t_i)$$

# Direct Modeling of Counterfactuals in PO

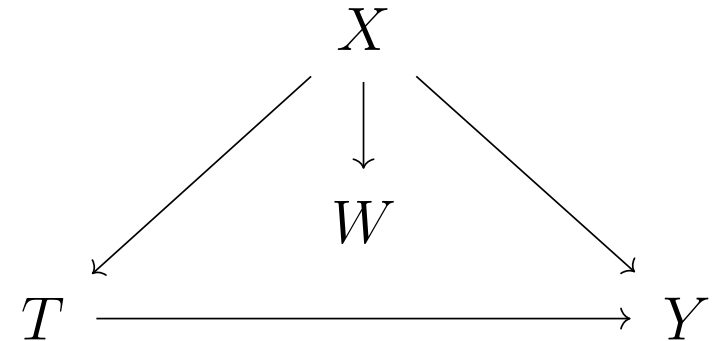
## *Learning Representations for Counterfactual Inference*

Johansson, Shalit, Sontag

- We need to fill in missing values “smartly”.
- Distill confounder so it is balanced across  $T=0$ ,  $T=1$ .
- Easier to learn a function  $h$  on a balanced representation.
- Use deep learning to obtain a representation with these objectives.
- But why would this even work? Or when should it work?

# Adjustment with Representation

- Learnt representation is a child of confounder  $X$  in the *true causal graph*.
- $W$  is **not** a valid adjustment set!
- Maybe under some assumptions?



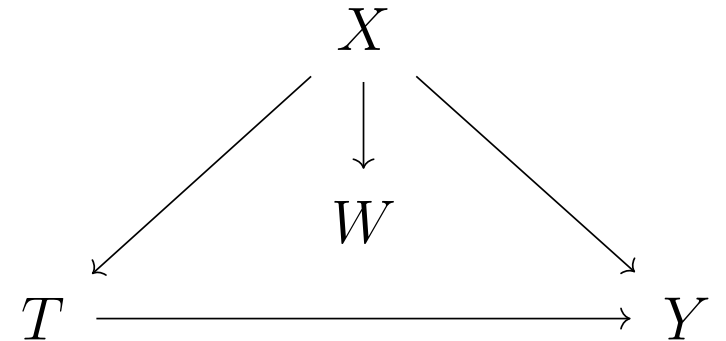
# A Good Representation

- Balanced:  $p(W|T=0) = p(W|T=1)$

$$W \perp\!\!\!\perp T$$

- Informative:  $p(Y|W, X=x) = p(Y|W, X=x')$

$$Y \perp\!\!\!\perp X \mid W$$



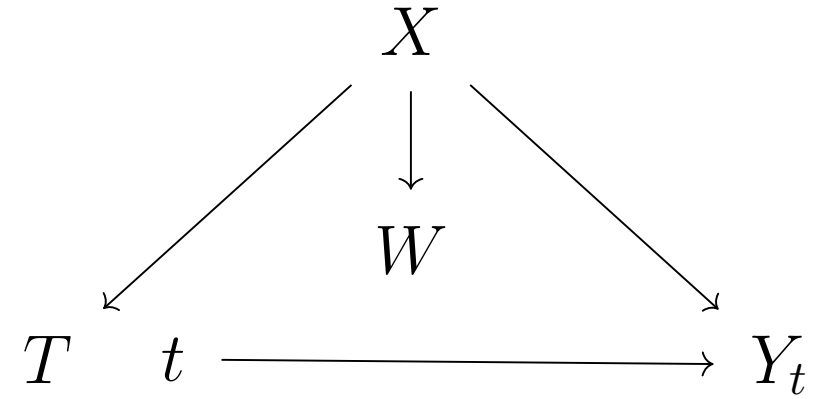
This removes trivial representations, such as a constant.

Are these enough for  $W$  to be a valid adjustment?



# Recall the adjustment condition

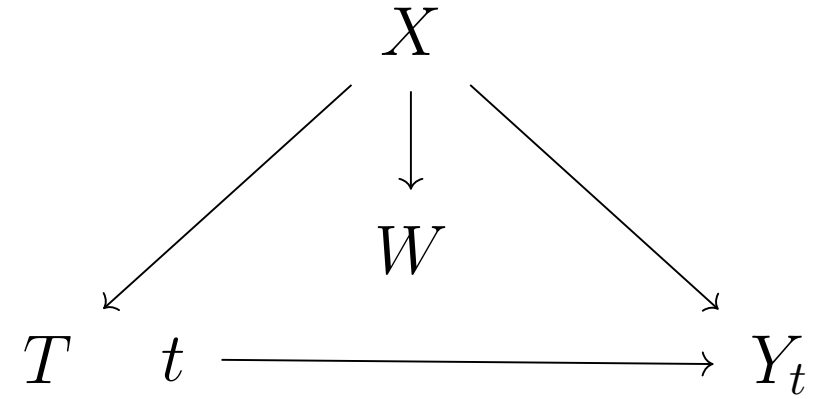
$$Y_t \perp\!\!\!\perp T \mid X$$



# Recall the adjustment condition

$$Y_t \perp\!\!\!\perp T \mid X$$

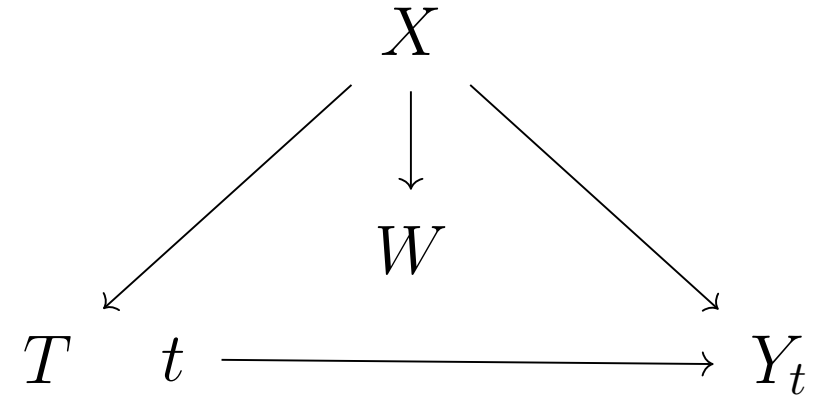
$$p(Y_t) = \sum_{x, t'} p(Y_t \mid X = x, T = t') p(X = x, T = t')$$



# Recall the adjustment condition

$$Y_t \perp\!\!\!\perp T \mid X$$

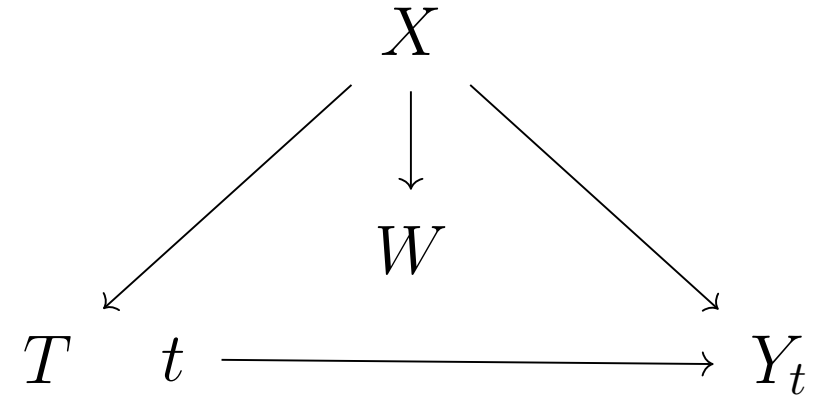
$$\begin{aligned} p(Y_t) &= \sum_{x,t'} \underline{p(Y_t \mid X = x, T = t')} p(X = x, T = t') \\ &= \sum_{x,t'} \underline{p(Y_t \mid X = x, T = t)} p(X = x, T = t') \end{aligned}$$



# Recall the adjustment condition

$$Y_t \perp\!\!\!\perp T \mid X$$

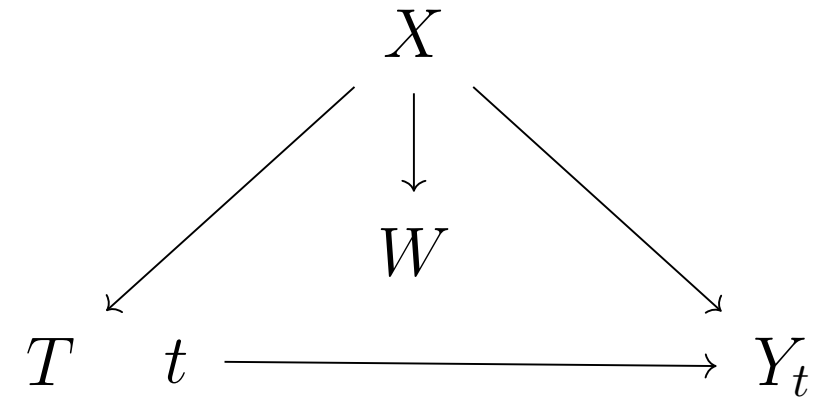
$$\begin{aligned} p(Y_t) &= \sum_{x,t'} \underline{p(Y_t | X = x, T = t')} p(X = x, T = t') \\ &= \sum_{x,t'} \underline{p(Y_t | X = x, T = t)} p(X = x, T = t') \\ &= \sum_x p(Y_t | X = x, T = t) \sum_{t'} p(X = x, T = t') \end{aligned}$$



# Recall the adjustment condition

$$Y_t \perp\!\!\!\perp T \mid X$$

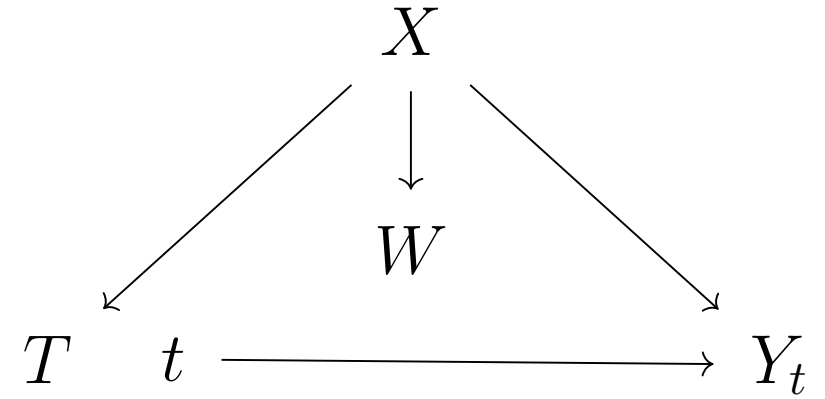
$$\begin{aligned} p(Y_t) &= \sum_{x,t'} p(Y_t \mid X = x, T = t') p(X = x, T = t') \\ &= \sum_{x,t'} p(Y_t \mid X = x, T = t) p(X = x, T = t') \\ &= \sum_x p(Y_t \mid X = x, T = t) \sum_{t'} p(X = x, T = t') \\ &= \sum_x p(Y \mid X = x, T = t) p(X = x) \end{aligned}$$



[Only a function of obs]

# Adjustment condition for representation

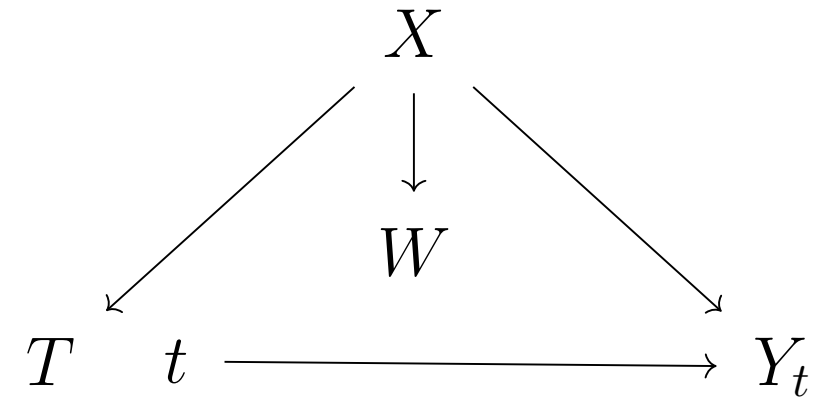
$$Y_t \perp\!\!\!\perp T \mid W$$



# Adjustment condition for representation

$$Y_t \perp\!\!\!\perp T \mid W$$

$$\begin{aligned} p(Y_t) &= \sum_{w,t'} p(Y_t \mid W = w, T = t') p(W = w, T = t') \\ &= \sum_{w,t'} p(Y_t \mid W = w, T = t) p(W = w, T = t') \\ &= \sum_w p(Y_t \mid W = w, T = t) \sum_{t'} p(W = w, T = t') \\ &= \sum_w p(Y \mid W = w, T = t) p(W = w) \end{aligned}$$



[Only a function of obs]

We can use the same derivation replacing X with the representation W!

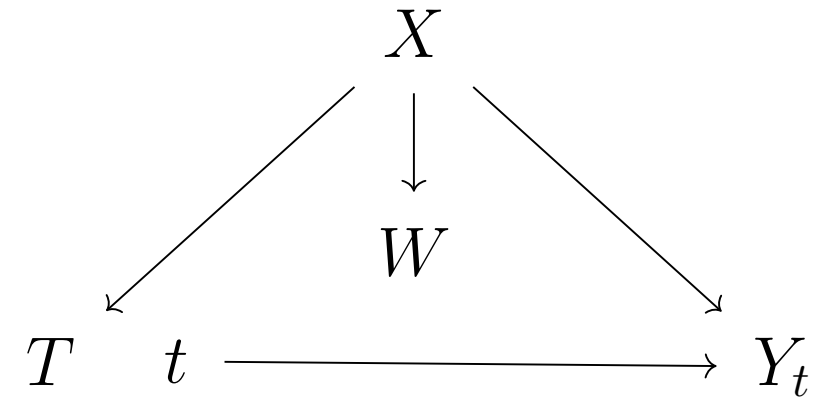
# When does $Y_t \perp\!\!\!\perp T | W$ hold?

- Balanced:  $p(W|T=0) = p(W|T=1)$

$$W \perp\!\!\!\perp T$$

- Informative:  $p(Y|W, X=x) = p(Y|W, X=x')$

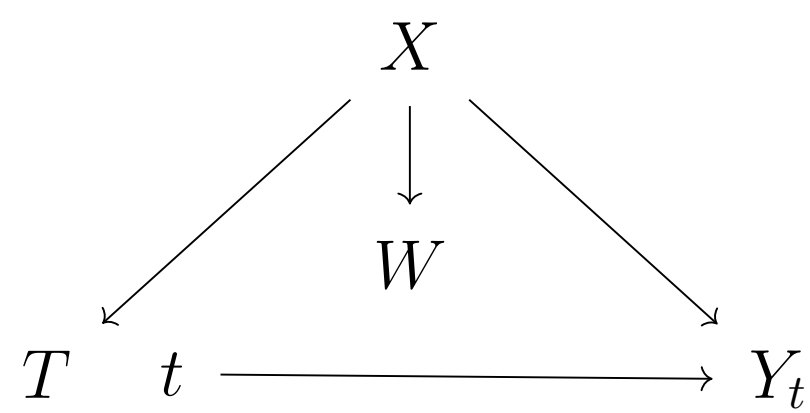
$$Y \perp\!\!\!\perp X | W$$



$$\{Y \perp\!\!\!\perp X | W\}, \{W \perp\!\!\!\perp T\} \stackrel{?}{\Rightarrow} Y_t \perp\!\!\!\perp T | W$$
$$\{Y_t \perp\!\!\!\perp T | X\}$$



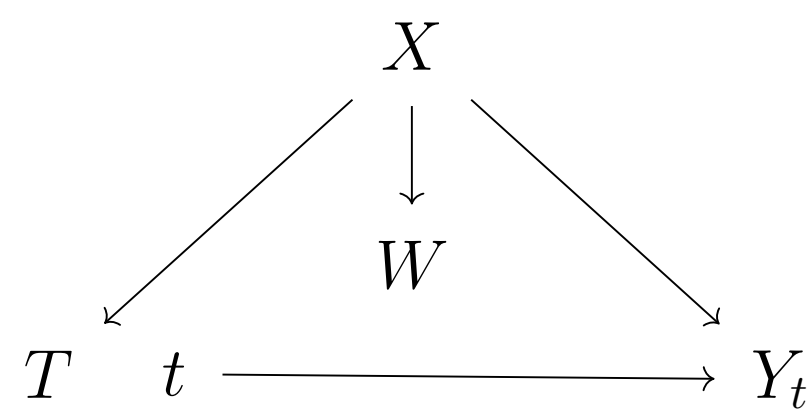
# Proof using Graphoid Axioms



1.  $Y_t \perp\!\!\!\perp T \mid X, W$  [d-separation]

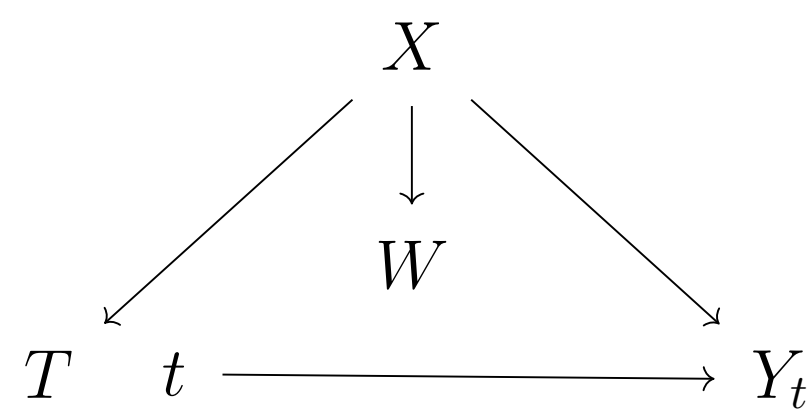
2.  $Y \perp\!\!\!\perp X \mid W \Rightarrow Y_t \perp\!\!\!\perp X \mid W$  [Assumption]

# Proof using Graphoid Axioms



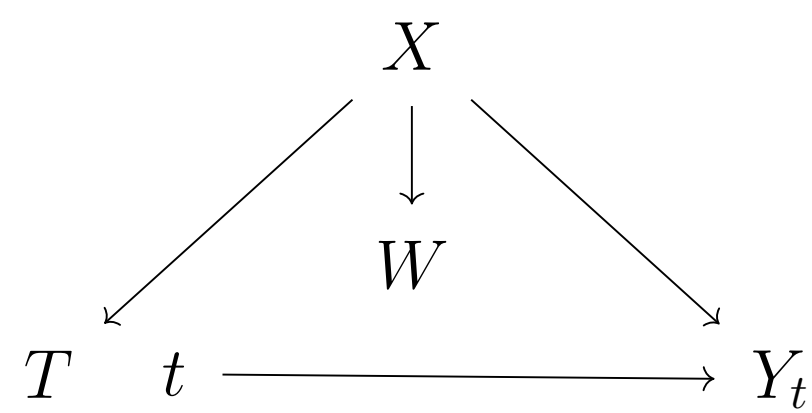
1.  $\underline{Y_t \perp\!\!\!\perp T \mid X, W}$  [d-separation]
2.  $Y \perp\!\!\!\perp X \mid W \Rightarrow \underline{Y_t \perp\!\!\!\perp X \mid W}$  [Assumption]
3.  $\underline{Y_t \perp\!\!\!\perp X \mid W} \ \& \ \underline{Y_t \perp\!\!\!\perp T \mid X, W} \Rightarrow Y_t \perp\!\!\!\perp T, X \mid W$  [Contraction]

# Proof using Graphoid Axioms



1.  $Y_t \perp\!\!\!\perp T \mid X, W$  [d-separation]
2.  $Y \perp\!\!\!\perp X \mid W \Rightarrow Y_t \perp\!\!\!\perp X \mid W$  [Assumption]
3.  $Y_t \perp\!\!\!\perp X \mid W \ \& \ Y_t \perp\!\!\!\perp T \mid X, W \Rightarrow Y_t \perp\!\!\!\perp T, X \mid W$  [Contraction]
4.  $Y_t \perp\!\!\!\perp T, X \mid W \Rightarrow Y_t \perp\!\!\!\perp T \mid W$  [Decomposition]

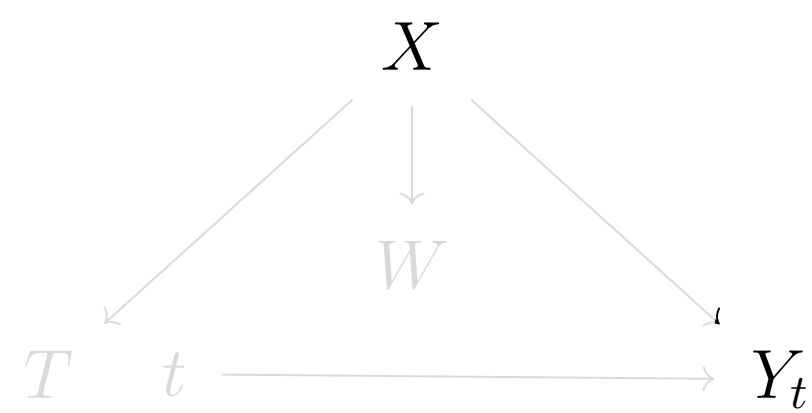
# Proof using Graphoid Axioms



1.  $Y_t \perp\!\!\!\perp T \mid X, W$  [d-separation]
2.  $Y \perp\!\!\!\perp X \mid W \Rightarrow Y_t \perp\!\!\!\perp X \mid W$  [Assumption]
3.  $Y_t \perp\!\!\!\perp X \mid W \ \& \ Y_t \perp\!\!\!\perp T \mid X, W \Rightarrow Y_t \perp\!\!\!\perp T, X \mid W$  [Contraction]
4.  $Y_t \perp\!\!\!\perp T, X \mid W \Rightarrow Y_t \perp\!\!\!\perp T \mid W$  [Decomposition]

This does not even use balanced treatment assumption!

# Proof using Graphoid Axioms



$$1. \quad Y_t \perp\!\!\!\perp T \mid X, W \quad [\text{d-separation}]$$

$$2. \quad Y \perp\!\!\!\perp X \mid W \Rightarrow Y_t \perp\!\!\!\perp X \mid W \quad [\text{Assumption}]$$

Any informative representation that satisfies  $Y \perp\!\!\!\perp X \mid W$  can be used for adjustment.

$$3. \quad Y_t \perp\!\!\!\perp X \mid W \ \& \ Y_t \perp\!\!\!\perp T \mid X, W \Rightarrow Y_t \perp\!\!\!\perp T, X \mid W \quad [\text{Contraction}]$$

$$4. \quad Y_t \perp\!\!\!\perp T, X \mid W \Rightarrow Y_t \perp\!\!\!\perp T \mid W \quad [\text{Decomposition}]$$

This does not even use balanced treatment assumption!

# Use of Balanced Representation

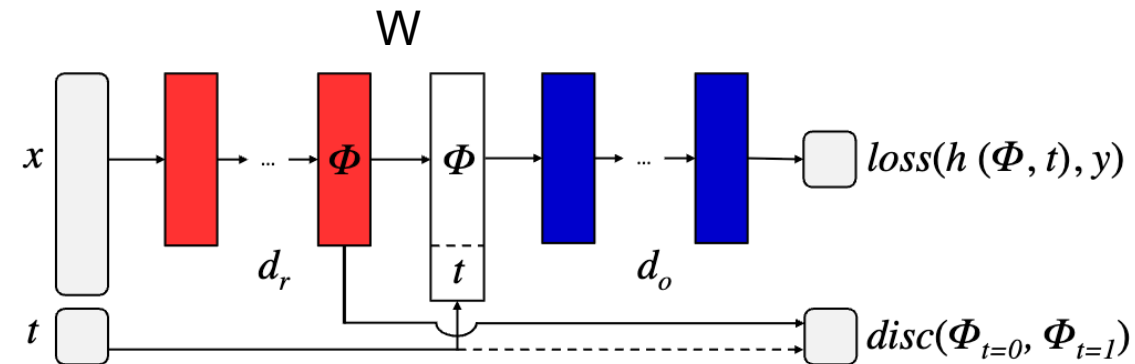
- Goal of the paper is NOT to do backdoor adjustment.
- Instead learn a balanced representation so we DO NOT need to do any adjustment!
- How does that work?

# Direct Modeling of Counterfactuals in PO

## *Learning Representations for Counterfactual Inference*

Johansson, Shalit, Sontag

- Use a neural network for learning representations that are balanced across treatment, similar to domain adaptation in ML.
- Similar to matching, make CF estimate “close” to the most similar observed (covariate, treatment) pair.
- $W = \Phi(X)$  a representation that has balanced distribution of the pairs  $(W(x_i), t_i)$  and  $(W(x_i), 1-t_i)$



# Direct Modeling of Counterfactuals in PO

## *Learning Representations for Counterfactual Inference*

Johansson, Shalit, Sontag

- Hope is that  $R(X)$  is indep from  $T$  in the induced distribution.
- Balanced representation itself is not useful as a constant  $\phi$  is also balanced, but would not be predictive.
- Three objectives:

$$B_{\mathcal{H},\alpha,\gamma}(\Phi, h) = \underbrace{\frac{1}{n} \sum_{i=1}^n |h(\Phi(x_i), t_i) - y_i^F|}_{\text{predictive}} + \underbrace{\alpha \text{disc}_{\mathcal{H}}(\hat{P}_{\Phi}^F, \hat{P}_{\Phi}^{CF})}_{\text{balanced}} + \underbrace{\frac{\gamma}{n} \sum_{i=1}^n |h(\Phi(x_i), 1 - t_i) - y_{j(i)}^F|}_{\text{matching}}$$

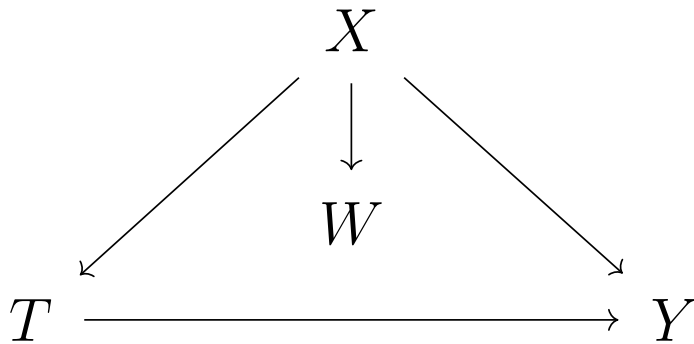


# Direct Modeling of Counterfactuals in PO

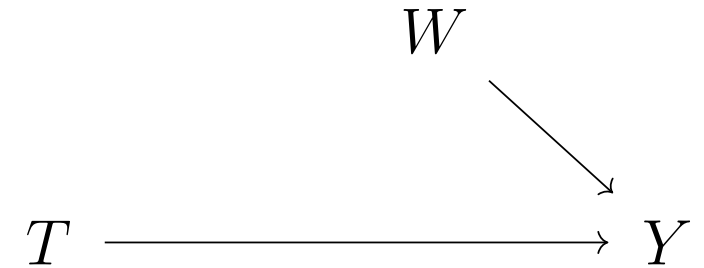
## *Learning Representations for Counterfactual Inference*

Johansson, Shalit, Sontag

- What do we really want from  $W$  here?



True SCM



What we wish the true SCM was

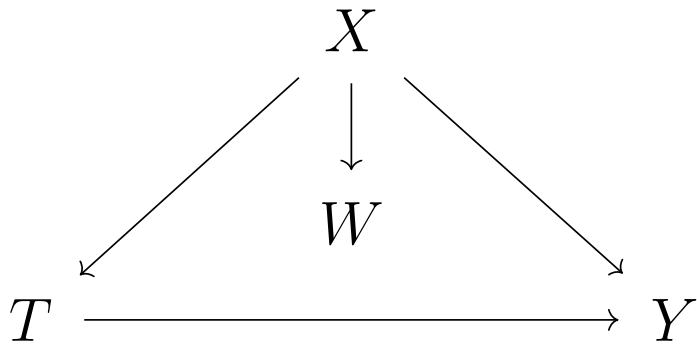
- With the causal graph on the right, we can then learn a function  $h(t, w(x))$  where  $h(1-t, w(x))$  gives the CF.

# Direct Modeling of Counterfactuals in PO

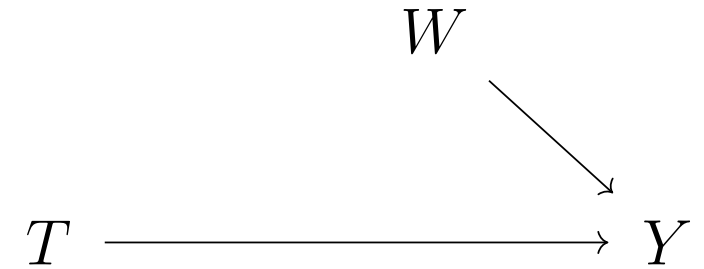
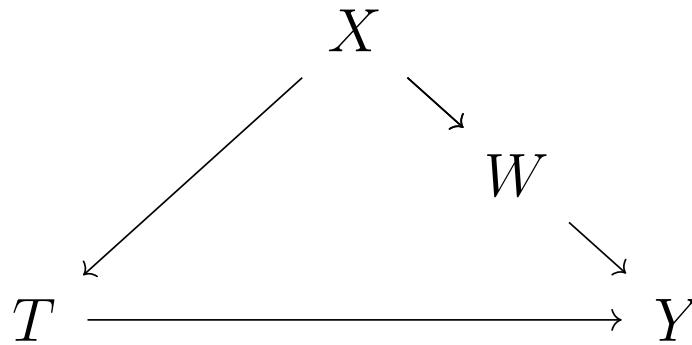
## *Learning Representations for Counterfactual Inference*

Johansson, Shalit, Sontag

- What do we really want from  $W$  here?



True SCM



What we wish the true SCM was

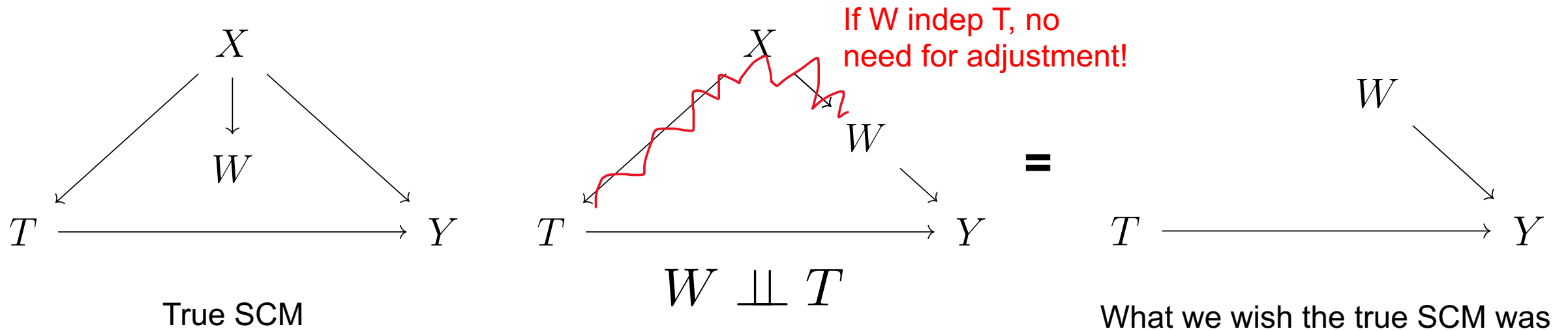
- With the causal graph on the right, we can then learn a function  $h(t, w(x))$  where  $h(1-t, w(x))$  gives the CF.
- If we assume there is a mediator  $W$  (the graph on the right), we can hope to recover the true  $W$  or a version of it.

# Direct Modeling of Counterfactuals in PO

## *Learning Representations for Counterfactual Inference*

Johansson, Shalit, Sontag

- What do we really want from  $W$  here?



- With the causal graph on the right, we can then learn a function  $h(t, w(x))$  where  $h(1-t, w(x))$  gives the CF.
- If we assume there is a mediator  $W$  (the graph on the right), we can hope to recover the true  $W$  or a version of it.

# Direct Modeling of Counterfactuals in PO

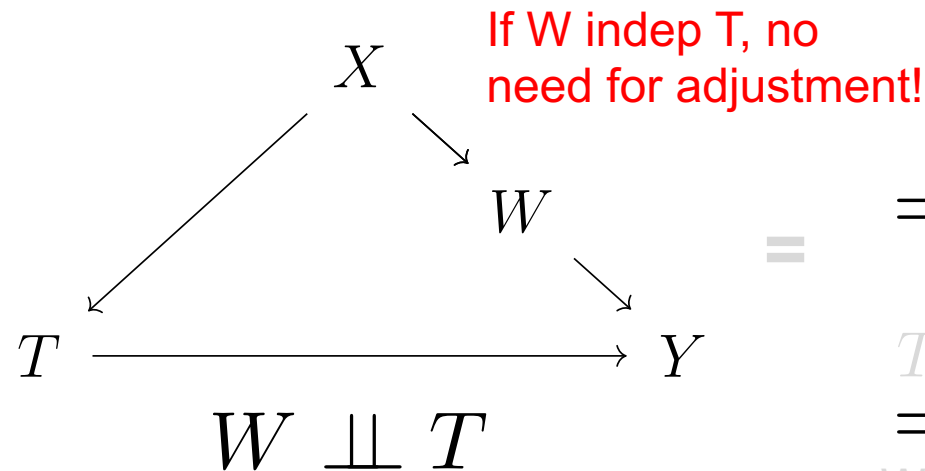
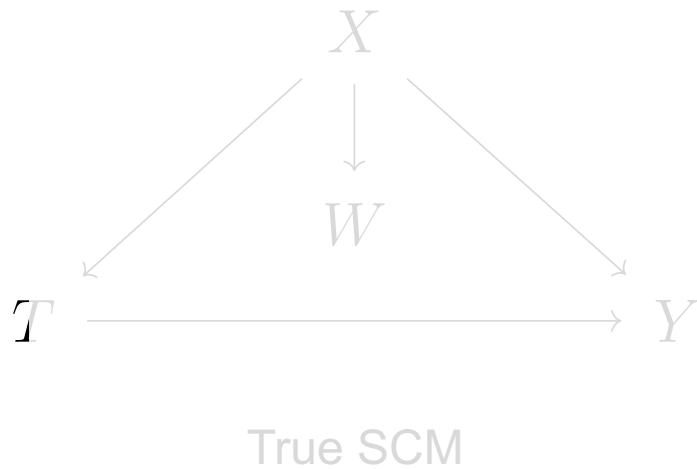
## Learning Representations for Counterfactual Inference

Johansson, Shalit, Sontag

Proof.

$$p(y|do(t)) = \sum_w p(y|t, w)p(w)$$

- What do we really want from  $W$  here?



$$\begin{aligned} &= \sum_w p(y|t, w)p(w|t) \\ &= \sum_w p(y, w|t) \\ &= p(y|t) \end{aligned}$$

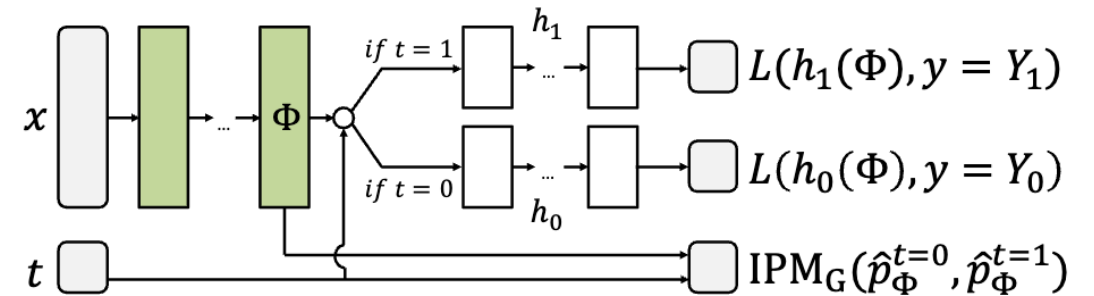
What we wish the true SCM was

- With the causal graph on the right, we can then learn a function  $h(t, w(x))$  where  $h(1-t, w(x))$  gives the CF.
- If we assume there is a mediator  $W$  (the graph on the right), we can hope to recover the true  $W$  or a version of it.

# TARNET

Shalit, Johansson, Sontag

- A follow-up to the previous paper.
- Some improvements such as the use of an integral probability metric (IPM) and multiple heads for different treatments.
- Same concept: Learn a balanced representation, learn a mapping to obtain counterfactuals.



# Adapting Neural Networks for the Estimation of Treatment Effects

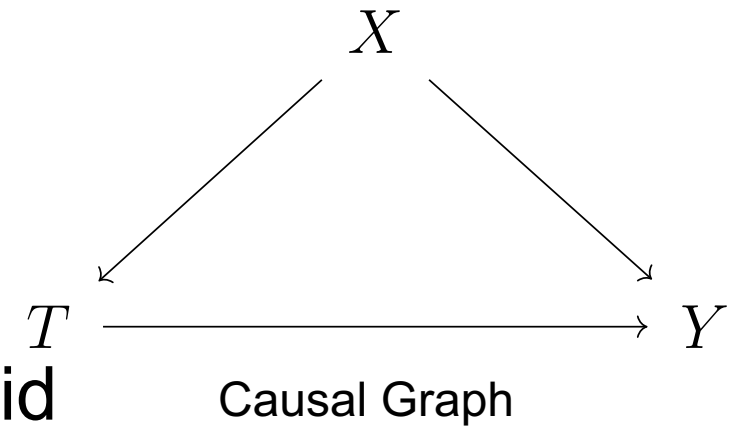
Shi, Blei, Veitch

- The following theorem tells us that a specific low dimensional representation is sufficient for adjustment:

**Theorem<sup>1</sup>:** Let  $X$  be a valid adjustment set. Then the propensity score  $g(X)$  is also a valid adjustment set, where

$$g(x) = \mathbb{P}(T = 1 | X = x)$$

Then  $\text{ATE} = \mathbb{E}[\mathbb{E}[Y | g(X), T = 1]] - \mathbb{E}[\mathbb{E}[Y | g(X), T = 0]]$



<sup>1</sup> Rosenbaum, Rubin, The central role of the propensity score in observational studies for causal effect, 1983

# Adapting Neural Networks for the Estimation of Treatment Effects

Shi, Blei, Veitch

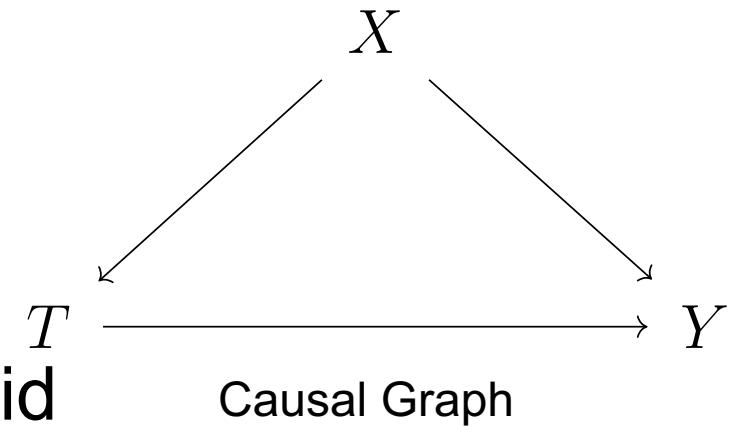
- The following theorem tells us that a specific low dimensional representation is sufficient for adjustment:

**Theorem<sup>1</sup>:** Let  $X$  be a valid adjustment set. Then the propensity score  $g(X)$  is also a valid adjustment set, where

$$g(x) = \mathbb{P}(T = 1 | X = x)$$

Then  $\text{ATE} = \mathbb{E}[\mathbb{E}[Y | g(X), T = 1]] - \mathbb{E}[\mathbb{E}[Y | g(X), T = 0]]$

Not exactly  
what they do!



<sup>1</sup> Rosenbaum, Rubin, The central role of the propensity score in observational studies for causal effect, 1983

# *Adapting Neural Networks for the Estimation of Treatment Effects*

Shi, Blei, Veitch

- Use neural net to model conditional outcome

$$Q(t, x) = \mathbb{E}[Y | x, t]$$

and propensity score

$$g(x) = \mathbb{P}(T = 1 | X = x)$$

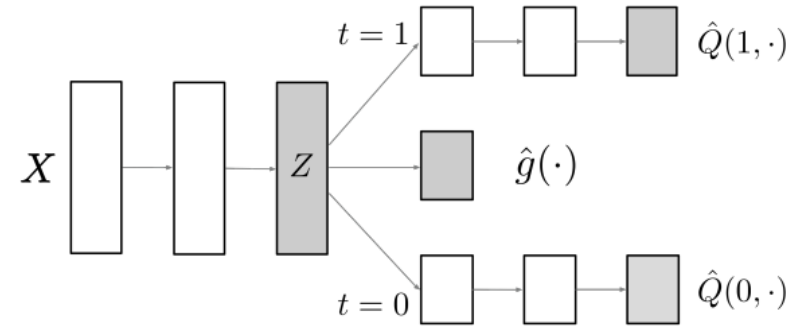
- The hope is to tap into NNs ability to distill relevant information



# Adapting Neural Networks for the Estimation of Treatment Effects

Shi, Blei, Veitch

- Find a good representation  $Z(X)$  that can be used to estimate both propensity score AND conditional outcome.



