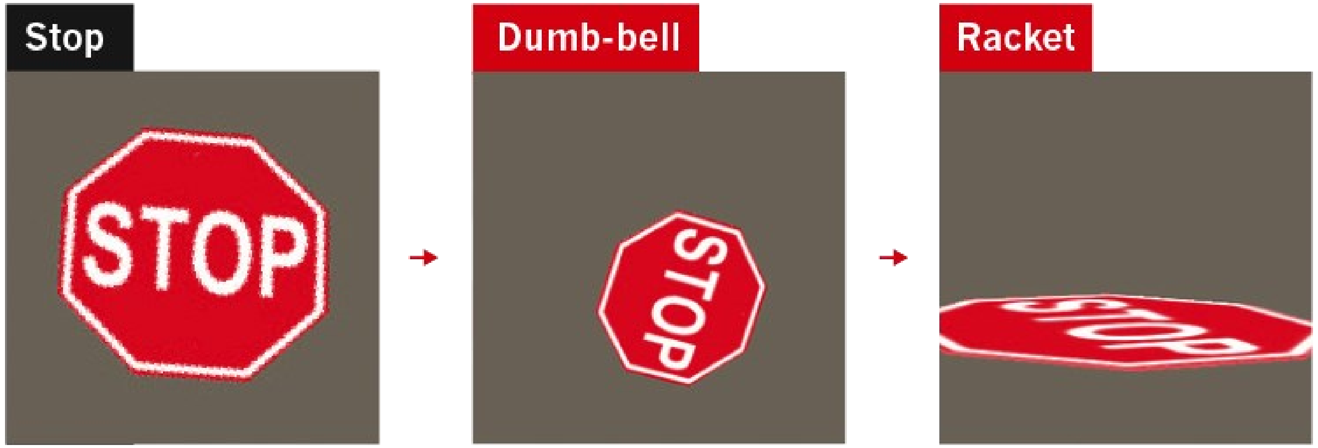
Causal Machine Learning:
Necessary Ingredient for building generalizable models
+
Intro to decision-making using DoWhy

Amit Sharma
Microsoft Research India
Twitter: @amt_shrma

www.amitsharma.in

(A) **Cow: 0.99**, Pasture: 0.99, Grass: 0.99, No Person: 0.98, Mammal: 0.98

(B) No Person: 0.99, Water: 0.98, Beach: 0.97, Outdoors: 0.97, Seashore: 0.97

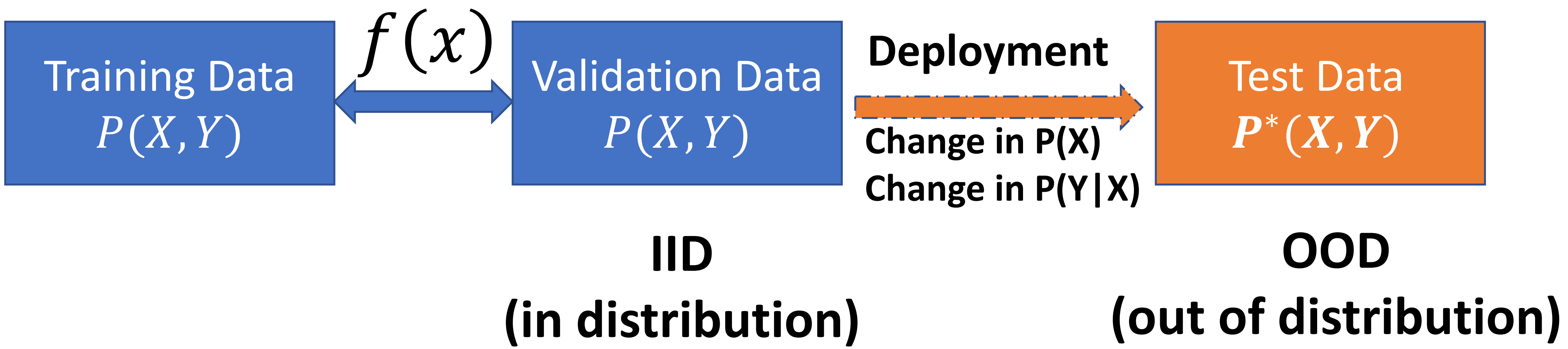


ClarifAI.com. Beery et al. (2018)
Alcorn et al. (2019)

Machine learning has a correlation problem

ML models should have captured the **causal** features
(e.g., cow's pixels, stop sign)

Failure Reason: Independent and identically distributed (IID) assumption.



Learnt correlations become a bigger problem for decision-making

Prediction: If we obtain a new input, what will be the outcome?

E.g., *what will be the heart attack risk for a new person?*

Decision-making: If we change a feature for a given input, how will that impact the outcome?

E.g., *if a person starts exercising, how much does it change the heart attack risk?*

Training Data $P(X,Y)$ $\xleftrightarrow{f(x)}$ Validation Data $P(X,Y)$ $\xrightarrow{\text{Decision}}$ Test Data $\boldsymbol{P^*(X,Y)}$

Change in P(X)
Change in P(Y|X)

IID (in distribution)

OOD (out of distribution)

Today’s session

PART I:

- Out-of-distribution: A key problem for machine learning
- Why causality is necessary for OOD generalization?
- Causal prediction in practice
  - (Conditional) independence regularization
  - Counterfactual augmentations
  - Domain knowledge regularization

PART II:

- Decision-making: A classic causal inference problem
- Important to explicitly state and validate assumptions
- Four steps of causal inference: Model, Identify, Estimate, Refute
  - Code demo using DoWhy

Part I: Causal reasoning is necessary for out-of-distribution generalization

Mahajan, Tople, Sharma. **ICML 2021**. Domain generalization using causal matching.

Kaur, Kiciman, Sharma. [2206.07837] Modeling the Data-Generating Process is Necessary for Out-of-Distribution Generalization (arxiv.org)

State-of-the-art for OOD generalization

Domain generalization

Multiple domains: Assume access to data from multiple distributions

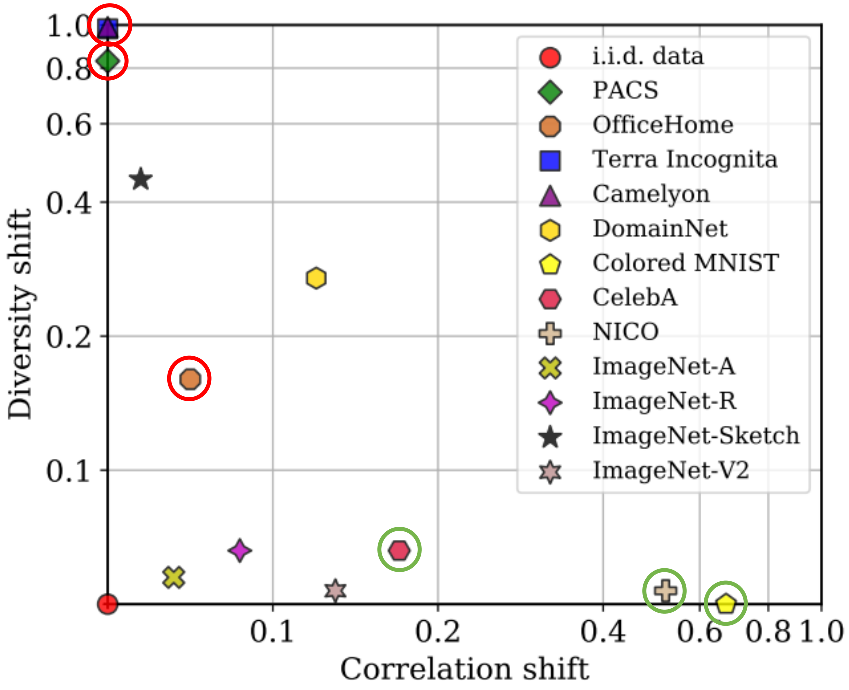
- Learn invariant patterns across the different sources
  - Invariant Risk Minimization (Arjovsky et al., 2019)
  - (Krueger et al. 2020, Ganin et al. 2016, Gulrajani & Lopez-Paz 2021, Nam et al. 2021)

Group generalization

Single domain: Assume access to group attributes for each input

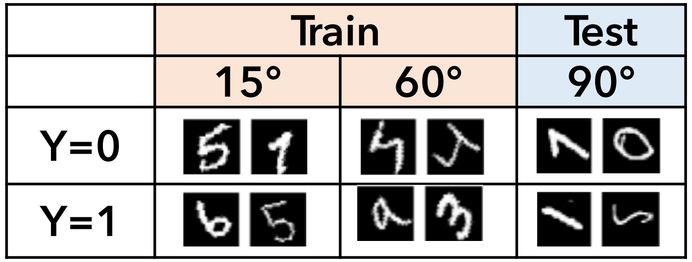
- Equalize accuracy across groups/maximize worst-group accuracy
  - Group-DRO (Sagawa et al., 2020), (Ahmed et al. 2021)