- A regularization scheme motivated by the properties of good estimators (fast convergence and lowest variance).

$$0 = \frac{1}{n} \sum \varphi(y_i, t_i, x_i; \hat{Q}, \hat{g}, \hat{\psi})$$

$$\varphi(y, t, x; Q, g, \psi) = Q(1, x) - Q(0, x) + \left( \frac{t}{g(x)} - \frac{1-t}{1-g(x)} \right) \{y - Q(t, x)\} - \psi.$$

- Construct the loss function so that the stationary point enforces the above equality!

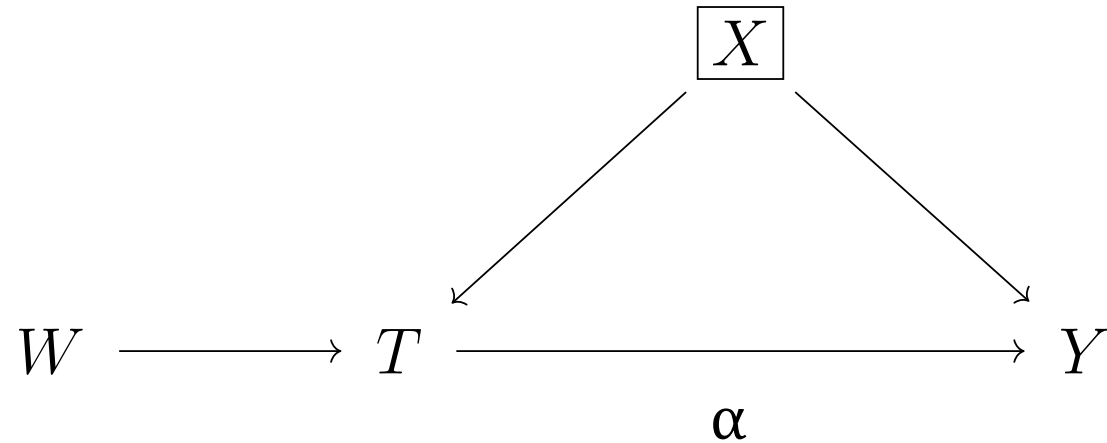
# Instrumental Variables

**W:** Fuel Price

**T:** Ticket Price

**Y:** # of Sales

**X:** Holiday Season



- Causal effect is identifiable under linearity assumption, e.g., by using 2SLS (two stage least squares): Regress  $T$  on  $W$  to get  $T'$ . Regress  $Y$  on  $T'$  to get  $\alpha$ .
- Tighter bounds can be obtained using instrument  $W$  in general.

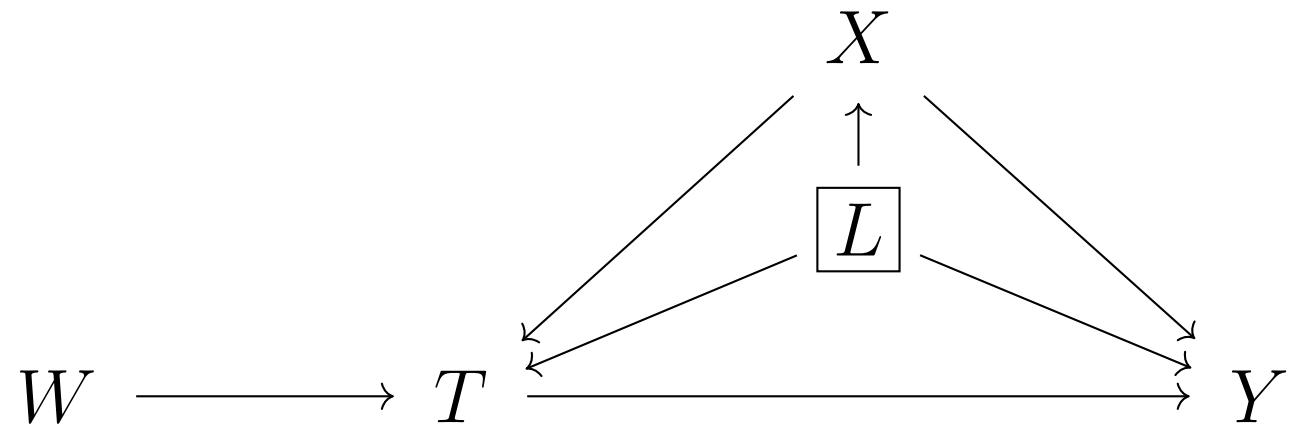
# Deep IV: A Flexible Approach for Counterfactual Pred.

Hartford, Lewis, Leyton-Brown, Taddy

- Assume the structural equation of the form.

$$Y = g(T, X) + L$$

- Given  $X=x$ , we want the “*counterfactual*”  $\text{do}(T = t)$

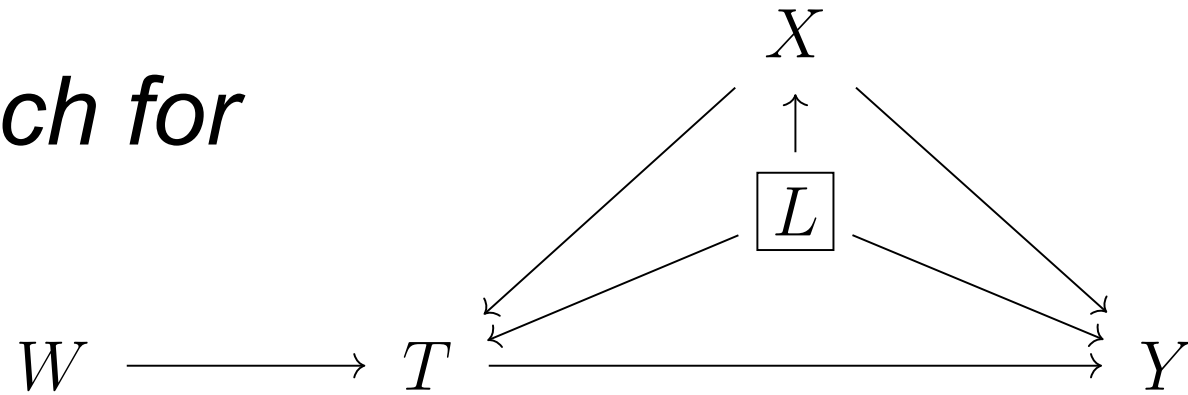


$$h(t, x) = g(t, x) + \mathbb{E}[L|X = x] \quad \leftarrow \text{Notice no } T = t \text{ here!}$$

- Learning  $g$  is enough to decide better intervention:  $h(t_1, x) - h(t_2, x) = g(t_1, x) - g(t_2, x)$

# Deep IV: A Flexible Approach for Counterfactual Pred.

Hartford, Lewis, Leyton-Brown, Taddy



$$\begin{aligned}\mathbb{E}[Y|x, w] &= \mathbb{E}[g(T, X) + L|x, w] = \mathbb{E}[g(T, X)|x, w] + \mathbb{E}[L|x, w] \\ &= \mathbb{E}[g(T, X)|x, w] + \mathbb{E}[L|x]\end{aligned}$$

- Recall:

$$h(t, x) = g(t, x) + \mathbb{E}[L|X = x]$$

- Then

$$\mathbb{E}[Y|x, w] = \int h(t, x) dF(t|x, w)$$

We want solution to this integral equation!

# *Deep IV: A Flexible Approach for Counterfactual Pred.*

Hartford, Lewis, Leyton-Brown, Taddy

$$\mathbb{E}[Y|x, w] = \int h(t, x) dF(t|x, w)$$

- Solve this integral equation in two stages.
- First do a density estimate of treatment given covariate and instrument  $f(t|x, w)$ .
  - If  $t$  is discrete, use categorical distribution with softmax output as parameter.
  - If  $t$  is continuous, model as mixture of Gaussians via mixture density network.

# *Deep IV: A Flexible Approach for Counterfactual Pred.*

Hartford, Lewis, Leyton-Brown, Taddy

- After neural density estimation of  $f(t|x,w)$  we focus on  $h$ .
- $h$  is parameterized by a neural network as well.
- Then minimize the loss function over  $h$ .

$$\min_h \sum_{i=1}^n \left( y_i - \int h(t, x_i) dF(t|x_i, w_i) \right)^2$$

- Integral is replaced by Monte Carlo averaging

# *Deep IV: A Flexible Approach for Counterfactual Pred.*

Hartford, Lewis, Leyton-Brown, Taddy

## Author notes:

- A challenge with using deep learning is extensive reliance on hyperparameters.
- Lack of ground truth is a problem for choosing the best hyperparameter set.
- Objectives that only rely on fitting observational data are more powerful as they do not require test set.

# Causal Inference with Deep Learning and Generative Models

## *Outline*

- Background
  - Causal Inference Basics
  - Neural Network Basics
- A Taxonomy of Deep Learning Approaches for Causal Inference
  - Function Modeling (a.k.a. Curve Fitting)
  - Feature Extraction
  - Generative Causal Inference

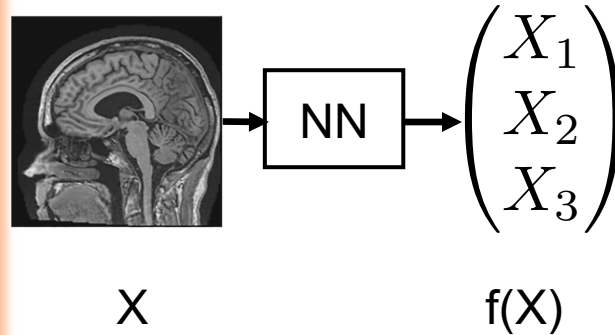


# A Taxonomy of Deep Learning Approaches for Causal Inference

Function Modeling

$$f : \mathbb{R}^k \rightarrow [0, 1]$$

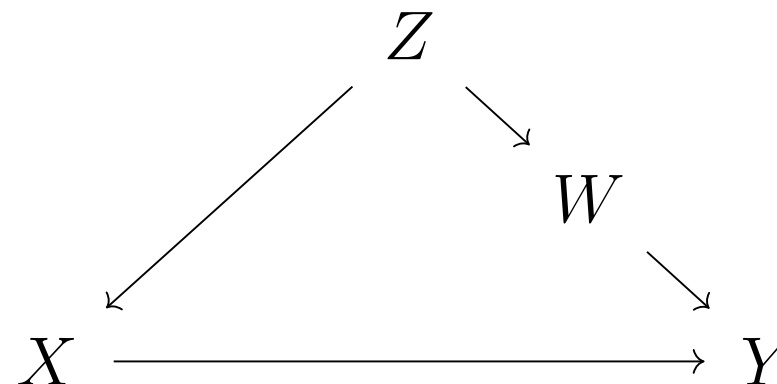
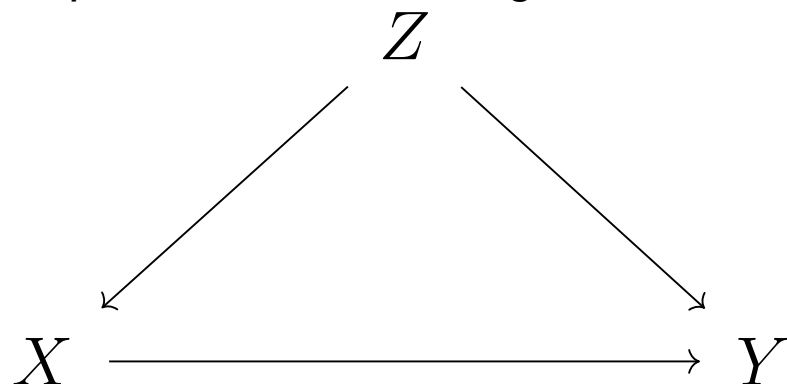
Feature Extraction



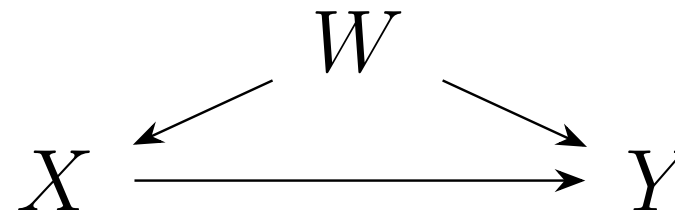
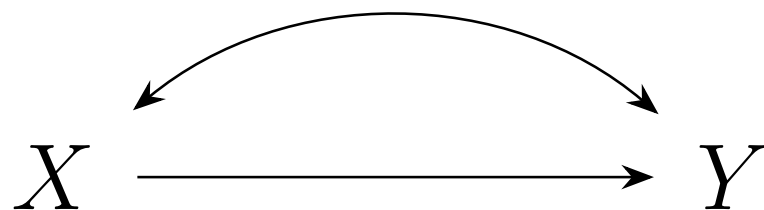
Generative Modeling

# Feature Extraction

*Representation Learning*



*Latent Feature Construction*



# Latent Feature Extraction

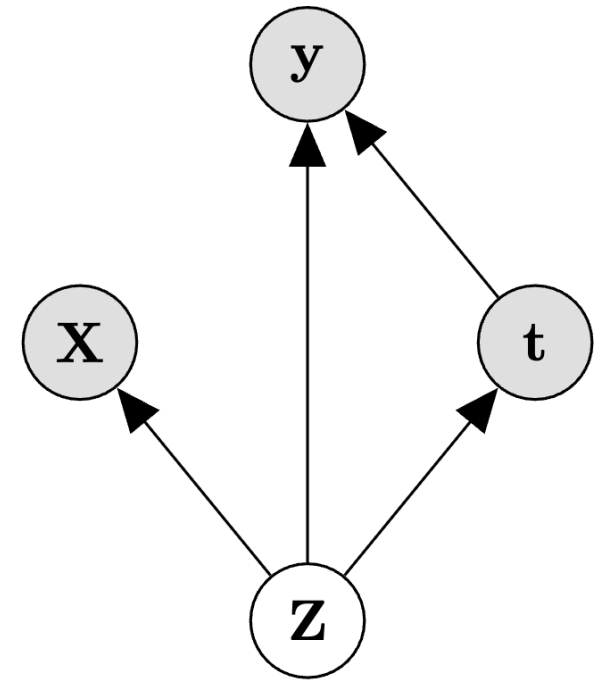
*Wishful Thinking? Or “Shorter” Leap of Faith?*

- Extracting latent features using neural networks seems more acceptable in academic circles.
- Usually no guarantee that the extracted latents are correct/sufficient for adjustment.
- Does a VAE extract causal features? Usually no.

# *Causal Effect Inference with Deep Latent-Variable Models*

Louizos, Shalit, Mooij, Sontag, Zemel, Welling

- Unobserved confounder  $Z$  prevent sound adjustment.
- Suppose a proxy  $X$  is available.
- Ex.: Genetic factors  $\rightarrow$  Blood sugar  
Socioeconomic  $\rightarrow$  Zip code

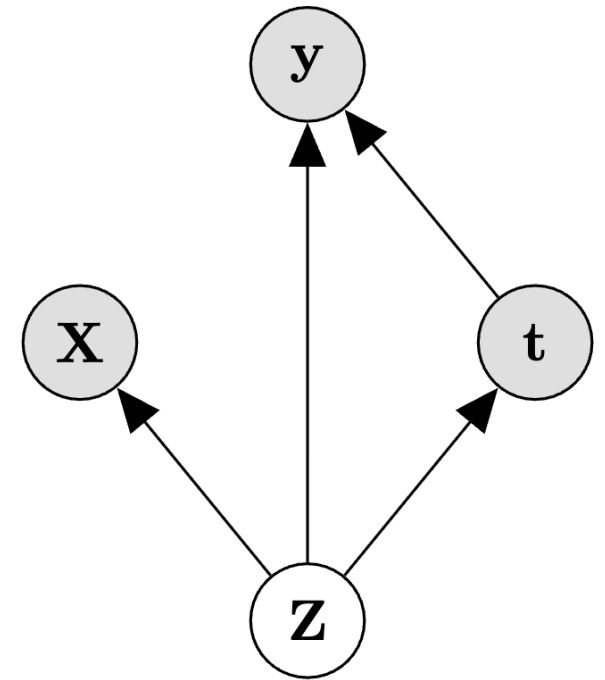


*$X$  is proxy of  
unobserved  
confounder  $Z$ .*

# *Causal Effect Inference with Deep Latent-Variable Models*

Louizos, Shalit, Mooij, Sontag, Zemel, Welling

- Use VAE to recover latents and use them for adjustment.
- When would this be sound? A sufficient condition is when we recover the correct latents. When does that happen? Unclear.



- Metric: ITE which in PO is defined as

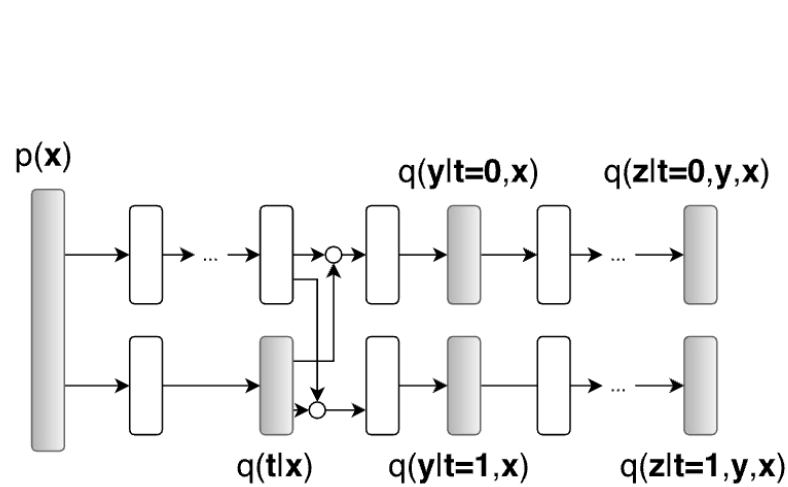
$$ITE(x) := \mathbb{E}[\mathbf{y} | \mathbf{X} = x, do(\mathbf{t} = 1)] - \mathbb{E}[\mathbf{y} | \mathbf{X} = x, do(\mathbf{t} = 0)], \quad ATE := \mathbb{E}[ITE(x)]$$

- A misnomer since stratifying with **X** usually does not limit cohort to an individual.

# CEVAE Architecture

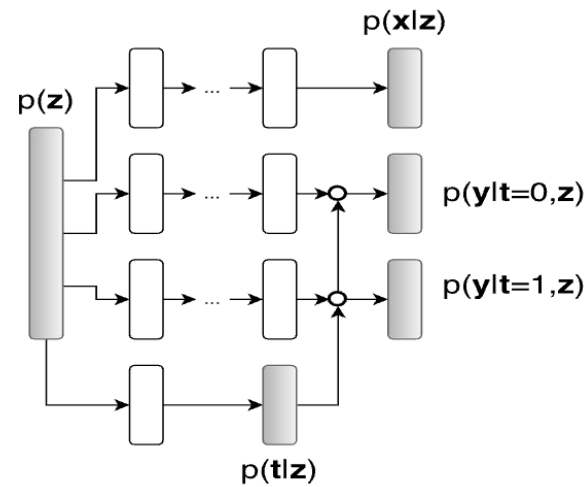
## *Causal Effect Inference with Deep Latent-Variable Models*

Louizos, Shalit, Mooij, Sontag, Zemel, Welling



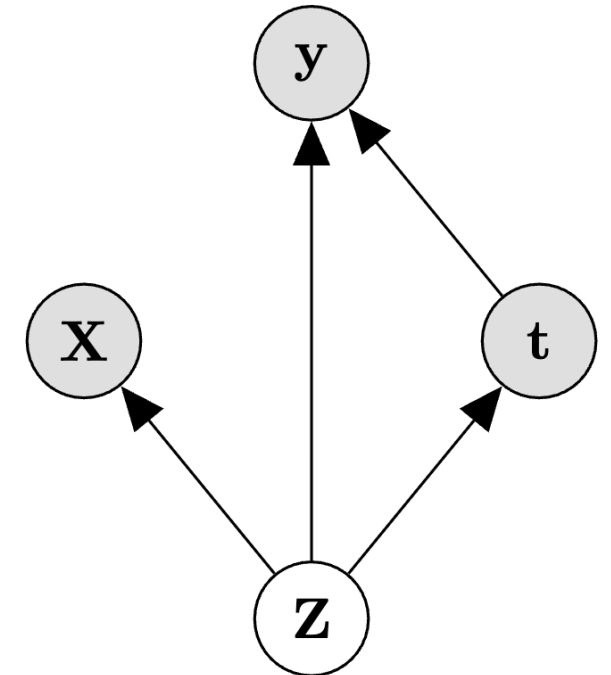
(a) Inference network,  $q(\mathbf{z}, t, y | \mathbf{x})$ .

Predict  $T$  from  $X$ ,  
 $Y$  from  $X, T$   
 $Z$  from  $X, T, Y$



(b) Model network,  $p(\mathbf{x}, \mathbf{z}, t, y)$ .

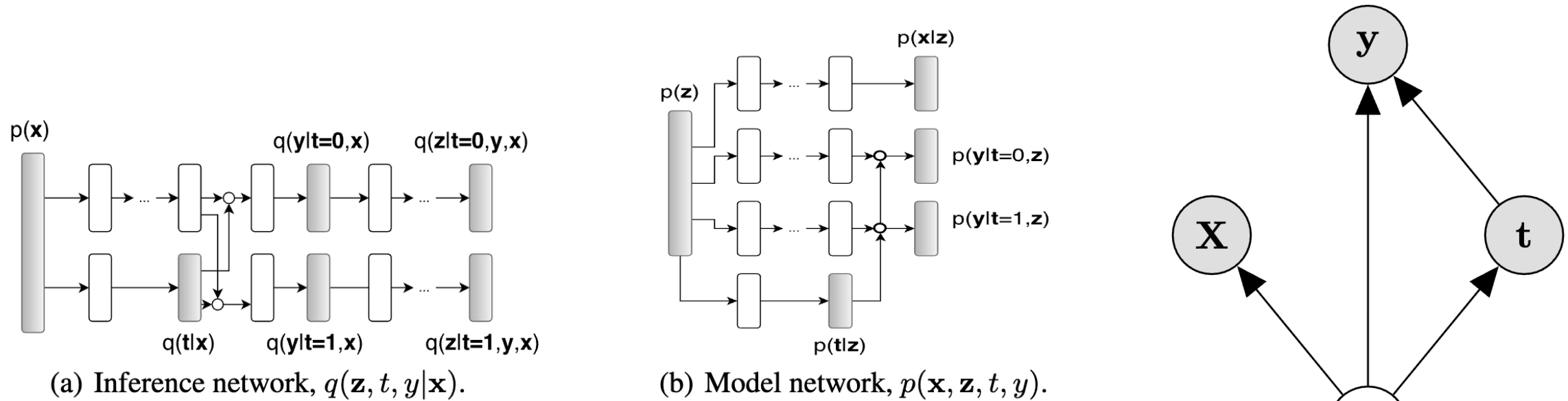
Predict  $T$  from  $Z$ ,  
 $X$  from  $Z$   
 $Y$  from  $T, Z$



# CEVAE Architecture

## *Causal Effect Inference with Deep Latent-Variable Models*

Louizos, Shalit, Mooij, Sontag, Zemel, Welling



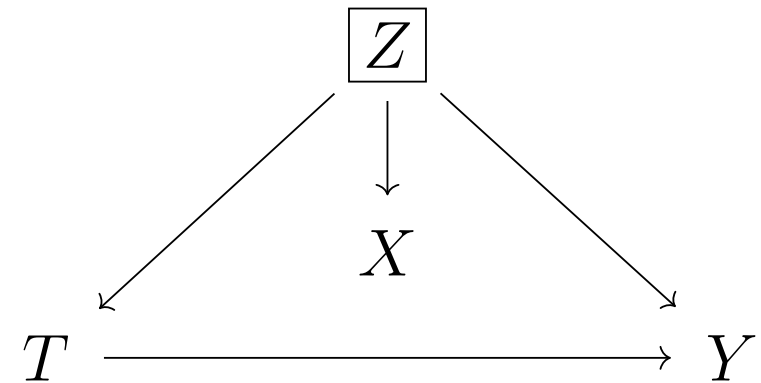
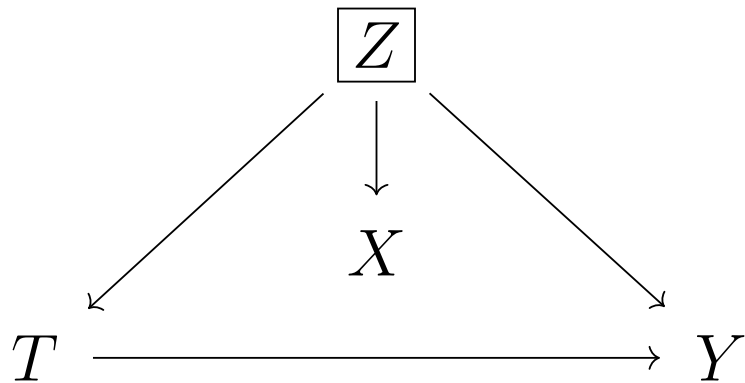
$$p(\mathbf{z}_i) = \prod_{j=1}^{D_z} \mathcal{N}(z_{ij} | 0, 1); \quad p(\mathbf{x}_i | \mathbf{z}_i) = \prod_{j=1}^{D_x} p(x_{ij} | \mathbf{z}_i); \quad p(t_i | \mathbf{z}_i) = \text{Bern}(\sigma(f_1(\mathbf{z}_i))),$$

Marginal dist. of unobs. conf. is assumed to be unit Gaussian.

How bad is this assumption really? The choice of function  $f$  can compensate for the mismatch but only sometimes for specific graphs/queries.

# The Identifiability Problem

- Given  $p(T, Y, X)$ , can we construct two SCMs with different causal effects?

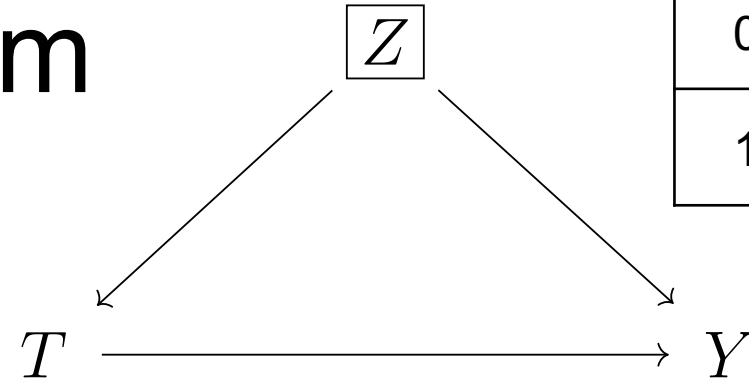


- Say  $T, Y, X$  are discrete for simplicity,  $Z$  scalar  $\text{Uniform}[0,1]$ .
- Drop  $X$  for now to see how we can do this always.



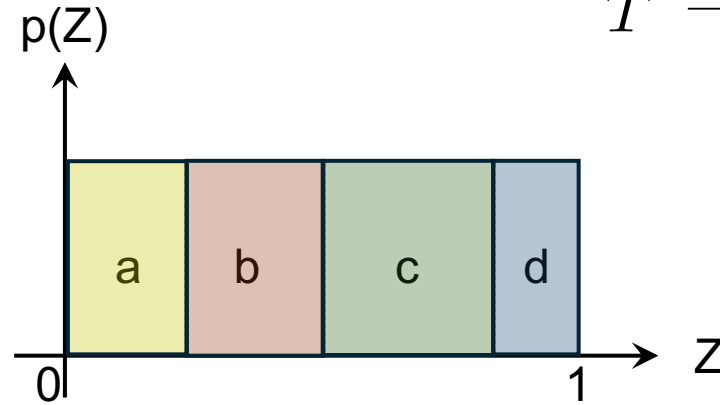
# The Identifiability Problem

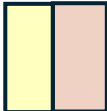
$T \setminus Y$	0	1
0	a	b
1	c	d





- Given  $p(T, Y)$ , can we construct two SCMs with different causal effects?

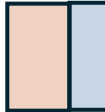
- SCM1:



$T = 0$  if  $Z \in$  

$Y = 0$  if  $Z \in$  

$T = 1$  if  $Z \in$  

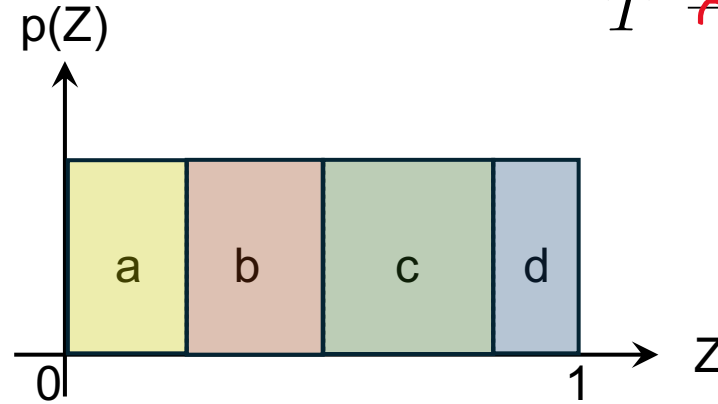
$Y = 1$  if  $Z \in$  


$$p(y|\text{do}(T = 1)) = p(y)$$


# The Identifiability Problem

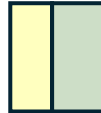
- Given  $p(T, Y)$ , can we construct two SCMs with different causal effects?

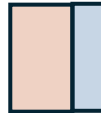
- SCM1:



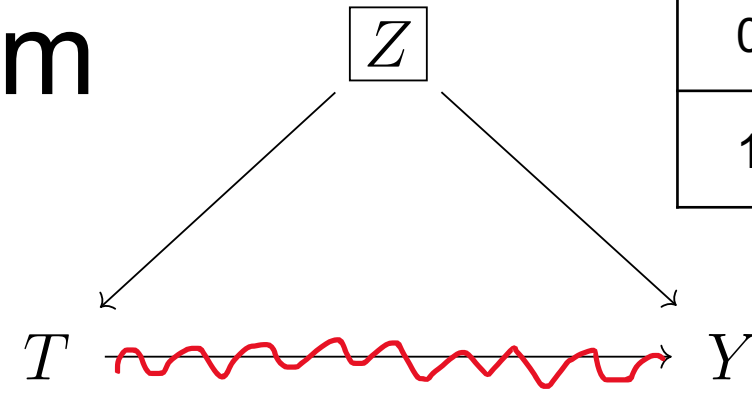
$T = 0$  if  $Z \in$  

$T = 1$  if  $Z \in$  

$Y = 0$  if  $Z \in$  

$Y = 1$  if  $Z \in$  

$$p(y|\text{do}(T = 1)) = p(y)$$

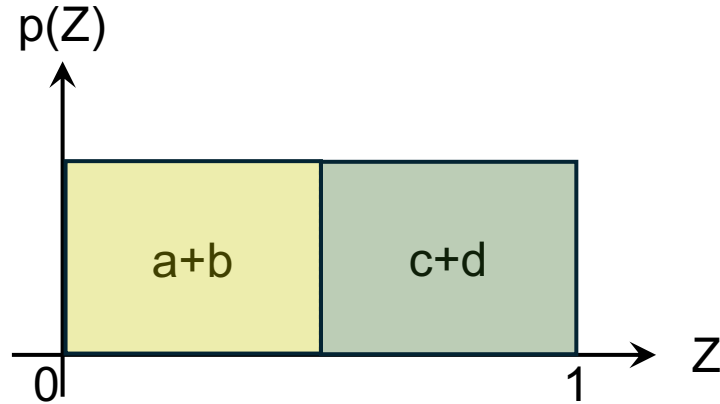


$T \setminus Y$	0	1
0	a	b
1	c	d

# The Identifiability Problem

- Given  $p(T, Y)$ , can we construct two SCMs with different causal effects?

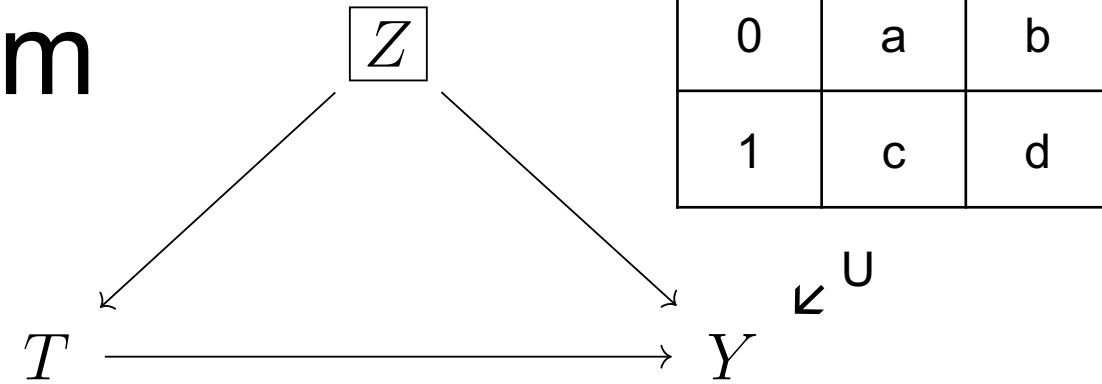
- SCM2:



$T = 0$  if  $Z \in$

$T = 1$  if  $Z \in$

$$p(y|\text{do}(T = 1)) = p(y|T = 1)$$

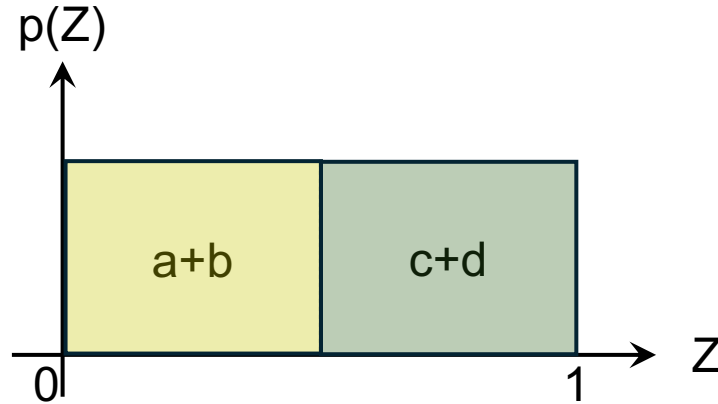


$T \setminus Y$	0	1
0	a	b
1	c	d

# The Identifiability Problem

- Given  $p(T, Y)$ , can we construct two SCMs with different causal effects?

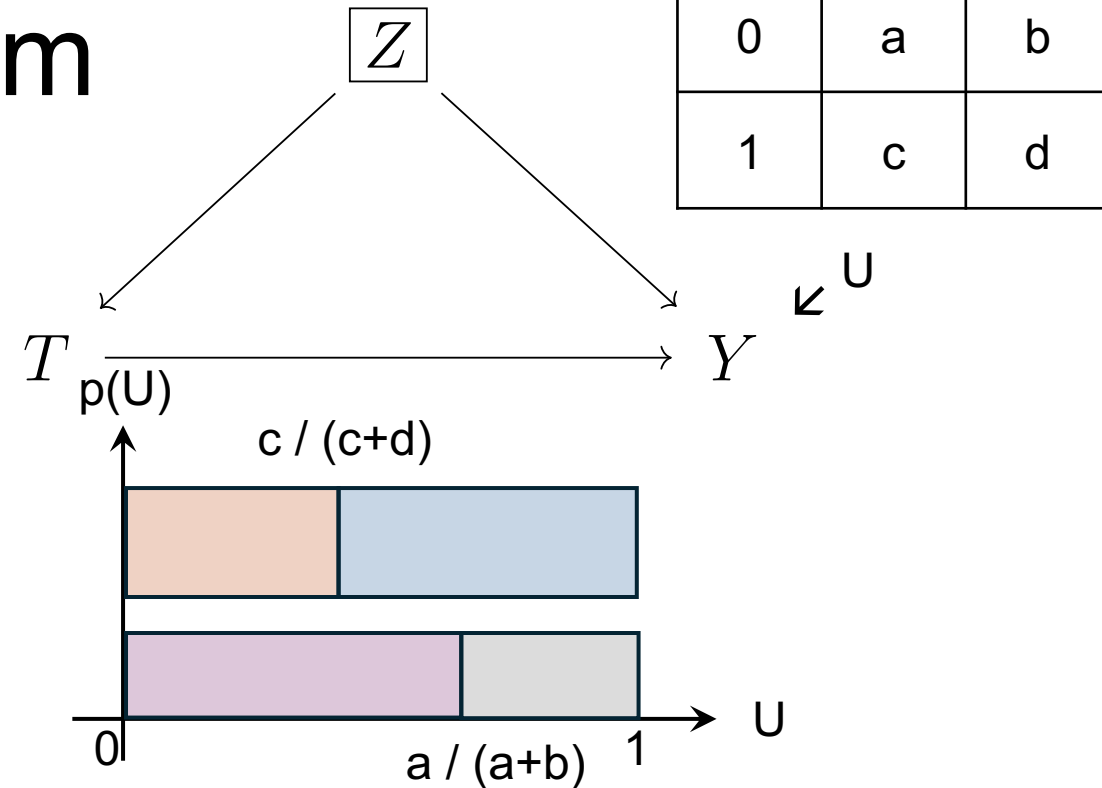
- SCM2:



$T = 0$  if  $Z \in$

$T = 1$  if  $Z \in$

$$p(y|\text{do}(T = 1)) = p(y|T = 1)$$



$Y = 0$  if  $U \in$   &  $T = 0$

$Y = 1$  if  $U \in$   &  $T = 0$

$Y = 0$  if  $U \in$   &  $T = 1$

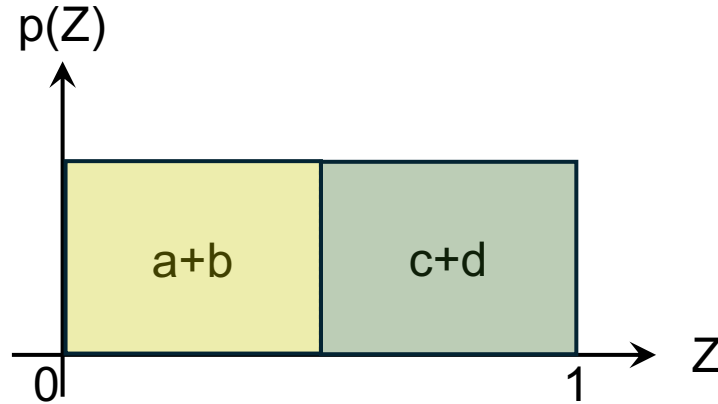
$Y = 1$  if  $U \in$   &  $T = 1$

$T \backslash Y$	0	1
0	a	b
1	c	d

# The Identifiability Problem

- Given  $p(T, Y)$ , can we construct two SCMs with different causal effects?

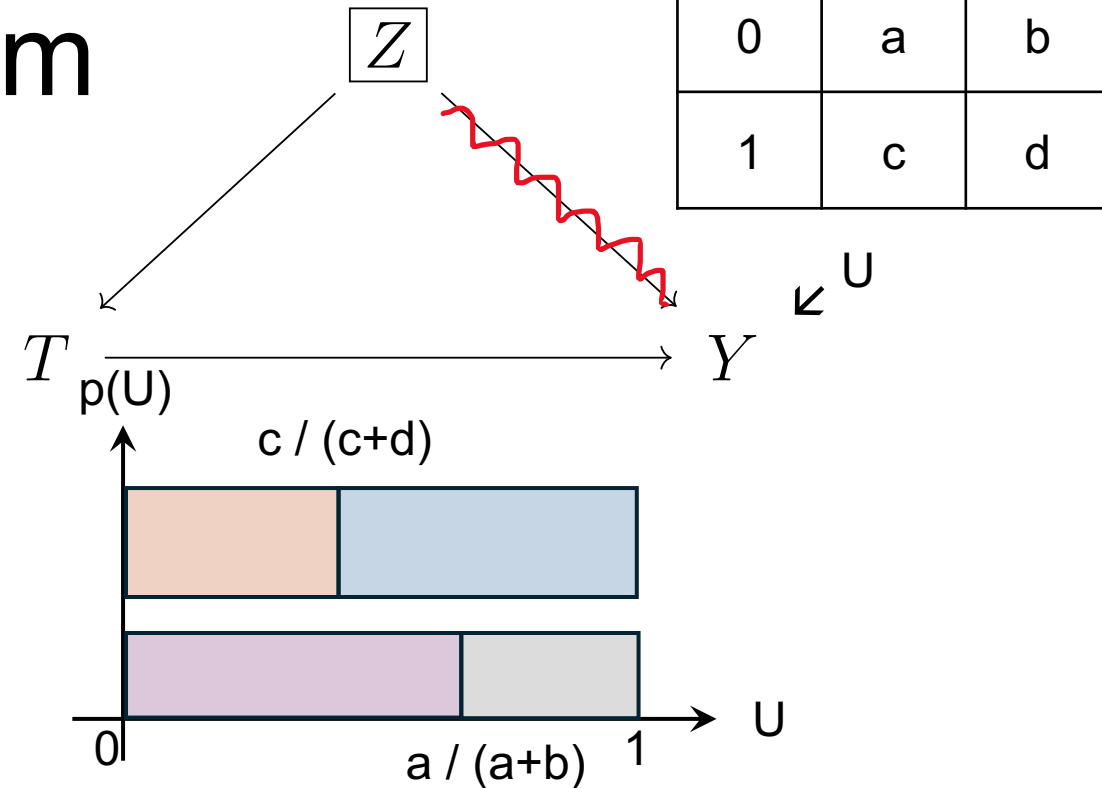
- SCM2:



$T = 0$  if  $Z \in$

$T = 1$  if  $Z \in$

$$p(y|\text{do}(T = 1)) = p(y|T = 1)$$



$T \backslash Y$	0	1
0	a	b
1	c	d

$Y = 0$  if  $U \in$   &  $T = 0$

$Y = 1$  if  $U \in$   &  $T = 0$

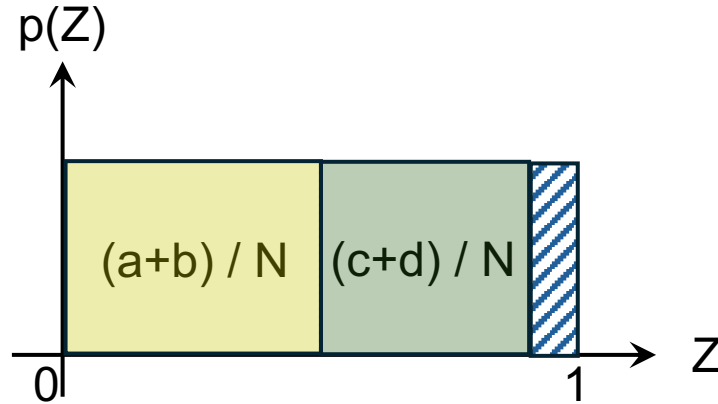
$Y = 0$  if  $U \in$   &  $T = 1$

$Y = 1$  if  $U \in$   &  $T = 1$

# The Identifiability Problem

- Given  $p(T, Y)$ , can we construct two SCMs with different causal effects?

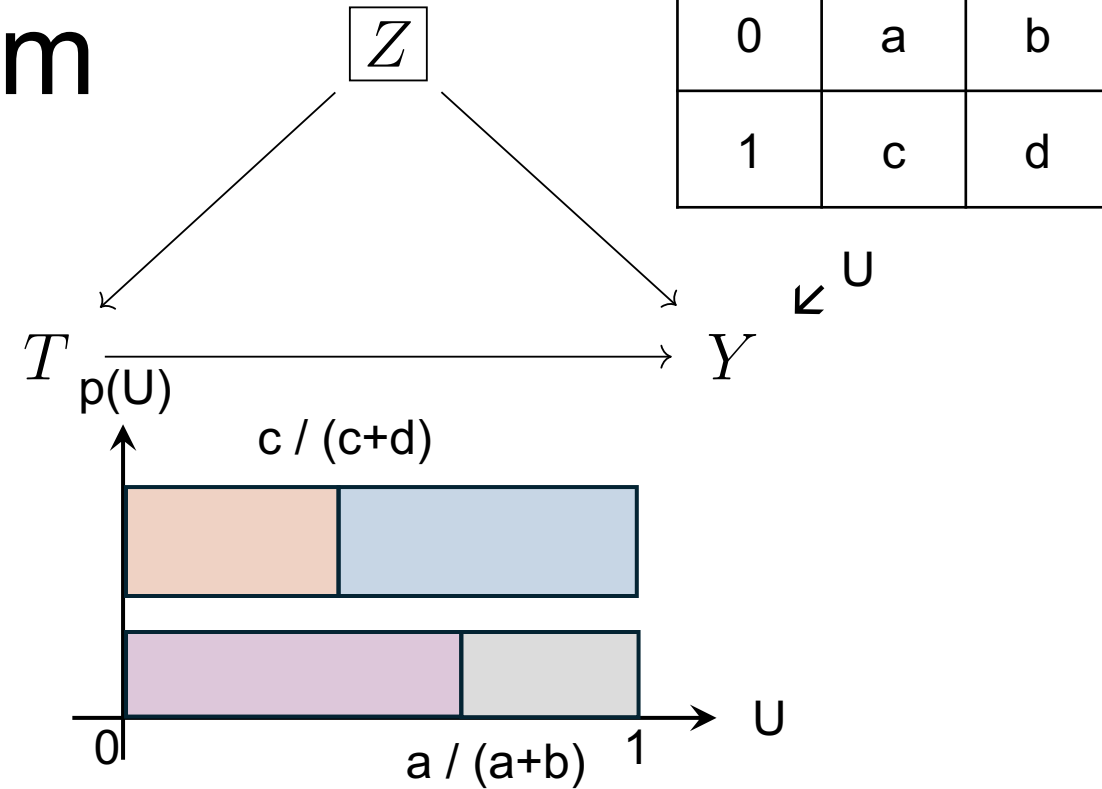
- SCM3:



- In between SCM1 and SCM2

If  $Z \in \text{blue hatched}$ , SCM1

Else, SCM2

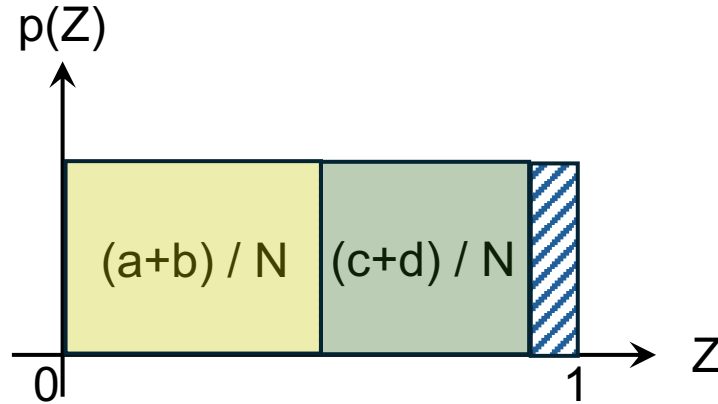


$T \setminus Y$	0	1
0	a	b
1	c	d

# The Identifiability Problem

- Given  $p(T, Y)$ , can we construct two SCMs with different causal effects?

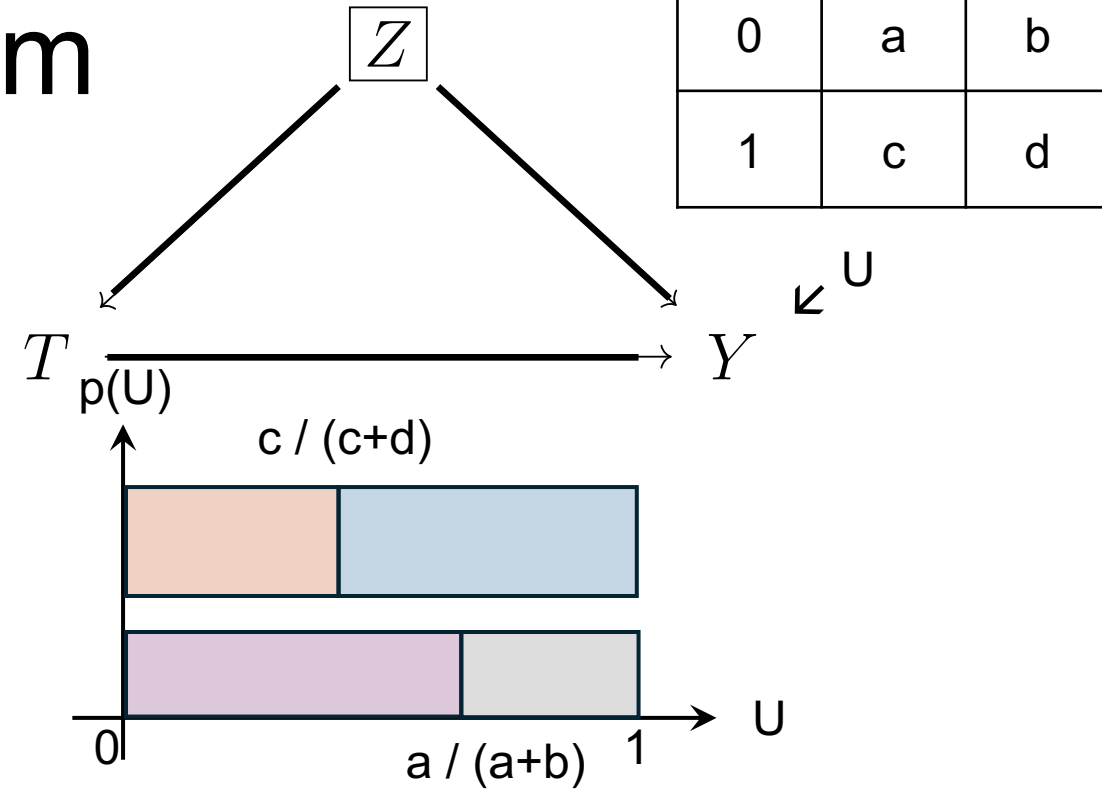
- SCM3:



- In between SCM1 and SCM2

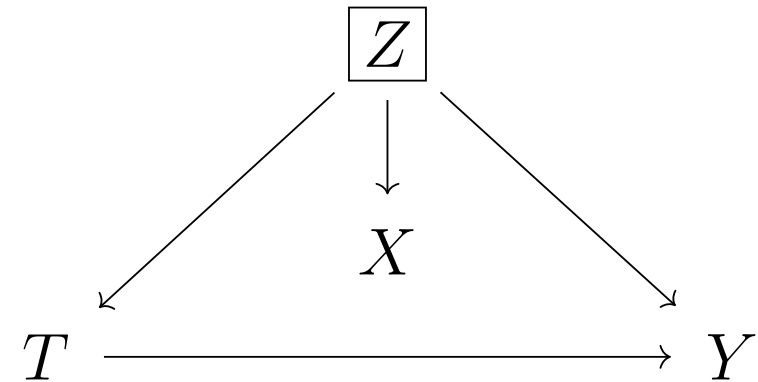
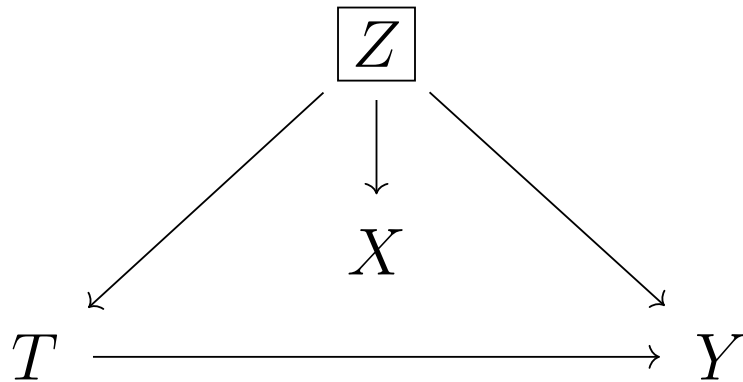
If  $Z \in \text{blue hatched bar}$ , SCM1

Else, SCM2



# The Identifiability Problem

- Given  $p(T, Y, X)$ , can we construct two SCMs with different causal effects?



- Does presence of  $X$  change anything?
- Not really! Just do the same thing to attain  $p(T, Y|X)$  for different values of  $X$  to show non-identifiability of  $p(y|do(t))$ .



# CEVAE Training

## *Causal Effect Inference with Deep Latent-Variable Models*

Louizos, Shalit, Mooij, Sontag, Zemel, Welling

- Use variational lower bound to minimize the cross entropy loss

$$\mathcal{L} = \sum_{i=1}^N \mathbb{E}_{q(\mathbf{z}_i|\mathbf{x}_i, t_i, y_i)} [\log p(\mathbf{x}_i, t_i|\mathbf{z}_i) + \log p(y_i|t_i, \mathbf{z}_i) + \log p(\mathbf{z}_i) - \log q(\mathbf{z}_i|\mathbf{x}_i, t_i, y_i)]$$

But for inference we would need to estimate  $y$ ,  $t$  given  $x$  and we can learn these during training using the following model:

$$q(t_i|\mathbf{x}_i) = \text{Bern}(\pi = \sigma(g_4(\mathbf{x}_i)))$$

$$q(y_i|\mathbf{x}_i, t_i) = \mathcal{N}(\mu = \bar{\mu}_i, \sigma^2 = \bar{v}) \quad \bar{\mu}_i = t_i(g_6 \circ g_5(\mathbf{x}_i)) + (1 - t_i)(g_7 \circ g_5(\mathbf{x}_i))$$

$$q(y_i|\mathbf{x}_i, t_i) = \text{Bern}(\pi = \bar{\pi}_i) \quad \bar{\pi}_i = t_i(g_6 \circ g_5(\mathbf{x}_i)) + (1 - t_i)(g_7 \circ g_5(\mathbf{x}_i)),$$

Composite loss then becomes:

$$\mathcal{F}_{\text{CEVAE}} = \mathcal{L} + \sum_{i=1}^N (\log q(t_i = t_i^*|\mathbf{x}_i^*) + \log q(y_i = y_i^*|\mathbf{x}_i^*, t_i^*)),$$

# CEVAE Training

## *Causal Effect Inference with Deep Latent-Variable Models*

Louizos, Shalit, Mooij, Sontag, Zemel, Welling

- Use variational lower bound to minimize the cross entropy loss

$$\mathcal{L} = \sum_{i=1}^N \mathbb{E}_{q(\mathbf{z}_i | \mathbf{x}_i, t_i, y_i)} [\log p(\mathbf{x}_i, t_i | \mathbf{z}_i) + \log p(y_i | t_i, \mathbf{z}_i) + \log p(\mathbf{z}_i) - \log q(\mathbf{z}_i | \mathbf{x}_i, t_i, y_i)]$$

### Questions.

But for inference we would need to estimate  $y, t$  given  $x$  and we can learn these during training using the following model:

$$\begin{aligned} q(t_i | \mathbf{x}_i) &= \text{Bern}(\pi = \sigma(g_4(\mathbf{x}_i))) \\ q(y_i | \mathbf{x}_i, t_i) &= \mathcal{N}(\mu = \mu_i, \sigma = \sigma) \quad \mu_i = t_i(g_6 \circ g_5(\mathbf{x}_i)) + (1 - t_i)(g_7 \circ g_5(\mathbf{x}_i)) \\ q(y_i | \mathbf{x}_i, t_i) &= \text{Bern}(\pi = \bar{\pi}_i) \quad \bar{\pi}_i = t_i(g_6 \circ g_5(\mathbf{x}_i)) + (1 - t_i)(g_7 \circ g_5(\mathbf{x}_i)), \end{aligned}$$

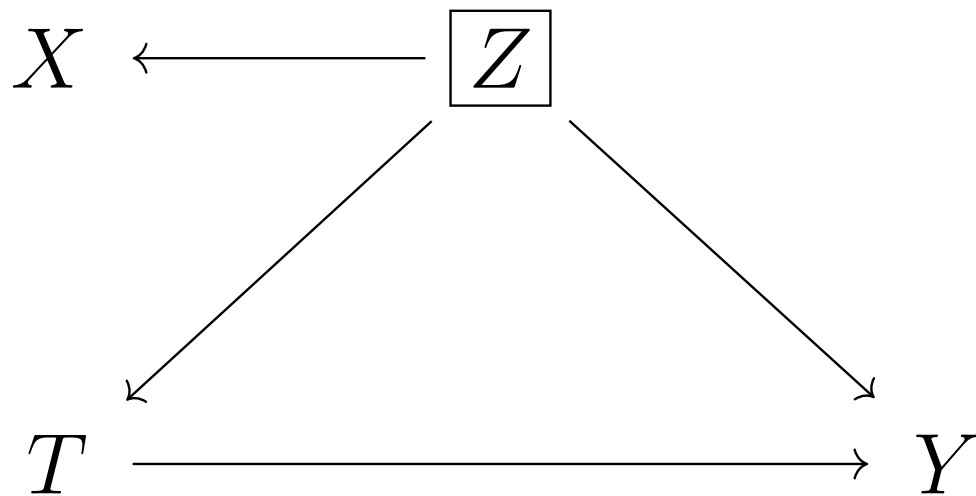
Composite loss then becomes:

$$\mathcal{F}_{\text{CEVAE}} = \mathcal{L} + \sum_{i=1}^N (\log q(t_i = t_i^* | \mathbf{x}_i^*) + \log q(y_i = y_i^* | \mathbf{x}_i^*, t_i^*)),$$

# A Special Case of Identifiable Latent Variable Models

## *Measurement bias and effect restoration in causal inference*

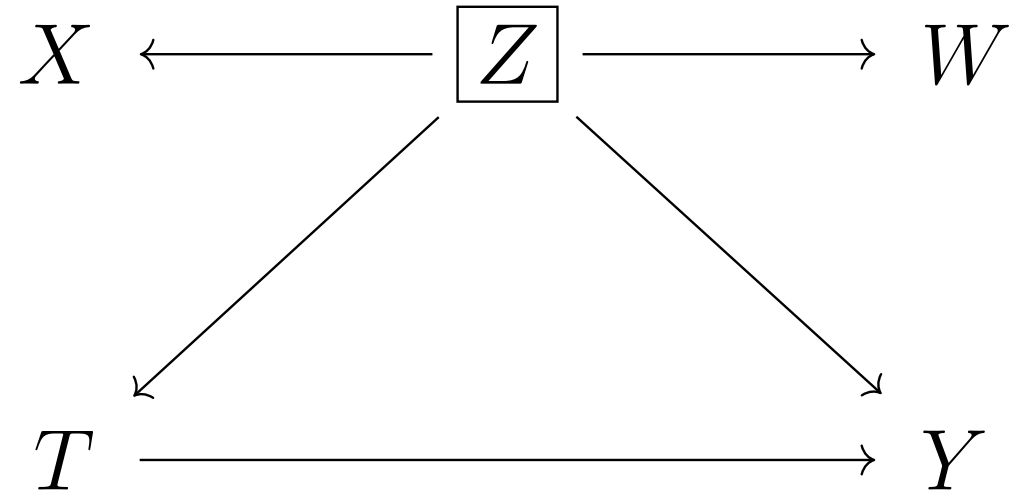
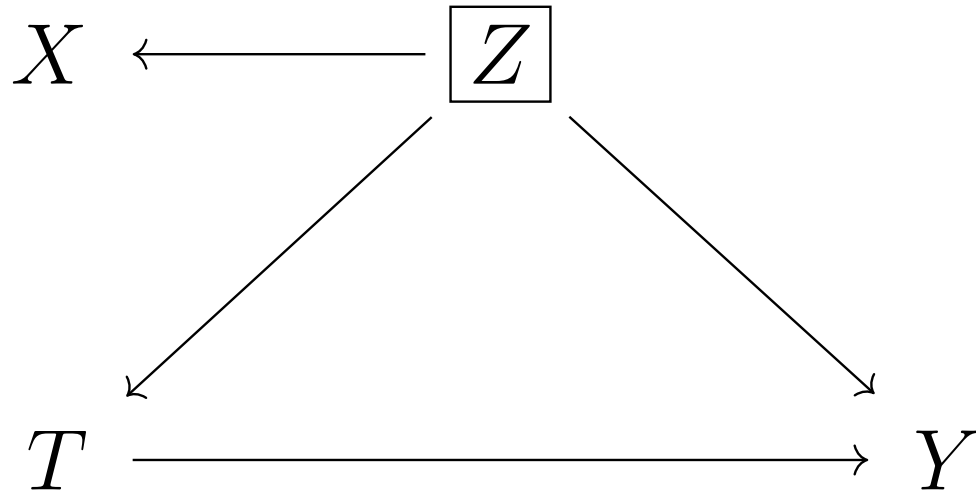
Kuroki, Pearl



# A Special Case of Identifiable Latent Variable Models

## *Measurement bias and effect restoration in causal inference*

Kuroki, Pearl

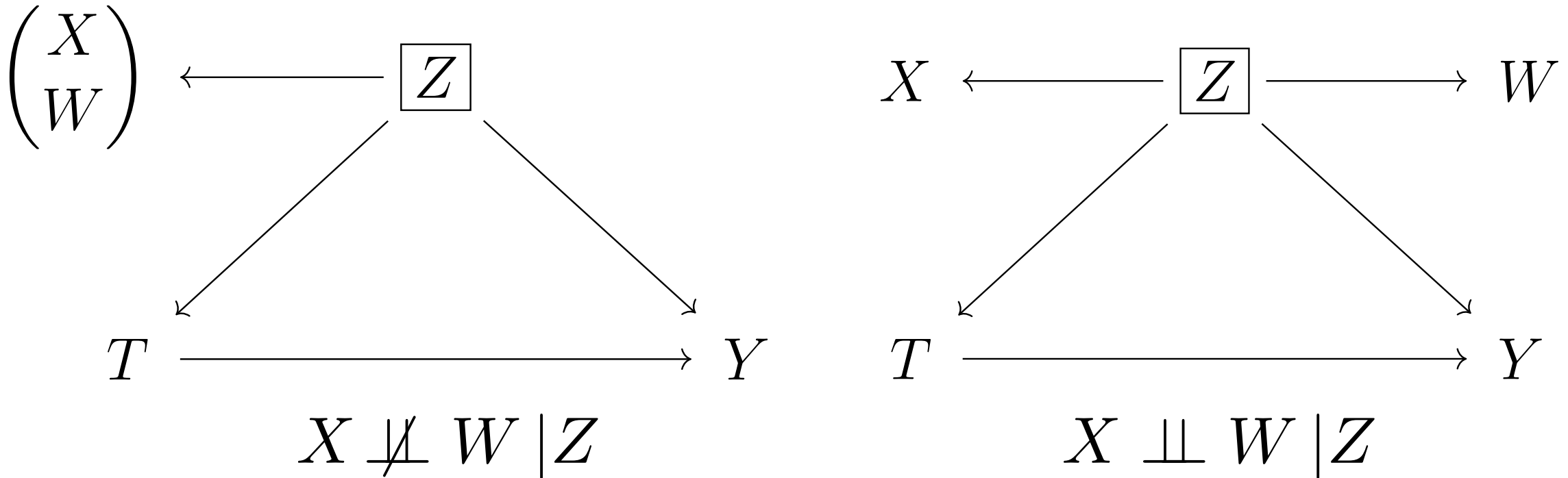


- Suppose we have two proxies  $X$ ,  $W$ .

# A Special Case of Identifiable Latent Variable Models

## *Measurement bias and effect restoration in causal inference*

Kuroki, Pearl



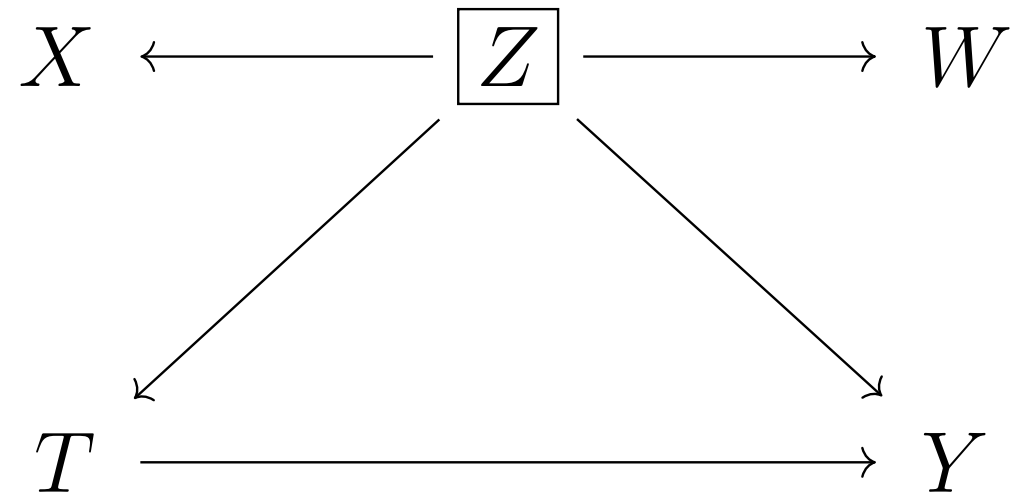
- Suppose we have two proxies  $X$ ,  $W$ .
- Note that this is different from having a single multivariate proxy.

# A Special Case of Identifiable Latent Variable Models

## *Measurement bias and effect restoration in causal inference*

Kuroki, Pearl 2014

- Under certain regularity conditions (assumptions) causal effect of  $T$  on  $Y$  is identifiable!
- Finite support discrete  $Z$ , or linear SCM.
- Expanded upon in 2018 by Miao, Geng, Tchetgen with relaxed assumptions.



# Testing CEVAE on Identifiable Two Proxy Setup

## *A Critical Look at the Consistency of Causal Estimation with Deep Latent Variable Models*

Rissanen, Marttinen

### **Proposition.**

*A linear CEVAE with a one-dimensional latent space estimates the causal effect correctly, given that it reaches the global optimum of the ELBO with infinite data.*

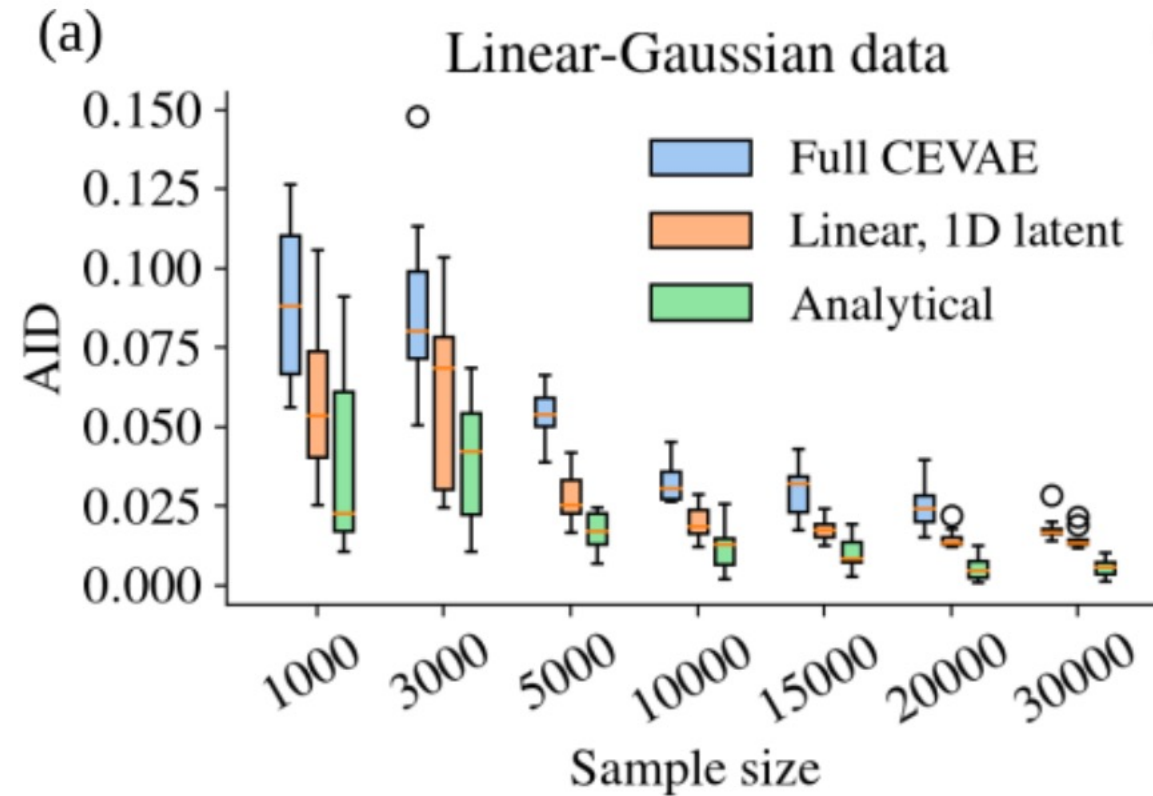
Correct VAE training → Correct causal effect estimation  
when latent is 1D in linear SCM

# Testing CEVAE on Identifiable Two Proxy Setup

## *A Critical Look at the Consistency of Causal Estimation with Deep Latent Variable Models*

Rissanen, Marttinen

Correct VAE training  
→  
Correct causal effect  
estimation  
when latent is 1D in linear  
SCM





# Testing CEVAE on Identifiable Two Proxy Setup

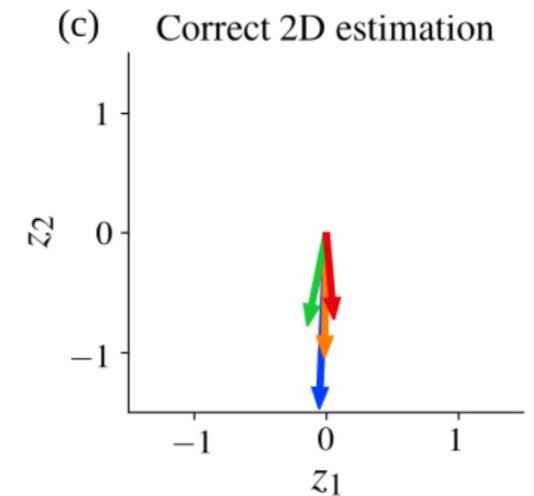
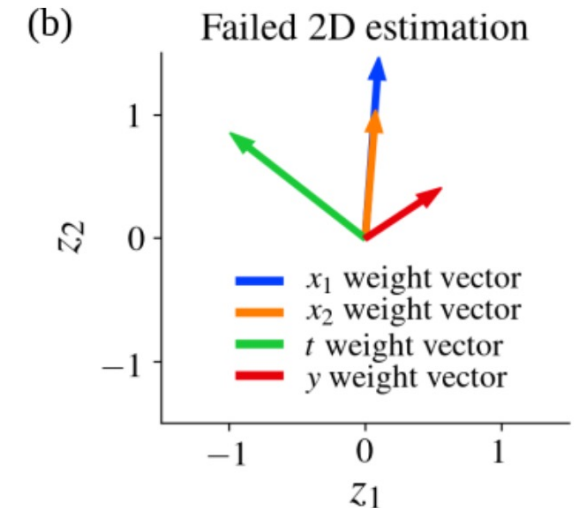
## *A Critical Look at the Consistency of Causal Estimation with Deep Latent Variable Models*

Rissanen, Marttinen

When VAE latent space is overparameterized, CEVAE may fail:

If we pick two dimensional CEVAE latent space, authors show ELBO minimization may be insufficient for correct causal effect estimation.

Posterior collapse helps pick one out of two latent dimensions but not always.



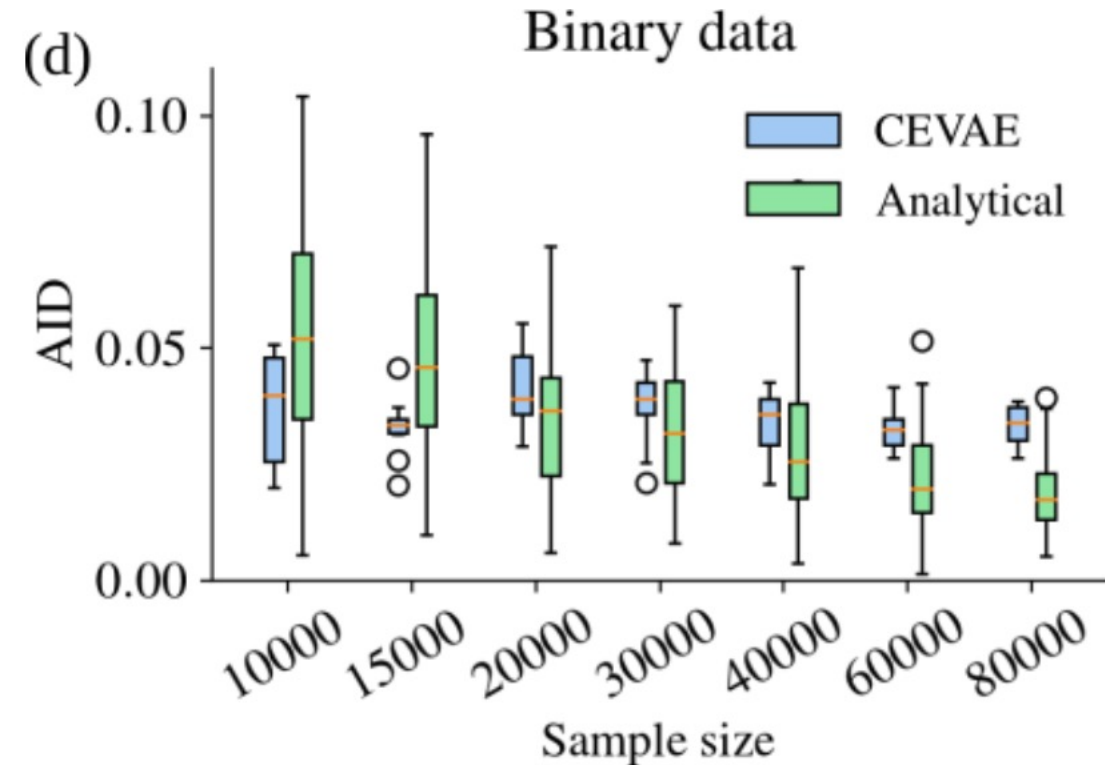
# Testing CEVAE on Identifiable Two Proxy Setup

## *A Critical Look at the Consistency of Causal Estimation with Deep Latent Variable Models*

Rissanen, Marttinen

When VAE latent space is misspecified, CEVAE may fail:

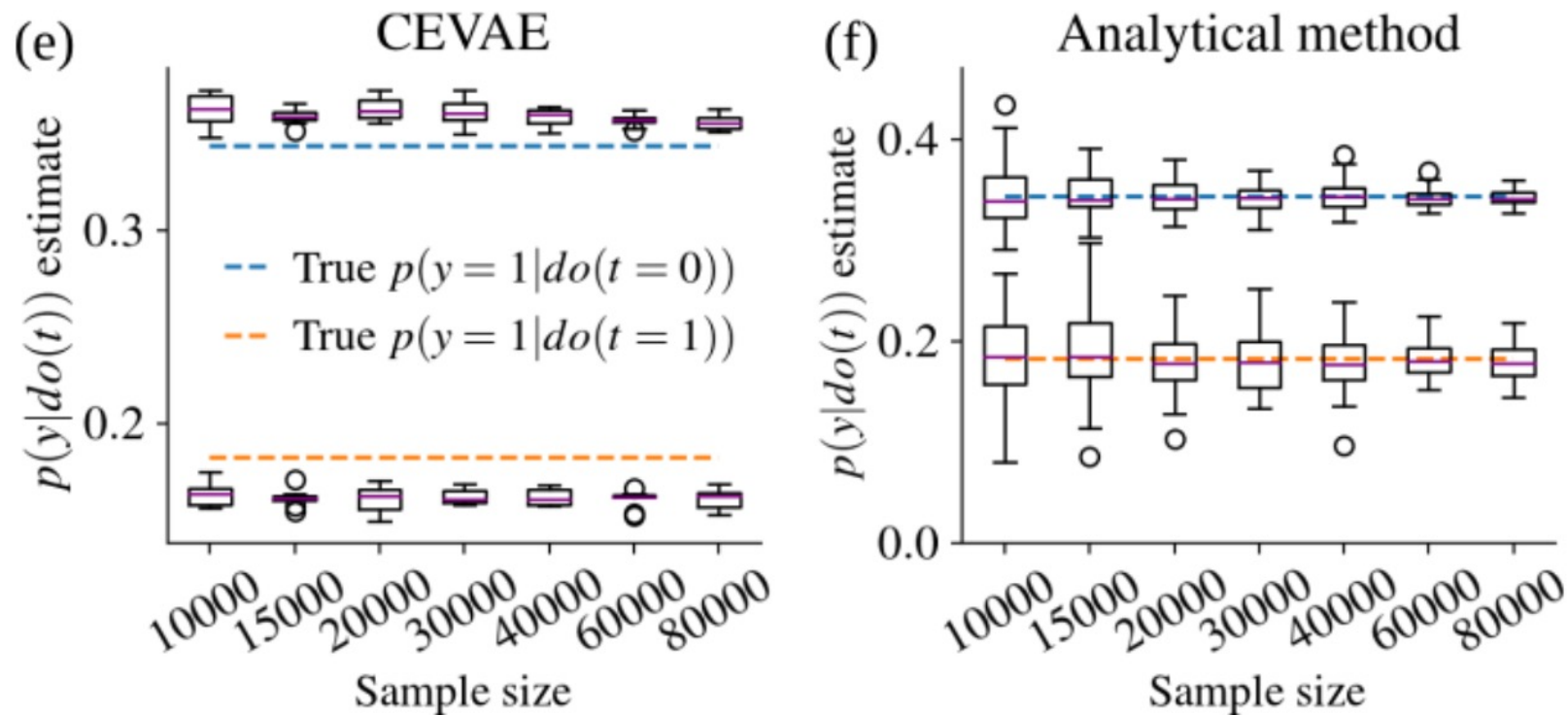
If true latent is binary but CEVAE assumes Gaussian latent, it may fail to converge.



# Testing CEVAE on Identifiable Two Proxy Setup

## *A Critical Look at the Consistency of Causal Estimation with Deep Latent Variable Models*

Rissanen, Marttinen

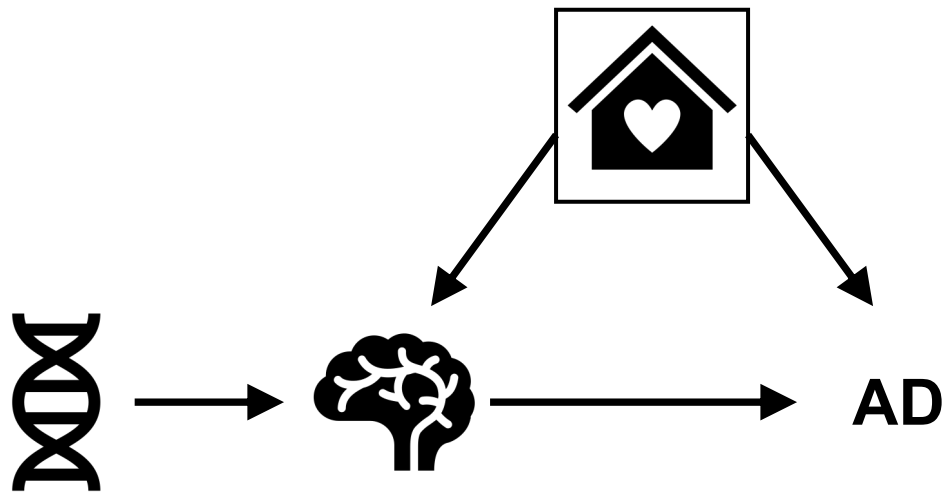


CEVAE may still be useful for decision-making even though effect estimates are not precise under misspecification.

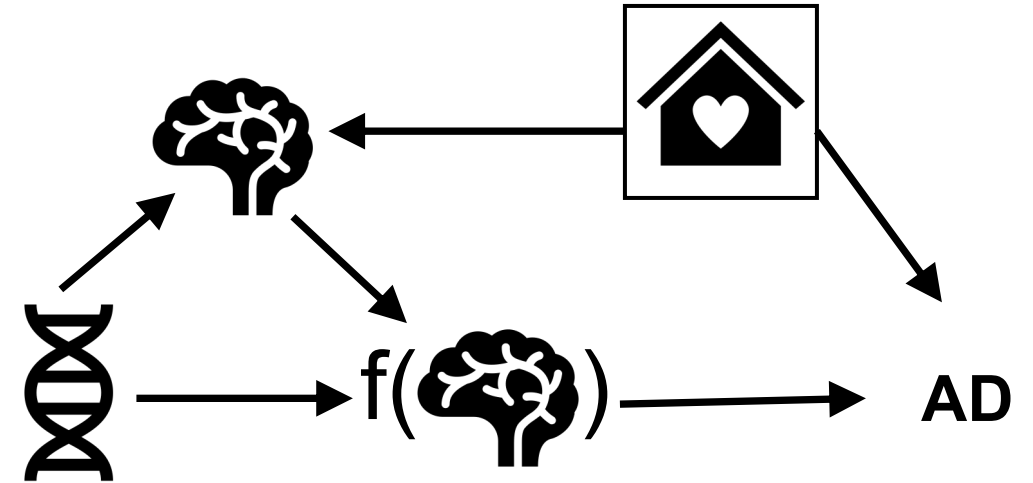
# *Deep causal feature extraction and inference with neuroimaging genetic data*

Yao, Chakraborty, Zhang, Shen, Pan

- Gene – Brain MRI – Alzheimer's Disease (AD)



*IV setup*

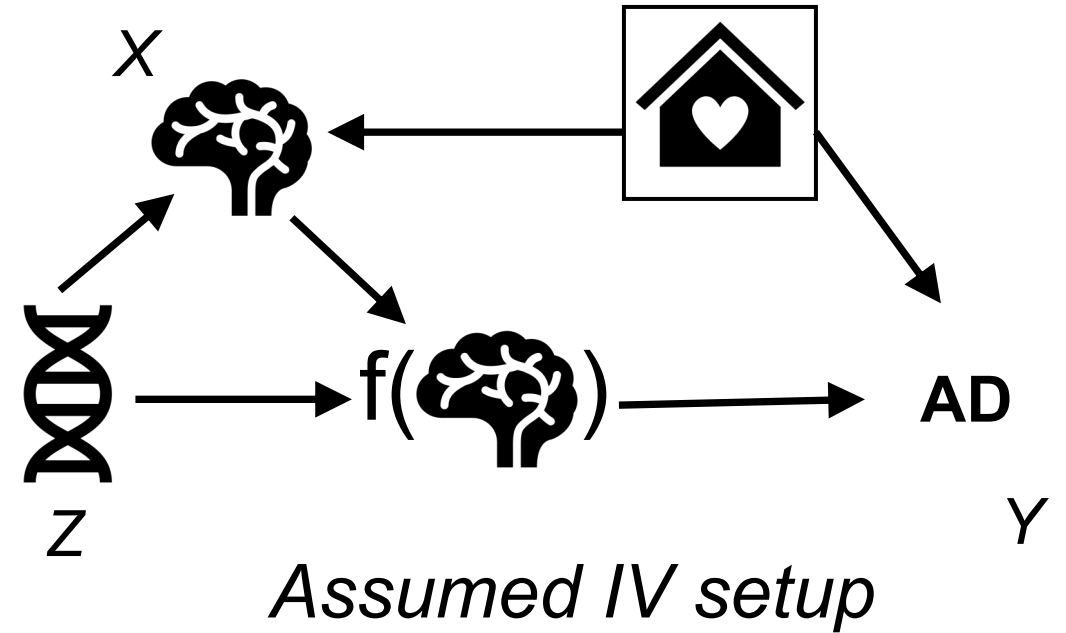


*Assumed IV setup*

# Deep causal feature extraction and inference with neuroimaging genetic data

Yao, Chakraborty, Zhang, Shen, Pan

- Gene – Brain MRI – Alzheimer's Disease (AD)
- High-dimensional brain imaging data not suitable as treatment in regular IV setting.
- Extract causal features from brain MRI images associated with causal genetic factors for AD.