Sobering state of SoTA algorithms

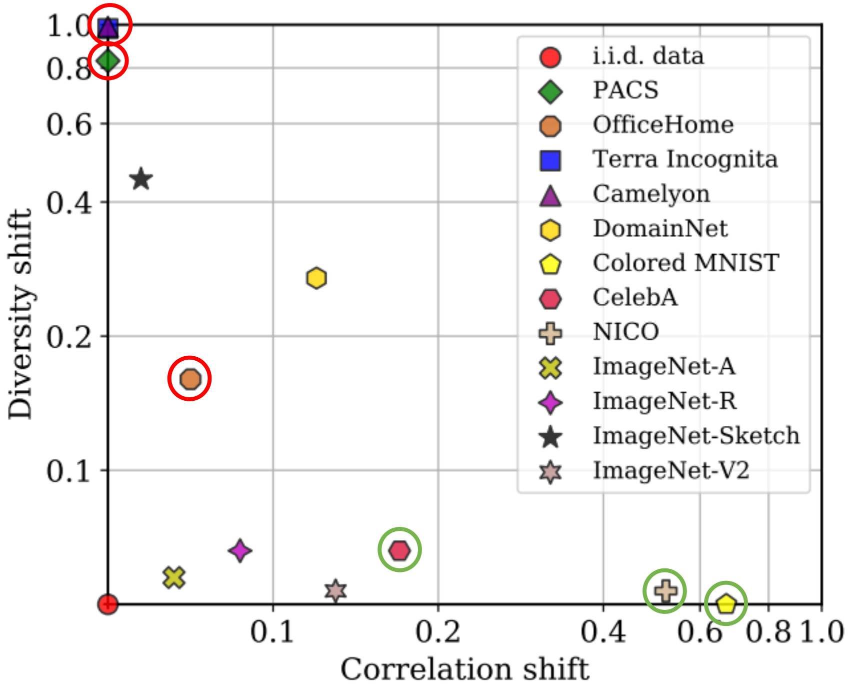


Ye et al., OoD-Bench, CVPR 2022

Sobering state of SoTA algorithms

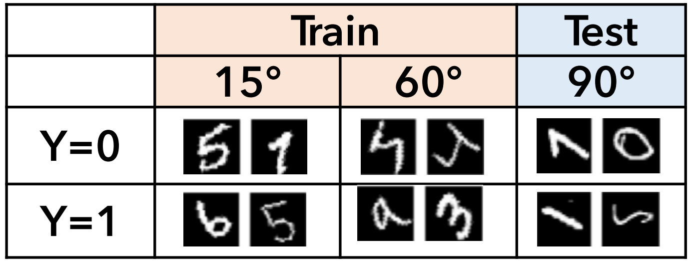


Rotated MNIST

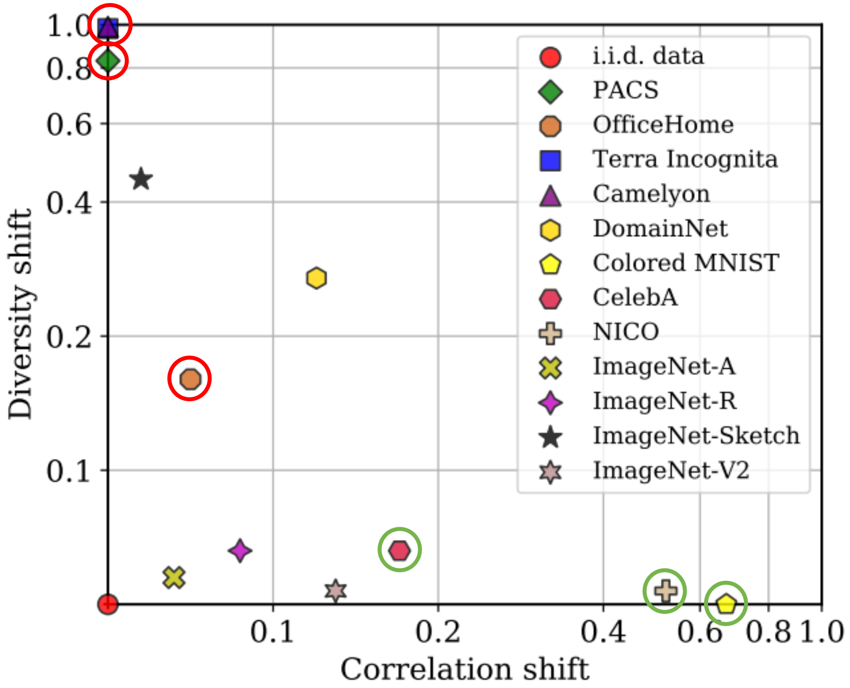


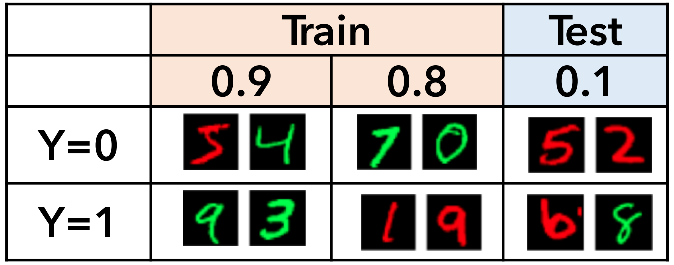
Ye et al., OoD-Bench, CVPR 2022

Sobering state of SoTA algorithms



Rotated MNIST

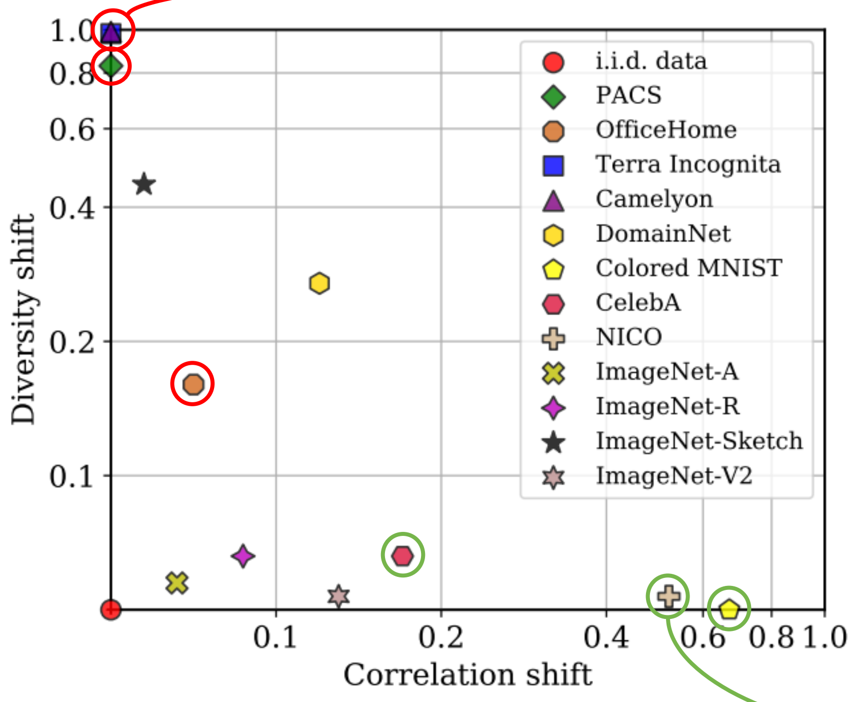




Colored MNIST

Ye et al., OoD-Bench, CVPR 2022

Sobering state of SoTA algorithms

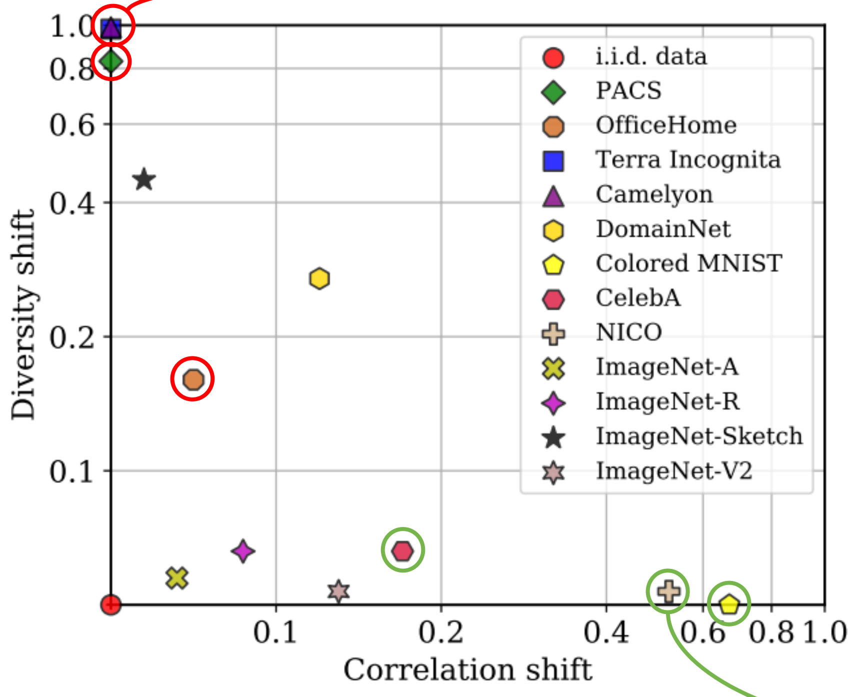


Algorithm	PACS	OfficeHome	TerraInc	Camelyon	Ranking score
MMD [42]	$81.7 \pm 0.2^{\uparrow}$	$63.8 \pm 0.1^{\uparrow}$	$38.3 \pm 0.4^{\downarrow}$	$94.9 \pm 0.4^{\uparrow}$	+2
ERM [69]	81.5 ± 0.0	63.3 ± 0.2	42.6 ± 0.9	94.7 ± 0.1	0
VREx [38]	$81.8 \pm 0.1^{\uparrow}$	63.5 ± 0.1	$40.7 \pm 0.7^{\downarrow}$	$94.1 \pm 0.3^{\downarrow}$	-1
GroupDRO [63]	$80.4 \pm 0.3^{\downarrow}$	63.2 ± 0.2	$36.8 \pm 1.1^{\downarrow}$	$95.2 \pm 0.2^{\uparrow}$	-1

Algorithm	Colored MNIST	CelebA	NICO	Prev score	Ranking score
VREx [38]	$56.3 \pm 1.9^{\uparrow}$	87.3 ± 0.2	71.0 ± 1.3	-1	+1
GroupDRO [63]	$32.5 \pm 0.2^{\uparrow}$	87.5 ± 1.1	71.8 ± 0.8	-1	+1
ERM [69]	29.9 ± 0.9	87.2 ± 0.6	71.4 ± 1.3	0	0
MMD [42]	$50.7 \pm 0.1^{\uparrow}$	$86.0 \pm 0.5^{\downarrow}$	$68.3 \pm 1.0^{\downarrow}$	+2	-1

Ye et al., OoD-Bench, CVPR 2022

Sobering state of SoTA algorithms



Algorithm	**PACS**	**OfficeHome**	**TerraInc**	**Camelyon**	**Ranking score**
MMD [42]	$81.7 \pm 0.2^{\uparrow}$	$63.8 \pm 0.1^{\uparrow}$	$38.3 \pm 0.4^{\downarrow}$	$94.9 \pm 0.4^{\uparrow}$	+2
ERM [69]	81.5 ± 0.0	63.3 ± 0.2	42.6 ± 0.9	94.7 ± 0.1	0
VREx [38]	$81.8 \pm 0.1^{\uparrow}$	63.5 ± 0.1	$40.7 \pm 0.7^{\downarrow}$	$94.1 \pm 0.3^{\downarrow}$	-1
GroupDRO [63]	$80.4 \pm 0.3^{\downarrow}$	63.2 ± 0.2	$36.8 \pm 1.1^{\downarrow}$	$95.2 \pm 0.2^{\uparrow}$	-1

No method can surpass ERM on all kinds of shifts!

Algorithm	**Colored MNIST**	**CelebA**	**NICO**	**Prev score**	**Ranking score**
VREx [38]	$56.3 \pm 1.9^{\uparrow}$	87.3 ± 0.2	71.0 ± 1.3	-1	+1
GroupDRO [63]	$32.5 \pm 0.2^{\uparrow}$	87.5 ± 1.1	71.8 ± 0.8	-1	+1
ERM [69]	29.9 ± 0.9	87.2 ± 0.6	71.4 ± 1.3	0	0
MMD [42]	$50.7 \pm 0.1^{\uparrow}$	$86.0 \pm 0.5^{\downarrow}$	$68.3 \pm 1.0^{\downarrow}$	+2	-1

Ye et al., OoD-Bench, CVPR 2022

Sobering state of SoTA algorithms

IID

[Correlation Shift]

Spurious correlation
b/w category and lighting

[Diversity Shift]

Unseen data shift
unseen azimuth values

Best methods are not consistent over different datasets and shifts

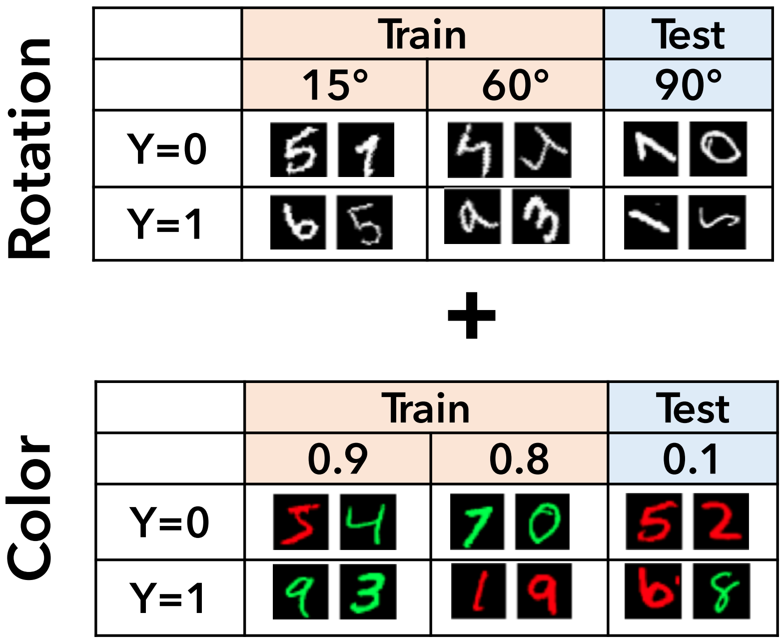
Wiles et al., ICLR 2022

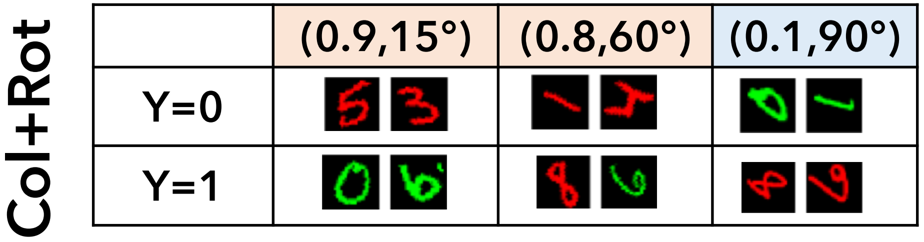
What if different distribution shifts co-exist?

	Train			Test	
Satellite Image (x)					
Year / Region (d)	2002 / Americas	2009 / Africa	2012 / Europe	2016 / Americas	2017 / Africa
Building / Land Type (y)	shopping mall	multi-unit residential	road bridge	recreational facility	educational institution

Koh et al., WILDS, ICML 2021

What if different distribution shifts co-exist?





Accuracy decreases further for all algorithms.

Algorithm	Color	Rotation	Col+Rot
ERM	30.9 ± 1.6	61.9 ± 0.5	25.2 ± 1.3
IRM	50.0 ± 0.1	61.2 ± 0.3	39.6 ± 6.7
MMD	29.7 ± 1.8	62.2 ± 0.5	24.1 ± 0.6
C-MMD	29.4 ± 0.2	62.3 ± 0.4	32.2 ± 7.0

I. Causal reasoning can explain this failure

[single shift] Explain results from causal perspective

- Different distribution shifts arise due to differences in data-generating process (DGP)
  - Leading to different independence constraints
- No single independence constraint can work for all shifts

II. Causal reasoning can provide a better algorithm

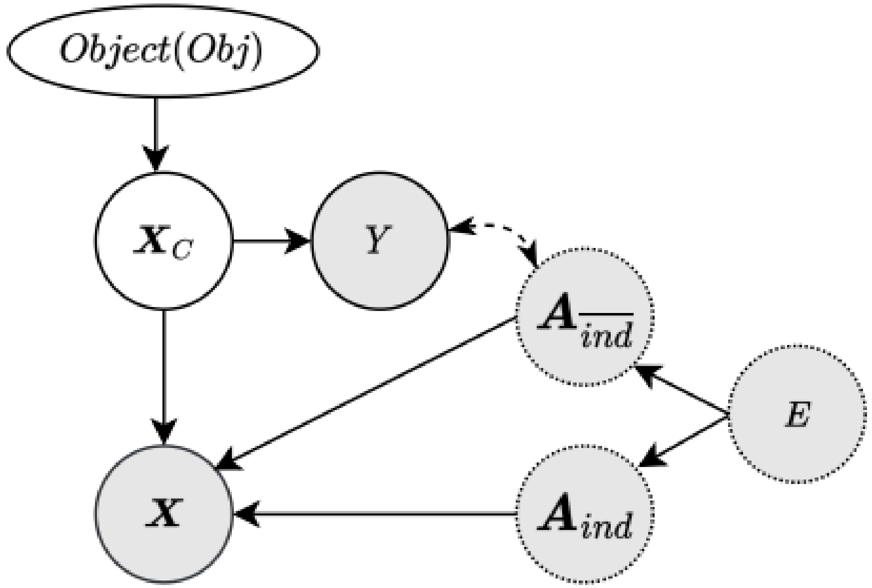
[single shift] Explain results from causal perspective

- Different distribution shifts arise due to differences in data-generating process (DGP)
    - Leading to different independence constraints
- No single independence constraint can work for all shifts

[multi-shift] Can we develop an algorithm that generalizes to individual as well as multi-attribute shifts?

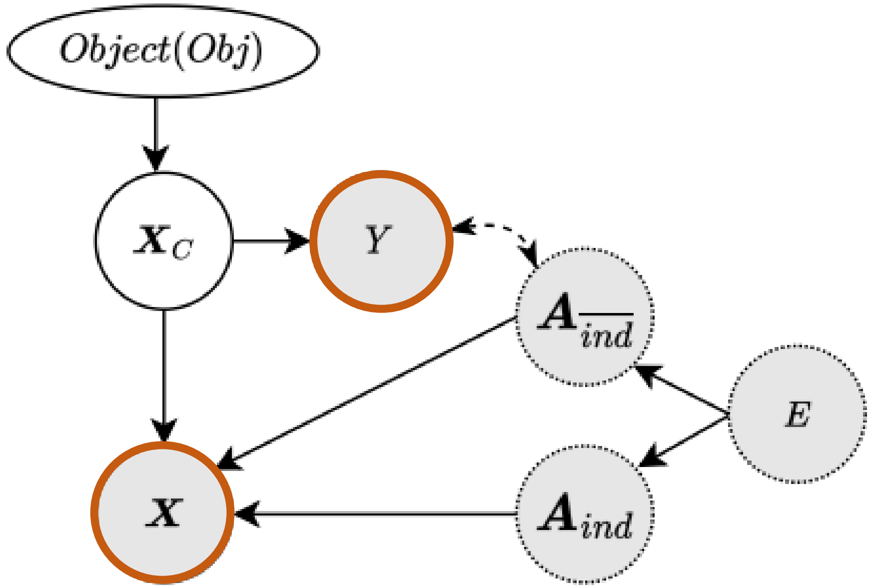
- We propose *Causally Adaptive Constraint Minimization (CACM)* to model the causal relationships in DGP

Representation of shifts using causal graph



Causal DAG to specify multi-attribute shifts

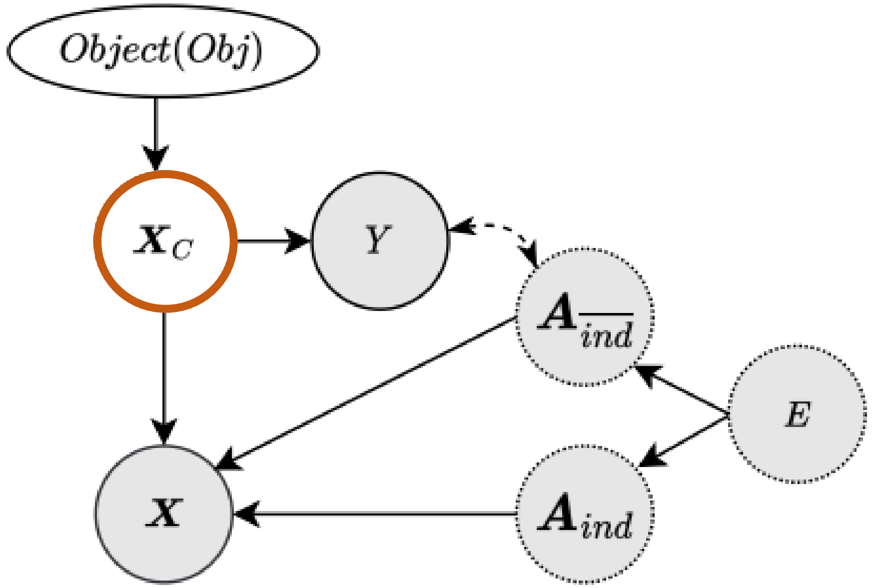
Representation of shifts using causal graph



Observed variables $\boldsymbol{X}, Y$

Causal DAG to specify multi-attribute shifts

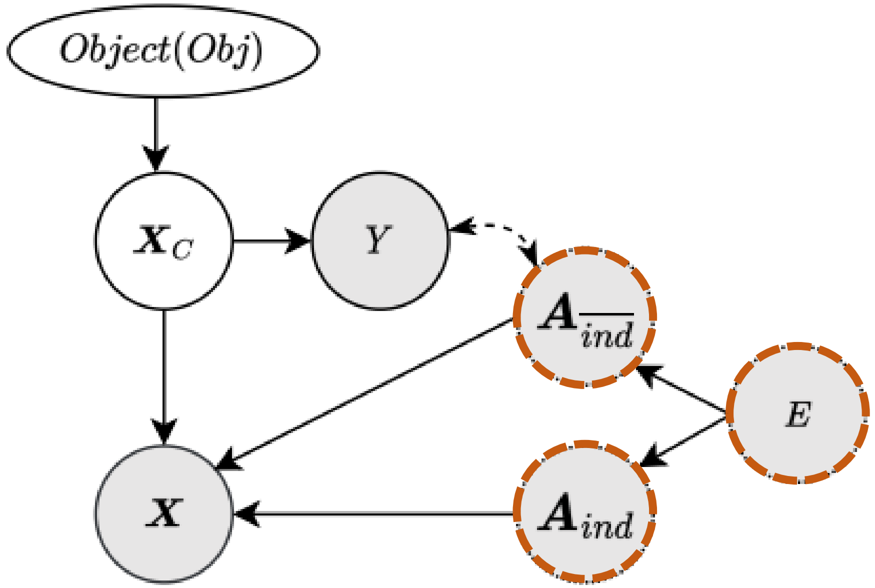
Representation of shifts using causal graph



Observed variables $\boldsymbol{X}, Y$
Causal features $\boldsymbol{X_c}$

Causal DAG to specify multi-attribute shifts

Representation of shifts using causal graph



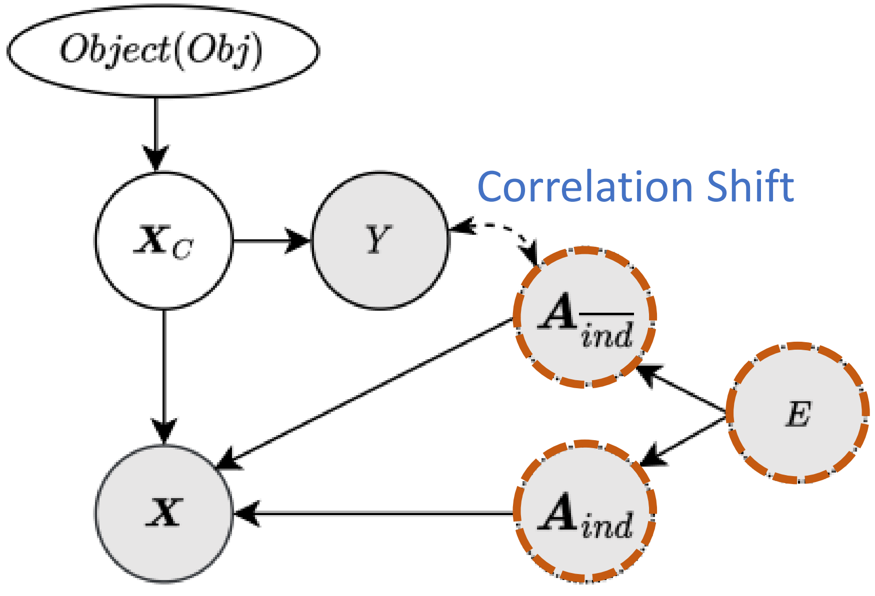
Causal DAG to specify multi-attribute shifts

Observed variables $\boldsymbol{X}, Y$

Causal features $\boldsymbol{X_c}$

Attributes $\boldsymbol{A_{ind}}, \boldsymbol{A_{\overline{ind}}}, E$ st $\boldsymbol{A_{ind}} \cup \boldsymbol{A_{\overline{ind}}} \cup \{E\} = \boldsymbol{A}$

Representation of shifts using causal graph



Causal DAG to specify multi-attribute shifts

Observed variables $\boldsymbol{X}, Y$

Causal features $\boldsymbol{X_c}$

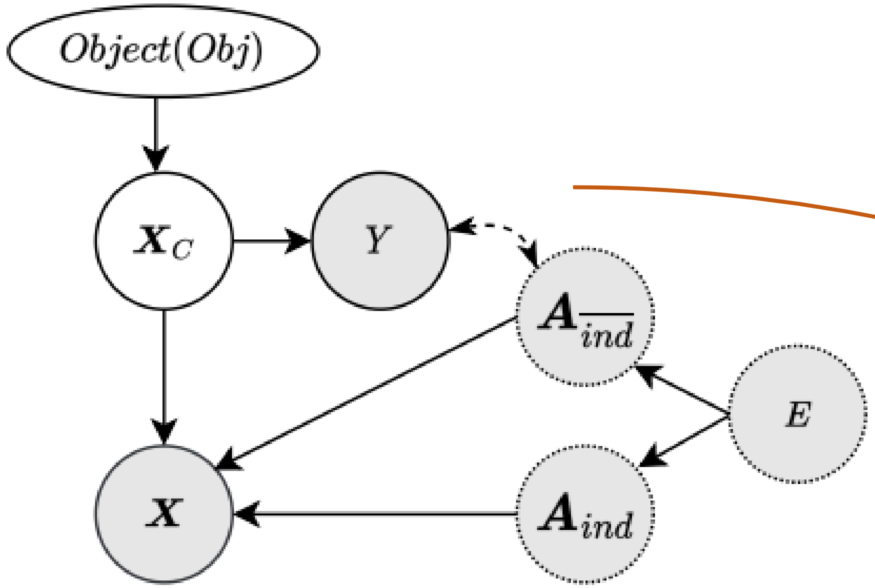
Attributes $\boldsymbol{A_{ind}}, \boldsymbol{A_{\overline{ind}}}, E$ st $\boldsymbol{A_{ind}} \cup \boldsymbol{A_{\overline{ind}}} \cup \{E\} = \boldsymbol{A}$

independent of label

correlated with label

domain attribute

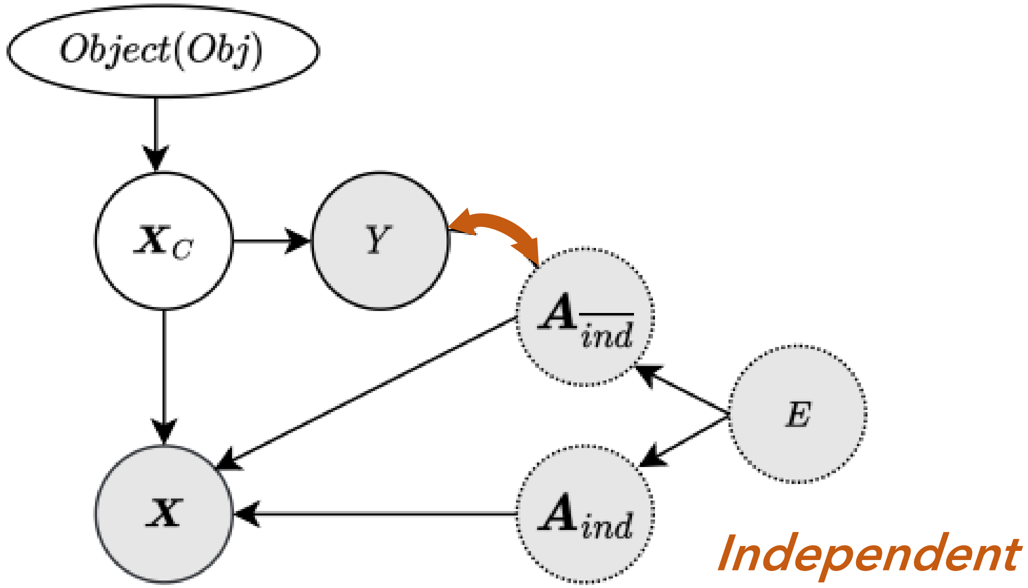
Representation of shifts using causal graph



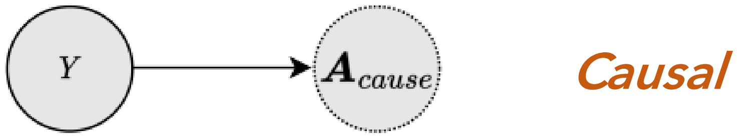
Causal DAG to specify multi-attribute shifts