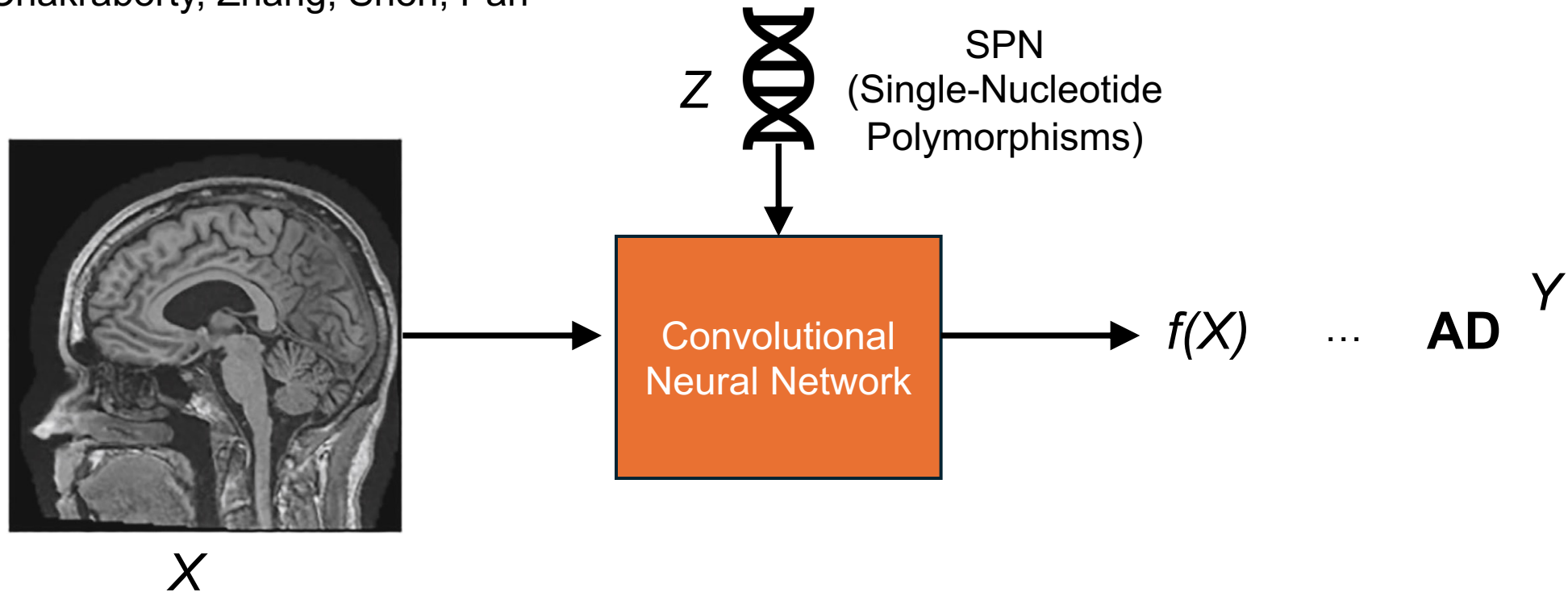


$$\text{stage 1 : } f(X) = B^T Z + U_1 + \Delta_1$$

$$\text{stage 2 : } Y = \beta^T f(X) + U_2 + \Delta_2$$

# Deep causal feature extraction and inference with neuroimaging genetic data

Yao, Chakraborty, Zhang, Shen, Pan



$$\min_{\theta, \beta} \frac{1}{n_b} \|\mathbf{Y}_b - \mathbf{Z}_b \hat{B}_\theta^b \beta\|_2^2 + \Omega(\theta, \beta),$$

$$\hat{B}_\theta = \min_B \frac{1}{n_b} \|f_\theta(\mathbf{X}_b) - \mathbf{Z}_b B\|_2^2 + \lambda \|B\|_2^2$$

# Many more interesting papers

Taghados *et al. BMC Cancer* (2025) 25:607  
<https://doi.org/10.1186/s12885-025-13926-2>

BMC Cancer

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. X, NO. X, XXX

1

## RESEARCH

## Open Access



# CausalCervixNet: convolutional neural networks with causal insight (CICNN) in cervical cancer cell classification—leveraging deep learning models for enhanced diagnostic accuracy

Zahra Taghados<sup>1</sup>, Zohreh Azimifar<sup>1\*</sup>, Malihezaman Monsefi<sup>2</sup> and Mojgan Akbarzadeh Jahromi<sup>3</sup>

## Deep Causal Reasoning for Recommendations

YAOCHEN ZHU, School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, China

JING YI, School of Computer Science, Wuhan University, Wuhan, China

JIAYI XIE, School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, China

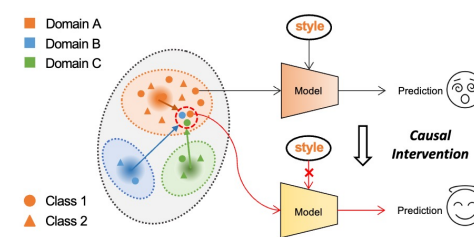
ZHENZHONG CHEN, School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, China

Traditional recommender systems aim to estimate a user's rating to an item based on observed ratings from the population. As with all observational studies, hidden confounders, which are factors that affect both item exposures and user ratings, lead to a systematic bias in the estimation. Consequently, causal inference has been introduced in recommendations to address the influence of unobserved confounders. Observing that confounders in recommendations are usually shared among items and are therefore multi-cause confounders, we model the recommendation as a multi-cause multi-outcome (MCMO) inference problem. Specifically, to remedy the confounding bias, we estimate user-specific latent variables that render the item exposures in-

## Casual Inference via Style Bias Deconfounding for Domain Generalization

Jiaxi Li, Di Lin, *Member, IEEE*, Hao Chen, *Senior Member, IEEE*, Hongying Liu, *Member, IEEE*, Liang Wan, *Member, IEEE*, and Wei Feng, *Member, IEEE*

**Abstract**—Deep neural networks (DNNs) often struggle with out-of-distribution data, limiting their reliability in diverse real-world applications. To address this issue, domain generalization methods have been developed to learn domain-invariant features from single or multiple training domains, enabling generalization to unseen testing domains. However, existing approaches usually overlook the impact of style frequency within the training set. This oversight predisposes models to capture spurious visual correlations caused by style confounding factors, rather than learning truly causal representations, thereby undermining inference reliability. In this work, we introduce Style Deconfounding Causal Learning (SDCL), a novel causal inference-based framework designed to explicitly address style as a confounding factor. Our approach begins with constructing a structural causal model (SCM) tailored to the domain generalization problem and applies a backdoor adjustment strategy to account for style influence. Building on this foundation, we design a style-guided expert



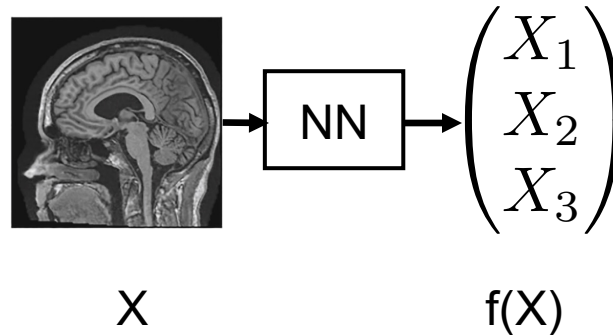
**Fig. 1** A schematic before and after causal intervention. Before intervention: The model relies on frequently occurring style types (orange oval) to make predictions. After intervention: Different style features from the source domain (green, blue) are fairly incorporated into the prediction of the current sample (orange), enabling the model to consider global styles comprehensively, thus eliminating style bias.

# A Taxonomy of Deep Learning Approaches for Causal Inference

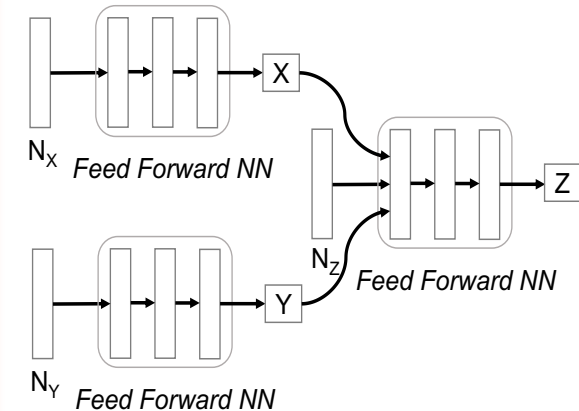
## Function Modeling

$$f : \mathbb{R}^k \rightarrow [0, 1]$$

## Feature Extraction



## Generative Modeling



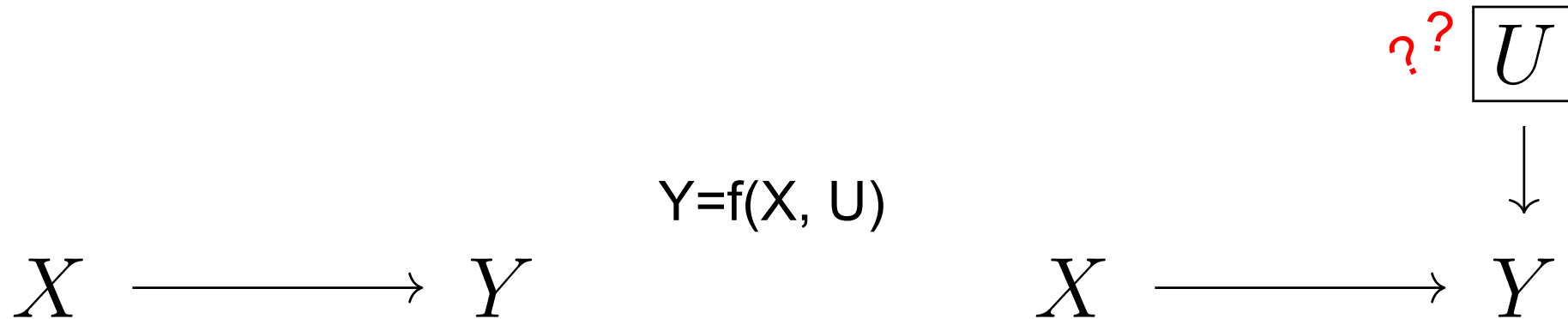


# Generative Causal Modeling

- A structural causal model contains a bunch of functions, read as assignment operators.
- We can try to approximate these functions to simulate the full SCM.
- But this is tricky because we do not have access to all variables affecting the system.

# Generative Causal Modeling

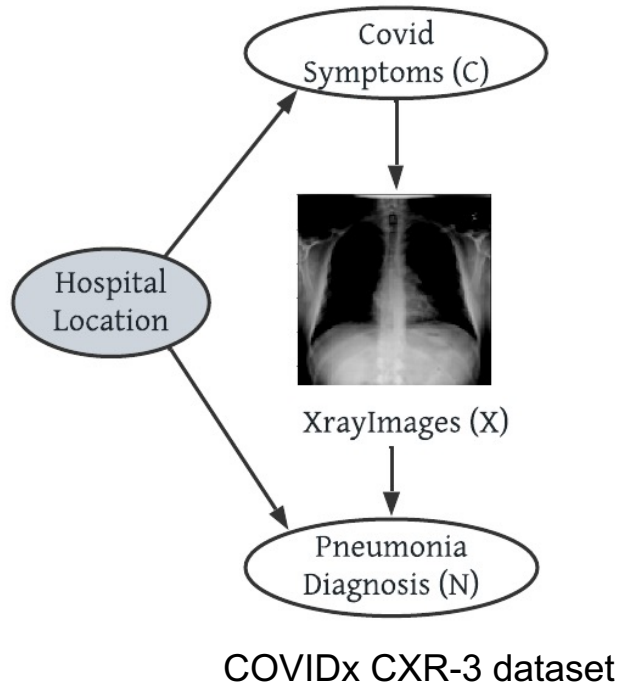
- Even without latent confounders, there are exogenous variables.



- Trying to learn true function in SCM is an impossible task!
- Maybe we can get away by choosing our own noise variable instead of true exogenous variable?
- Can we guarantee anything though?

# Need for Neural Nets with High-dimensional Data

- Estimating probability distributions in ID expressions is impractical

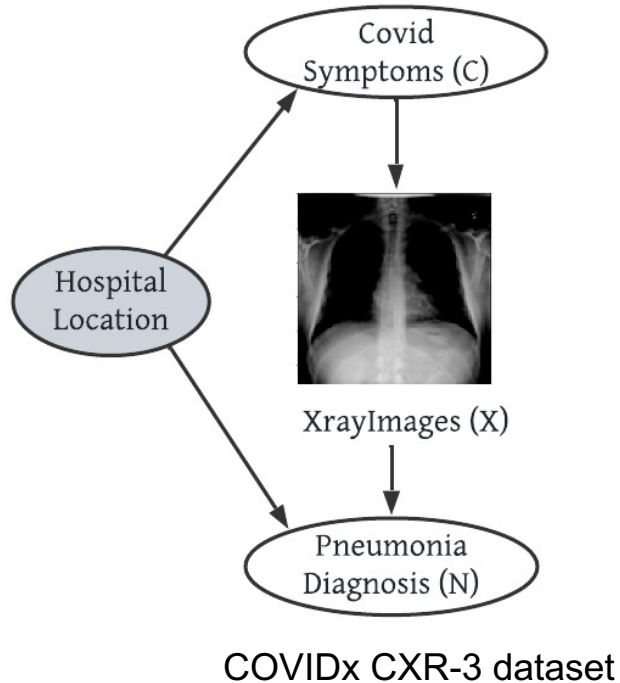


$$P(N|do(C)) = \sum_X P(X|C) \sum_{C'} P(N|X, C') P(C')$$

$$P(X|C)$$

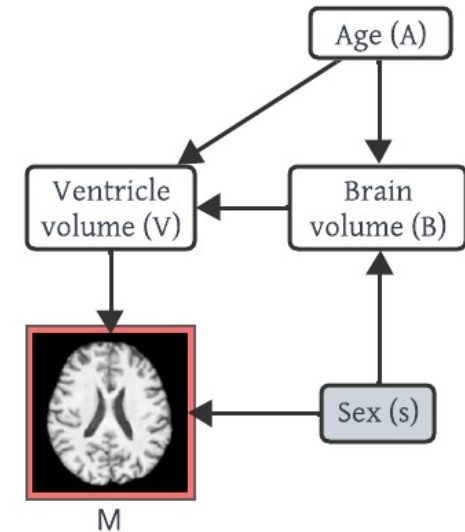
# Need for Neural Nets with High-dimensional Data

- Estimating probability distributions in ID expressions is impractical



$$P(N|do(C)) = \sum_X P(X|C) \sum_{C'} P(N|X, C') P(C')$$

$$P(X|C)$$



$$P(M|do(V)) = \sum_{A,B} P(M|V, A, B) P(A, B)$$

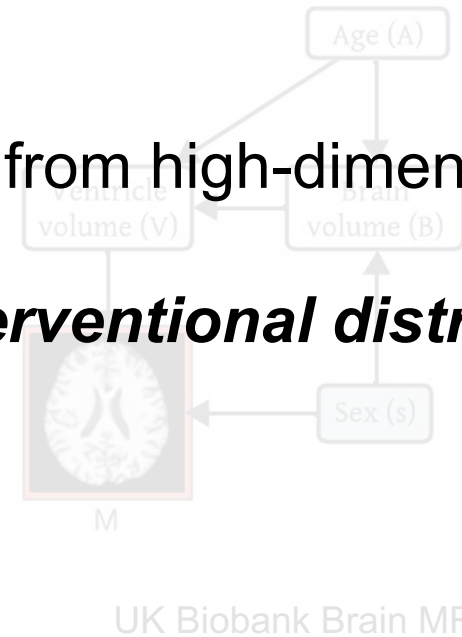
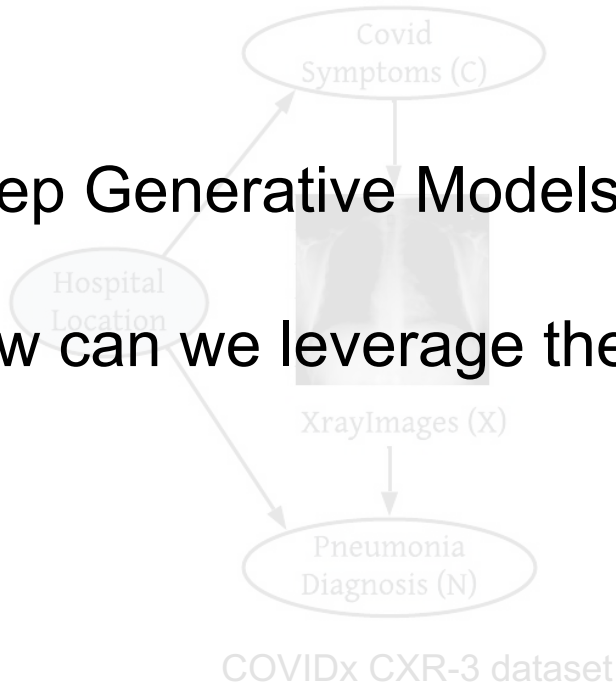
$$P(M|V, A, B)$$

# Need for Neural Nets with High-dimensional Data

- Estimating probability distributions in ID expressions is impractical

Deep Generative Models are great at **sampling** from high-dimensional distributions.

How can we leverage them to **sample from interventional distributions**?



$$P(N|do(C)) = \sum_X P(X|C) \sum_{C'} P(N|X, C') P(C')$$

$$P(X|C) = ? \text{ 😞}$$

$$P(M|do(V)) = \sum_{A,B} P(M|V, A, B) P(A, B)$$

$$P(M|V, A, B) = ? \text{ 😞}$$

# Key Idea

A structural causal model sequentially generates data in the causal order.

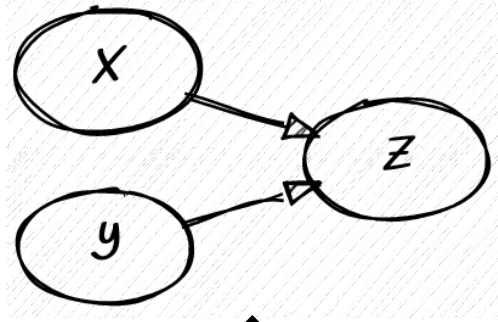
Each structural equation uses some observed and unobserved variables.

We can mimic this process using neural networks.

# Key Idea

## Real Data-generating Process

Causal Graph



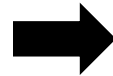
Structural Equations

(UNKNOWN)

$$X = f_X(E_X)$$

$$Y = f_Y(E_Y)$$

$$Z = f_Z(X, Y, E_Z)$$



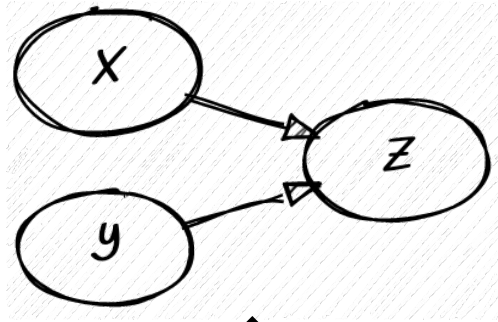
Real Data

$$\sim p(X, Y, Z)$$

# Key Idea

## Real Data-generating Process

Causal Graph

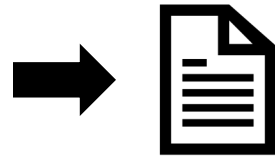


Structural Equations  
(UNKNOWN)

$$X = f_X(E_X)$$

$$Y = f_Y(E_Y)$$

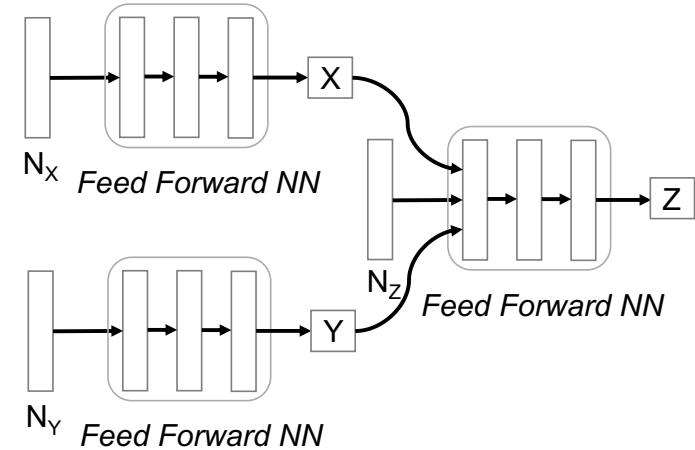
$$Z = f_Z(X, Y, E_Z)$$



Real Data

$$\sim p(X, Y, Z)$$

## Causal Simulator

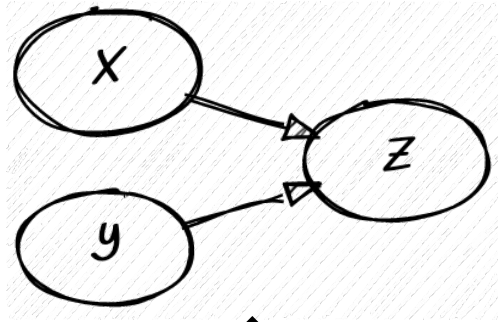




# Key Idea

## Real Data-generating Process

Causal Graph

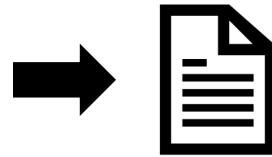


Structural Equations  
(UNKNOWN)

$$X = f_X(E_X)$$

$$Y = f_Y(E_Y)$$

$$Z = f_Z(X, Y, E_Z)$$



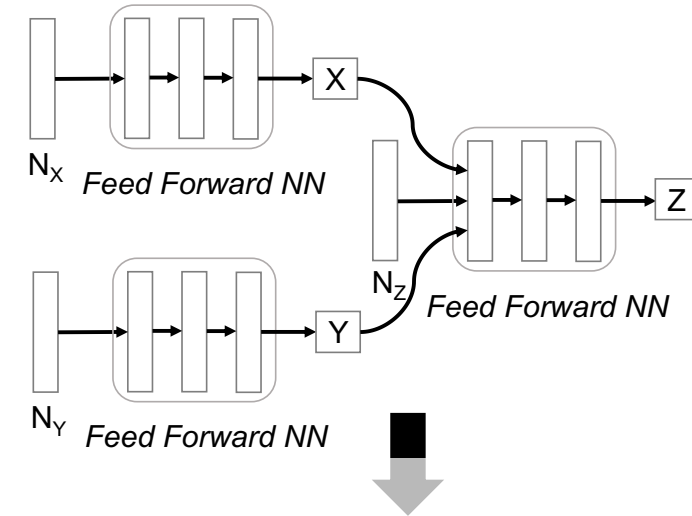
Real Data



Fake Data

$$p(X, Y, Z) \sim q(X, Y, Z)$$

## Causal Simulator



$$X = f'_X(E'_X)$$

$$Y = f'_Y(E'_Y)$$

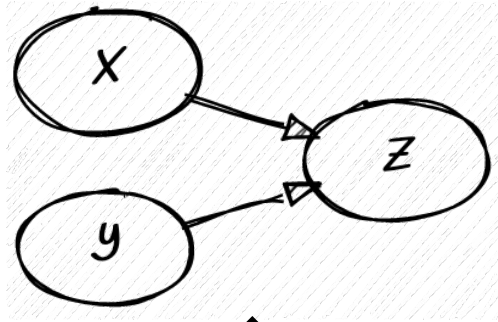
$$Z = f'_Z(X, Y, E'_Z)$$

UNKNOWN

# Key Idea

## Real Data-generating Process

Causal Graph

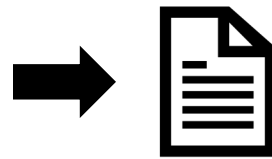


Structural Equations  
(UNKNOWN)

$$X = f_X(E_X)$$

$$Y = f_Y(E_Y)$$

$$Z = f_Z(X, Y, E_Z)$$



Real Data

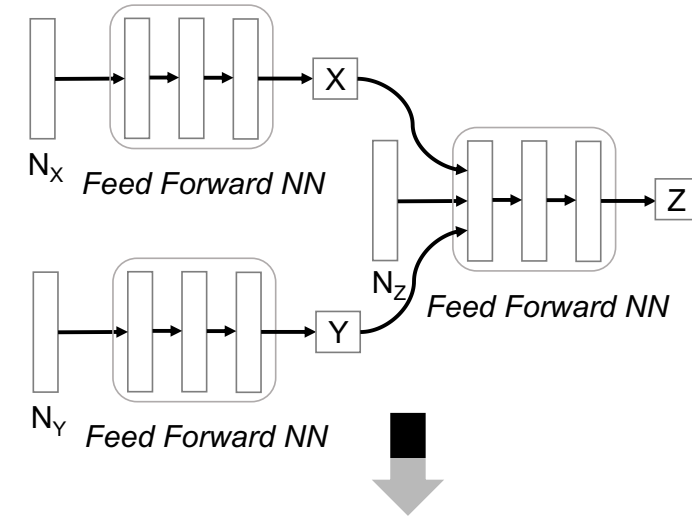


Fake Data

$$p(X, Y, Z) \text{ ..... } q(X, Y, Z)$$

How can we ensure  $p = q$  ?

## Causal Simulator



$$X = f'_X(E'_X)$$

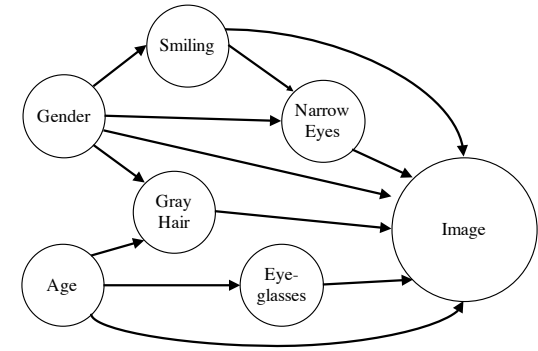
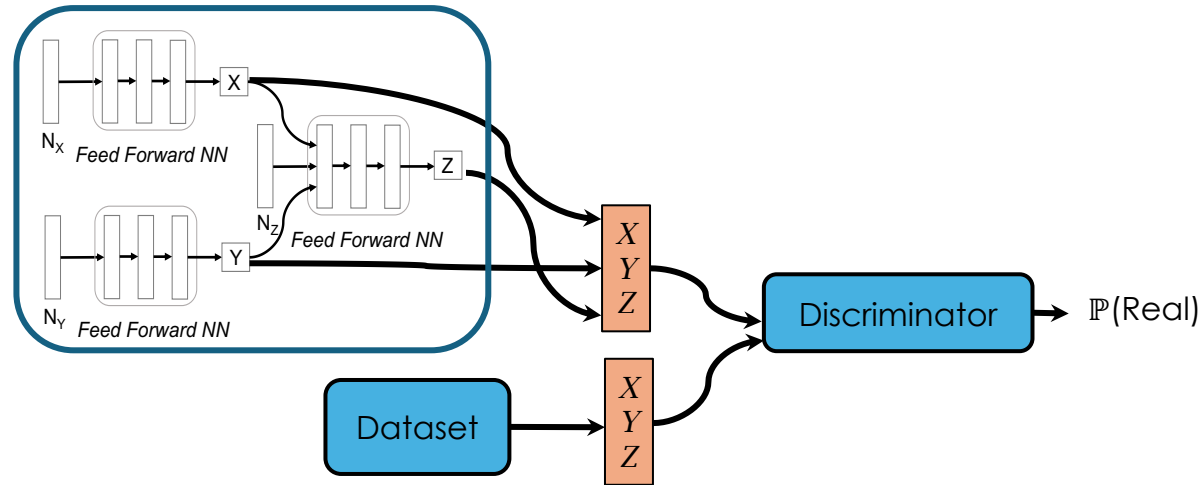
$$Y = f'_Y(E'_Y)$$

$$Z = f'_Z(X, Y, E'_Z)$$

**UNKNOWN**

# CausalGAN

Use GAN Training to Fit to the Observed Data



Conditioning vs. Intervening on *Mustache*

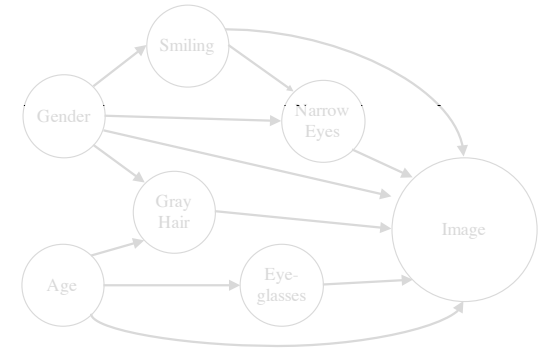
Assumption 1: Causal graph is given and has no latent variables.

Assumption 2: Distribution is strictly positive.

**Theorem:** *The optimal generator can be used to sample from any interventional distribution.*  
**Kocaoglu et al.**  
**ICLR'18**

# CausalGAN

Use GAN Training to Fit to the Observed Data



Conditioning vs. Intervening on *Mustache*

*Note that this simple idea extends to when latent confounders are present.*

*It extends even beyond interventional distributions (e.g., counterfactuals!).*

Assumption 1: Causal graph is given and has no latent variables.

Assumption 2: Distribution is strictly positive.

**Theorem:** *The optimal generator can be used to sample from any interventional distribution.*  
Kocaoglu et al.  
ICLR'18

## Claim:

*Any DCM (deep causal model) that entails observational distribution can be used to sample from any **identifiable** interventional distribution.*

## Claim:

*Any DCM (deep causal model) that entails observational distribution can be used to sample from any **identifiable** interventional distribution.*

## Proof

## Claim:

*Any DCM (deep causal model) that entails observational distribution can be used to sample from any **identifiable** interventional distribution.*

## Proof

i) Any identifiable  $p(y|\text{do}(x))$  is a fixed function of obs. joint  $p(v)$  given the causal graph:

$$p(y|\text{do}(x)) = g(p(v))$$

## Claim:

*Any DCM (deep causal model) that entails observational distribution can be used to sample from any **identifiable** interventional distribution.*

## Proof

i) Any identifiable  $p(y|\text{do}(x))$  is a fixed function of obs. joint  $p(v)$  given the causal graph:

$$p(y|\text{do}(x)) = g(p(v))$$

ii) DCM is just another SCM that entails the same causal graph as true SCM.



## Claim:

*Any DCM (deep causal model) that entails observational distribution can be used to sample from any **identifiable** interventional distribution.*

## Proof

i) Any identifiable  $p(y|\text{do}(x))$  is a fixed function of obs. joint  $p(v)$  given the causal graph:

$$p(y|\text{do}(x)) = g(p(v))$$

ii) DCM is just another SCM that entails the same causal graph as true SCM.

iii) Int. dist. induced by DCM  $q(y|\text{do}(x))$  is the same function of joint.

## Claim:

*Any DCM (deep causal model) that entails observational distribution can be used to sample from any **identifiable** interventional distribution.*

## Proof

i) Any identifiable  $p(y|\text{do}(x))$  is a fixed function of obs. joint  $p(v)$  given the causal graph:

$$p(y|\text{do}(x)) = g(p(v))$$

ii) DCM is just another SCM that entails the same causal graph as true SCM.

iii) Int. dist. induced by DCM  $q(y|\text{do}(x))$  is the same function of joint.

iv) If  $q(v) = p(v)$ , we have  $q(y|\text{do}(x)) = g(q(v)) = g(p(v)) = p(y|\text{do}(x))$



# Generative Causal Modeling

## *Takeaways*

- Given the causal graph of the system, parameterize the generative function for each observed variable with a neural net.
- Model latent confounders with high dimensional Gaussian noise.
- Fit (e.g., via GANs) the joint distribution as you optimize the neural networks that generate the observed variables.
- Any identifiable causal query is correctly sampled from after convergence.

# Pros and Cons of Generative Causal Modeling

## Pros

Non-parametric, can model a rich, non-restrictive class of structural causal models.

A trained model can be used to answer any identifiable causal question.

Pretty much only way to handle the presence of high-dim variables (e.g., images)

## Cons

Fitting high-dimensional distribution is often challenging (e.g., GAN convergence issues).

Sampling based estimation, no closed-form expression.

Feedforward nature makes anti-causal sampling challenging (rejection sampling).

**End of Part I.**

**Come back for Part II for some cool image problems and  
state of the art generative models!**

**Questions?**

# UAI 2025 Tutorial (Part 2)

Murat Kocaoglu  
Purdue University  
(@JHU in Fall 2025)

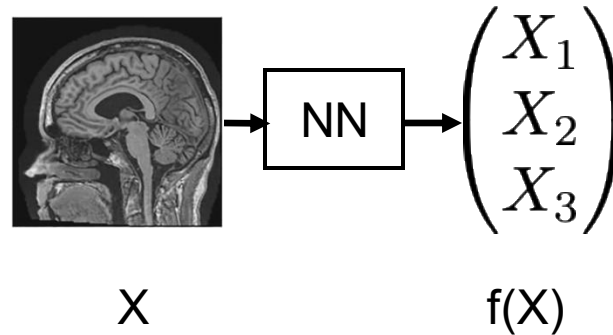
**Md Musfiqur Rahman**  
**Purdue University**

# A Taxonomy of Deep Learning Approaches for Causal Inference

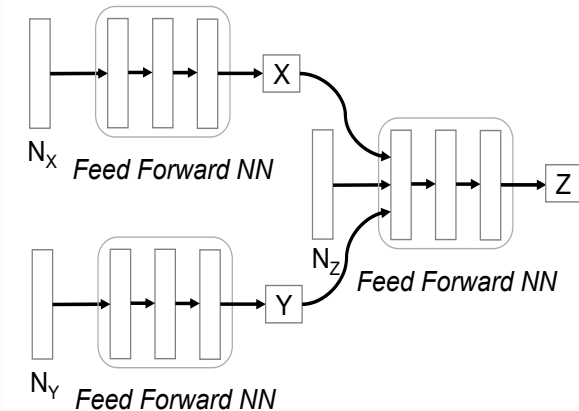
## Function Modeling

$$f : \mathbb{R}^k \rightarrow [0, 1]$$

## Feature Extraction



## Generative Modeling



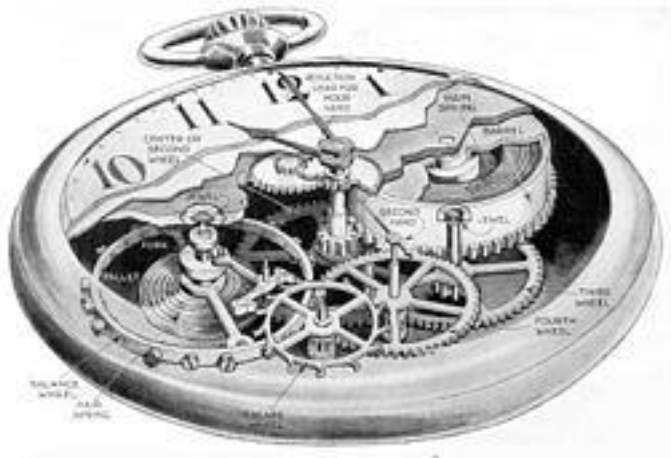
# Outline

- Structural Causal Models (SCMs)
- Neural Networks with SCMs
- Causal Inference with Tabular Data – Interventional Sampling
- Causal Inference with Images – Interventional Sampling
- Causal Inference with Tabular Data – Counterfactuals
- Counterfactual Inference with Empirical Success – Relaxing Theoretical Guarantees
- Causal Inference with Images – Counterfactuals

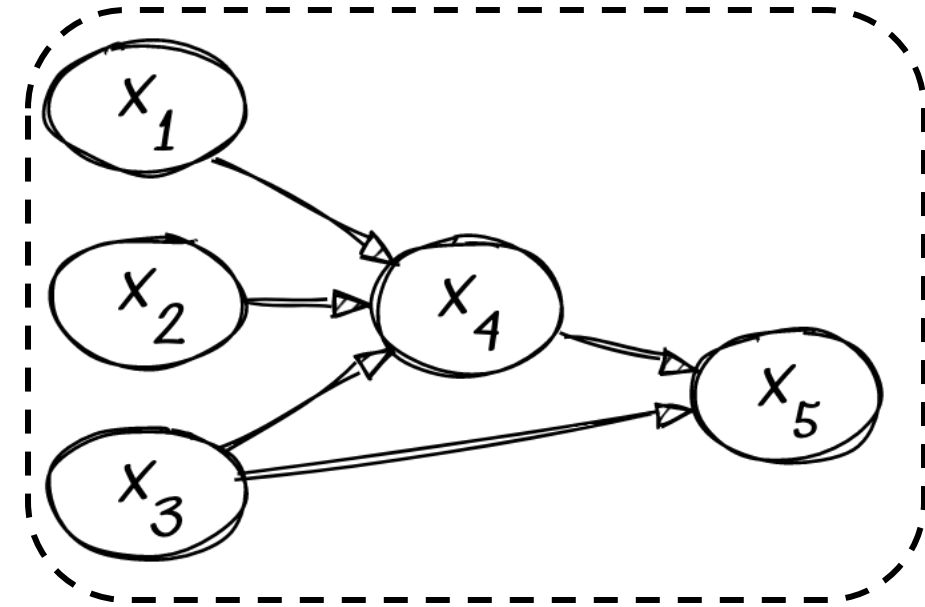


# Structural Causal Models (SCM)

# Structural Causal Models (SCM)



Causal Graph



**Vertices:** Random variables

**Edges:** Causal relations

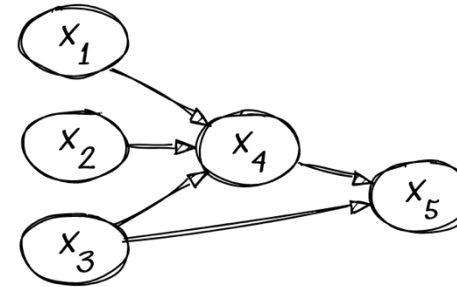
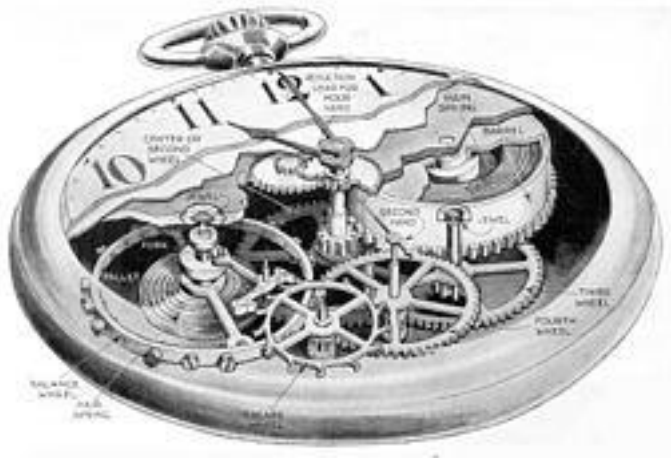
$$X_i = f_i(Pa_i, E_i)$$

$Pa_i$  : Set of parents of  $X_i$  in the causal graph

**"Structural Causal Model" (SCM):** A model in which each variable is a function of its parent variables and independent "exogenous" random variables.