Different $Y - \boldsymbol{A}_{\overline{ind}}$ relationships

Representation of shifts using causal graph

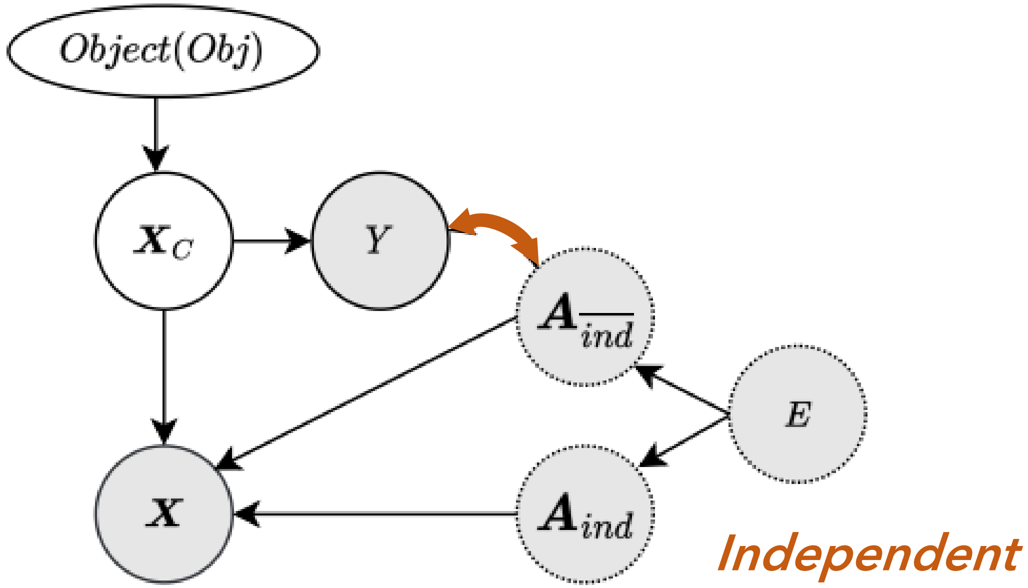


Causal DAG to specify multi-attribute shifts

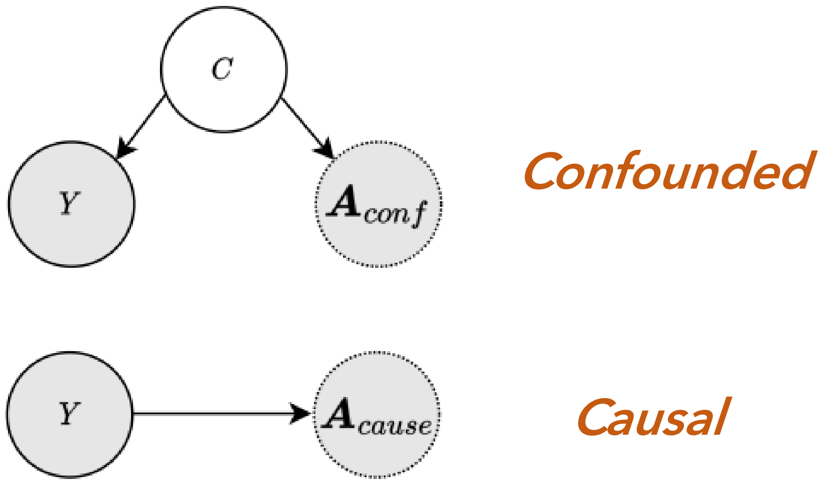


Different $Y - A_{\overline{ind}}$ relationships

Representation of shifts using causal graph

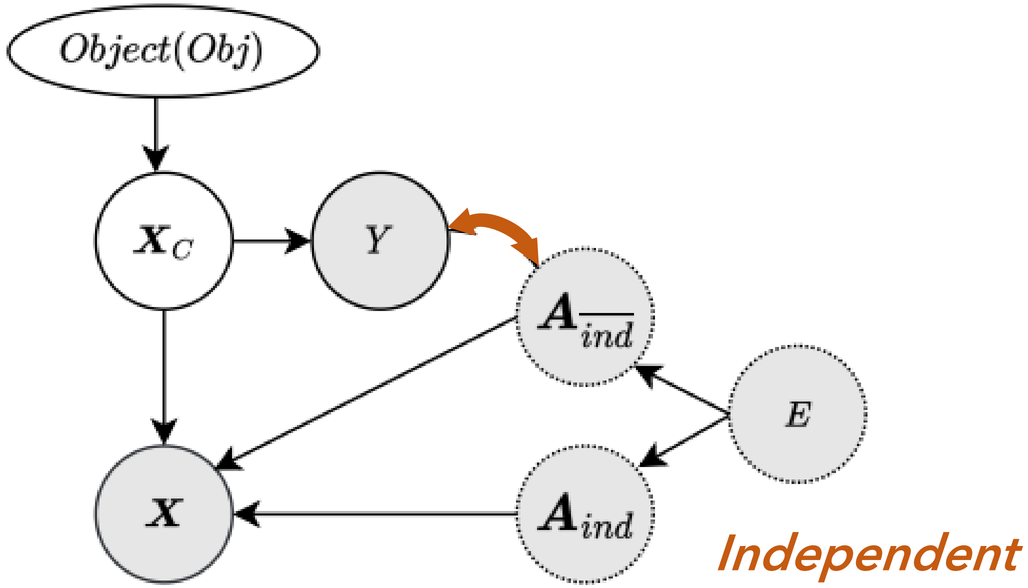


Causal DAG to specify multi-attribute shifts

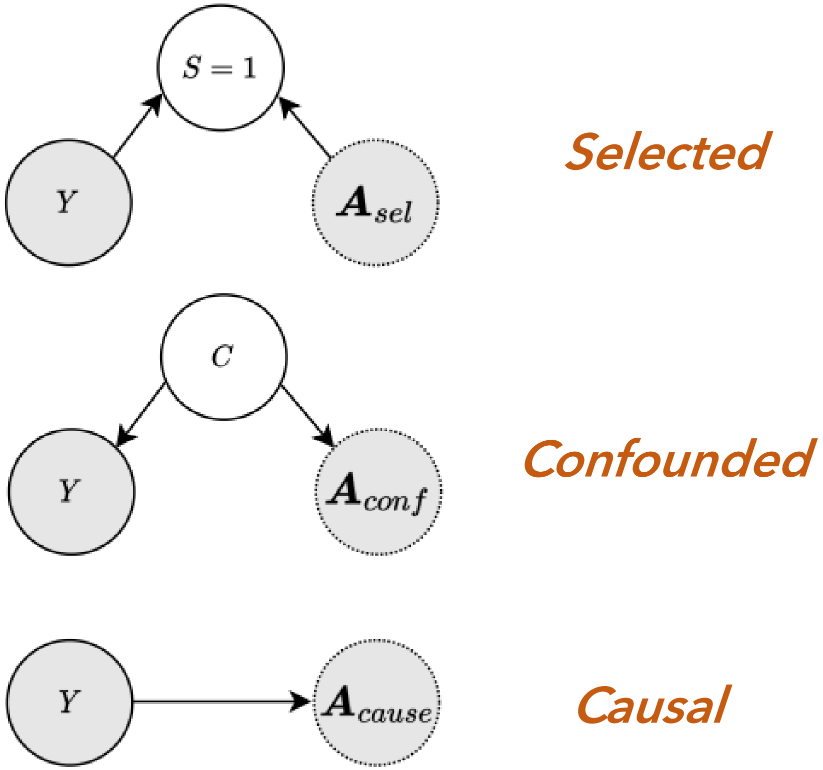


Different $Y - \boldsymbol{A}_{\overline{ind}}$ relationships

Representation of shifts using causal graph



Causal DAG to specify multi-attribute shifts



Different $Y - A_{\overline{ind}}$ relationships

Back to the MNIST example

Rotation

	Train		Test
	15°	60°	90°
Y=0			
Y=1			

A_{ind}

+

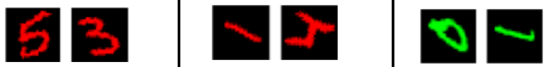
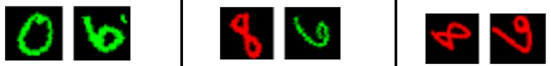
Color

	Train		Test
	0.9	0.8	0.1
Y=0			
Y=1			

A_{cause}

$(A_{\overline{ind}})$

Col+Rot

	(0.9,15°)	(0.8,60°)	(0.1,90°)
Y=0			
Y=1			

Causal + Independent

$A_{cause} \cup A_{ind}$

Generalization to multi-attribute shifts

Algorithm	Color	Rotation	Col+Rot
ERM	30.9 ± 1.6	61.9 ± 0.5	25.2 ± 1.3
IRM	50.0 ± 0.1	61.2 ± 0.3	39.6 ± 6.7
MMD	29.7 ± 1.8	62.2 ± 0.5	24.1 ± 0.6
C-MMD	29.4 ± 0.2	62.3 ± 0.4	32.2 ± 7.0
CACM	**70.4 ± 0.5**	**62.4 ± 0.4**	**54.1 ± 0.3**

CACM outperforms on individual as well as combination of shifts

The *CACM* Approach

Identifying the correct regularizer under multi-attribute shifts

The *CACM* Approach

Identifying the correct regularizer under multi-attribute shifts

I. Derive correct independence constraints for $\boldsymbol{X_c}$ based on causal graph

II. Apply the constraints as regularizer to standard ERM loss.

Step I: Deriving independence constraints

Predictor $g(\boldsymbol{x}) = g_1(\phi(\boldsymbol{x}))$

Representation ϕ should follow same conditional independence constraints as $\boldsymbol{X_c}$

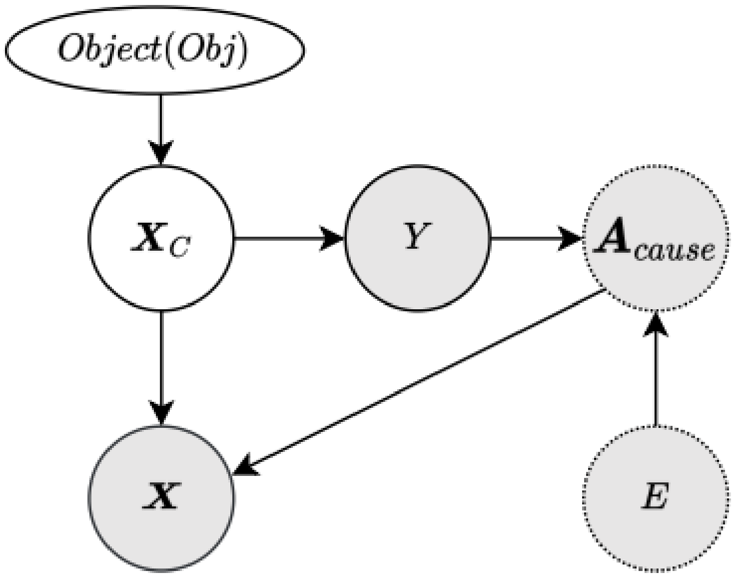
Mahajan et al., ICML 2021; Veitch et al., NeurIPS 2021; Makar et al., AISTATS 2022

Step I: Deriving independence constraints

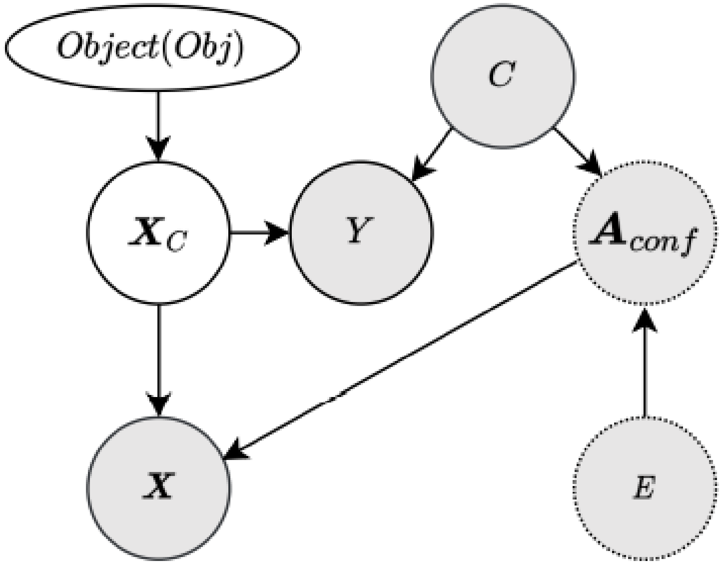
Predictor $g(\boldsymbol{x}) = g_1(\phi(\boldsymbol{x}))$
Representation ϕ should follow same conditional independence constraints as $\boldsymbol{X_c}$

Proposition 3.1. Given a dataset $(\boldsymbol{x}_i, \boldsymbol{a}_i, y_i)_{i=1}^{n}$ and a causal DAG over $\langle\, \boldsymbol{X_c}, \boldsymbol{X}, \boldsymbol{A}, Y \,\rangle$ such that $\boldsymbol{X_c}$ is the only variable (or set of variables) that causes Y and is not independent of $\boldsymbol{X}$, then the conditional independence constraints satisfied by $\boldsymbol{X_c}$ are necessary for a risk-invariant predictor.

Step I: Deriving independence constraints



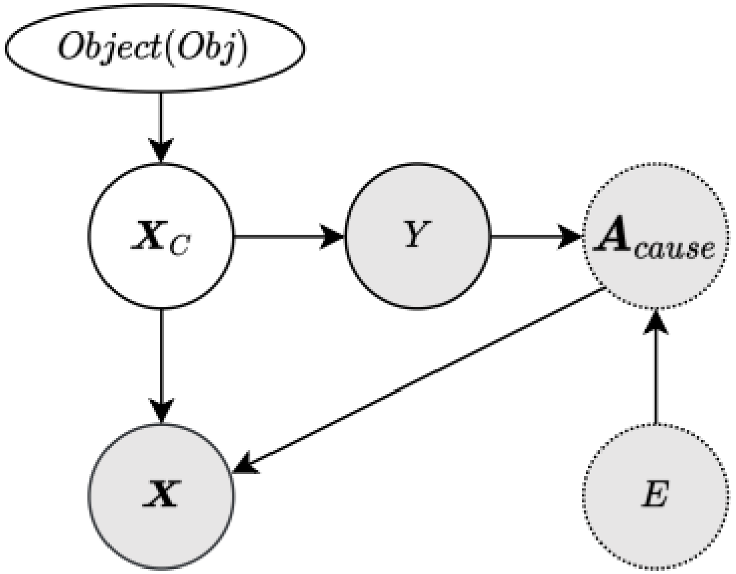
Causal



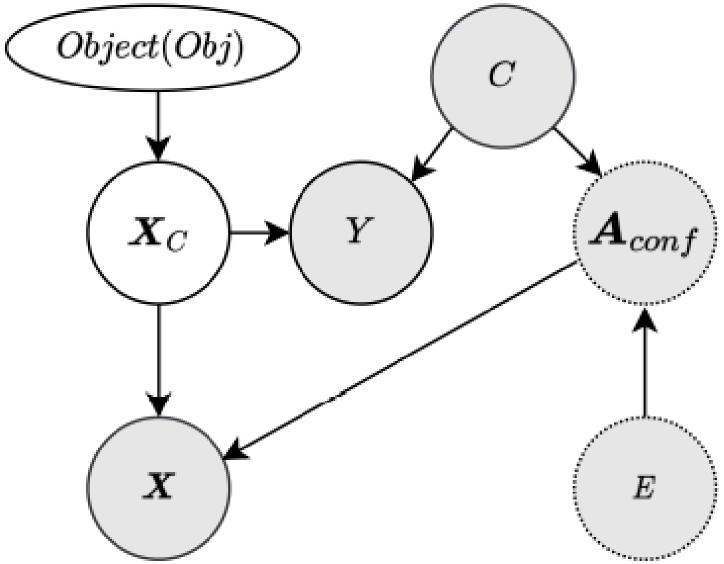
Confounded

Different $Y - \boldsymbol{A}_{\overline{ind}}$ relationships lead to different constraints

Step I: Deriving independence constraints



Causal



Confounded

$\boldsymbol{X_c} \perp\!\!\!\perp \boldsymbol{A_{cause}} \mid Y, E$ ✓
$\boldsymbol{X_c} \perp\!\!\!\perp \boldsymbol{A_{cause}} \mid E$ ✗

$\boldsymbol{X_c} \perp\!\!\!\perp \boldsymbol{A_{conf}} \mid Y, E$ ✗
$\boldsymbol{X_c} \perp\!\!\!\perp \boldsymbol{A_{conf}} \mid E$ ✓

Step I: Deriving independence constraints

Theorem 3.1.

1. *Independent:* $\boldsymbol{X_c} \perp\!\!\!\perp \boldsymbol{A_{ind}}$; $\boldsymbol{X_c} \perp\!\!\!\perp E$; $\boldsymbol{X_c} \perp\!\!\!\perp \boldsymbol{A_{ind}} | Y$; $\boldsymbol{X_c} \perp\!\!\!\perp \boldsymbol{A_{ind}} | E$; $\boldsymbol{X_c} \perp\!\!\!\perp \boldsymbol{A_{ind}} | Y, E$
2. *Causal:* $\boldsymbol{X_c} \perp\!\!\!\perp \boldsymbol{A_{cause}} | Y$; $\boldsymbol{X_c} \perp\!\!\!\perp E$; $\boldsymbol{X_c} \perp\!\!\!\perp \boldsymbol{A_{cause}} | Y, E$
3. *Confounded:* $\boldsymbol{X_c} \perp\!\!\!\perp \boldsymbol{A_{conf}}$; $\boldsymbol{X_c} \perp\!\!\!\perp E$; $\boldsymbol{X_c} \perp\!\!\!\perp \boldsymbol{A_{conf}} | E$
4. *Selected:* $\boldsymbol{X_c} \perp\!\!\!\perp \boldsymbol{A_{sel}} | Y$; $\boldsymbol{X_c} \perp\!\!\!\perp \boldsymbol{A_{sel}} | Y, E$

Step I: Deriving independence constraints

Theorem 3.1.

1. *Independent:* $\boldsymbol{X_c} \perp\!\!\!\perp \boldsymbol{A_{ind}}$; $\boldsymbol{X_c} \perp\!\!\!\perp E$; $\boldsymbol{X_c} \perp\!\!\!\perp \boldsymbol{A_{ind}} | Y$; $\boldsymbol{X_c} \perp\!\!\!\perp \boldsymbol{A_{ind}} | E$; $\boldsymbol{X_c} \perp\!\!\!\perp \boldsymbol{A_{ind}} | Y, E$
2. *Causal:* $\boldsymbol{X_c} \perp\!\!\!\perp \boldsymbol{A_{cause}} | Y$; $\boldsymbol{X_c} \perp\!\!\!\perp E$; $\boldsymbol{X_c} \perp\!\!\!\perp \boldsymbol{A_{cause}} | Y, E$
3. *Confounded:* $\boldsymbol{X_c} \perp\!\!\!\perp \boldsymbol{A_{conf}}$; $\boldsymbol{X_c} \perp\!\!\!\perp E$; $\boldsymbol{X_c} \perp\!\!\!\perp \boldsymbol{A_{conf}} | E$
4. *Selected:* $\boldsymbol{X_c} \perp\!\!\!\perp \boldsymbol{A_{sel}} | Y$; $\boldsymbol{X_c} \perp\!\!\!\perp \boldsymbol{A_{sel}} | Y, E$

No (conditional) independence constraint valid for all shifts

Theoretical evidence for past work: *A fixed conditional independence constraint cannot work for all datasets*

Ye et al., OoD-Bench, CVPR 2022; Wiles et al., ICLR 2022

Theoretical evidence for previous results: *A fixed conditional independence constraint cannot work for all datasets*

Theorem 3.2. For any predictor algorithm for Y that uses a single type of (conditional) independence constraint, there exists a realized graph $\mathcal{G}$ and a corresponding training dataset such that the learned predictor cannot be a risk-invariant predictor across distributions in $\mathcal{P}_{\mathcal{G}}$.

Ye et al., OoD-Bench, CVPR 2022; Wiles et al., ICLR 2022

Step II: Applying regularization penalty

Constraint: $\boldsymbol{X_c} \perp\!\!\!\perp \boldsymbol{A_{cause}} \mid Y, E$ [***Causal*** shift]

$$RegPenalty_{\boldsymbol{A}_{cause}} = \sum_{|E|} \sum_{y \in Y} \sum_{i=1}^{|\boldsymbol{A_{cause}}|} \sum_{j>i} \mathrm{MMD}\left(P\left(g_1(\phi(\boldsymbol{x}))\middle|a_{i,cause}, y\right), P\left(g_1(\phi(\boldsymbol{x}))\middle|a_{j,cause}, y\right)\right)$$

$$\boldsymbol{g_1}, \boldsymbol{\phi} = \mathrm{argmin}_{g_1,\phi}\ \mathrm{L}\left(g_1\left(\phi(\boldsymbol{x})\right), y\right) + \lambda^*(RegPenalty_{\boldsymbol{A}_{cause}})$$

Finally, *CACM* Algorithm for general graphs

Phase I: Derive correct independence constraints

1. For every observed variable $A \in \mathcal{A}$ in the graph, check whether $(\boldsymbol{X_c}, A)$ are d-separated.
   => $\boldsymbol{X_c} \perp\!\!\!\perp A$ is a valid constraint

2. If not, check whether $(\boldsymbol{X_c}, A)$ are d-separated conditioned on any subset $\boldsymbol{A_s}$ of the remaining observed variables in $\mathcal{A} \setminus \{A\}$.
   => $\boldsymbol{X_c} \perp\!\!\!\perp A \,|\boldsymbol{A_s}$ is a valid constraint

Finally, *CACM* Algorithm for general graphs

Phase II: Apply regularization penalty using constraints derived

If $\boldsymbol{X_c} \perp\!\!\!\perp A$

$$RegPenalty_A = \sum_{|E|} \sum_{i=1}^{|A|} \sum_{j>i} \text{MMD}\left(P(g_1(\phi(\boldsymbol{x}))|A_i), P\big(g_1(\phi(\boldsymbol{x}))\big|A_j\big)\right)$$

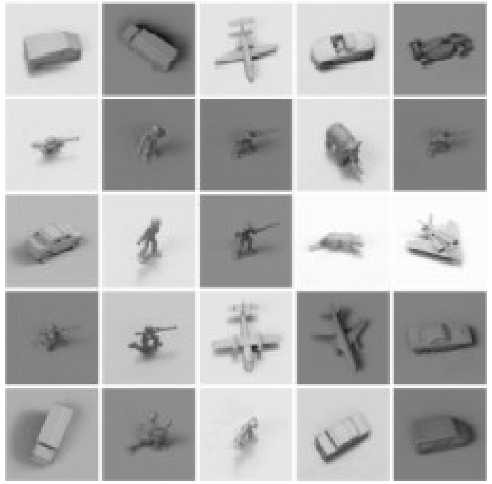
If $\boldsymbol{X_c} \perp\!\!\!\perp A \,|\boldsymbol{A_s}$

$$RegPenalty_A = \sum_{|E|} \sum_{a \in \boldsymbol{A_s}} \sum_{i=1}^{|A|} \sum_{j>i} \text{MMD}\left(P(g_1(\phi(\boldsymbol{x}))|A_i, a), P\big(g_1(\phi(\boldsymbol{x}))\big|A_j, a\big)\right)$$

$$\boldsymbol{RegPenalty} = \sum_{A \in \boldsymbol{A}} Penalty_A$$

$$\boldsymbol{g_1}, \boldsymbol{\phi} = \text{argmin}_{g_1, \phi}\ \text{L}\big(g_1\big(\phi(\boldsymbol{x})\big), y\big) + \lambda^*(RegPenalty)$$

Empirical evaluation



Spurious correlation
b/w category and lighting
(A_{cause})

+

Unseen data shift
unseen azimuth values
(A_{ind})

small NORB dataset

- Multi-class (5 classes)
- Multi-valued attributes
- Real objects

Wiles et al., ICLR 2022

Correct constraint derived from CG matters

Algorithm	lighting A_{cause}	azimuth A_{ind}	lighting+azimuth $A_{cause} \cup A_{ind}$
ERM	65.5 ± 0.7	**78.6 ± 0.7**	**64.0 ± 1.2**
IRM	**66.7 ± 1.5**	75.7 ± 0.4	61.7 ± 1.5
VREx	64.7 ± 1.0	77.6 ± 0.5	62.5 ± 1.6
MMD	66.6 ± 1.6	76.7 ± 1.1	62.5 ± 0.3
CORAL	64.7 ± 1.5	77.2 ± 0.7	62.9 ± 0.3
DANN	64.6 ± 1.4	**78.6 ± 0.7**	60.8 ± 0.7
C-MMD	65.8 ± 0.8	76.9 ± 1.0	61.0 ± 0.9
CDANN	64.9 ± 0.5	77.3 ± 0.3	60.8 ± 0.9

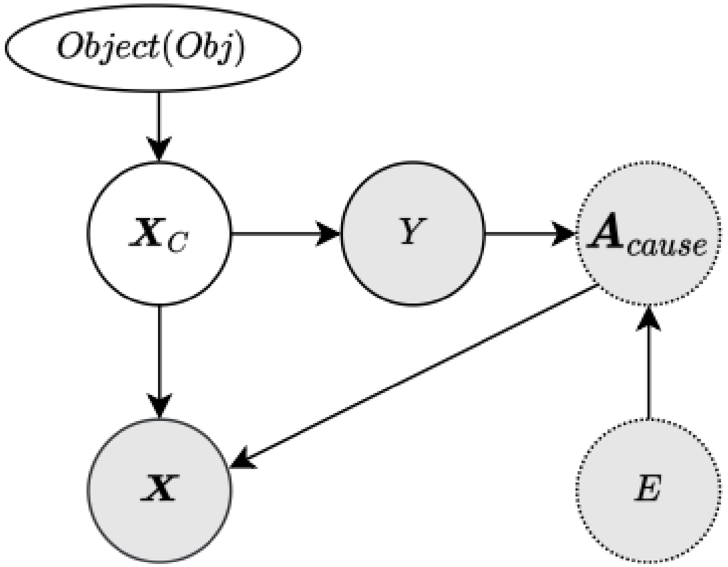
ERM outperforms all DG algorithms!

Correct constraint derived from CG matters

Algorithm	lighting A_{cause}	azimuth A_{ind}	lighting+azimuth $A_{cause} \cup A_{ind}$
ERM	65.5 ± 0.7	78.6 ± 0.7	64.0 ± 1.2
IRM	66.7 ± 1.5	75.7 ± 0.4	61.7 ± 1.5
VREx	64.7 ± 1.0	77.6 ± 0.5	62.5 ± 1.6
MMD	66.6 ± 1.6	76.7 ± 1.1	62.5 ± 0.3
CORAL	64.7 ± 1.5	77.2 ± 0.7	62.9 ± 0.3
DANN	64.6 ± 1.4	78.6 ± 0.7	60.8 ± 0.7
C-MMD	65.8 ± 0.8	76.9 ± 1.0	61.0 ± 0.9
CDANN	64.9 ± 0.5	77.3 ± 0.3	60.8 ± 0.9
CACM	**85.4 ± 0.5**	**80.5 ± 0.6**	**69.6 ± 1.6**

CACM provides upto 20% improvement

Incorrect constraints hurt generalization!