# Structural Causal Models (SCM)

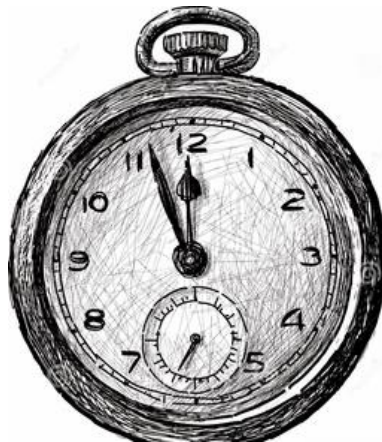
Unknown  
ground truth  
SCM



$$X_i = f_i(Pa_i, E_i)$$

$Pa_i$  : Set of parents of  $X_i$  in the causal graph

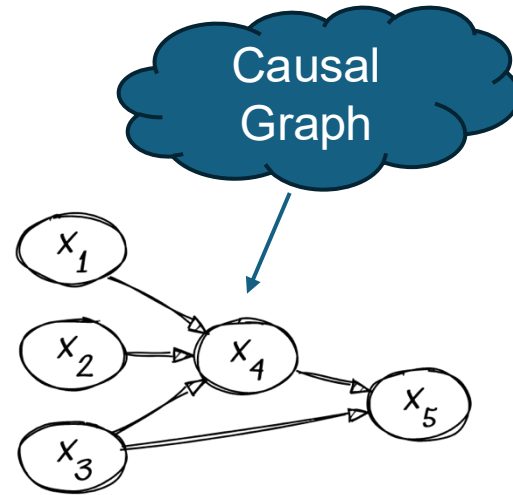
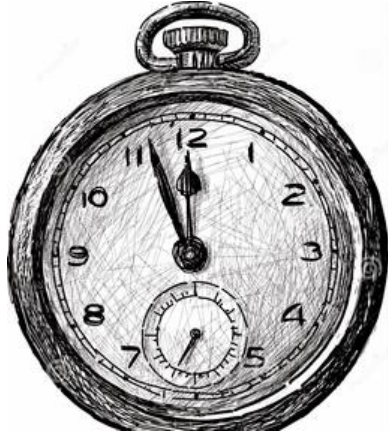
Observed



Observational data  
generated from the SCM

X1	X2	X3	X4	X5
...	...	...	...	...
...	...	...	...	...

# Structural Causal Models (SCM)

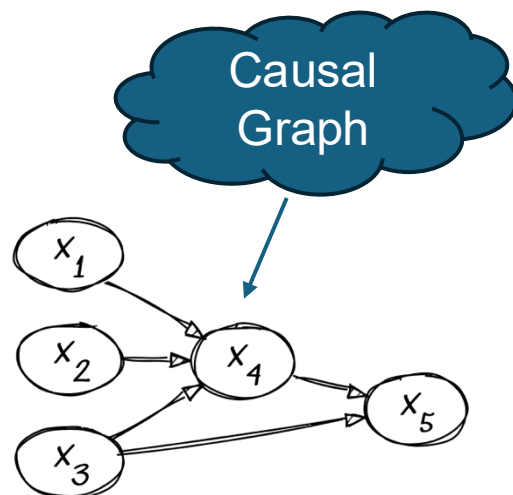
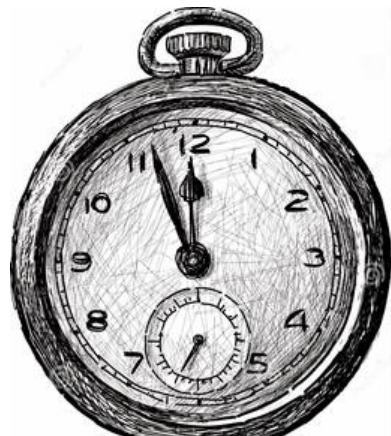


Learning

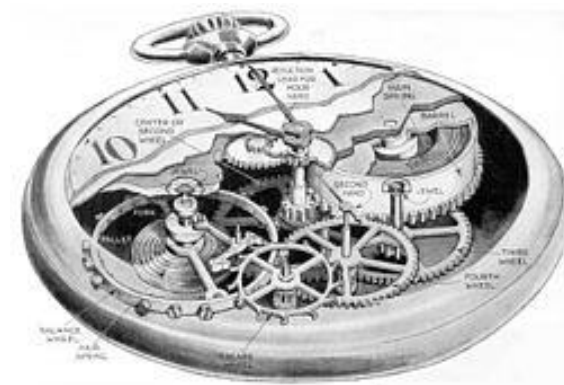
X1	X2	X3	X4	X5
...	...	...	...	...
...	...	...	...	...



# Structural Causal Models (SCM)



Learning



$$X_i = f_i(Pa_i, E_i)$$

X1	X2	X3	X4	X5
...	...	...	...	...
...	...	...	...	...

Training Data

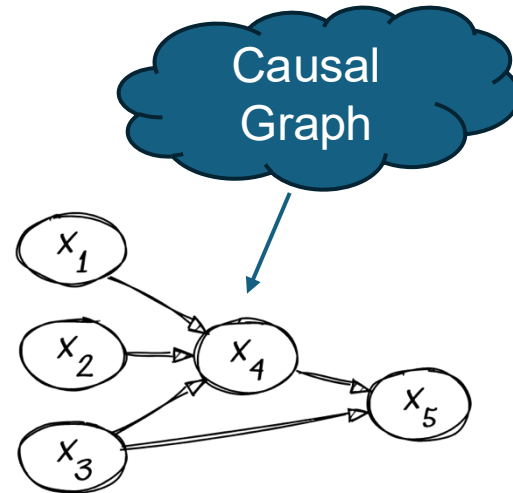
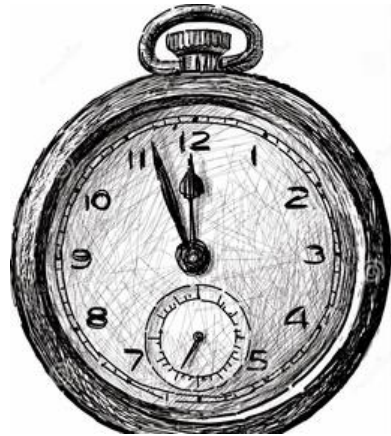
Approximate Structural functions

$$X_i = \hat{f}_i(Pa_i, *)$$

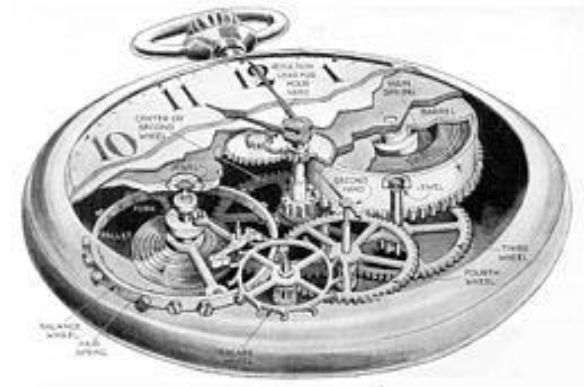
# Proxy of the Data Generating Process

- We can learn a proxy of the structural causal model.
- Why is it a proxy? Why can't we learn the original SCM?

# Structural Causal Models (SCM)



Learning



$$X_i = \boxed{f_i}(Pa_i, E_i)$$

X1	X2	X3	X4	X5
...	...	...	...	...
...	...	...	...	...

Training Data

Approximate Structural functions

$$X_i = \hat{f}_i(Pa_i, *)$$

Its challenging because we **can never know** the true environment variables  $E_i$  and thus the true functions  $f_i$  !

# Structural Causal Models (SCM)

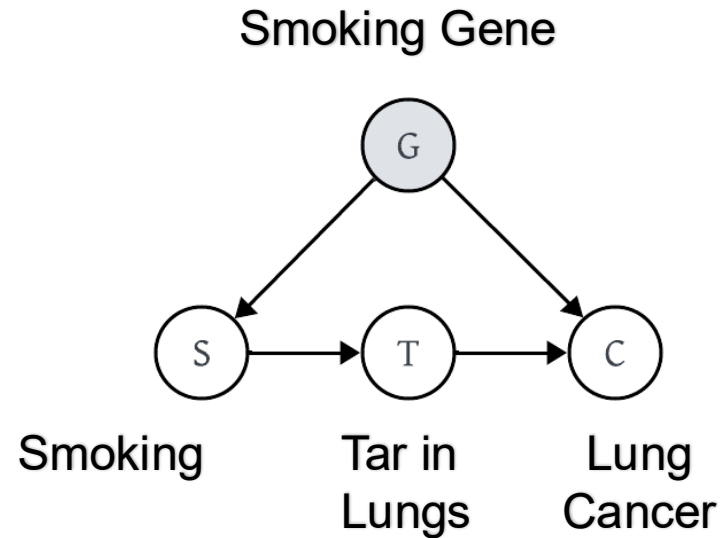
After learning an approximation of the SCM, we can estimate the associated

- Observational distribution
- Interventional distribution
- Counterfactual distribution



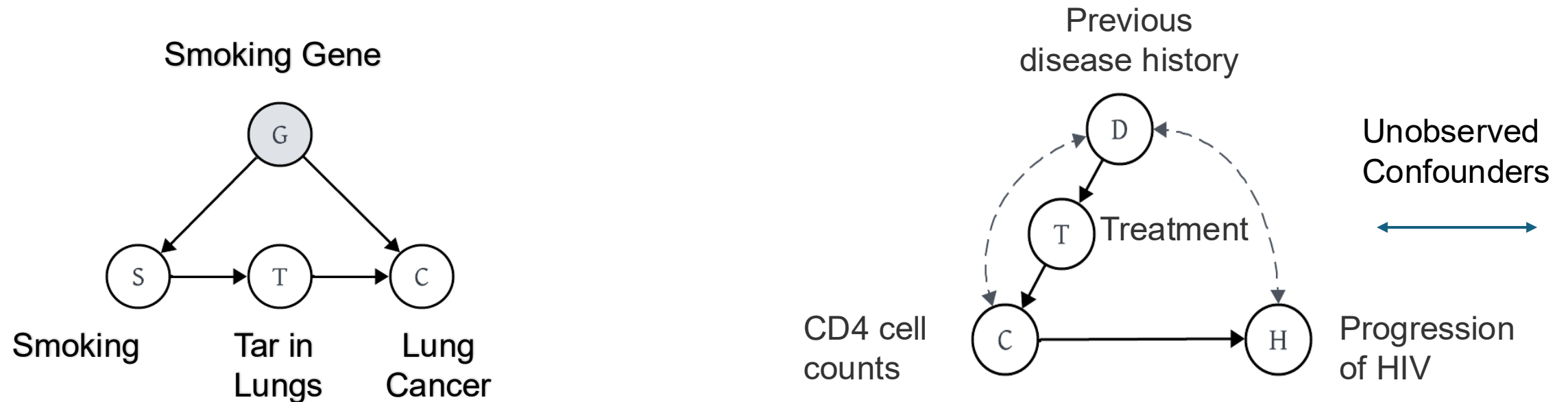
# SCMs with Neural Networks

# Estimating causal effect from data $D \sim P(V)$



Causal effect of smoking on Lung cancer,  $P(C|do(S)) = ?$

# Estimating causal effect from data $D \sim P(V)$

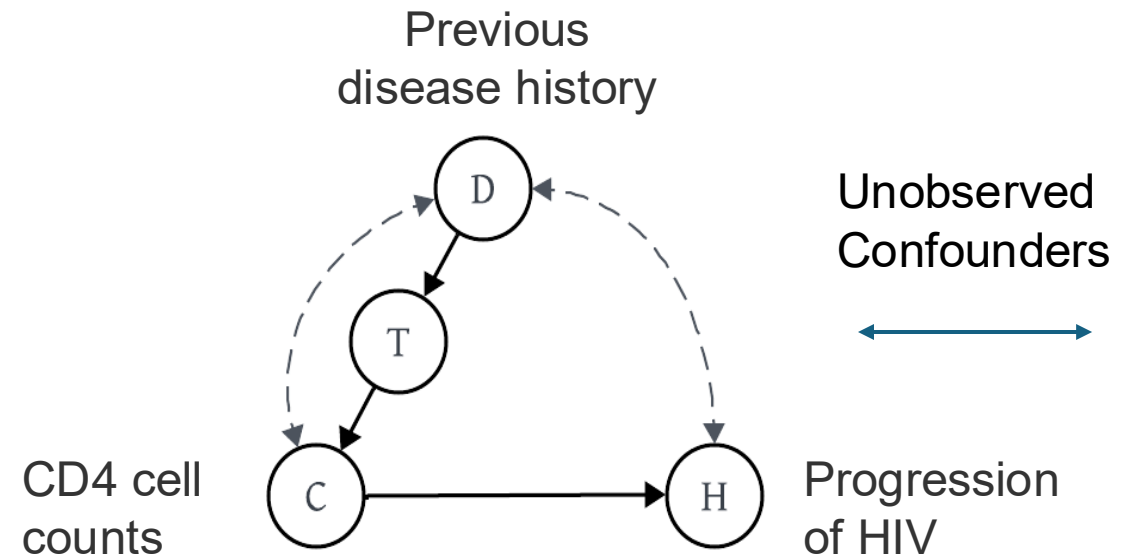
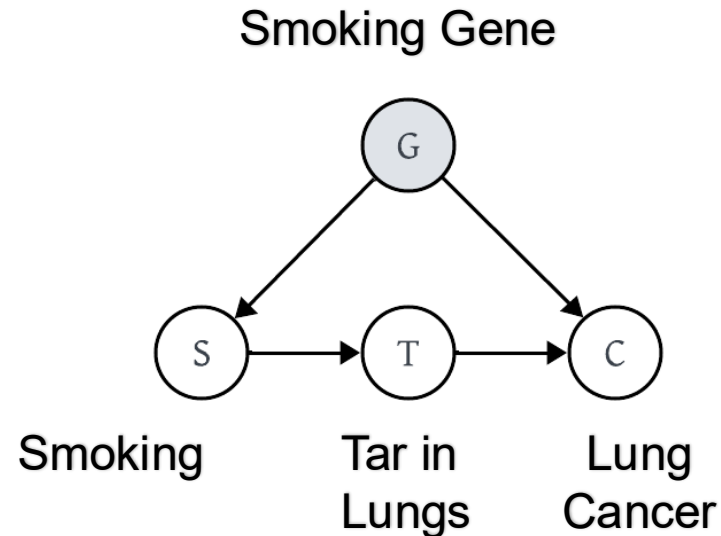


Learning causal effects via weighted empirical risk minimization, jung2020learning

Causal effect of CD4 cell counts on progression of HIV,  $P(H|do(C)) = ?$

# Estimating causal effect

Identification algorithms by Shpitser et al 2008, Tian et al 2002.

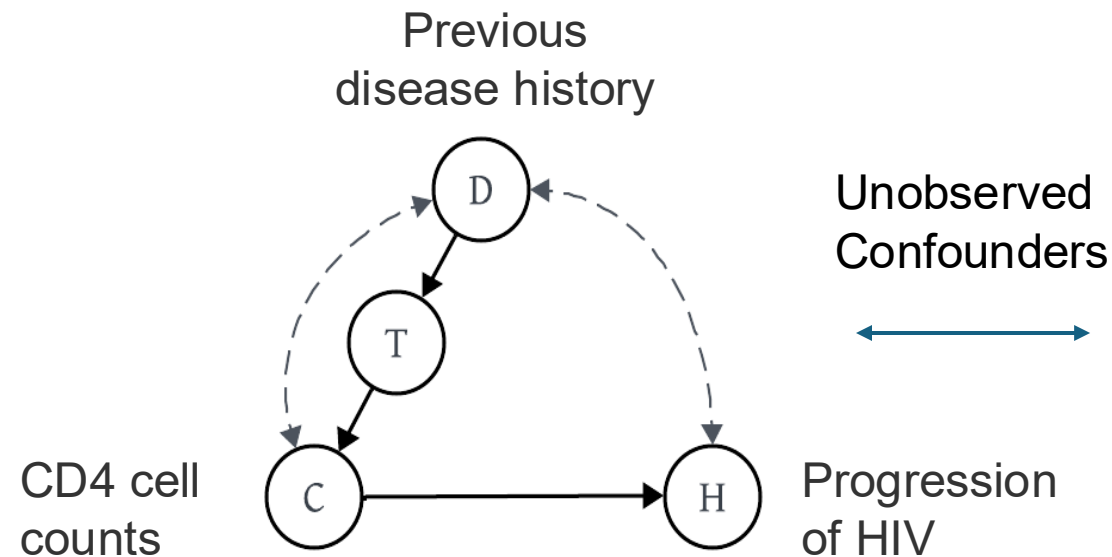
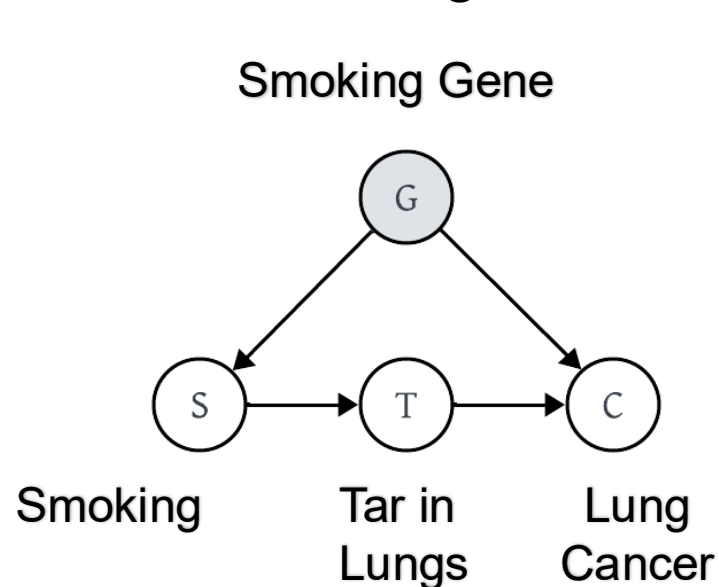


Learning causal effects via weighted empirical risk minimization, jung2020learning

$$P(C|do(S)) = \sum_T P(T|S) \sum_{S'} P(C|S', T) P(S')$$

# Estimating causal effect

Identification algorithms by Shpitser et al 2008, Tian et al 2002.



Learning causal effects via weighted empirical risk minimization, jung2020learning

$$P(C|do(S)) = \sum_T P(T|S) \sum_{S'} P(C|S', T) P(S')$$

$$P(H|do(C)) = \frac{\sum_D P(C, H|D, T) P(D)}{\sum_D P(C|D, T) P(D)}$$

# Why do we want to learn the SCM?

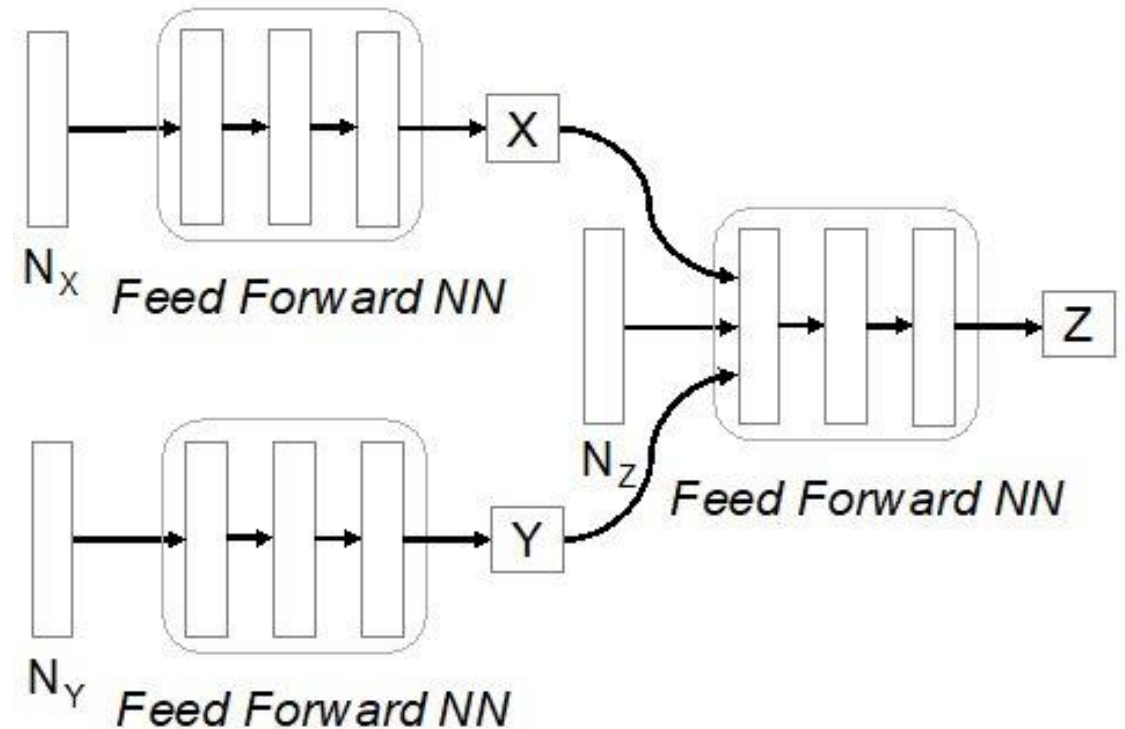
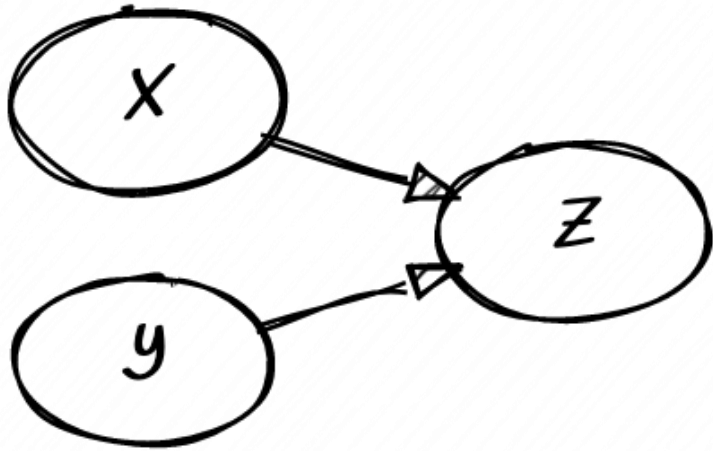
- Identification algorithms are **query-specific**.
- For a new query, the estimation might need to be done **from scratch**.
- With a learned SCM, we can answer any identifiable causal questions.

# Why do we use Neural Networks for learning SCMs?

- Due to the expressive power of neural networks, we can match arbitrary observational distributions
- *(e.g., data might come from linear, nonlinear, additive, non-additive, or nonparametric SCMs).*

# How do we use Neural Networks for SCMs?

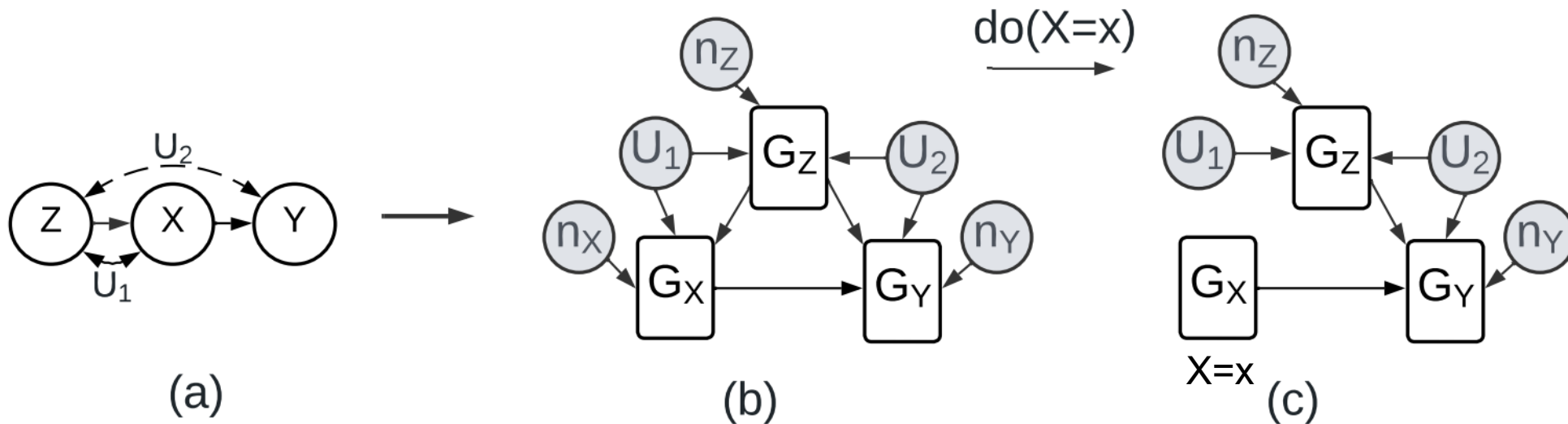
- No unobserved confounders





# Structural Causal Model with Neural Networks

- Unobserved confounders ( represented as bi-directed edges)



Before and after intervention.

# Causal Inference with Tabular Data

*Interventional Sampling*

# Problem Definition

- Suppose we have collected a dataset.

If we train an estimator on it, the prediction is unfair (e.g., gives more priority to a specific subgroup of the population due to their specific attributes or variables).

- Can we learn an SCM from the unfair data and perform interventions to generate fair synthetic data?

# DECAF: Generating Fair Synthetic Data Using Causally-Aware Generative Networks

Van Breugel, B., Kyono, T., Berrevoets, J., & Van der Schaar, M , 2021

- Generates fair synthetic data from unfair data
- By learning underlying data-generating process (DGP/SCM) with GAN training on tabular data.

# Conditional Fairness

- $X$ : random variables with distribution  $P(X)$
- $A$ : protected/sensitive attributes
- $Y$ : Target variable

**Conditional Fairness (CF):** A predictor  $\hat{Y}$  is said to be *conditionally fair* if the sensitive attribute  $A$  is conditionally independent of the prediction  $\hat{Y}$  given  $R$ , i.e.,

$$A \perp\!\!\!\perp \hat{Y} \mid R,$$

which implies:

$$\forall r, a, a' : \quad P(\hat{Y} \mid R = r, A = a) = P(\hat{Y} \mid R = r, A = a').$$

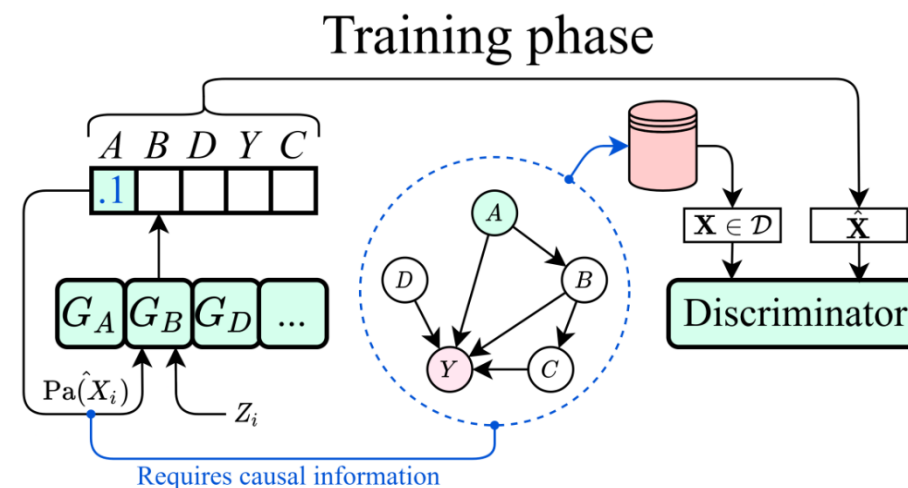
# DECAF

- Features are generated sequentially following the topological ordering of the underlying causal DAG
- $\hat{Pa}(X_i)$  are generated causal parents
- $Z_i$  is independently sampled from  $P(Z)$ . (e.g. standard Gaussian)

$$\hat{X}_i = G_i(\hat{Pa}(X_i), Z_i) \quad \forall i,$$

# DECAF: Model Training

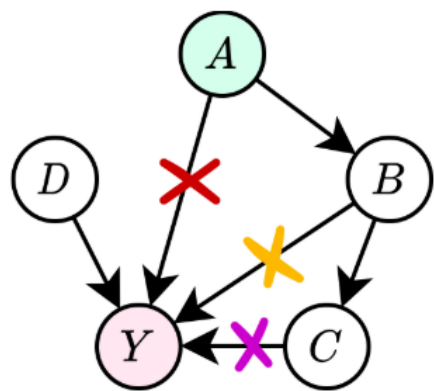
- Generates Fake Data.
- Compares with the real data with a discriminator.



$$\max_{\{G_i\}_{i=1}^d} \min_D \mathbb{E}[\log D(G(Z)) + \log(1 - D(X))],$$

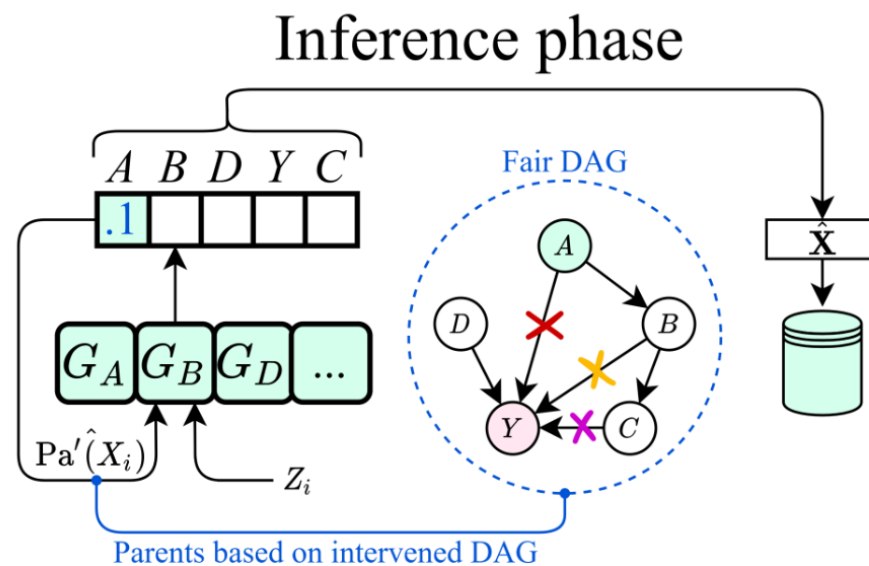
# DECAF: Inference

Uniformly Sample parents of the deleted edges.



when  $R = C$ : XX;  
when  $B \in R$ : X

$$A \perp\!\!\!\perp \hat{Y} \mid R,$$





# Problem Definition

- Can we use neural networks to understand if a causal effect can be estimated uniquely from given observational data? i.e., test for identifiability.
- If so, can we obtain the estimate at the same time?

# **The Causal-Neural Connection: Expressiveness, Learnability, and Inference**

Xia, K., Lee, K. Z., Bengio, Y., & Bareinboim, E. (2021)

- Follows the same architecture proposed by CausalGAN.
- Performs identification of interventional queries with minimization and-maximization.
- Identification of Interventional queries in presence of unobserved confounders.

# NCM Identifiability

We search for our solution SCMs in two different runs.

Each run maintains two loss functions:

- Loss 1: Matches observational distribution.
- Loss 2 : Optimizes the interventional query (maximize or minimize)

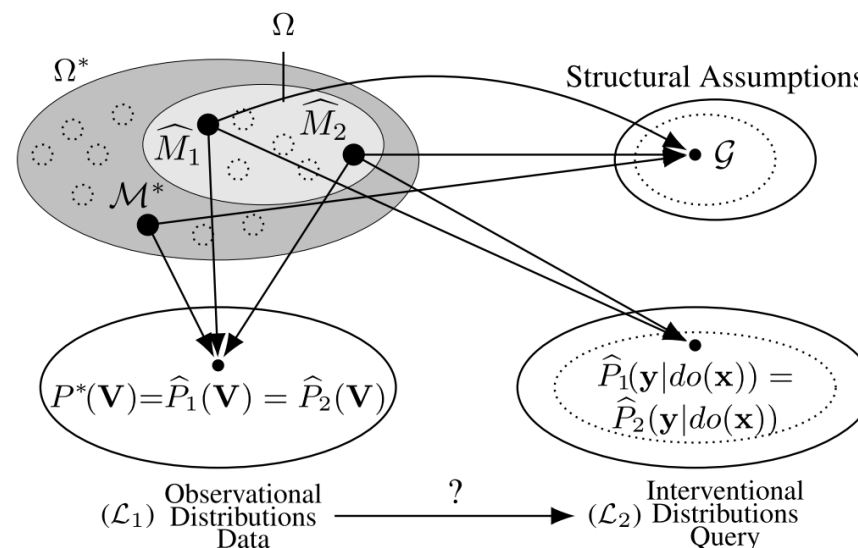


Figure :  $P(\mathbf{y} | do(\mathbf{x}))$  is identifiable from  $P(\mathbf{V})$  and  $\Omega(\mathcal{G})$  if for any SCM  $\mathcal{M}^* \in \Omega^*$  and NCMs  $\widehat{M}_1, \widehat{M}_2 \in \Omega$  (top left),  $\widehat{M}_1, \widehat{M}_2, \mathcal{M}^*$  match in  $P(\mathbf{V})$  (bottom left) and  $\mathcal{G}$  (top right), then the NCMs  $\widehat{M}_1, \widehat{M}_2$  also match in  $P(\mathbf{y} | do(\mathbf{x}))$  (bottom right).

# NCM: Architecture & loss/objective

- Matches both observational and maximizes/minimizes the interventional distribution at the same time.

Loss 1:

$$\begin{aligned}\boldsymbol{\theta} &\in \arg \min_{\boldsymbol{\theta}} \mathbb{E}_{P^*(\mathbf{v})} \left[ -\log P^{\widehat{M}(\mathcal{G};\boldsymbol{\theta})}(\mathbf{v}) \right] \\ &\approx \arg \min_{\boldsymbol{\theta}} \frac{1}{n} \sum_{k=1}^n -\log \widehat{P}_m^{\widehat{M}(\mathcal{G};\boldsymbol{\theta})}(\mathbf{v}_k). \quad (4)\end{aligned}$$

Loss 2:

To simultaneously maximize  $P^{\widehat{M}}(\mathbf{y} \mid do(\mathbf{x}))$ , we subtract a weighted second term  $\log \widehat{P}_m^{\widehat{M}}(\mathbf{y} \mid do(\mathbf{x}))$ , resulting in the objective  $\mathcal{L}(\{\mathbf{v}_k\}_{k=1}^n)$  equal to

$$\frac{1}{n} \sum_{k=1}^n -\log \widehat{P}_m^{\widehat{M}}(\mathbf{v}_k) - \lambda \log \widehat{P}_m^{\widehat{M}}(\mathbf{y} \mid do(\mathbf{x})), \quad (5)$$

# NCM: Architecture & loss/objective

- Two solutions (let SCM1 and SCM2) are found from the independent runs and both have matched the training distribution.
- What about their causal effects?
  - SCM 1 maximizes the effect.
  - SCM 2 minimizes the effect.

## Algorithm Sketch: NeuralID

- **Input:** Query  $Q = P(y_* \mid \mathbf{x}_*)$ , dataset  $\mathcal{Z}(\mathcal{M}^*)$ , causal graph  $\mathcal{G}$
- **Train NCM:**  $\widehat{M} \leftarrow \text{NCM}(\mathcal{V}, \mathcal{G})$
- **Search:**
  - $\theta_{\min}^* \leftarrow \arg \min_{\theta} P^{\widehat{M}(\theta)}(y_* \mid \mathbf{x}_*)$
  - $\theta_{\max}^* \leftarrow \arg \max_{\theta} P^{\widehat{M}(\theta)}(y_* \mid \mathbf{x}_*)$
  - subject to  $\mathcal{Z}(\widehat{M}(\theta)) = \mathcal{Z}(\mathcal{M}^*)$
- **Decision:**
  - If  $P^{\widehat{M}(\theta_{\min}^*)}(y_* \mid \mathbf{x}_*) \neq P^{\widehat{M}(\theta_{\max}^*)}(y_* \mid \mathbf{x}_*)$ , then FAIL
  - Else, return  $P^{\widehat{M}(\theta^*)}(y_* \mid \mathbf{x}_*)$  (choose min or max arbitrarily)
- **Output:** Estimated  $P(y_* \mid \mathbf{x}_*)$  if identifiable; otherwise FAIL

# NCM: Architecture & loss/objective

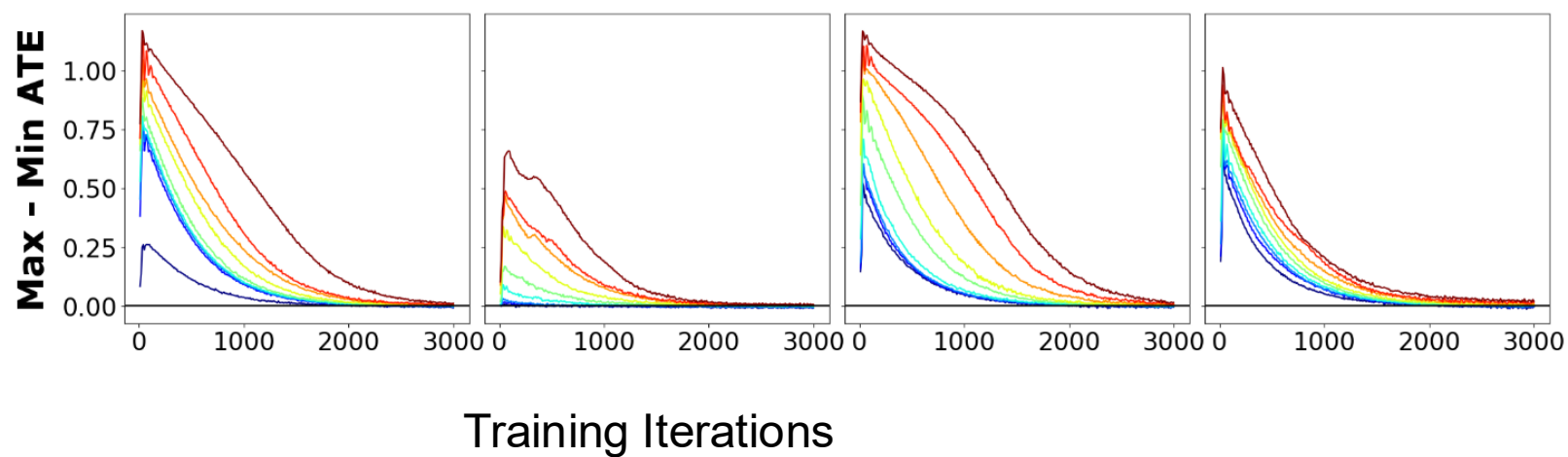
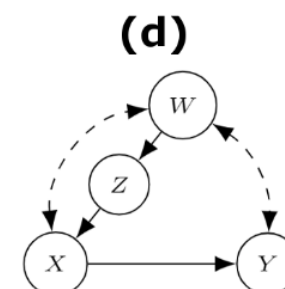
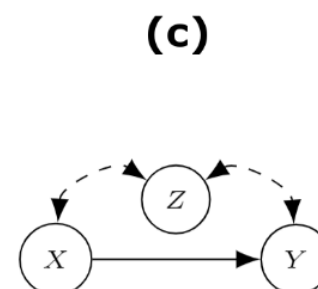
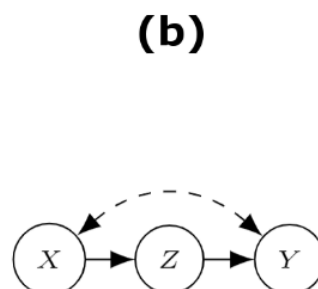
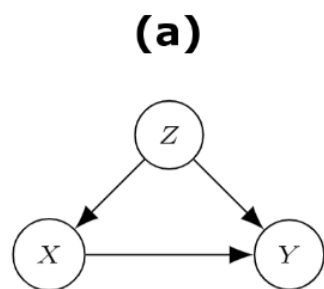
- What about their causal effects?
  - SCM 1 maximizes the effect.
  - SCM 2 minimizes the effect.
- If there exists a gap between the maximized causal effect and the minimized causal effect,
- The effect is non-identifiable.
- Otherwise report the estimate.

## Algorithm Sketch: NeuralID

- **Input:** Query  $Q = P(y_* \mid \mathbf{x}_*)$ , dataset  $\mathcal{Z}(\mathcal{M}^*)$ , causal graph  $\mathcal{G}$
- **Train NCM:**  $\widehat{M} \leftarrow \text{NCM}(\mathcal{V}, \mathcal{G})$
- **Search:**
  - $\theta_{\min}^* \leftarrow \arg \min_{\theta} P^{\widehat{M}(\theta)}(y_* \mid \mathbf{x}_*)$
  - $\theta_{\max}^* \leftarrow \arg \max_{\theta} P^{\widehat{M}(\theta)}(y_* \mid \mathbf{x}_*)$
  - subject to  $\mathcal{Z}(\widehat{M}(\theta)) = \mathcal{Z}(\mathcal{M}^*)$
- **Decision:**
  - If  $P^{\widehat{M}(\theta_{\min}^*)}(y_* \mid \mathbf{x}_*) \neq P^{\widehat{M}(\theta_{\max}^*)}(y_* \mid \mathbf{x}_*)$ , then FAIL
  - Else, return  $P^{\widehat{M}(\theta^*)}(y_* \mid \mathbf{x}_*)$  (choose min or max arbitrarily)
- **Output:** Estimated  $P(y_* \mid \mathbf{x}_*)$  if identifiable; otherwise FAIL

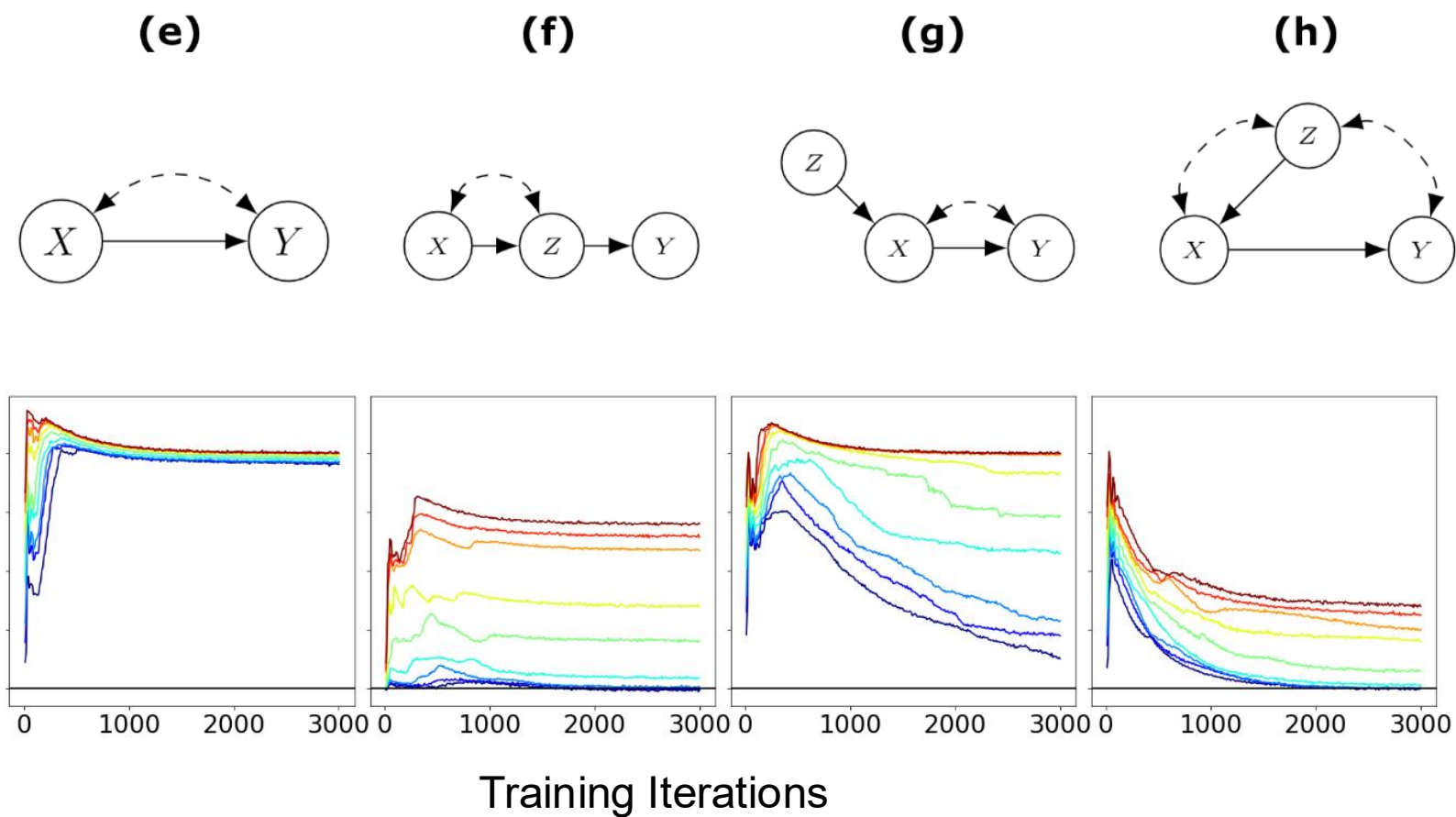
# NCM: Results

## Identifiable



# NCM: Results

## Non-Identifiable





# Limitations

- Presence of unobserved confounders
- Finite sample error.
- High-dimensional variables.
- Sampling variation.

# What happens if the causal effect is non identifiable?

- Even though we know the true graph and can match the observational training data distribution.

# Partial identification of treatment effects with implicit generative models

Balazadeh Meresht, V., Syrgkanis, V., & Krishnan, R. G. (2022).

- Addresses this problem by performing partial identification of average treatment effects for both discrete and continuous variables.
- Continuous treatments

# Partial identification of treatment effects

- Due to the continuous treatment, we need a new definition for ATE.

**Definition** (Average Treatment Derivative). *For the treatment regime  $f_T$  in SCM  $\mathcal{M}$ , we define the average treatment derivative (ATD) as*

$$\text{ATD}_{\mathcal{M}} = \mathbb{E}_{\mathbf{u} \sim P_U} \left[ \left. \frac{\partial Y_{\mathcal{M}(T=t)}(\mathbf{u})}{\partial t} \right|_{t=T(\mathbf{u})} \right],$$

**Definition 4** (Partial Identification of ATD). *Partial identification of ATD is the solution to the following optimization problem:*

$$\left( \min_{\mathcal{M}' \in \mathfrak{M}} \text{ATD}_{\mathcal{M}'}, \quad \max_{\mathcal{M}' \in \mathfrak{M}} \text{ATD}_{\mathcal{M}'} \right) \quad \text{s.t.} \quad P_{\mathcal{M}'} = P \quad \& \quad \mathcal{G}_{\mathcal{M}'} = \mathcal{G}$$

where  $\mathfrak{M}$  is the set of all SCMs on random variables  $\mathbf{V}$ .

Finite sampling error:

$$\left( \min_{\theta} \text{ATD}_{\mathcal{M}_{\mathcal{G}}^{\theta}}, \quad \max_{\theta} \text{ATD}_{\mathcal{M}_{\mathcal{G}}^{\theta}} \right) \quad \text{s.t.} \quad W_1 \left( P_{\mathcal{M}_{\mathcal{G}}^{\theta}}, P^n \right) \leq \alpha_n$$

where  $\alpha_n$  is a hyper-parameter that specifies the level of tightness of the bounds.

**Sinkhorn divergence**, a differentiable approximation to the Wasserstein distance, as the measure of distance between distributions and solve the following:

$$\min_{\theta} \max_{\lambda \geq 0} \text{ATD}_{\mathcal{M}_{\mathcal{G}}^{\theta}} + \lambda \left( S_{\epsilon} \left( P_{\mathcal{M}_{\mathcal{G}}^{\theta}}, P^n \right) - \alpha_n \right)$$

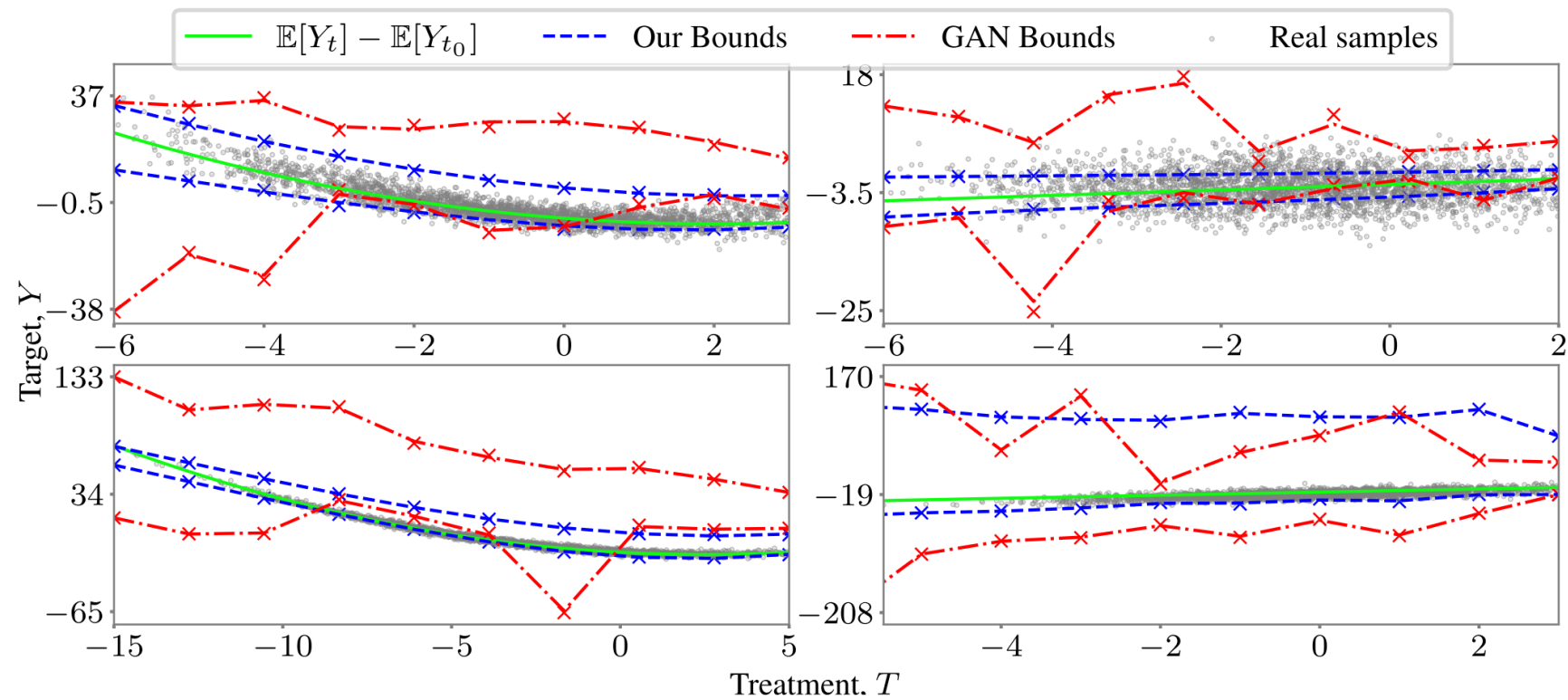
# Partial identification of treatment effects

The value of  $Y_{\mathcal{M}_\theta}(T = t)(\mathbf{u})$  can be calculated by hard intervention  $T = t$ , i.e., fixing the output of function  $f_T^\theta$  as  $t$  and computing  $Y$  through a topological order of calculations. Then,  $\text{ATD}_{\mathcal{M}_\theta}$  is estimated as follows:

$$\text{ATD}_{\mathcal{M}_\theta} \approx \frac{1}{n} \sum_{i=1}^n \frac{1}{\epsilon} \left[ Y_{\mathcal{M}_\theta}(T = t^{(i)} + \epsilon)(\mathbf{u}^{(i)}) - Y_{\mathcal{M}_\theta}(T = t^{(i)})(\mathbf{u}^{(i)}) \right]$$

where  $\{t^{(i)}\}_{i=1}^n$  are samples from the treatment variable, and  $\{\mathbf{u}^{(i)}\}_{i=1}^n$  are the latent variables generated from a *uniform distribution*.

# Results:



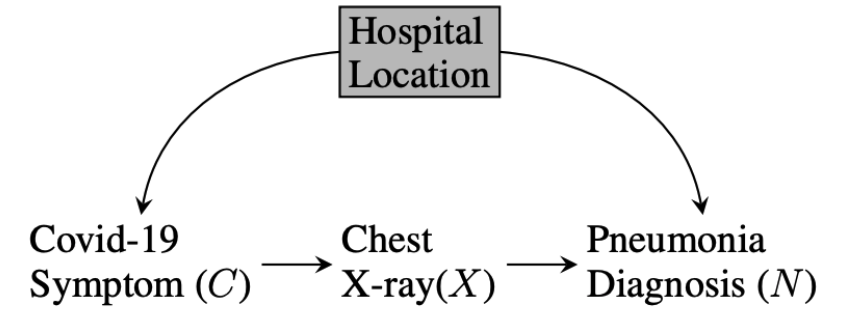
# Causal Inference with Images

## *Interventional sampling*



# Problem Definition

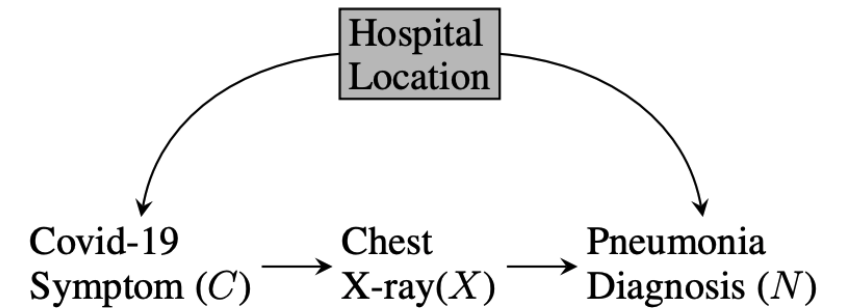
- 30,000 patients' data
  - Covid-19 Symptom
  - Chest x-ray images
  - Pneumonia Diagnosis
  - From 51 countries (but unobserved for each patient)



- Goal: how likely an average person is to be diagnosed with pneumonia if they have Covid symptoms. Predict Covid symptoms  $\rightarrow$  Pneumonia diagnosis
- Possible Solutions:
  - Covid  $\rightarrow$  Pneumonia diagnosis :  $P(N|C)$

# Problem Definition

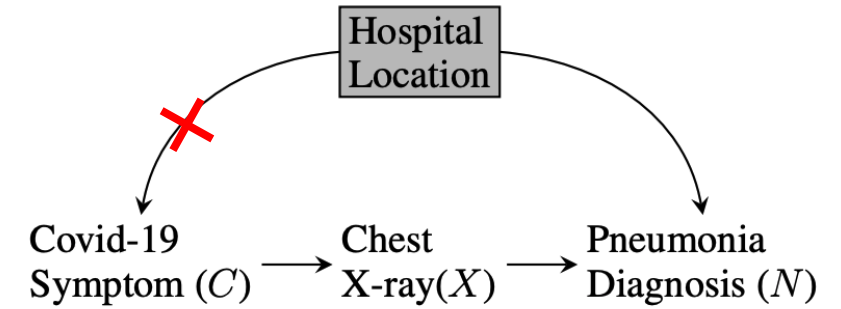
- 30,000 patients' data



- From 51 countries (but unobserved for each patient)
  - Socio-economic and health conditions in a specific location might affect both the likelihood of getting Covid and quality of local health care for Pneumonia diagnosis.
  - Shift in the location causes shift in associated mechanisms while keeping other mechanisms invariant
- Possible Solutions: (Biased prediction? Can we trust the model?)
  - Covid -> Pneumonia diagnosis :  $P(N|C)$

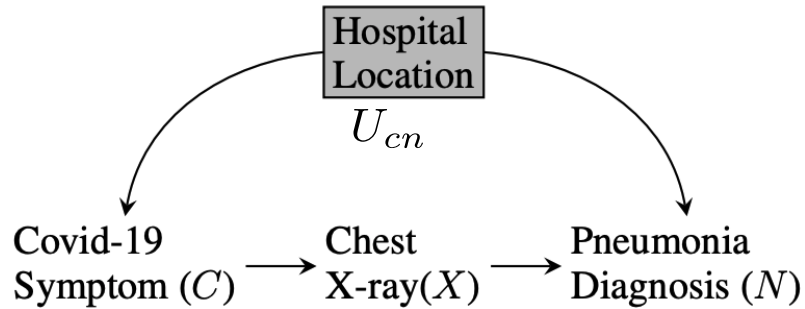
# Problem Definition

- 30,000 patients' data
  - Covid-19 Symptom
  - Chest x-ray images
  - Pneumonia Diagnosis
  - From 51 countries (but unobserved for each patient)
- Assumption: No edge to hospital location to Chest Xray, otherwise effect non-identifiable.



- Estimate the causal effect:  $P(N|do(C))$  instead of  $P(N|C)$

# Deep Causal Generative Models



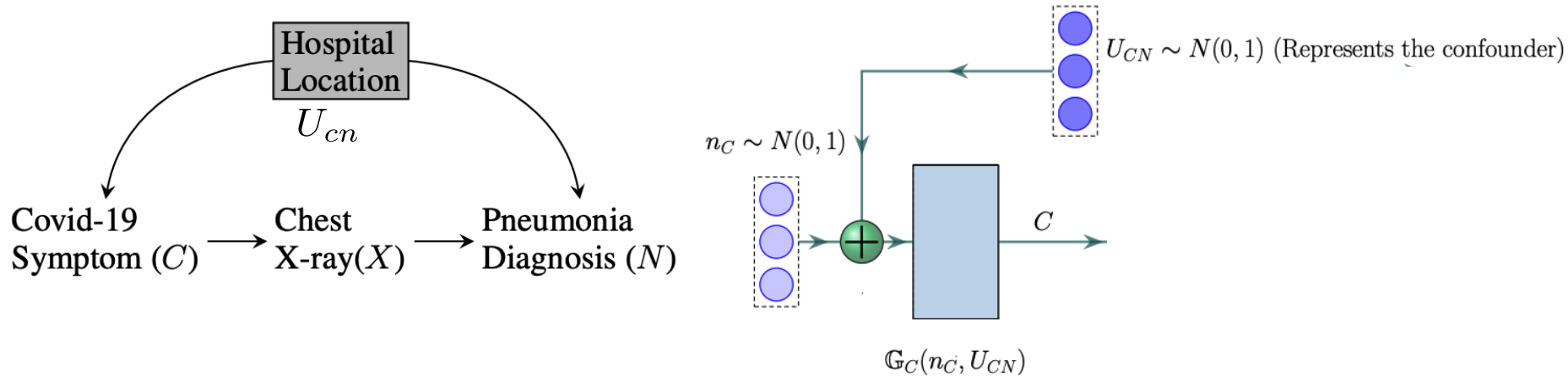
$$C = f_C(n_C, U_{cn})$$

$$X = f_X(C, n_X)$$

$$N = f_N(X, n_N, U_{cn})$$

How can we learn the unknown structural functions for high-dimensional variables?

# Deep Causal Generative Models (Similar to CausalGAN)

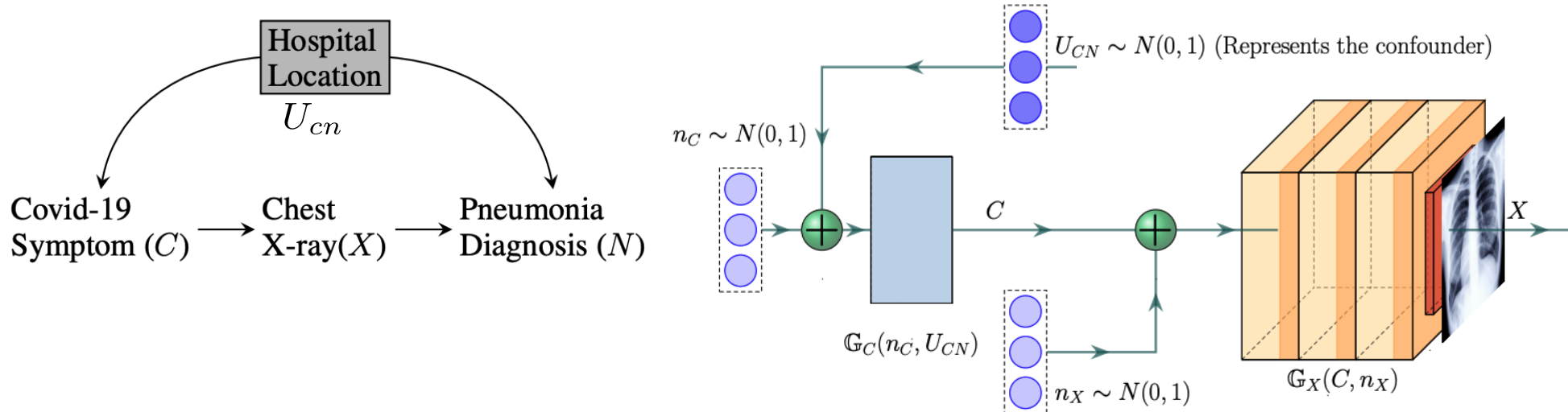


$$C = f_C(n_C, U_{cn})$$

$$X = f_X(C, n_X)$$

$$N = f_N(X, n_N, U_{cn})$$

# Deep Causal Generative Models (Similar to CausalGAN)

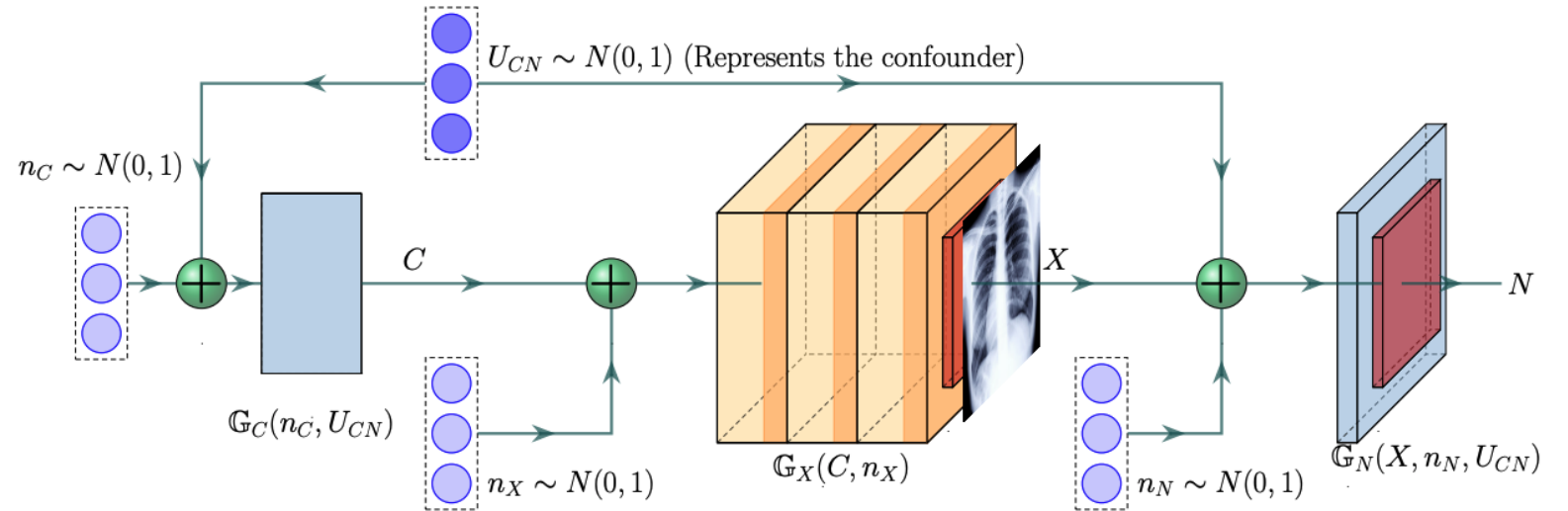
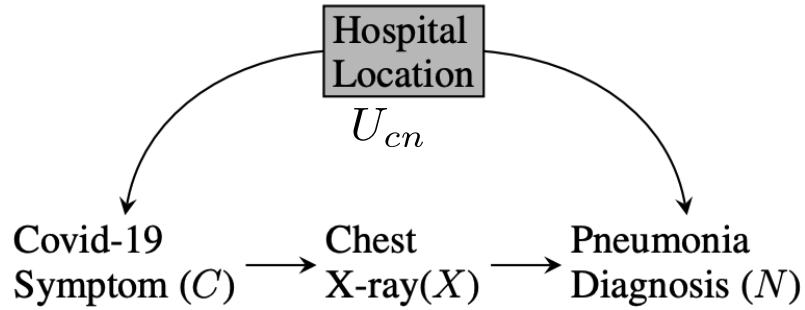


$$C = f_C(n_C, U_{cn})$$

$$X = f_X(C, n_X)$$

$$N = f_N(X, n_N, U_{cn})$$

# Deep Causal Generative Models (Similar to CausalGAN)

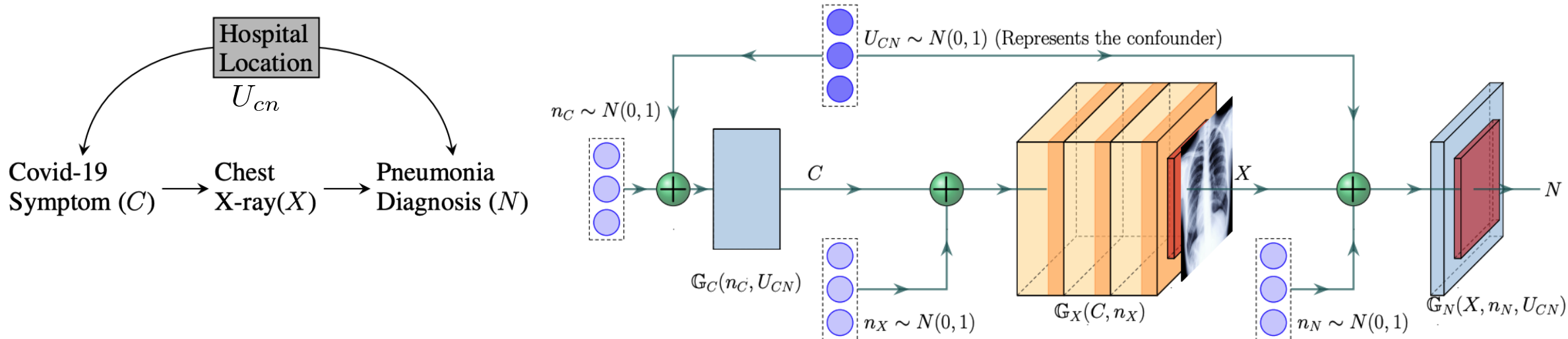


$$C = f_C(n_C, U_{cn})$$

$$X = f_X(C, n_X)$$

$$N = f_N(X, n_N, U_{cn})$$

# Deep Causal Generative Models (Similar to CausalGAN)



$$C = f_C(n_C, U_{cn})$$

$$X = f_X(C, n_X)$$

$$N = f_N(X, n_N, U_{cn})$$



Real Data



Fake Data

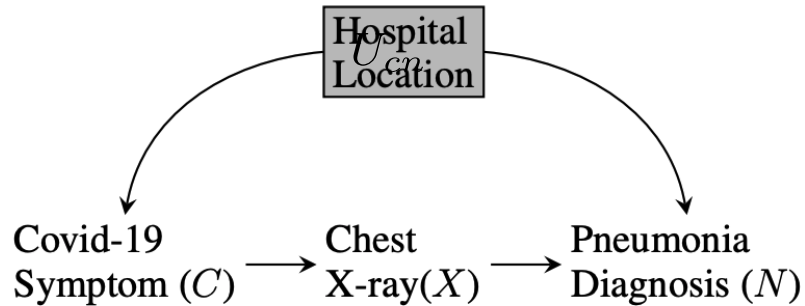
$p(C, Xray, N) \sim q(C, Xray, N)$  Need this but.

$p(Xray) \sim q(Xray)$  **Discriminator might do this!**



# Modular learning of deep causal generative models for high-dimensional causal inference.

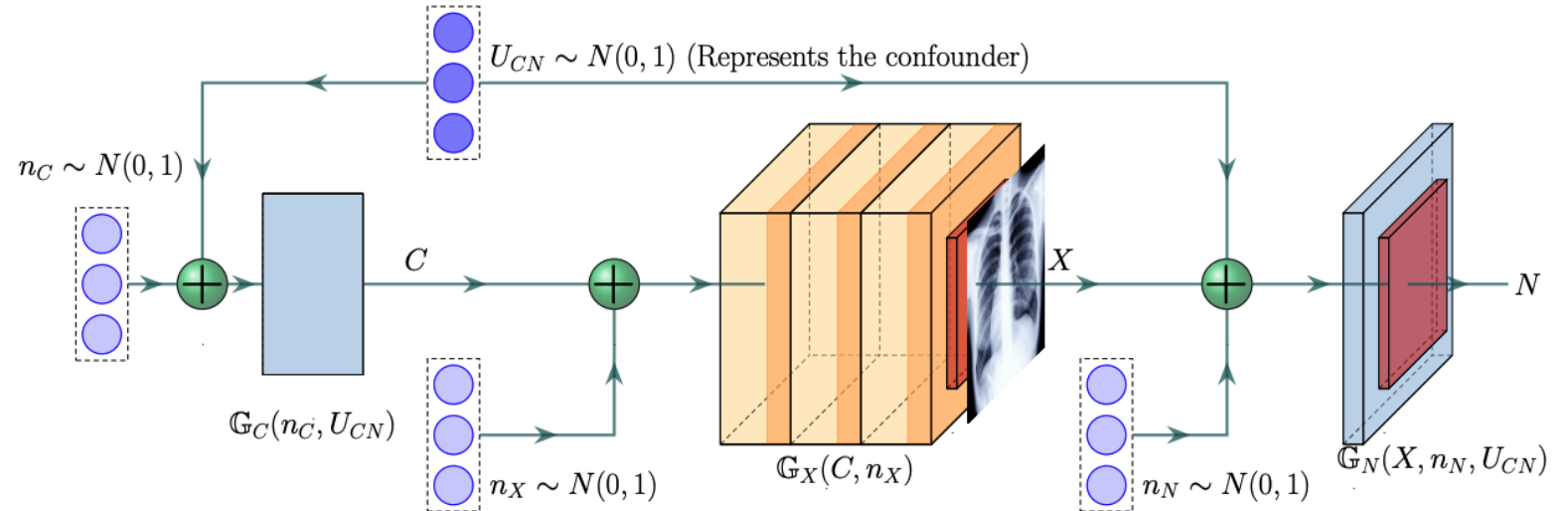
Rahman, M. M., & Kocaoglu, M. (2024).



$$C = f_C(n_C, U_{cn})$$

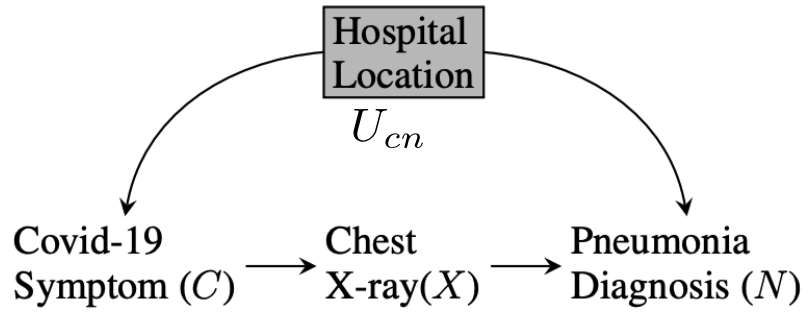
$$X = f_X(C, n_X)$$

$$N = f_N(X, n_N, U_{cn})$$

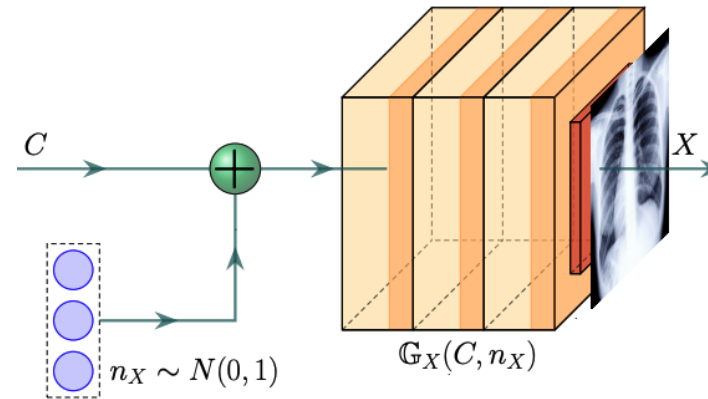


- Modular training based on c-component factorization is proposed. (confounded :  $\{C, N\}$ )
- Train models i)  $[G_X]$  and then ii)  $[G_C, G_N]$ .

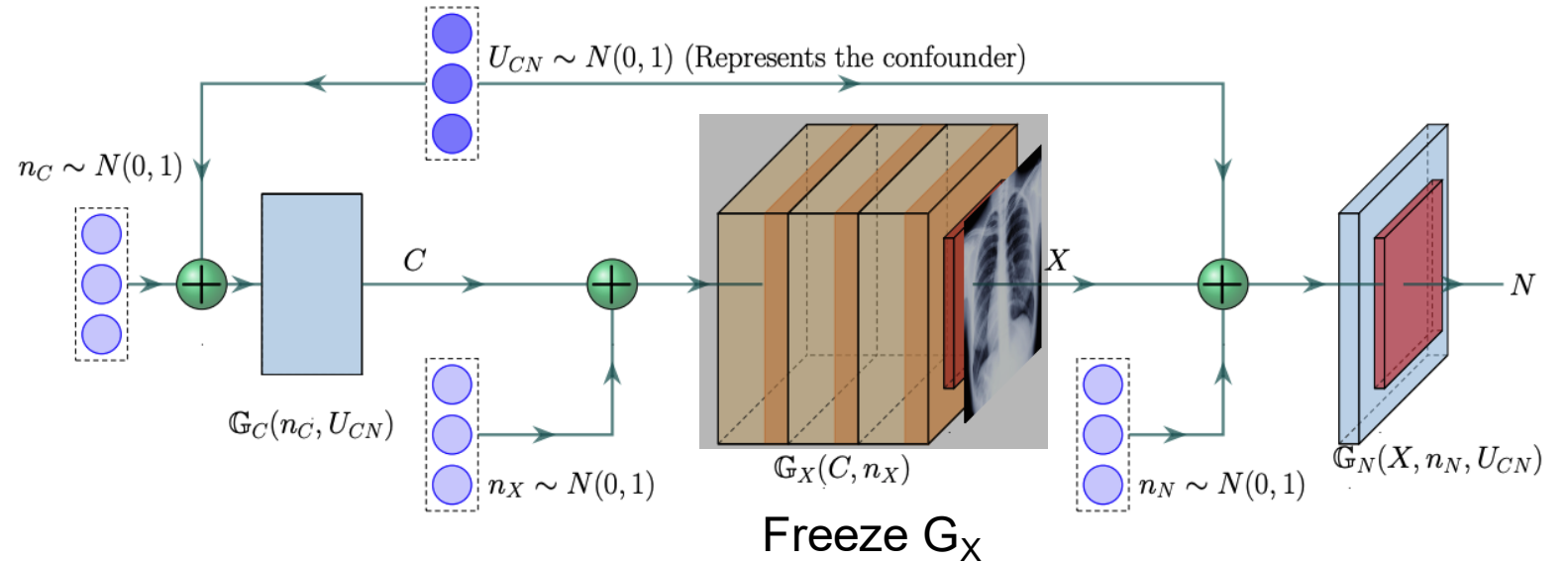
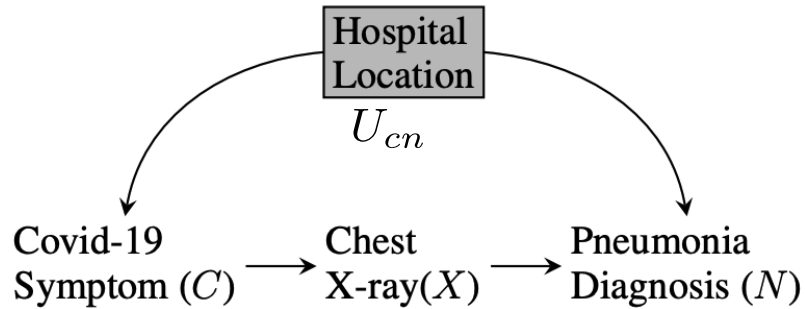
# Modular-DCM



$$X = f_X(C, n_X)$$



# Modular-DCM



$$C = f_C(n_C, U_{cn})$$

$$X = f_X(C, n_X)$$

$$N = f_N(X, n_N, U_{cn})$$

It is shown that this modularization provides

- Better convergence
- Theoretical guarantee of plugging-in pre-trained models.

# Open Problems

- Positivity violations
- Acyclicity violations / Feedback loops.
- Partial/No Causal graphs.

# Causal Inference with Tabular Data

*Counterfactuals*

# Three Steps of Counterfactual

**Theorem** Given a model  $\langle M, P(u) \rangle$ , the conditional probability  $P(B_X | e)$  of a counterfactual sentence “If it were  $X$  then  $B$ ,” given evidence  $e$ , can be evaluated using the following three steps:

1. **Abduction:** Update the distribution  $P(u)$  by the evidence  $e$  to obtain the posterior  $P(u | e)$ .
2. **Action:** Modify the model  $M$  by performing the intervention  $\text{do}(X)$ , where  $X$  is the antecedent of the counterfactual, resulting in the submodel  $M_X$ .
3. **Prediction:** Use the modified model  $\langle M_X, P(u | e) \rangle$  to compute the probability of  $B$ , the consequence of the counterfactual.

# Neural Causal Models for Counterfactual Identification and Estimation (Similar to NCM)

Xia, K., Pan, Y., & Bareinboim, E. (2022).

- Checks identifiability of counterfactual queries using learned NCMs.
- Trains two models with same observational fit and compares their counterfactual outputs after maximizing and minimizing.
- Returns prediction only if all consistent models agree.

## Algorithm Sketch: NeuralID

- **Input:** Query  $Q = P(\mathbf{y}_* \mid \mathbf{x}_*)$ , dataset  $\mathcal{Z}(\mathcal{M}^*)$ , causal graph  $\mathcal{G}$
- **Train NCM:**  $\widehat{M} \leftarrow \text{NCM}(\mathcal{V}, \mathcal{G})$
- **Search:**
  - $\theta_{\min}^* \leftarrow \arg \min_{\theta} P^{\widehat{M}(\theta)}(\mathbf{y}_* \mid \mathbf{x}_*)$
  - $\theta_{\max}^* \leftarrow \arg \max_{\theta} P^{\widehat{M}(\theta)}(\mathbf{y}_* \mid \mathbf{x}_*)$
  - *subject to*  $\mathcal{Z}(\widehat{M}(\theta)) = \mathcal{Z}(\mathcal{M}^*)$
- **Decision:**
  - If  $P^{\widehat{M}(\theta_{\min}^*)}(\mathbf{y}_* \mid \mathbf{x}_*) \neq P^{\widehat{M}(\theta_{\max}^*)}(\mathbf{y}_* \mid \mathbf{x}_*)$ , then FAIL
  - Else, return  $P^{\widehat{M}(\theta^*)}(\mathbf{y}_* \mid \mathbf{x}_*)$  (choose min or max arbitrarily)
- **Output:** Estimated  $P(\mathbf{y}_* \mid \mathbf{x}_*)$  if identifiable; otherwise FAIL

# GAN-NCMs

- Use rejection sampling:
  - Sample random noise from the model's input distribution.
  - Simulate the model; keep samples that match the desired condition.
  - Collect the corresponding output as a counterfactual sample.
- **Repeat** until we have enough valid samples.

## Algorithm Sketch: NCM Counterfactual Sampling

- **Goal:** Sample counterfactual outcomes  $\mathbf{Y}_*$  conditioned on  $\mathbf{X}_* = \mathbf{x}_*$  using a trained NCM  $\widehat{M}(\theta)$ .
- **Steps:**
  1. Initialize an empty set  $S \leftarrow \emptyset$ .
  2. **Repeat until**  $|S| = m$ :
    - Sample exogenous noise:  $\hat{\mathbf{u}} \sim P(\hat{\mathbf{U}})$ .
    - Simulate  $\hat{\mathbf{X}}_* = \mathbf{X}_*^{\widehat{M}(\theta)}(\hat{\mathbf{u}})$ .
    - **If**  $\hat{\mathbf{X}}_* = \mathbf{x}_*$ , then:
      - \* Compute counterfactual  $\hat{\mathbf{Y}}_* = \mathbf{Y}_*^{\widehat{M}(\theta)}(\hat{\mathbf{u}})$ .
      - \* Add to sample set:  $S \leftarrow S \cup \{\hat{\mathbf{Y}}_*\}$ .
  3. **Return**  $S$  as the final set of  $m$  counterfactual samples.



# GAN-NCMs: Limitations

- Although okay for binary, rejection sampling to collect exogenous noise might be infeasible for high dimensional variables
- We might have to wait infinite time to obtain the expected number of samples

# Counterfactual Inference with Empirical Success

*Relaxing Theoretical Guarantee*

# Abduction

- Diffusion models ?
- Variational auto encoders
- Normalizing flow.