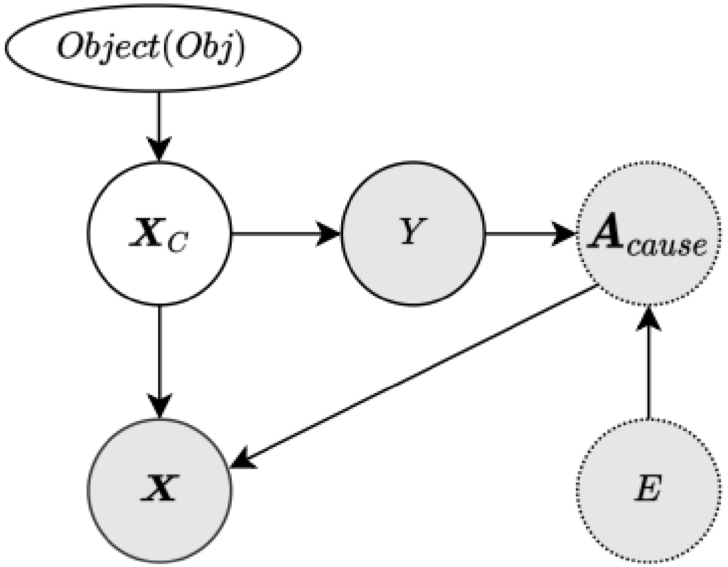


Causal

$$X_c \perp\!\!\!\perp A_{cause} \mid E$$

$$X_c \perp\!\!\!\perp A_{cause} \mid Y, E$$

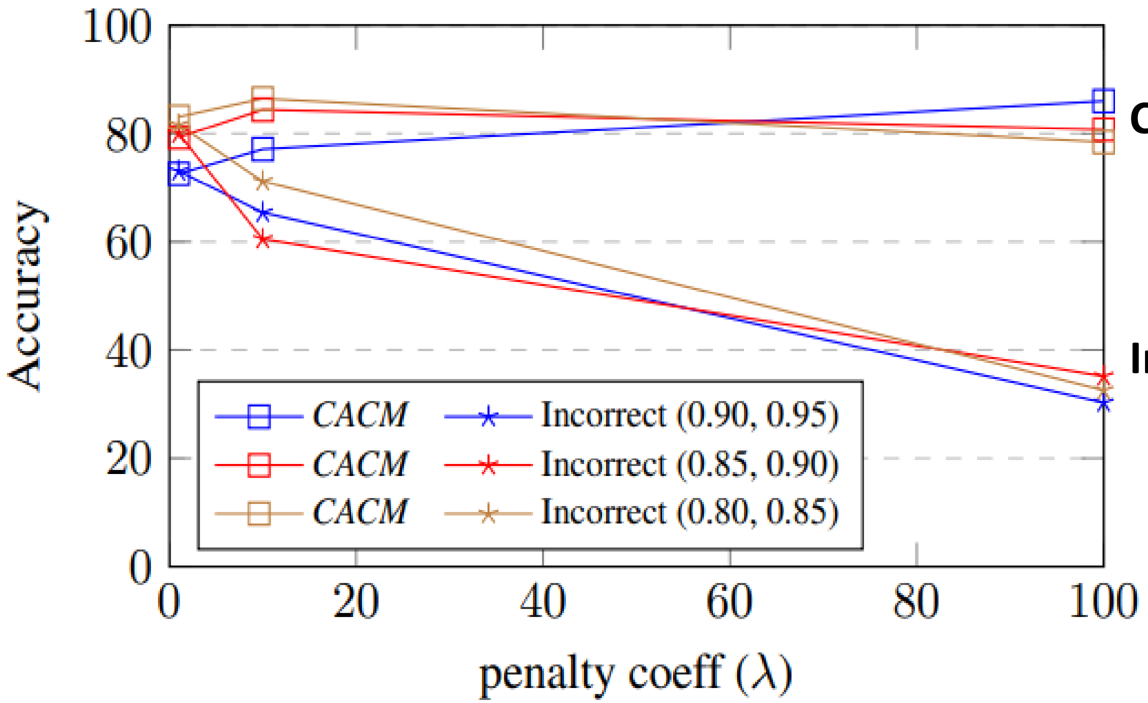
Incorrect constraints hurt generalization!



Causal

$X_c \perp\!\!\!\perp A_{cause} \mid E$ ✗

$X_c \perp\!\!\!\perp A_{cause} \mid Y, E$ ✓



Correct Constraint

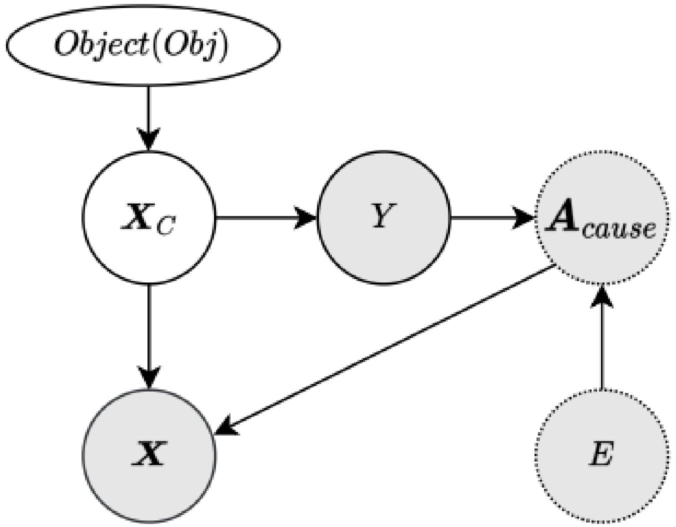
$X_c \perp\!\!\!\perp A_{cause} \mid Y, E$

Incorrect Constraint

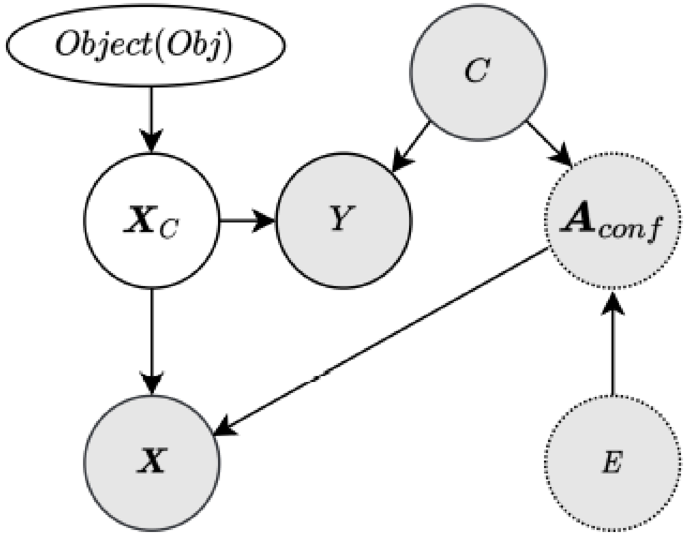
$X_c \perp\!\!\!\perp A_{cause} \mid E$

OOD Accuracy of incorrect constraint decreases as regularization penalty is increased

Incorrect constraints hurt generalization!

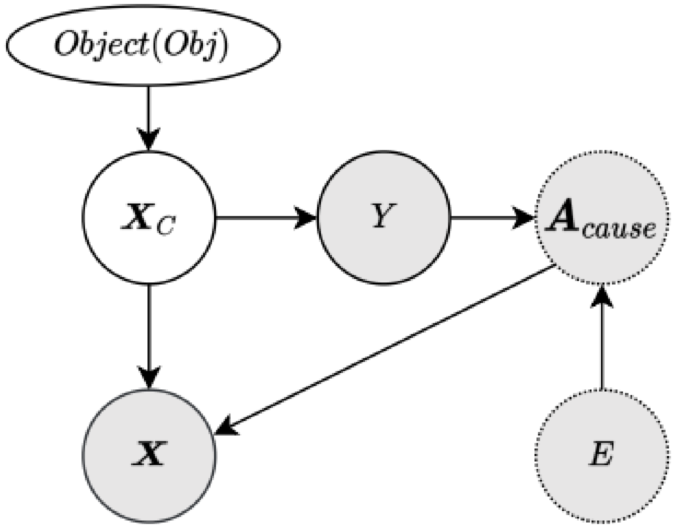


Causal

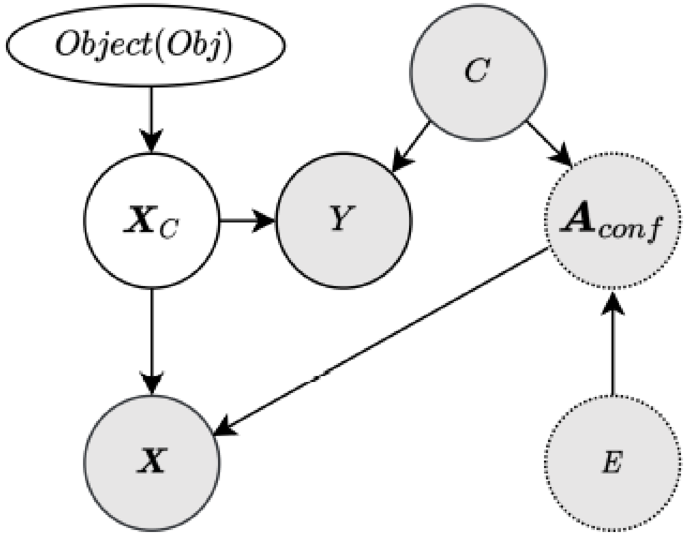


Confounded

Incorrect constraints hurt generalization!



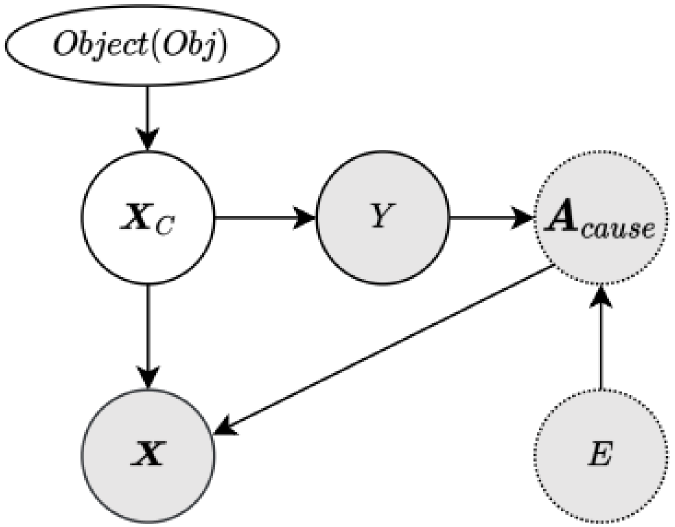
Causal

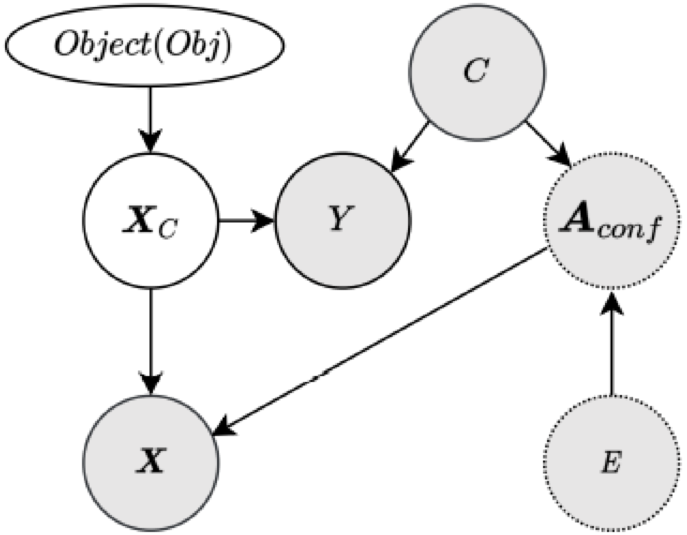


Confounded

Constraint	*Causal*	*Confounded*
$\boldsymbol{X}_c \perp\!\!\!\perp \boldsymbol{A} \mid E$	29.7 ± 3.8	**62.4 ± 1.9**
$\boldsymbol{X}_c \perp\!\!\!\perp \boldsymbol{A} \mid Y, E$	**94.1 ± 0.5**	56.0 ± 1.0

Incorrect constraints hurt generalization!





$X_c \perp\!\!\!\perp A_{cause} \mid E$ ✗

$X_c \perp\!\!\!\perp A_{cause} \mid Y, E$ ✓

$X_c \perp\!\!\!\perp A_{conf} \mid E$ ✓

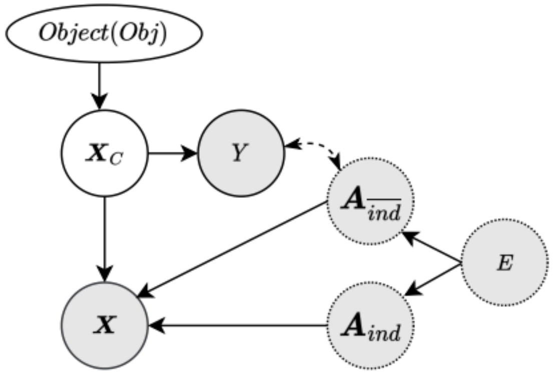
$X_c \perp\!\!\!\perp A_{conf} \mid Y, E$ ✗

Constraint	*Causal*	*Confounded*
$X_c \perp\!\!\!\perp A \mid E$	29.7 ± 3.8	**62.4 ± 1.9**
$X_c \perp\!\!\!\perp A \mid Y, E$	**94.1 ± 0.5**	56.0 ± 1.0

Takeaways

- Necessary to model causal relationships in the data-generating process for OOD generalization
  - Algorithms based on single, fixed constraint fail to generalize
- Do not need full causal graph
  - Only the attributes and their relationship with outcome variable
- Algorithm with causally adaptive constraints outperforms existing OOD algorithms
  - Works equally well on single dataset, datasets with multiple domains, etc.

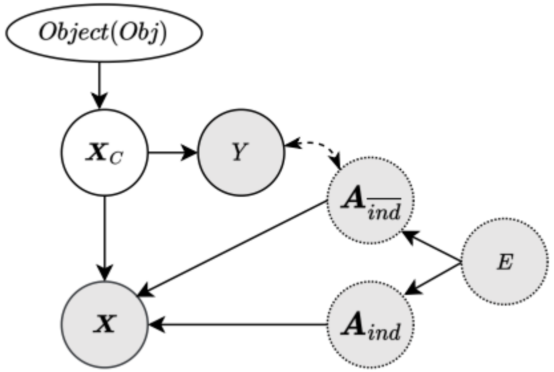
Beyond CACM: Counterfactual data augmentation



- Generate synthetic data with different attributes that breaks the correlation
- What if we change only the spurious attribute while keeping the rest of input identical?
- Theoretically consistent with recovering X_c
- In practice, use GANs/Adversarially learnt inference to build generative model

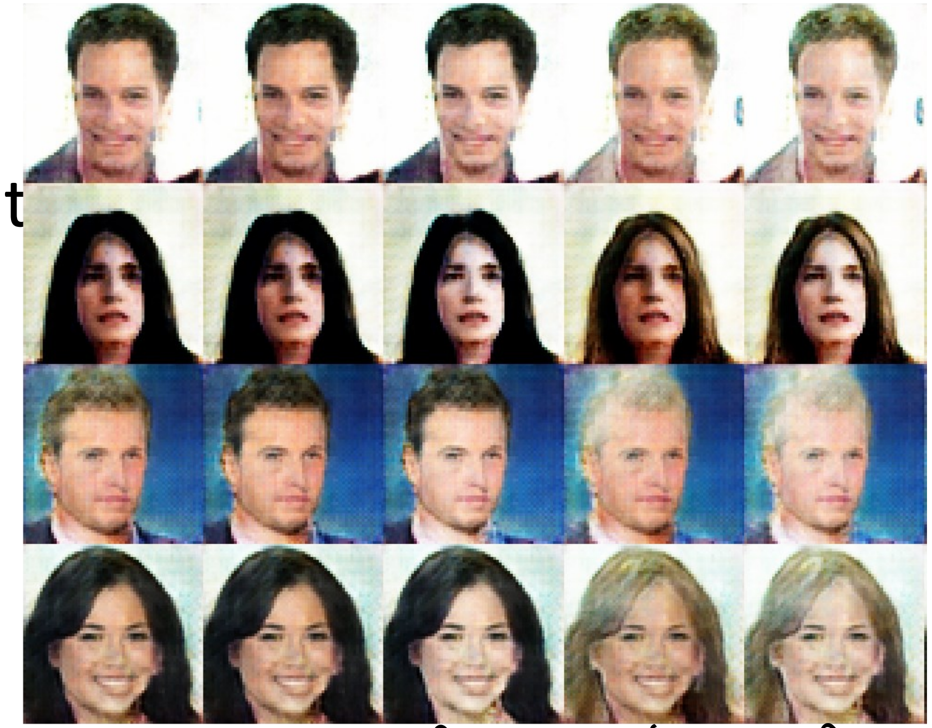
ICML 2021. *Domain generalization using causal matching. Mahajan, Tople, Sharma.*
WACV 2022. *Evaluating and Mitigating Bias in Image Classifiers: A Causal Perspective Using Counterfactuals.* Dash, Balasubramanian, Sharma.

Beyond CACM: Counterfactual data augmentation



- Generate synthetic data with different attributes that breaks the correlation
- What if we change only the spurious attribute while keeping the rest of input identical?
- Theoretically consistent with recovering X_c
- In practice, use GANs/Adversarially learnt inference to build generative model

$$\boldsymbol{g_1}, \boldsymbol{\phi} = \operatorname{argmin}_{g_1,\phi}\ \mathrm{L}(g_1(\phi(\boldsymbol{x})), y) + \lambda^* \sum_{x,x'} (\phi(\boldsymbol{x}) - \phi(\boldsymbol{x}')^2)$$



ICML 2021. *Domain generalization using causal matching. Mahajan, Tople, Sharma.*

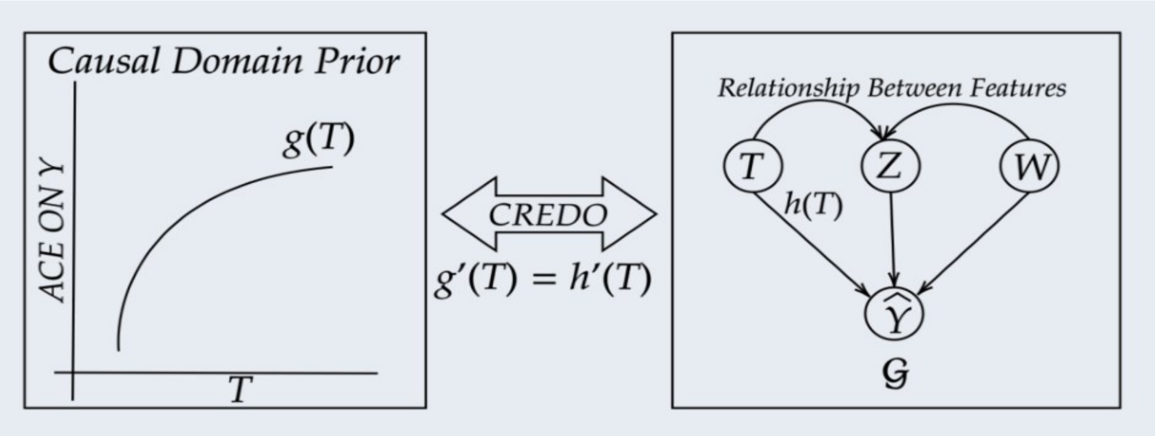
WACV 2022. *Evaluating and Mitigating Bias in Image Classifiers: A Causal Perspective Using Counterfactuals.* Dash, Balasubramanian, Sharma.

Beyond CACM: Using causal domain knowledge

In addition to structure, people may know the *shape* of causal effect function (causal prior).

Shape: *diminishing return, U-shaped, Z-shaped, etc.*

Type: *direct causal effect, indirect effect, total effect*



ICML 2022. *Matching learned causal effect of neural networks using domain priors. Kancheti, Abbavaram, Balasubramanian, Sharma.*

Beyond CACM: Using causal domain knowledge

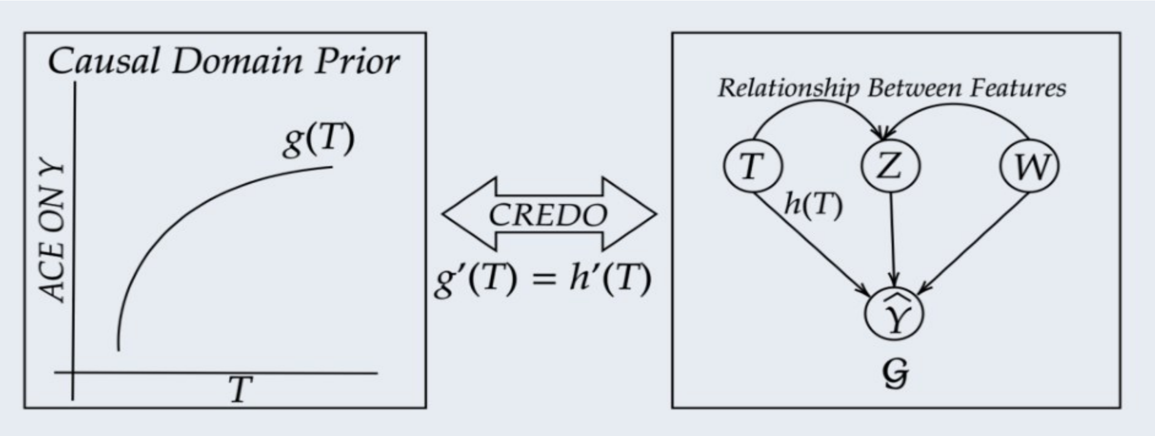
In addition to structure, people may know the *shape* of causal effect function (causal prior).

Shape: *diminishing return, U-shaped, Z-shaped, etc.*

Type: *direct causal effect, indirect effect, total effect*

Can enforce it by,

1. Measuring causal effect of a feature on the model's prediction
2. Matching the model's gradient to provided causal prior's gradient



ICML 2022. *Matching learned causal effect of neural networks using domain priors. Kancheti, Abbavaram, Balasubramanian, Sharma.*

PART II: Practical causal inference with DoWhy

- DoWhy Library: https://github.com/py-why/dowhy
- Arxiv paper on the four steps of causal inference: https://arxiv.org/abs/2011.04216

From prediction to decision-making

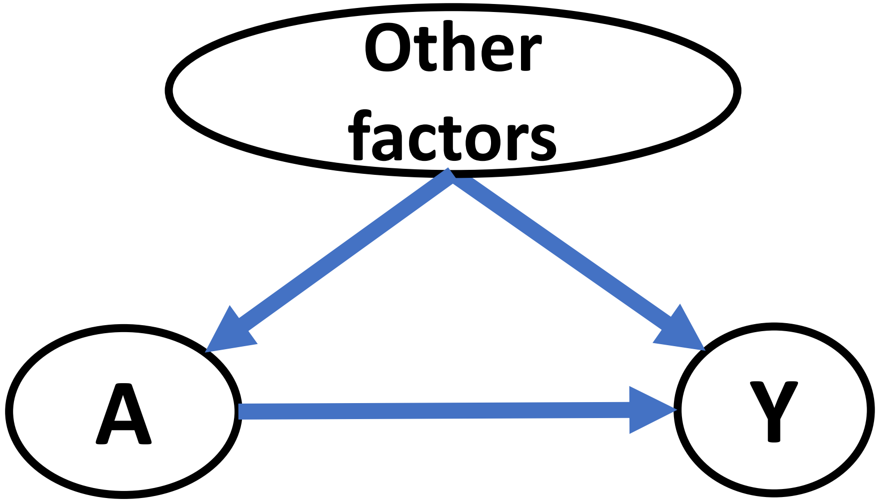
Decision-making: Acting/intervening on a feature

- Interventions break correlations used by supervised ML
    - Special kind of OOD generalization
- The feature with the highest importance score in a prediction model,
    - Need not be the best feature to act on
    - May not even affect the outcome at all!

For decision-making, need to find the features that **cause the outcome** & *estimate how the outcome would change if the features are changed.*