# Basics of Diffusion Models

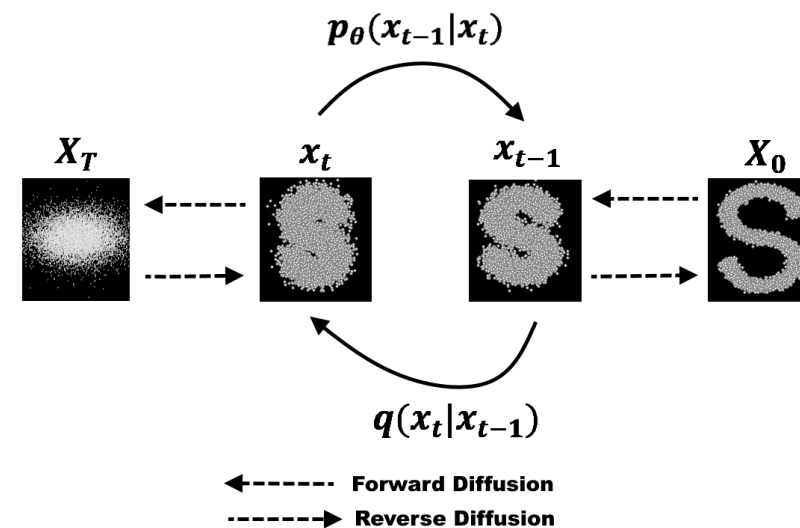
<https://medium.com/data-science/diffusion-models-made-easy-8414298ce4da>

We can represent diffusion models as a fixed Markov chain that adds Gaussian noise with variances  $\beta_1, \dots, \beta_T \in (0, 1)$ , generating latent variables  $X^1, \dots, X^T$ ,

$$q(X^t | x^{t-1}) = \mathcal{N}(X^t; \sqrt{1 - \beta_t} x^{t-1}, \beta_t I)$$

$$q(X^t | x^0) = \mathcal{N}(X^t; \sqrt{\alpha_t} x^0, (1 - \alpha_t)I)$$

where  $\alpha_t = \prod_{i=1}^t (1 - \beta_i)$ . For large  $T$  and  $\alpha_t \rightarrow 0$  we have  $X^T$  distributed as an isotropic Gaussian.



# Basics of Diffusion Models

<https://medium.com/data-science/diffusion-models-made-easy-8414298ce4da>

We can represent diffusion models as a fixed Markov chain that adds Gaussian noise with variances  $\beta_1, \dots, \beta_T \in (0, 1)$ , generating latent variables  $X^1, \dots, X^T$ ,

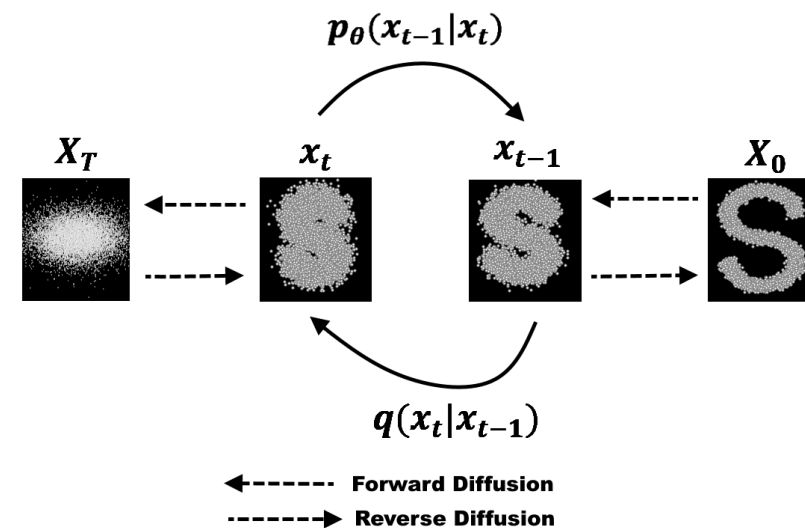
$$q(X^t | x^{t-1}) = \mathcal{N}(X^t; \sqrt{1 - \beta_t} x^{t-1}, \beta_t I)$$

$$q(X^t | x^0) = \mathcal{N}(X^t; \sqrt{\alpha_t} x^0, (1 - \alpha_t)I)$$

where  $\alpha_t = \prod_{i=1}^t (1 - \beta_i)$ . For large  $T$  and  $\alpha_t \rightarrow 0$  we have  $X^T$  distributed as an isotropic Gaussian.

- Reverse diffusion process:

$$p_\theta(X^{t-1} | x^t) = \mathcal{N}(X^{t-1}; \mu_\theta(x^t, t), \Sigma_\theta(x^t, t)).$$



# Basics of Diffusion Models

<https://medium.com/data-science/diffusion-models-made-easy-8414298ce4da>

We can represent diffusion models as a fixed Markov chain that adds Gaussian noise with variances  $\beta_1, \dots, \beta_T \in (0, 1)$ , generating latent variables  $X^1, \dots, X^T$ ,

$$q(X^t | x^{t-1}) = \mathcal{N}(X^t; \sqrt{1 - \beta_t} x^{t-1}, \beta_t I)$$

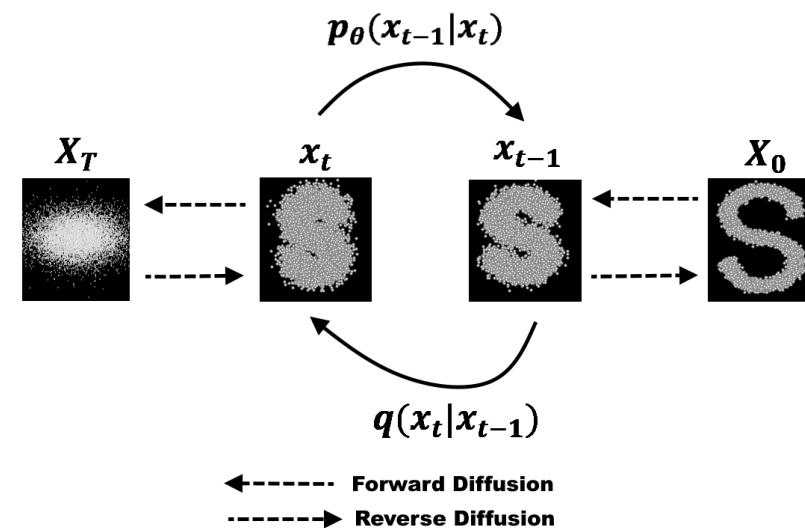
$$q(X^t | x^0) = \mathcal{N}(X^t; \sqrt{\alpha_t} x^0, (1 - \alpha_t)I)$$

where  $\alpha_t = \prod_{i=1}^t (1 - \beta_i)$ . For large  $T$  and  $\alpha_t \rightarrow 0$  we have  $X^T$  distributed as an isotropic Gaussian.

- Reverse diffusion process:

$$p_\theta(X^{t-1} | x^t) = \mathcal{N}(X^{t-1}; \mu_\theta(x^t, t), \Sigma_\theta(x^t, t)).$$

$$\mathbb{E}_{\substack{t \sim \text{Unif}\{[T]\} \\ X^0 \sim Q \\ \varepsilon \sim \mathcal{N}(0, I)}} [\|\varepsilon - \varepsilon_\theta(\sqrt{\alpha_t} X^0 + \sqrt{1 - \alpha_t} \varepsilon, t)\|^2],$$



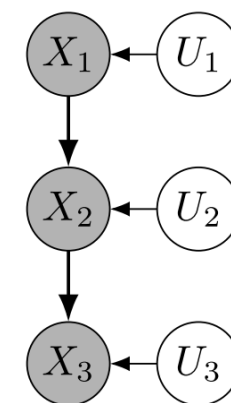
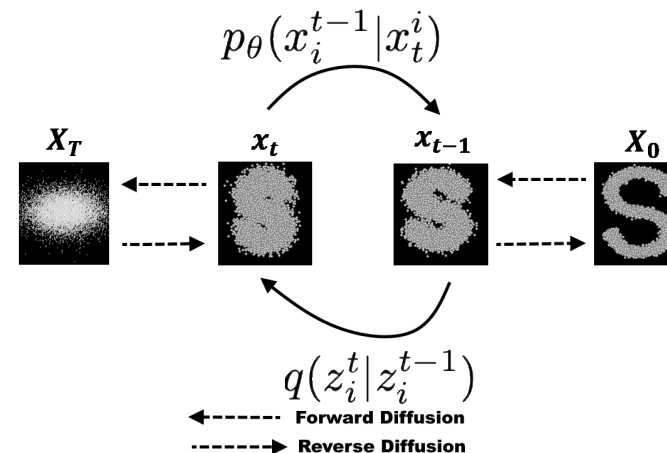
# Denoising Diffusion Implicit Models (DDIM)

- Use a pretrained DDPM model to obtain a deterministic sample given noise.

# Interventional and Counterfactual Inference with Diffusion Models

Chao, P., Blöbaum, P., & Kasiviswanathan, S. P. (2023)

- Given a causal graph over endogenous variables.  $(X_1, \dots, X_K)$
- $Z_i^t$  Each variable at diffusion step  $t$  of the forward implicit diffusion process.
- $\hat{X}_i^t$  Each variable at diffusion step  $t$  of the reverse implicit diffusion process



(a) Chain graph.



# DCM: Training for Each Node

- A diffusion model is trained for each node, with denoised parent values as input using classifier free guidance.

---

**Algorithm 1** DCM Training

---

**Input:** Distribution  $Q$ , scale factors  $\{\alpha_t\}_{t=1}^T$ , causal DAG  $\mathcal{G}$  with node  $i$  represented by  $X_i$

1: **while** not converged **do**

2:   Sample  $X^0 \sim Q$

3:   **for**  $i = 1, \dots, K$  **do**

4:      $t \sim \text{Unif}[\{1, \dots, T\}]$

5:      $\varepsilon \sim \mathcal{N}(0, I_{d_i})$

6:     Update parameters of node  $i$ 's diffusion model  $\varepsilon_\theta^i$ , by minimizing the following loss:

$$\|\varepsilon - \varepsilon_\theta^i(\sqrt{\alpha_t}X_i^0 + \sqrt{1 - \alpha_t}\varepsilon, X_{\text{pa}_i}^0, t)\|_2^2$$

7:   **end for**

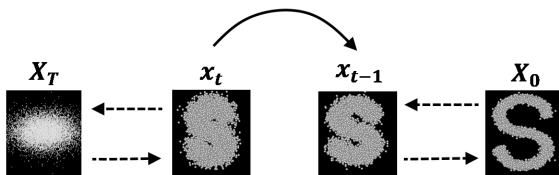
8: **end while**

---

# DCM: Observational Sampling/ Interventional Sampling

- Use a pretrained DDPM model to obtain a deterministic sample given noise.
- Reverse implicit diffusion process (Decoding)

$$\hat{X}_i^{t-1} \leftarrow \sqrt{\frac{\alpha_{t-1}}{\alpha_t}} \hat{X}_i^t - \varepsilon_{\theta}^i(\hat{X}_i^t, X_{\text{pa}_i}, t) \left( \sqrt{\frac{\alpha_{t-1}(1-\alpha_t)}{\alpha_t}} - \sqrt{1-\alpha_{t-1}} \right)$$




---

## Algorithm 2 Observational/Interventional Sampling

---

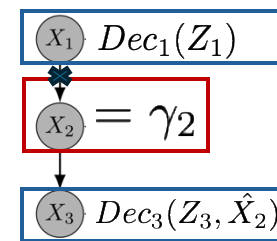
**Input:** Intervention set  $\mathcal{I}$  with values  $\gamma$  ( $I = \emptyset$  for observational sampling)

```

1: for  $i = 1, \dots, K$  do
2:    $Z_i \sim \mathcal{N}(0, I_{d_i})$ 
3:   if  $i \in \mathcal{I}$  then
4:      $\hat{X}_i \leftarrow \gamma_i$ 
5:   else
6:      $\hat{X}_i \leftarrow \text{Dec}_i(Z_i, \hat{X}_{\text{pa}_i})$ 
7:   end if
8: end for
9: Return  $\hat{X} := (\hat{X}_1, \dots, \hat{X}_K)$ 

```

---



# DDIM for Counterfactual Sampling

- Forward implicit diffusion process (Encoding)

$$\hat{X}_i^{t-1} \leftarrow \sqrt{\frac{\alpha_{t-1}}{\alpha_t}} \hat{X}_i^t - \varepsilon_{\theta}^i(\hat{X}_i^t, X_{\text{pa}_i}, t) \left( \sqrt{\frac{\alpha_{t-1}(1-\alpha_t)}{\alpha_t}} - \sqrt{1-\alpha_{t-1}} \right)$$

---

**Algorithm 3** Counterfactual Inference

---

**Input:** Intervention set  $\mathcal{I}$  with values  $\gamma$ , factual sample  $x^{\text{F}} := (x_1^{\text{F}}, \dots, x_K^{\text{F}})$

```
1: for  $i = 1, \dots, K$  do {in topological order}
2:   if  $i \in \mathcal{I}$  then Action
3:      $\hat{x}_i^{\text{CF}} \leftarrow \gamma_i$ 
4:   else if  $i$  is not a descendant of any intervened node in  $\mathcal{I}$  then
5:      $\hat{x}_i^{\text{CF}} \leftarrow x_i^{\text{F}}$ 
6:   else Abduction
7:      $z_i^{\text{F}} \leftarrow \text{Enc}_i(x_i^{\text{F}}, x_{\text{pa}_i}^{\text{F}})$  {abduction step}
8:      $\hat{x}_i^{\text{CF}} \leftarrow \text{Dec}_i(z_i^{\text{F}}, \hat{x}_{\text{pa}_i})$  Prediction {action and prediction steps}
9:   end if
10: end for
11: Return  $\hat{x}^{\text{CF}} := (\hat{x}_1^{\text{CF}}, \dots, \hat{x}_K^{\text{CF}})$ 
```

---

# Abduction

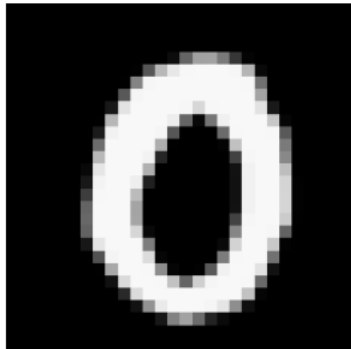
- Diffusion models 

# Causal Inference with Images

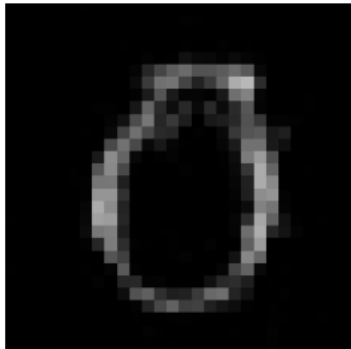
*Counterfactuals*

# What is a counterfactual question for images?

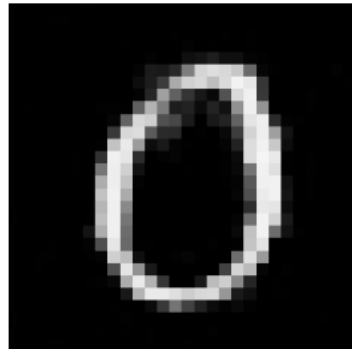
- How would a given image look like if we changed a specific feature.



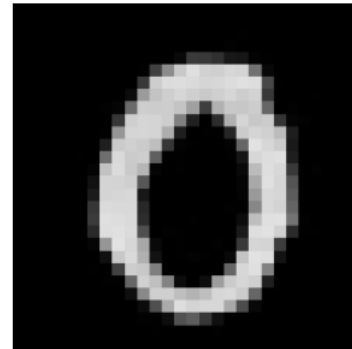
Original



$do(t = 1.5)$



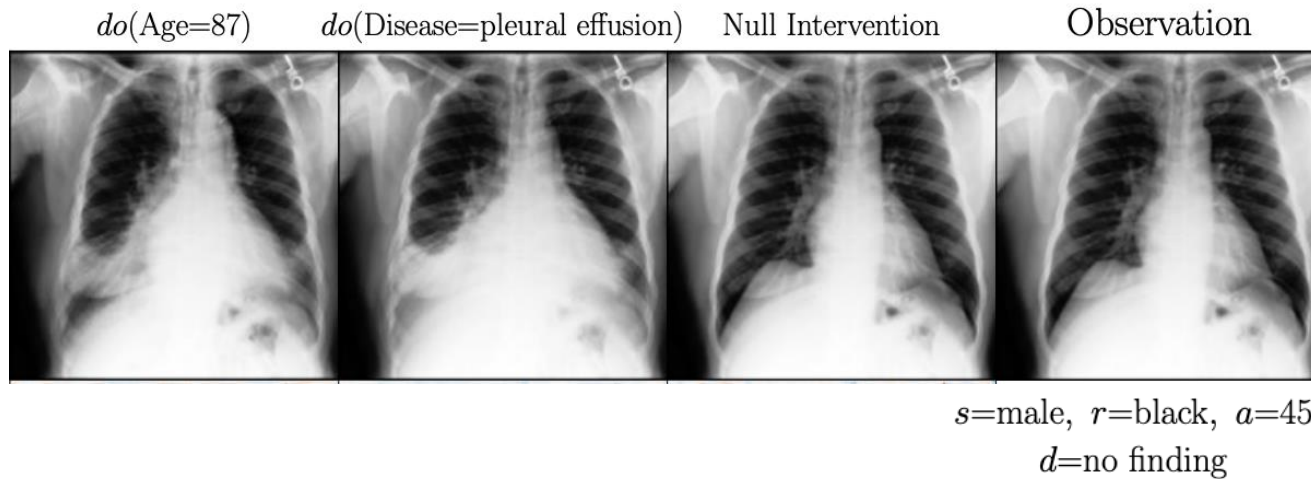
$do(t = 1.5, i = 224)$



$do(t = 3, i = 180)$

# What is a counterfactual question for images?

- How would a given image look like if we changed a specific feature.



High Fidelity Image Counterfactuals with Probabilistic Causal Models

# What is a counterfactual question for images?

- By training on a given dataset, possibly containing spurious correlations.



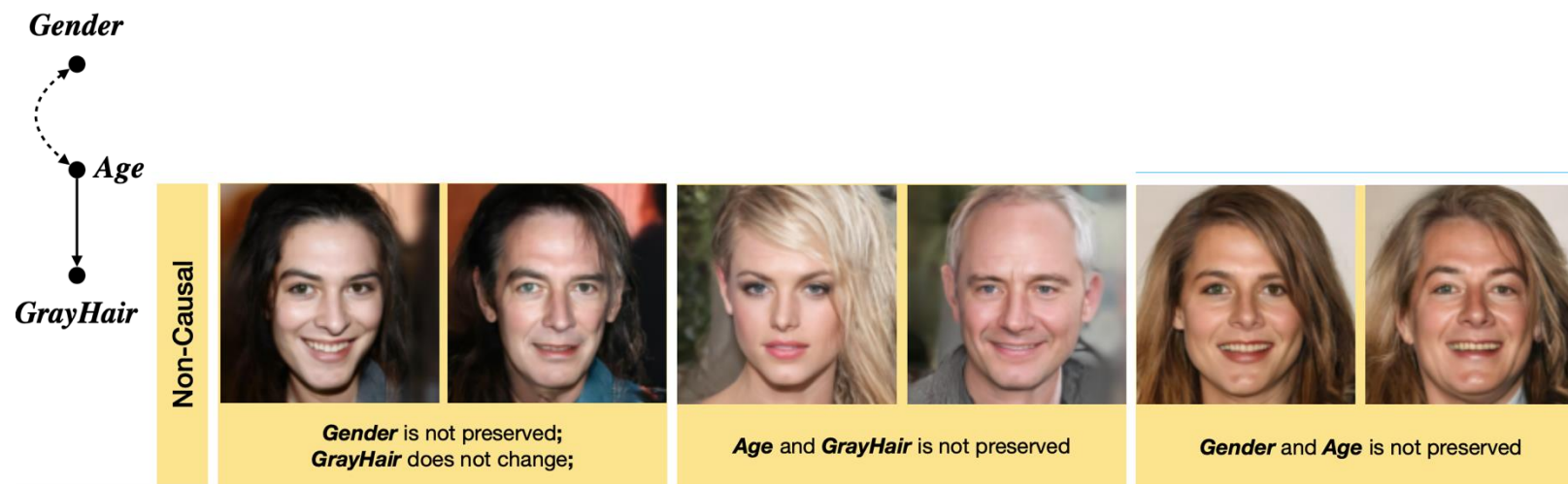
# What is a counterfactual question for images?

- By training on a given dataset, possibly containing spurious correlations.



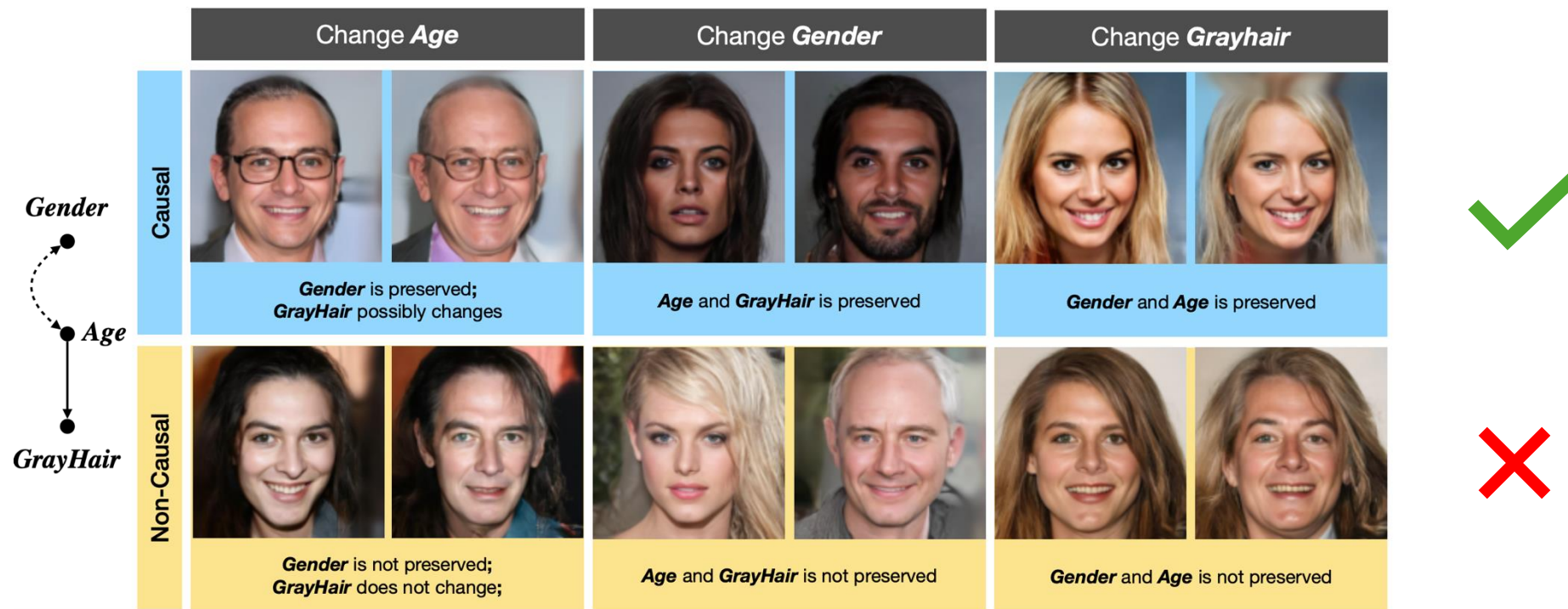
# What is a counterfactual question for images?

- By training on a given dataset, possibly containing spurious correlations.



# What is a counterfactual question for images?

- By training on a given dataset, possibly containing spurious correlations.



# Three Steps of Counterfactual with Images

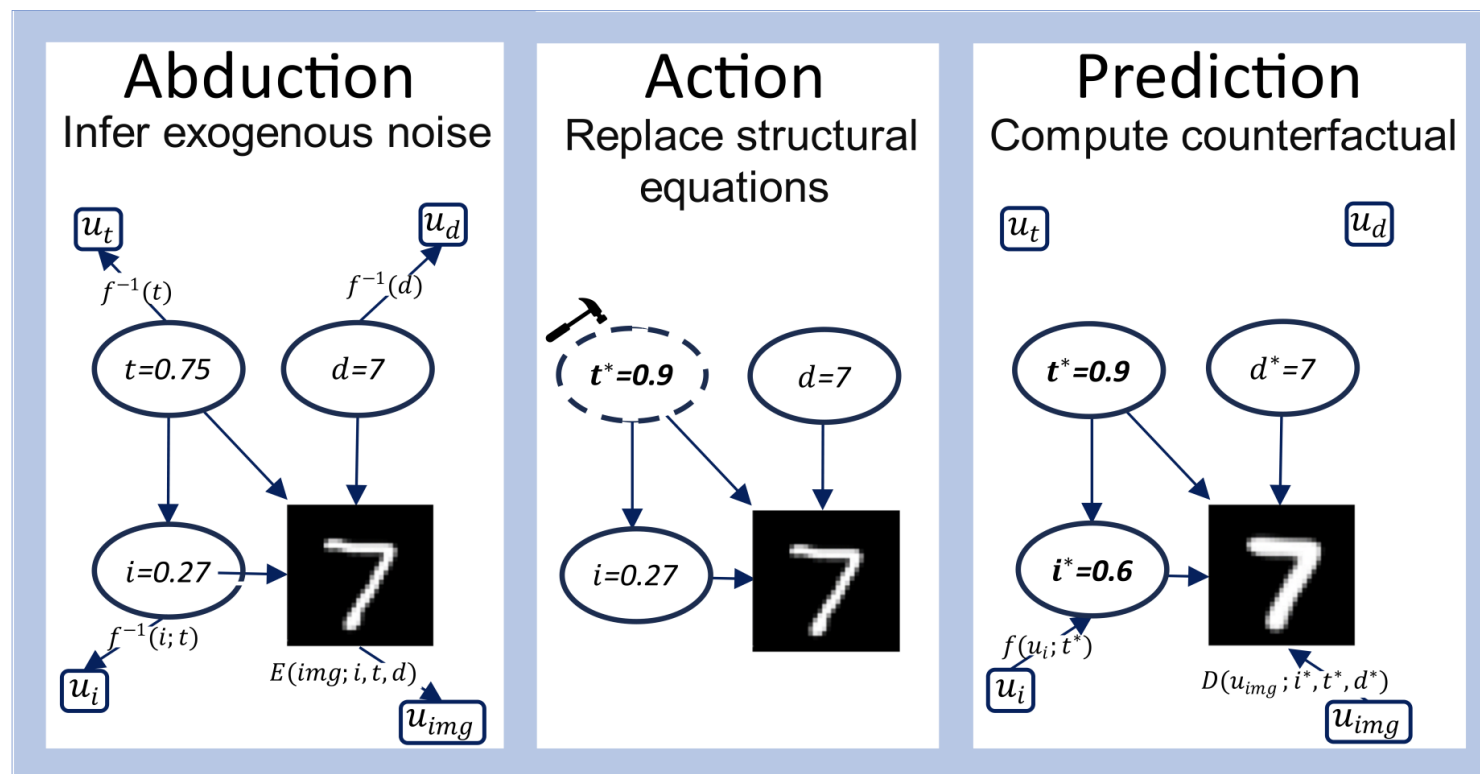
- Query:  $P(I_x, |I', x')$

- Evidence:

$$I', x'$$

- Action:  $do(x)$

- Outcome:  $I_x$



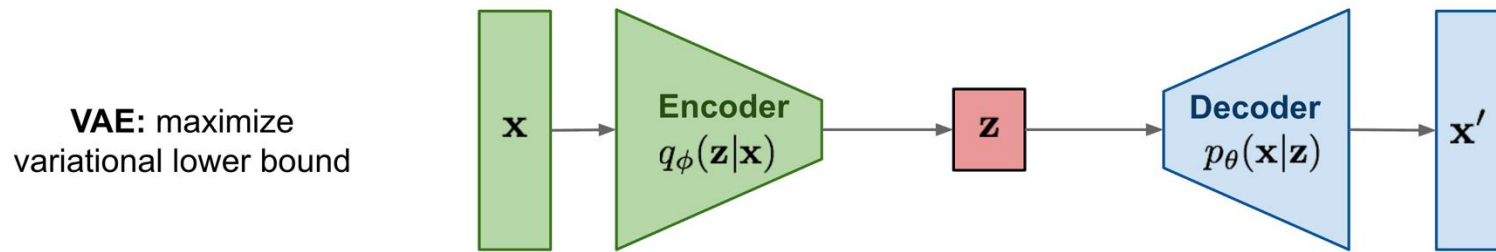
# Feature control requires disentanglement

- Generative models can reconstruct images with new features.
- But correct control over the features requires disentangling causal features from the exogenous/background properties.

# Abduction

- Diffusion models
- Variational auto encoders
- Normalizing flow.

# Basics of a VAE



$$\mathcal{L}_{\text{total}}(x) = \underbrace{-\mathbb{E}_{q_{\phi}(z|x)}[\log p_{\theta}(x|z)]}_{\text{Reconstruction Loss}} + \underbrace{\text{KL}(q_{\phi}(z|x) \parallel p(z))}_{\text{Regularization}}$$

# Problem Definition

- Variational autoencoder (VAE) can be used to disentangle independent factors from observations.
- What if the factors associated with semantics are not independent rather maintains an underlying causal structure?



# Problem Definition

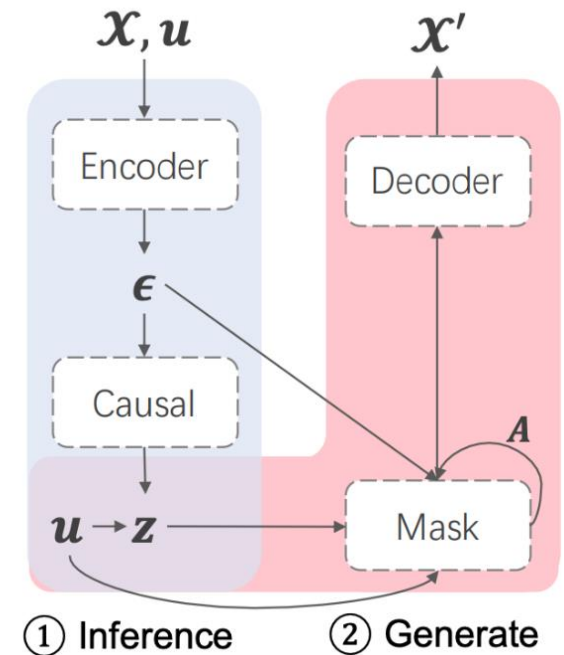
- i) Light position, ii) pendulum angle; causes shadow iii) position and iv) length. – All associated.
- Causal disentangled representation learning + "do-operation"  
= generate new images (ex: without any shadow)



# CausalVAE: Disentangled Representation Learning via Neural Structural Causal Models

Yang, M., Liu, F., Chen, Z., Shen, X., Hao, J., & Wang, J. (2021)

- The encoder takes observation  $x$  as inputs to generate independent exogenous variable  $\epsilon$
- With a prior distribution assumed to be standard Multivariate Gaussian



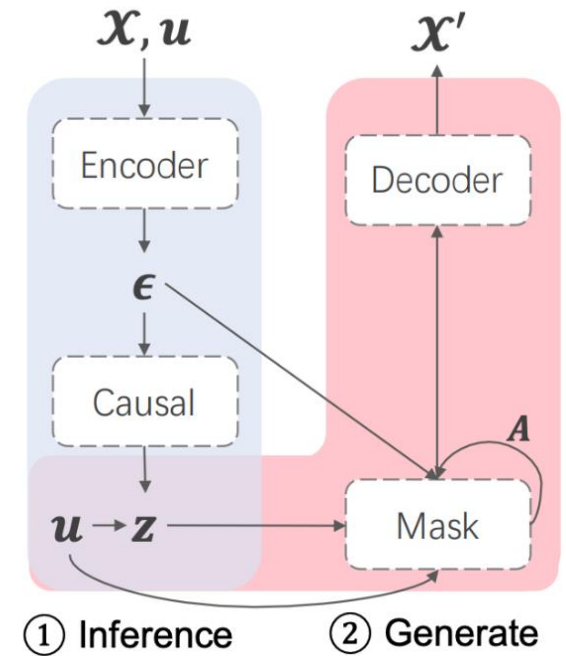
# CausalVAE

- A causal Layer implements a Linear SCM

$$\mathbf{z} = \mathbf{A}^T \mathbf{z} + \boldsymbol{\epsilon} = (\mathbf{I} - \mathbf{A}^T)^{-1} \boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}),$$

Diagram illustrating the equation  $\mathbf{z} = \mathbf{A}^T \mathbf{z} + \boldsymbol{\epsilon}$  with callouts:

- Gaussian Exogenous**: Points to  $\boldsymbol{\epsilon}$ .
- Learnable Parameters**: Points to  $\mathbf{A}$ .
- Causal Representation of n concepts**: Points to  $\mathbf{z}$ .



# CausalVAE

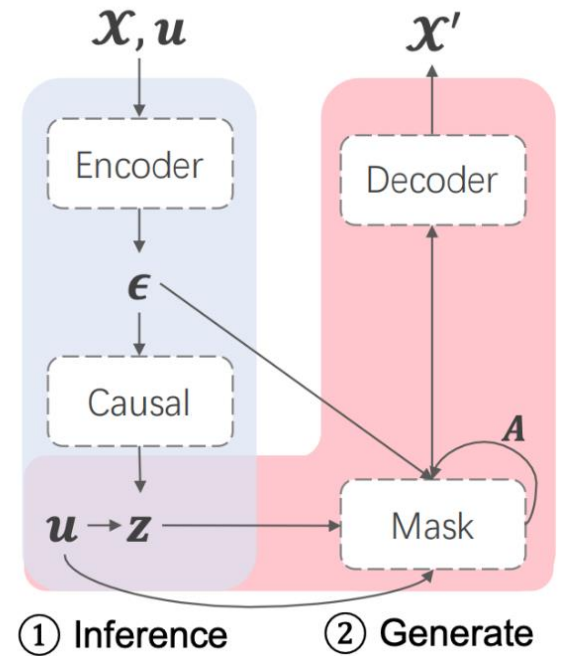
- A Mask Layer to generate children from their corresponding parent variables.

$$z_i = g_i(\mathbf{A}_i \circ \mathbf{z}; \boldsymbol{\eta}_i) + \epsilon_i,$$

Non-linear  
functions for stable  
performance

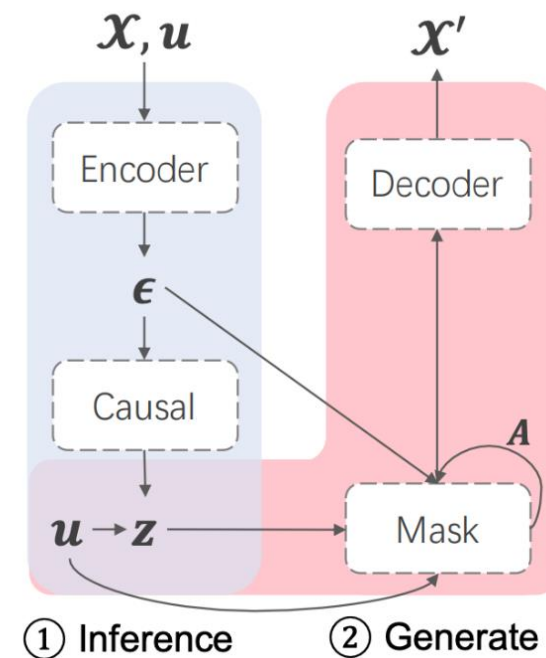
Parent weight  
vector of node i

Parameters  
of  $g_i$



# CausalVAE

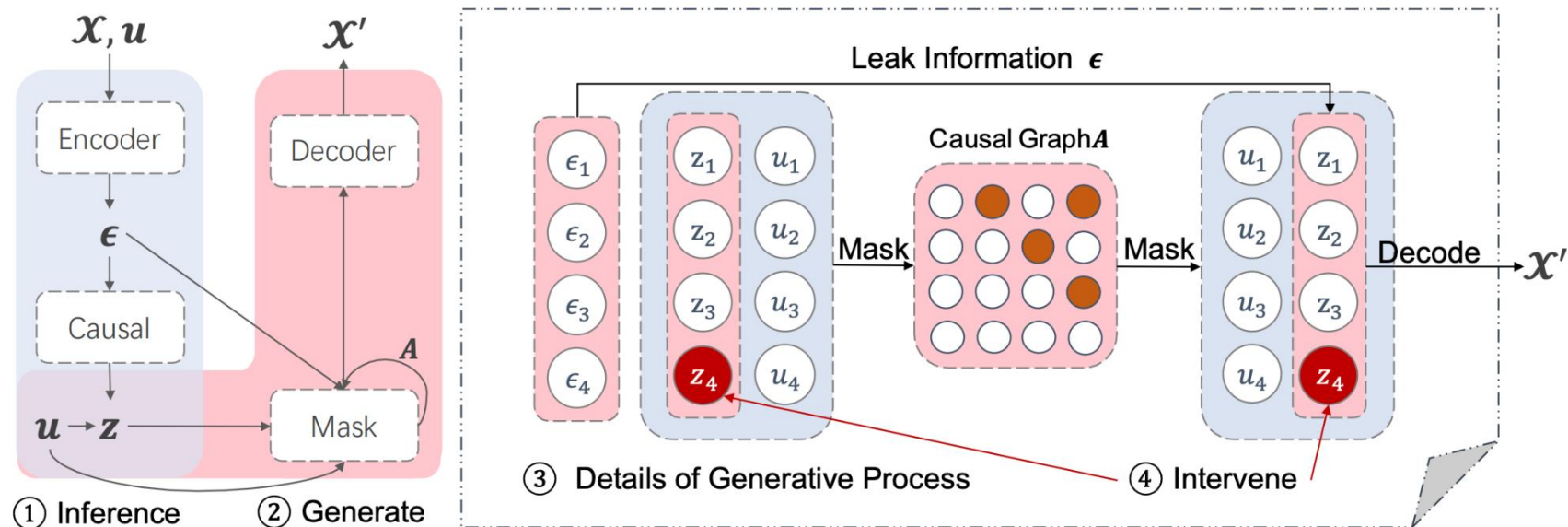
- Finally,  $z$  is taken as the input of the decoder to reconstruct the observation  $x$ .



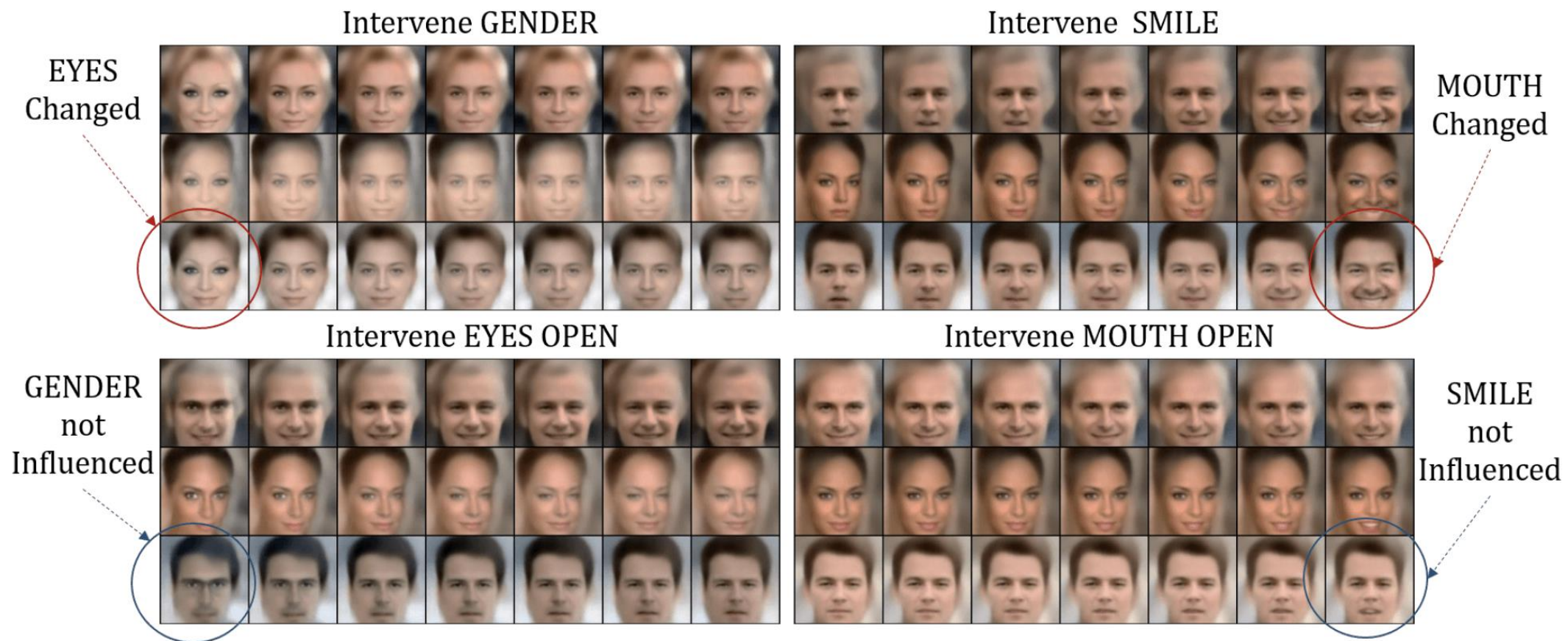
# CausalVAE

- Do-intervention are performed at the Mask layer to propagate the effect to child concepts.

$$z_i = g_i(\mathbf{A}_i \circ \mathbf{z}; \boldsymbol{\eta}_i) + \epsilon_i,$$



# CausalVAE: Results



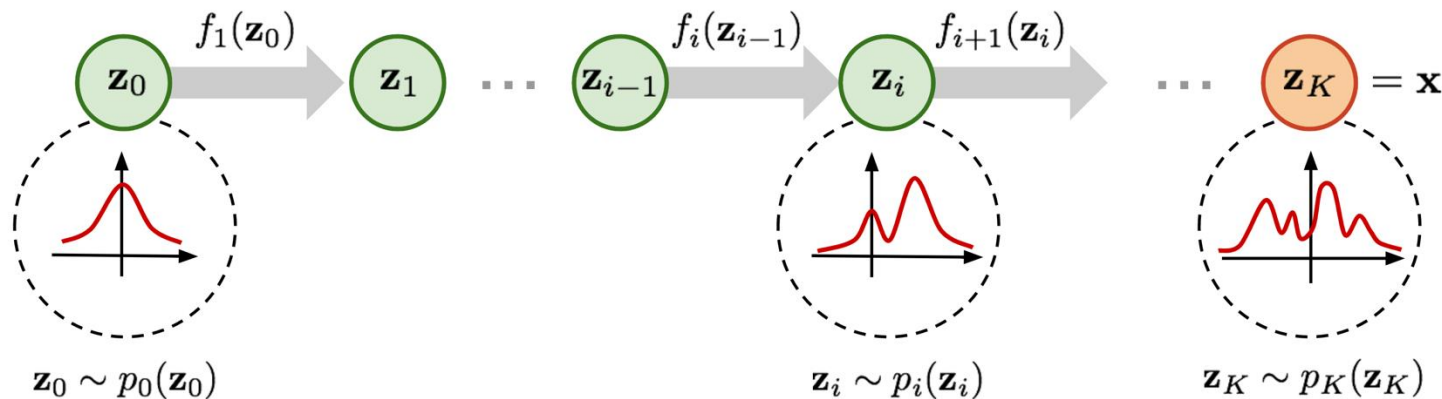
# Problem Definition

- Tractable inference of exogenous noise variables for the abduction step.
- Visualization of counterfactuals in more diverse applications.



# Basics of Normalizing Flow

- Change of variables theorem + A sequence of invertible transformation functions.
- A simple distribution  $\rightarrow$  complex distribution

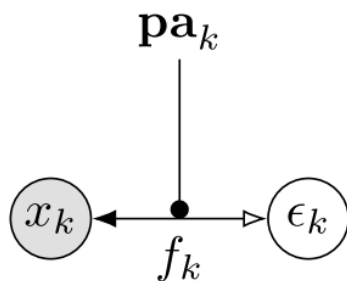


# Deep structural causal models for tractable counterfactual inference (DSCM)

Pawlowski, N., Coelho de Castro, D., & Glocker, B. (2020).

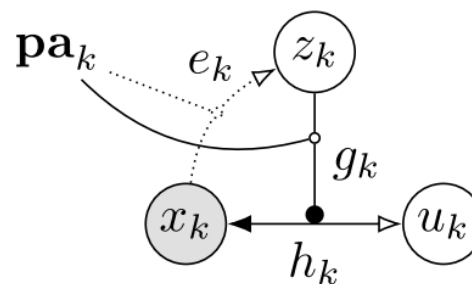
- **Modular Framework:** Unified SCM framework with deep neural mechanisms.
- **Efficient Inference:** Tractable counterfactuals via variational methods or normalizing flows making the complex abduction step efficient.
- **Diverse Applications:** Case studies on synthetic data and brain MRI scans.

# DSCM Architectures



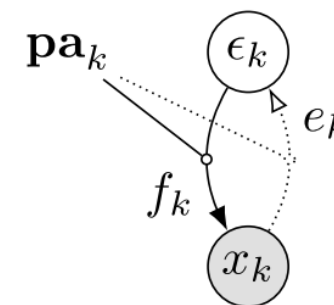
(a) Invertible explicit likelihood

Uses conditional  
Normalizing flow.



(b) Amortised explicit likelihood

Decomposes  
transformation  
complexity and uses  
convolutional neural  
network.



(c) Amortised implicit likelihood

Suggests using  
adversarial objective  
with discriminator  
encoder.

## DSCM: Abduction w/ Normalizing Flows (NF)

- In invertible mechanisms, the exogenous noise can be deterministically and exactly recovered by inverting the mechanism:

$$\epsilon_i = f_i^{-1}(x_i; \mathbf{pa}_i)$$

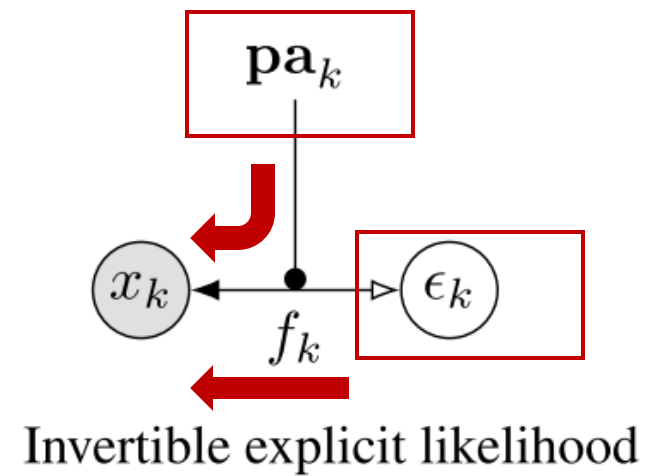
# DSCM Intervention

- A fixed intervention removes the dependency with parents and exogenous noise.
- Replaces the existing mechanism with a surrogate mechanism:

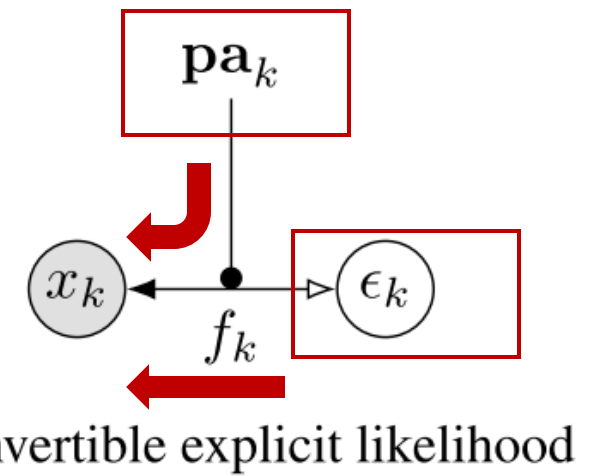
$$x_k := \tilde{f}_k(\epsilon_k; \widetilde{\mathbf{pa}}_k)$$

# DSCM Prediction

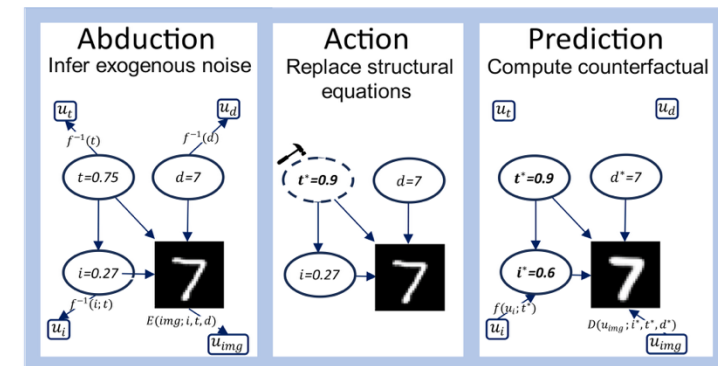
- We have deterministically inverted noise variables found during the abduction step.
- We have the new parent values propagated due to the intervention step.
- Plug back noise into the into the forward model along with new parent values.
- Redundant for not descendants of the intervened variable, as they will be unaffected by the changes



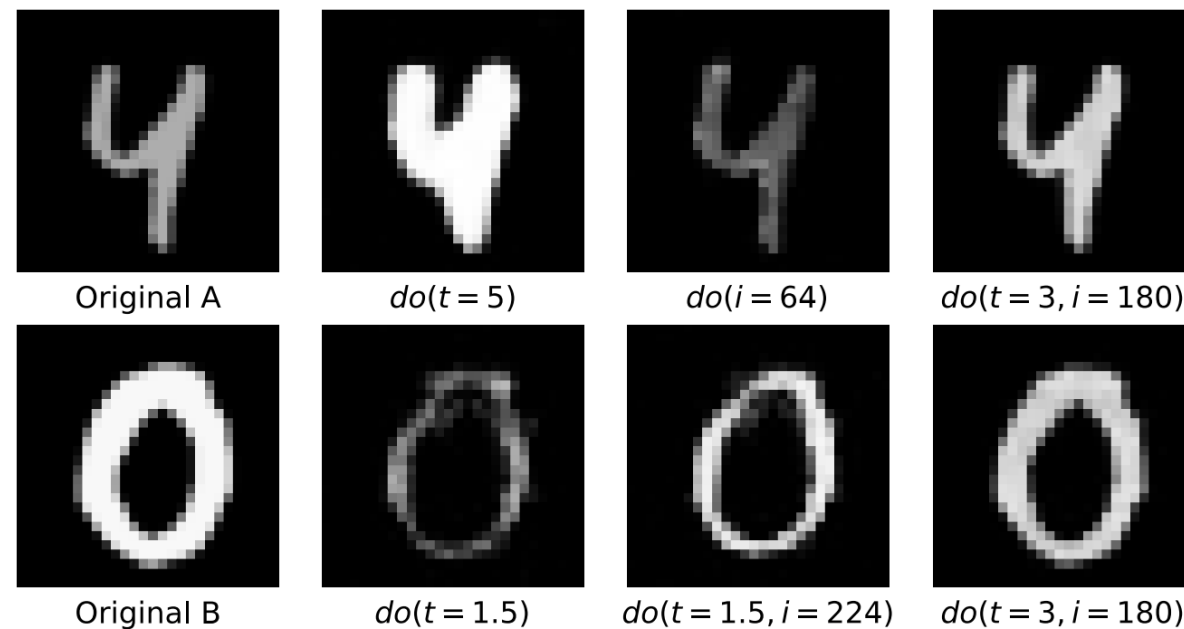
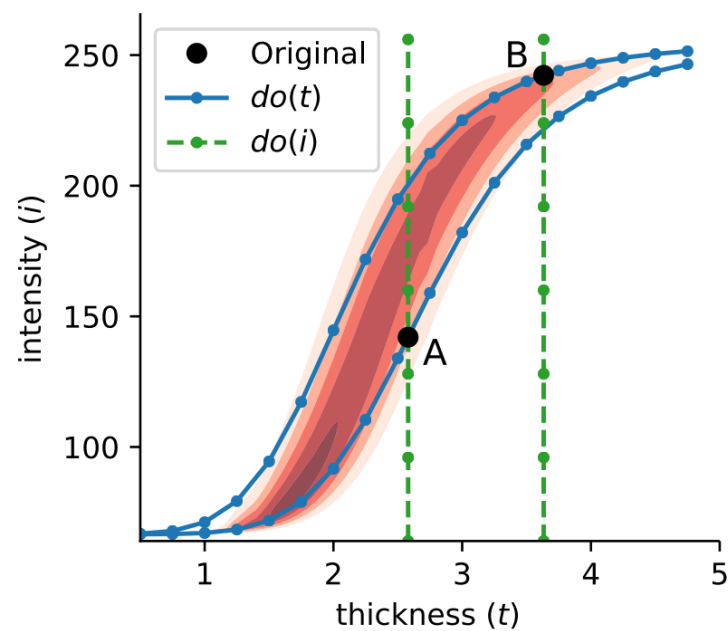
# DSCM Prediction



- Plug back noise into the forward model along with new parent values.
- Redundant for not descendants of the intervened variable, as they will be unaffected by the changes



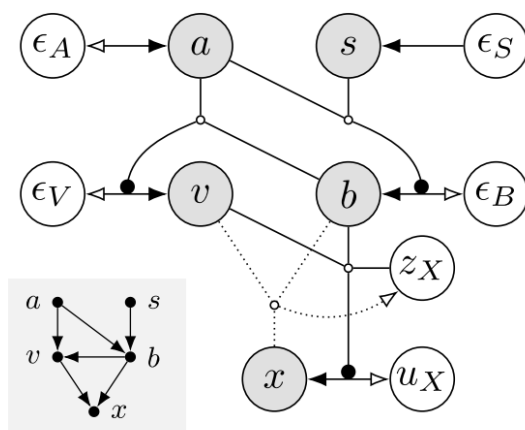
# DSCM Results: MNIST





# DSCM Results: Brain MRI

- Image ( $x$ ), age ( $a$ ), sex ( $s$ ), and brain ( $b$ ) and ventricle ( $v$ ) volumes.
- Different interventions on the same original brain.

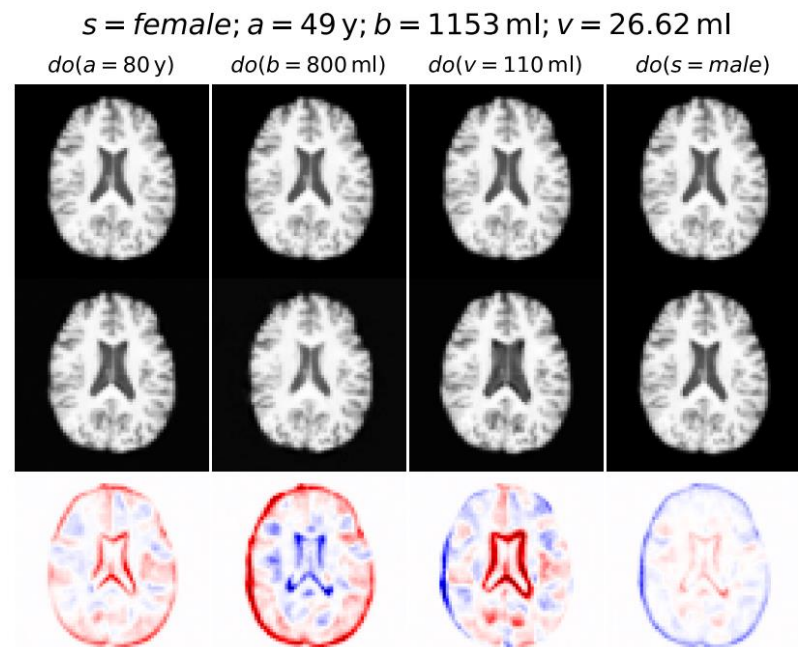


(a) Computational graph

Original image

Counterfactuals

Difference maps



(b) Original image, counterfactuals, and difference maps

## Problem: (Non-)identifiability of Image counterfactuals

- Obtaining unique exogenous noise from observations- challenging!
- Till now, empirical performances under different assumptions.
- Is it even possible to obtain unique counterfactual images in general?

# Counterfactual Image Editing

Pan, Y., & Bareinboim, E. (2024).

- Shows fundamental impossibility results for counterfactual editing.
- Approximates non-identifiable counterfactual distributions with counterfactual consistent estimators.

# ANCM

**Theorem (ID).** *The image counterfactual distribution  $P(\mathbf{I}, \mathbf{I}'_{x'})$  is not identifiable from any combination of  $\langle P(\mathbf{V}, \mathbf{I}), \mathcal{G} \rangle$ .*

- Any image counterfactual distribution is almost never uniquely computable from the observational distribution/samples and the given graph.

# ANCM: causal graph

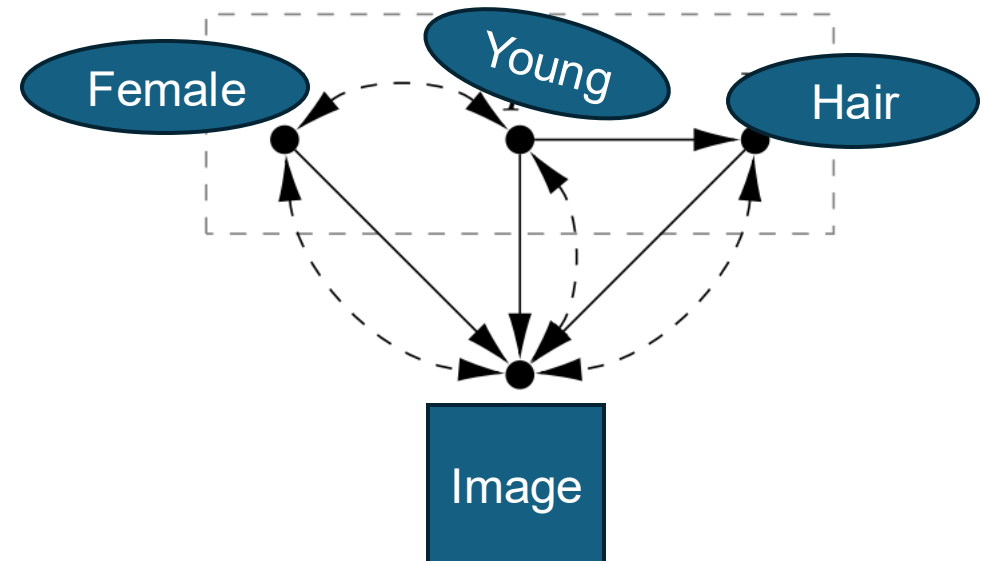
- male  $F = 0$ ; female  $F = 1$
- young  $Y = 0$ ; old  $Y = 1$
- gray  $H = 1$ ; non-gray  $H = 0$

$$U_F \rightarrow F$$

$$U_Y \rightarrow Y$$

$$U_{H_1} \rightarrow H$$

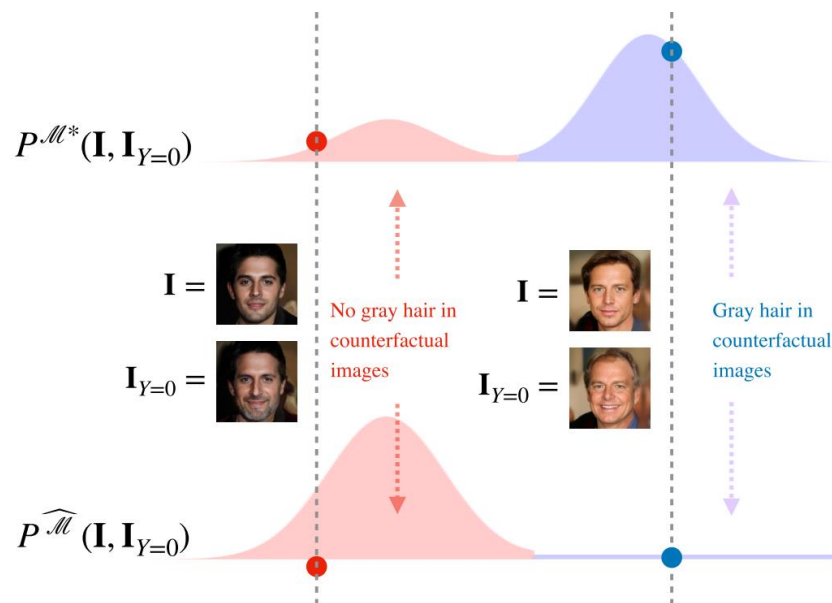
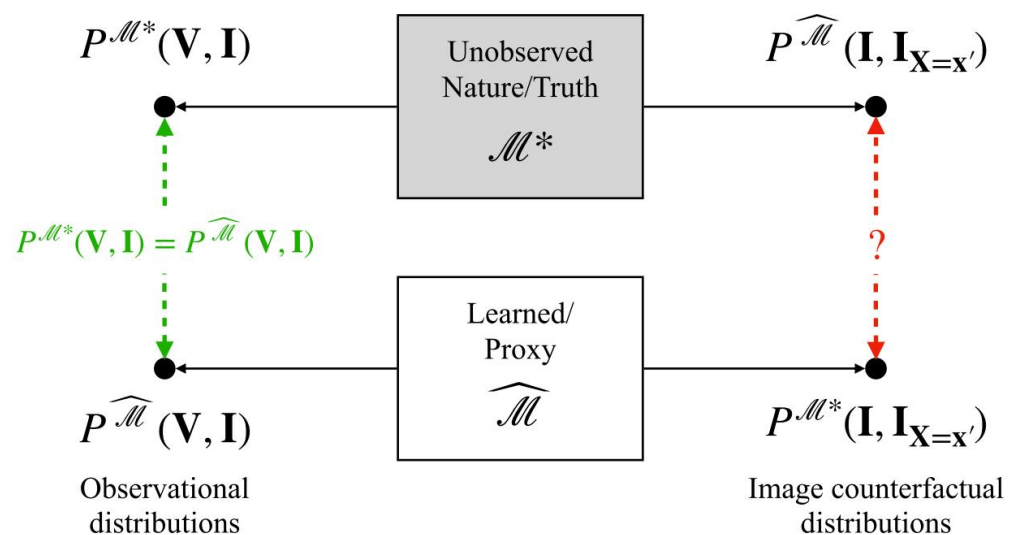
$$U_{H_2} \rightarrow H$$



# ANCM: non-identifiability

$\mathcal{M}^*, \hat{\mathcal{M}}$  Two SCMs similar except

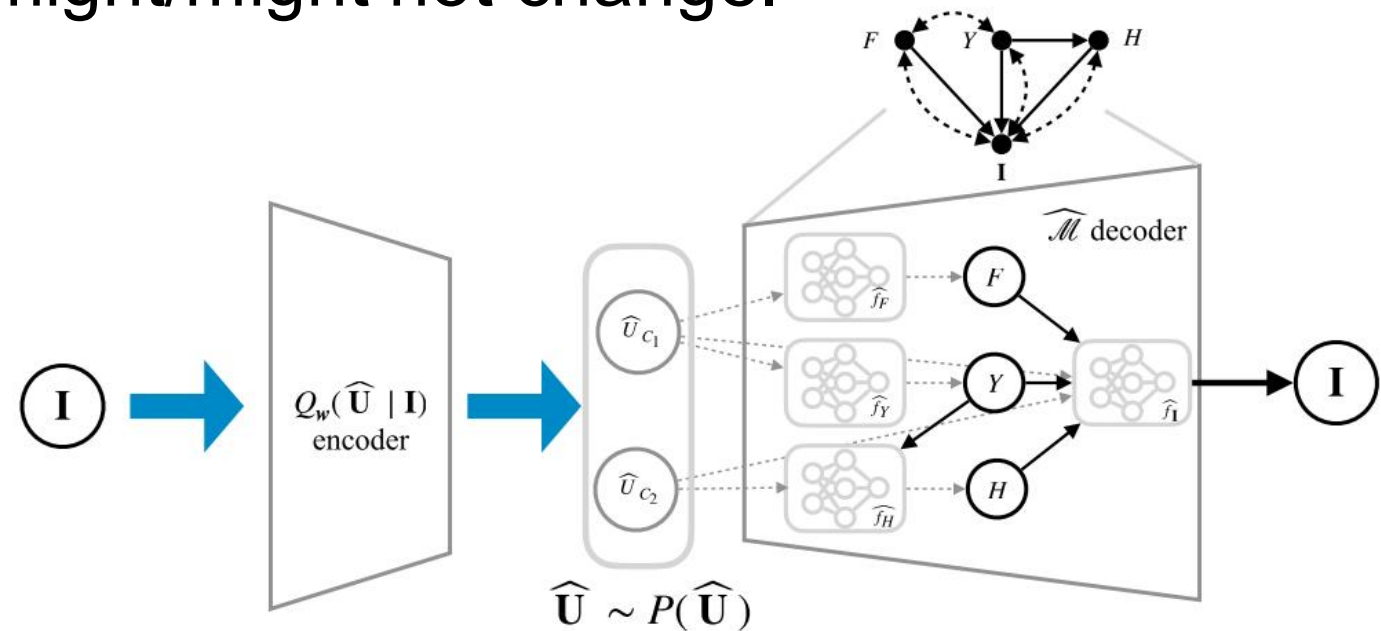
$$\hat{f}_H = (\neg U_Y \wedge U_{H_0}) \oplus (U_Y \wedge U_{H_1})$$



$Y = 0$ : young  
 $Y = 1$ : old

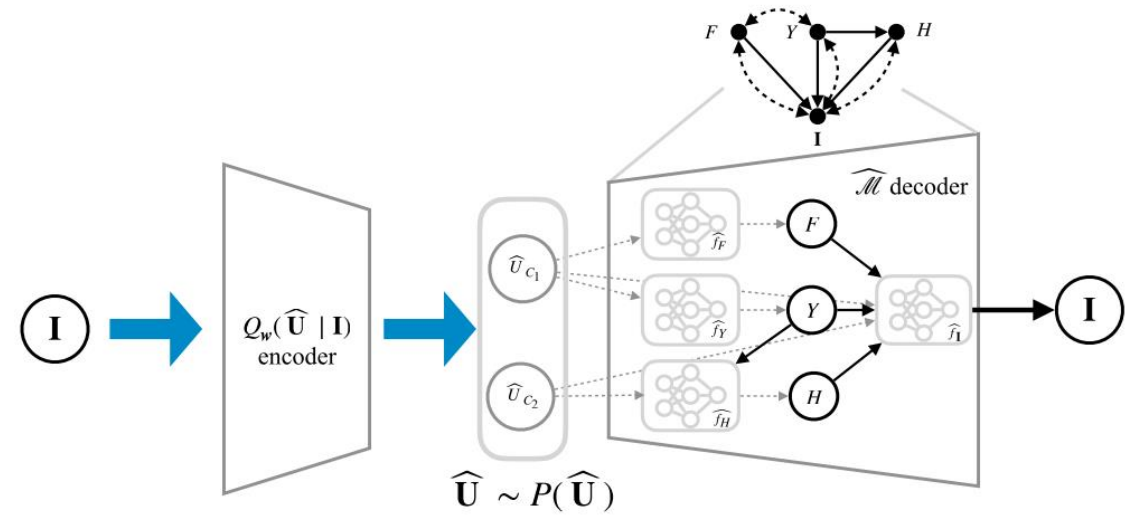
# ANCM: Architecture

- Care set  $W$ : The factors in the image we care to be changed after an intervention. Rest might/might not change.
- Training:
  - Encoder decoder to fit  $P(I)$
  - Neural nets to fit  $P(V|I)$



# ANCM: Inference

- Inference:  $P(I, I_{x'})$ 
  - Sample  $\hat{u} \sim P(\hat{U})$
  - Sample initial  $\hat{i} \sim I^{\hat{M}_{x'}}(\hat{u})$
  - Sample counterfactual  $\hat{i} \sim I^{\hat{M}_{x'}}(\hat{u})$   
where  $\hat{M}_{x'}(\hat{u})$  is sub-model after intervention

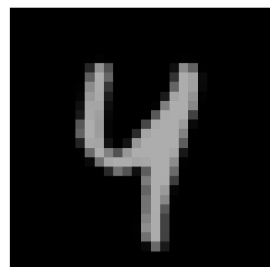




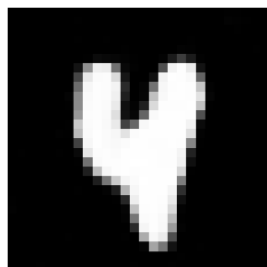
## ANCM: Limitations in Inference

- The proposed inference of the algorithm generates counterfactual pairs from the distribution  $P(I, I'_{x'})$
- Counterfactual Editing by existing work:  $P(I'_{x'}, |I)$

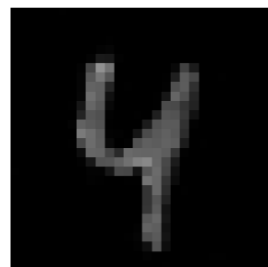
What would the image be had the bar been removed?



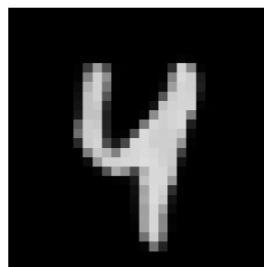
Original A



$do(t = 5)$

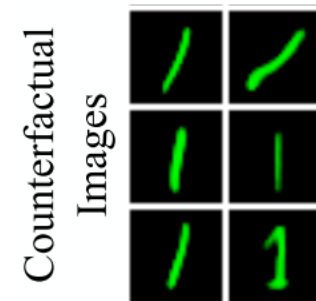
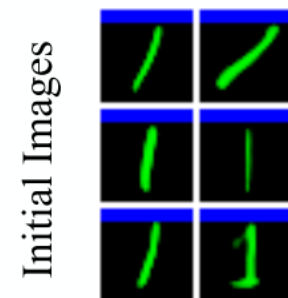


$do(i = 64)$



$do(t = 3, i = 180)$

$$P(I'_{x'}, |I)$$



$$P(I, I'_{x'})$$

# ANCM: Limitations in Inference

- The authors answer the following question in their paper FAQ:
- Q: "Is the editing goal of this paper to change the intervened features in the image and keep other features the same?"
- A: "not necessarily...The goal of this work is to provide causally consistent editing results with the underlying ground truth.  $P(I, I'_{x'})$

Many existing works try to edit certain features and prevent this edit from affecting other features."  $P(I'_{x'}, |I)$

# ANCM: Limitations in Inference

- Stable Diffusion[1], StyleGAN[2] might be strong competitor in empirical performance, but they do not contain any theoretical guarantee.
- Thus, **“counterfactual image editing while keeping background information unchanged with theoretically correctness”**- is still an open problem.

[1] Rombach, Robin, et al. "High-resolution image synthesis with latent diffusion models." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022.

[2] Karras, Tero, Samuli Laine, and Timo Aila. "A style-based generator architecture for generative adversarial networks." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019.

# Benchmarking Counterfactual Image Generation

Melistas, T., Spyrou, N., Gkouti, N., Sanchez, P., Vlontzos, A., Panagakis, Y., ... & Tsaftaris, S. (2024)

## Evaluation:

- Effectiveness: Intervening on a variable to have a specific value will cause the variable to take on that value.
- Composition: Intervening on a variable to have a value it would have had without our intervention will not affect the other variables in the system.
- Reversibility: Can the method reverse the counterfactual image to the original image?

# Evaluation on Images

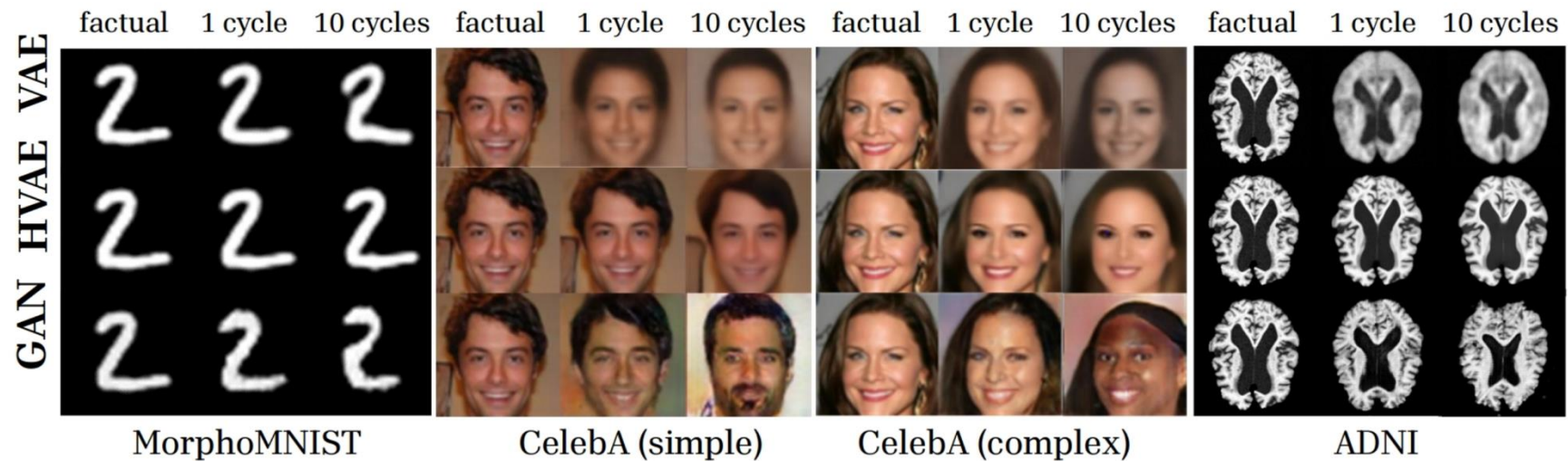
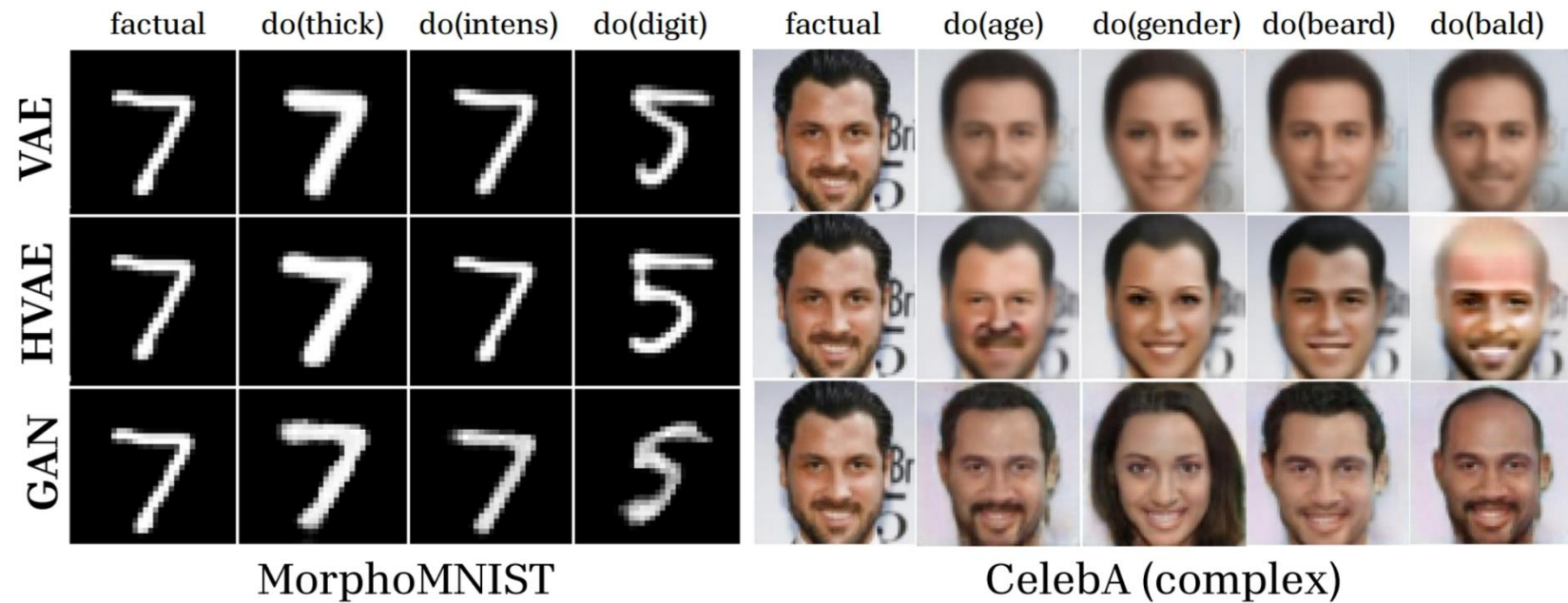
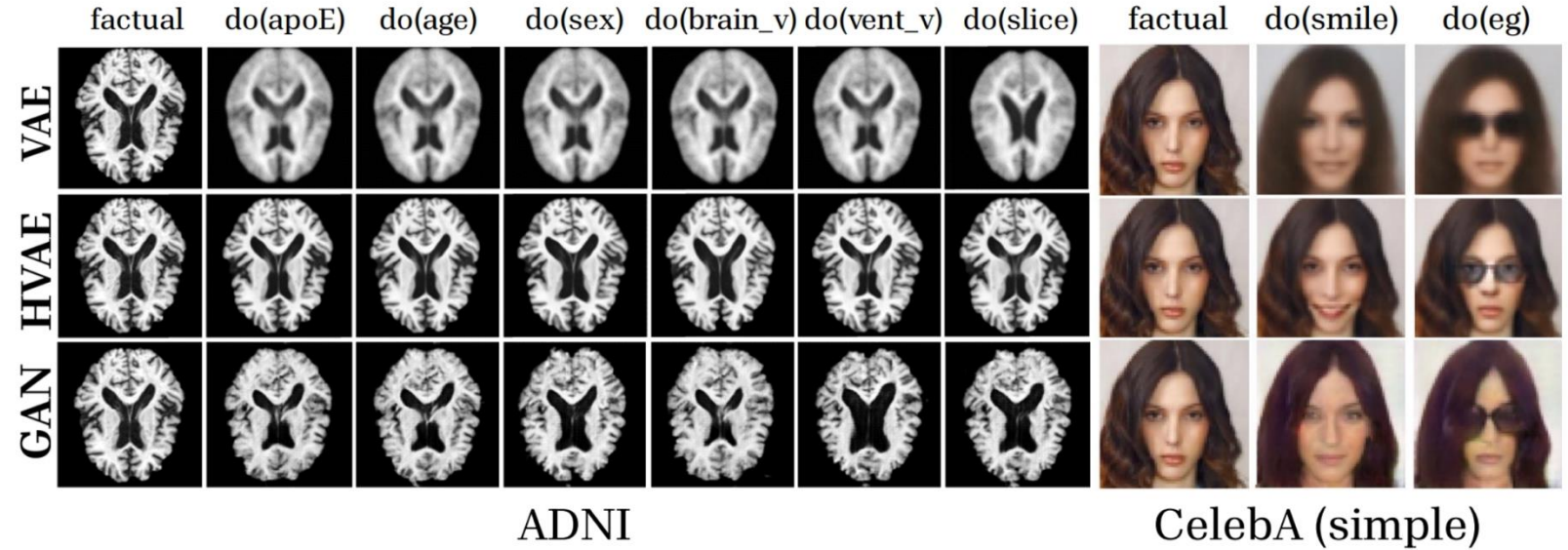


Figure : Qualitative evaluation of composition across all datasets/graphs. From left to right across all datasets: (i) factual, (ii) null-intervention (reconstruction) (iii) 10 cycles of null-intervention

# Evaluation on Images: Effectiveness



# Evaluation on Images: Effectiveness





# Which architecture does perform best?

Table 3: Effectiveness on CelebA test set for both graphs.

CelebA (simple)								
Model	Smiling (s) F1 ↑				Eyeglasses (e) F1 ↑			
	$do(s)$		$do(e)$		$do(s)$		$do(e)$	
VAE	0.897 <sub>0.02</sub>		0.987 <sub>0.01</sub>		0.938 <sub>0.05</sub>		0.810 <sub>0.02</sub>	
HVAE	<b>0.998</b> <sub>0.01</sub>		<b>0.997</b> <sub>0.01</sub>		0.883 <sub>0.06</sub>		<b>0.981</b> <sub>0.02</sub>	
GAN	0.819 <sub>0.02</sub>		0.873 <sub>0.01</sub>		<b>0.957</b> <sub>0.03</sub>		0.891 <sub>0.01</sub>	
CelebA (complex)								
	Age (a) F1 ↑				Gender (g) F1 ↑			
	$do(a)$	$do(g)$	$do(br)$	$do(bl)$	$do(a)$	$do(g)$	$do(br)$	$do(bl)$
VAE	0.35 <sub>0.04</sub>	0.782 <sub>0.02</sub>	0.816 <sub>0.02</sub>	0.819 <sub>0.02</sub>	0.977 <sub>0.01</sub>	0.909 <sub>0.02</sub>	0.959 <sub>0.02</sub>	<b>0.973</b> <sub>0.01</sub>
HVAE	<b>0.654</b> <sub>0.1</sub>	<b>0.893</b> <sub>0.04</sub>	<b>0.908</b> <sub>0.03</sub>	<b>0.899</b> <sub>0.03</sub>	<b>0.988</b> <sub>0.02</sub>	0.949 <sub>0.03</sub>	<b>0.994</b> <sub>0.01</sub>	0.95 <sub>0.03</sub>
GAN	0.413 <sub>0.04</sub>	0.71 <sub>0.02</sub>	0.818 <sub>0.02</sub>	0.799 <sub>0.01</sub>	0.952 <sub>0.01</sub>	<b>0.982</b> <sub>0.01</sub>	0.92 <sub>0.01</sub>	0.961 <sub>0.01</sub>
	Beard (br) F1 ↑				Bald (bl) F1 ↑			
	$do(a)$	$do(g)$	$do(br)$	$do(bl)$	$do(a)$	$do(g)$	$do(br)$	$do(bl)$
VAE	0.944 <sub>0.01</sub>	0.828 <sub>0.03</sub>	0.296 <sub>0.05</sub>	<b>0.945</b> <sub>0.02</sub>	<b>0.023</b> <sub>0.03</sub>	0.496 <sub>0.05</sub>	0.045 <sub>0.04</sub>	0.412 <sub>0.03</sub>
HVAE	<b>0.952</b> <sub>0.03</sub>	<b>0.951</b> <sub>0.03</sub>	<b>0.441</b> <sub>0.11</sub>	0.916 <sub>0.04</sub>	0.02 <sub>0.05</sub>	<b>0.86</b> <sub>0.05</sub>	0.045 <sub>0.07</sub>	<b>0.611</b> <sub>0.04</sub>
GAN	0.908 <sub>0.01</sub>	0.838 <sub>0.02</sub>	0.233 <sub>0.03</sub>	0.907 <sub>0.01</sub>	0.021 <sub>0.02</sub>	0.82 <sub>0.02</sub>	<b>0.055</b> <sub>0.02</sub>	0.492 <sub>0.02</sub>

**Thank You!**