Observed distribution

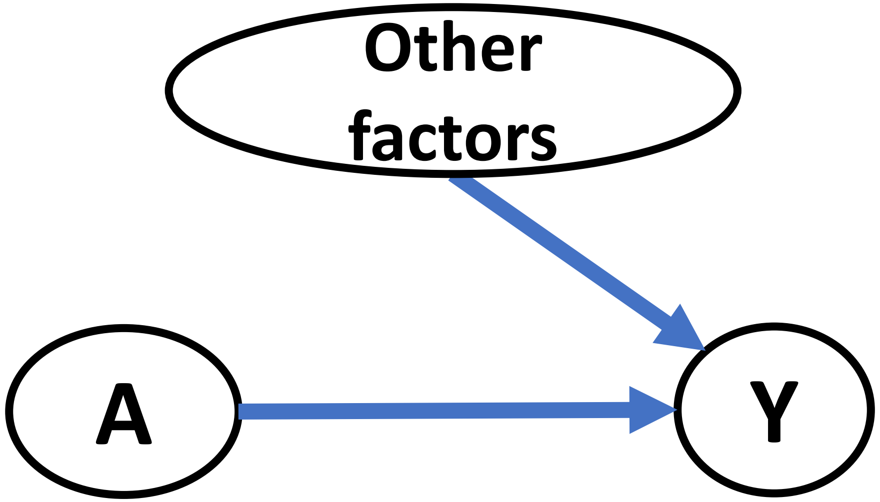
$P(Y|A)$



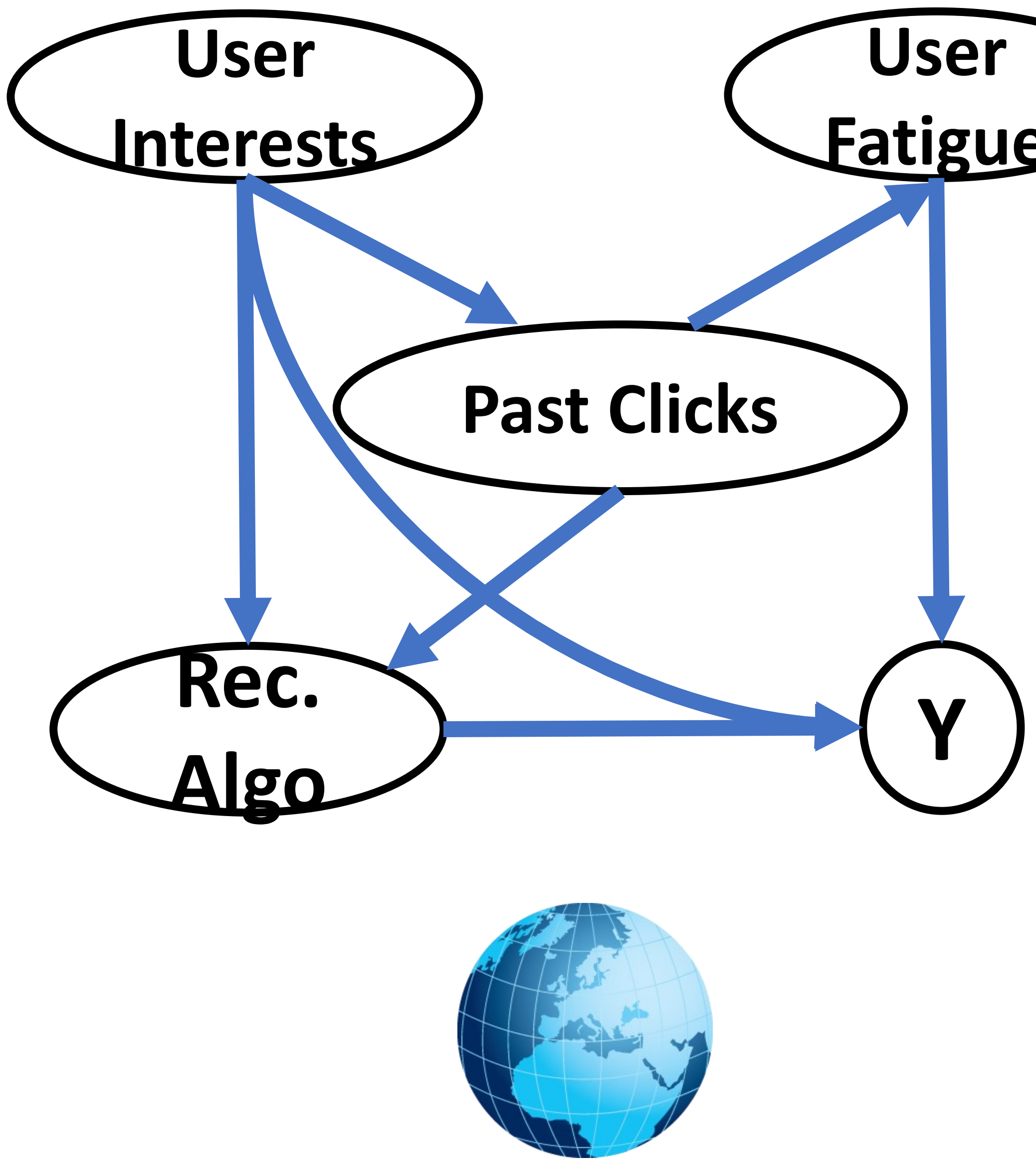
Real World

Interventional Distribution

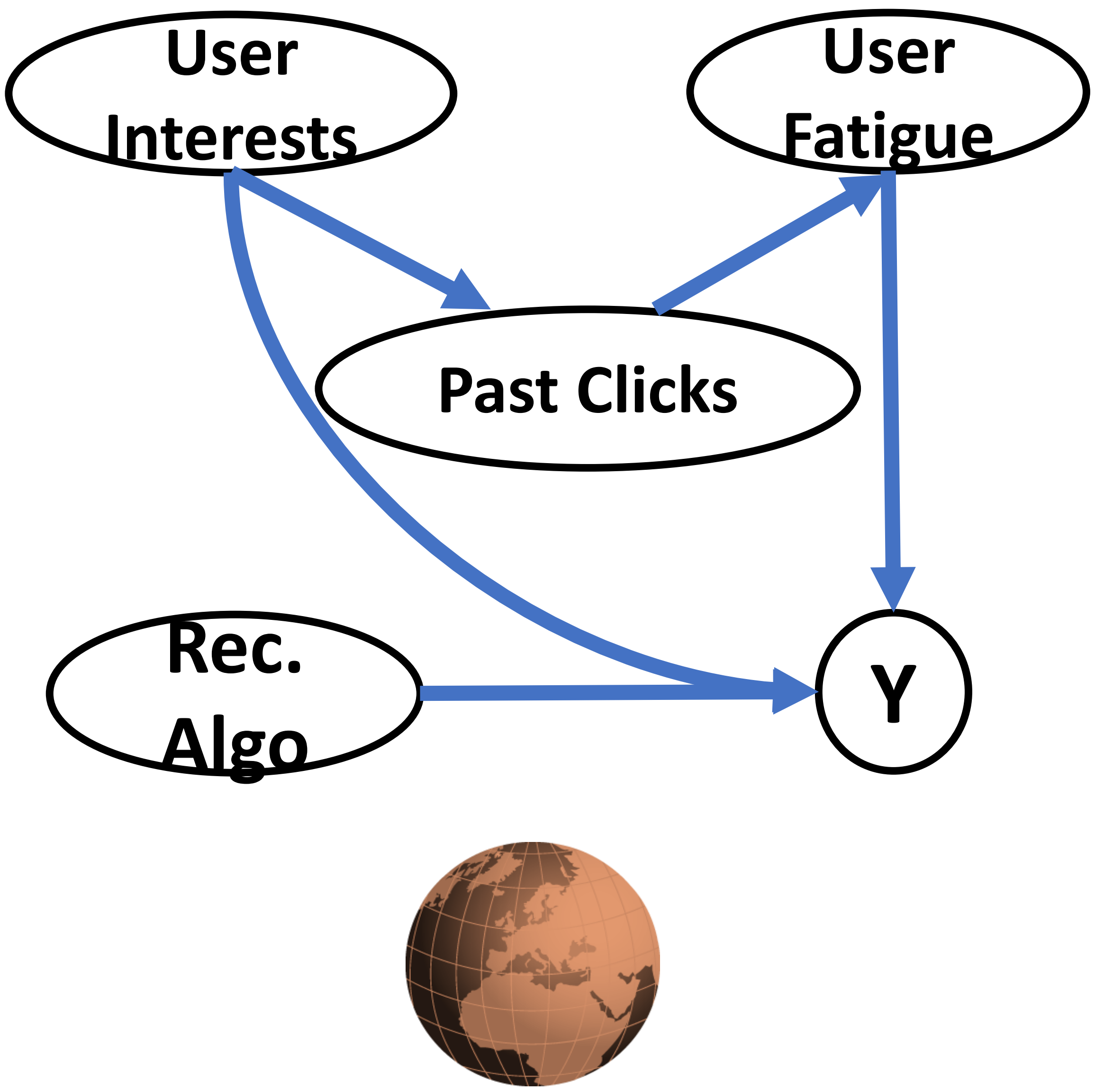
$P(Y|do(A = 1))$



Counterfactual World



Real World



Counterfactual World

Two Fundamental Challenges for Causal Inference

Multiple causal graphs can fit the same data distribution. **Do we have the right graph?**

→ **1. Assumptions**

Target distribution is unobserved. **No easy "cross-validation".**

→ **2. Evaluation**

We built DoWhy library to make assumptions front-and-center of any causal analysis.

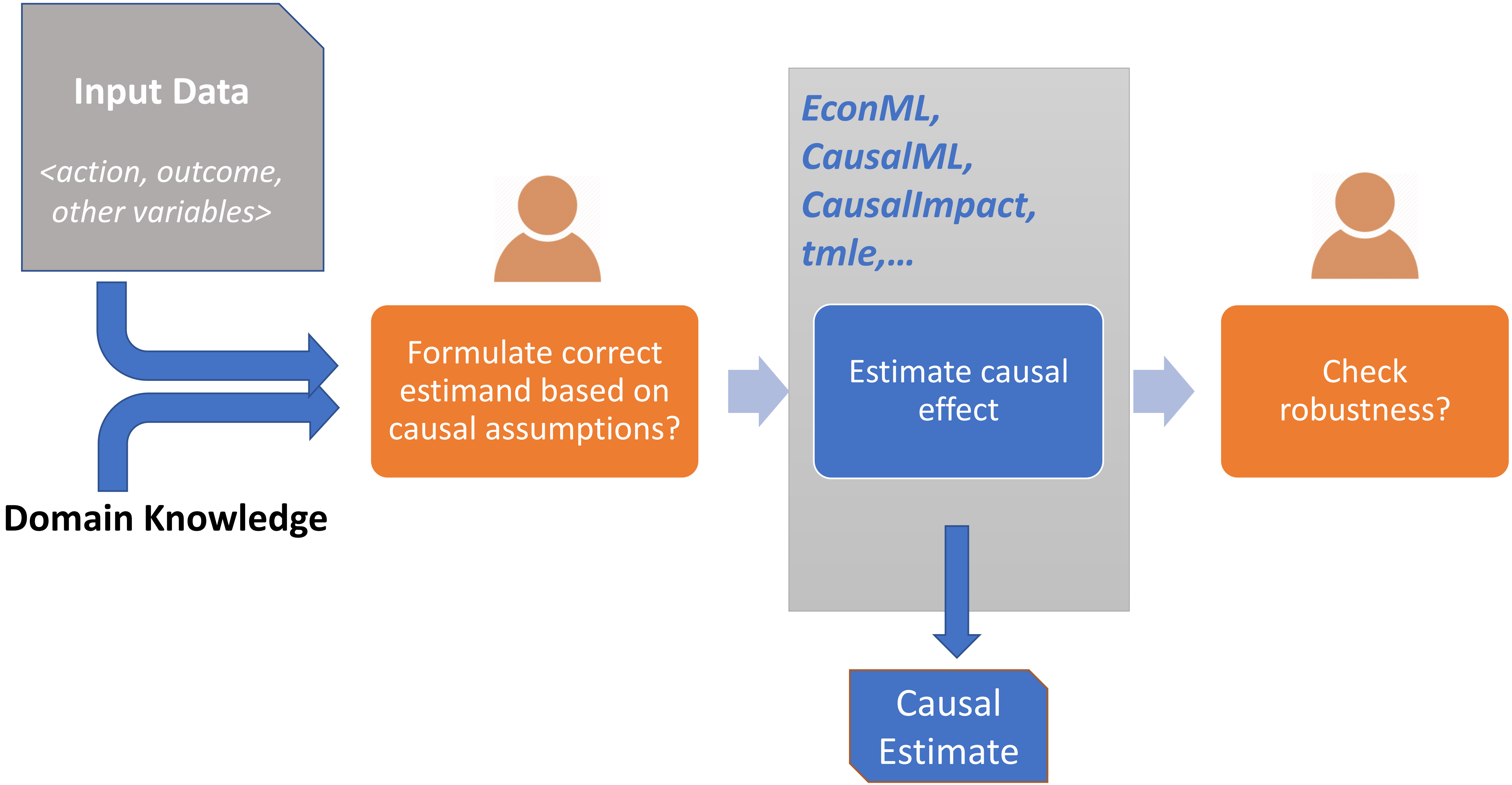
- Transparent declaration of assumptions
- Evaluation of those assumptions, to the extent possible

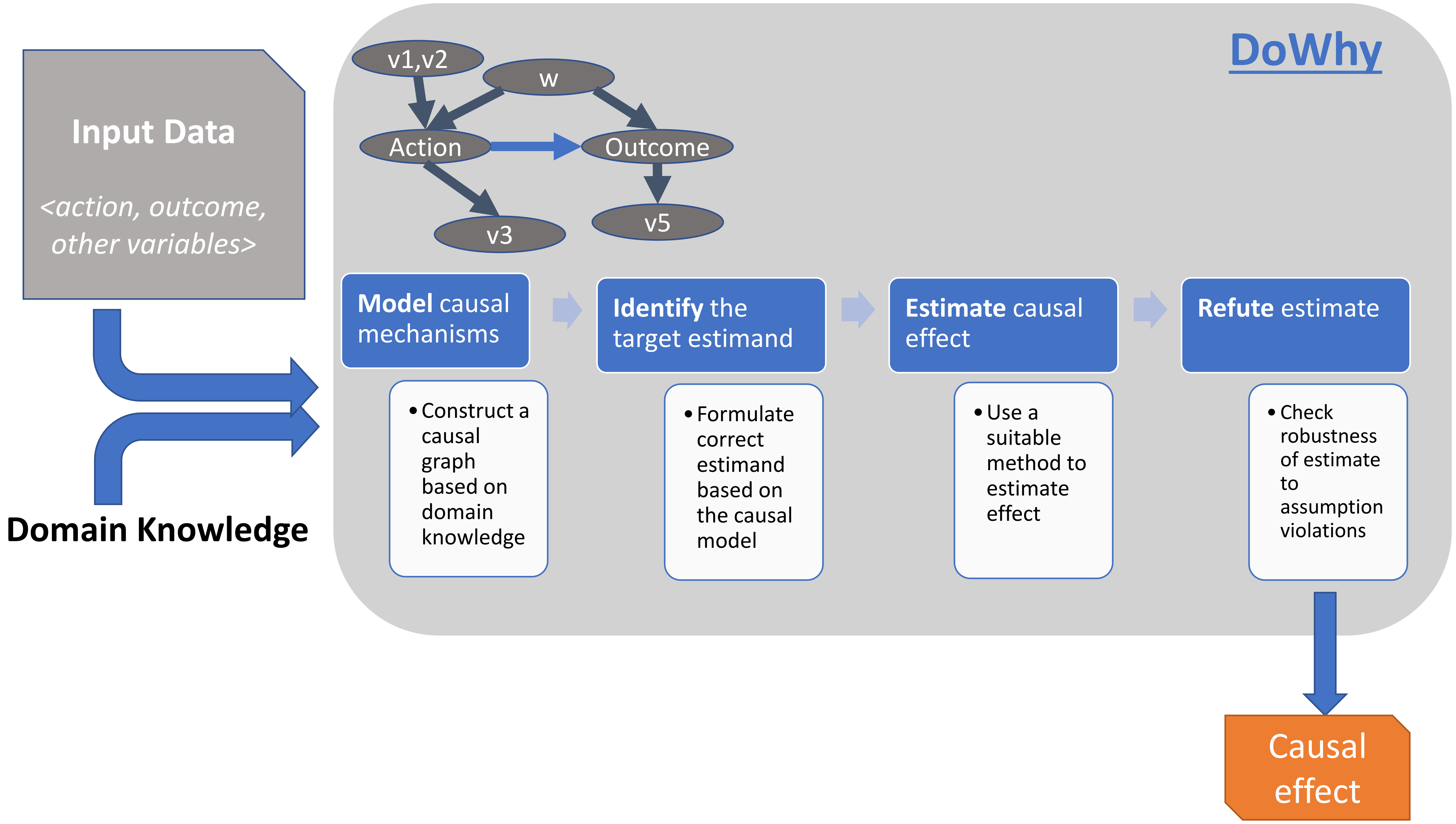
One of the most popular causal libraries on GitHub
(>1.3M downloads, 5K stars, 690+ forks)

Taught in third-party tutorials and courses: O'Reilly, PyData, Northeastern, …
Used by many companies and researchers.

Maintained by independent org py-why with >50 contributors

An end-to-end platform for doing causal inference


Input Data
<action, outcome, other variables>
Domain Knowledge
Formulate correct estimand based on causal assumptions?
EconML, CausalML, CausalImpact, tmle,...
Estimate causal effect
Causal Estimate
Check robustness?


DoWhy
Input Data
<action, outcome, other variables>
Domain Knowledge
v1,v2
w
Action
Outcome
v3
v5
Model causal mechanisms
Identify the target estimand
Estimate causal effect
Refute estimate
Construct a causal graph based on domain knowledge
Formulate correct estimand based on the causal model
Use a suitable method to estimate effect
Check robustness of estimate to assumption violations
Causal effect

DoWhy provides a general API for the four steps of causal inference

1. **Modeling:** Create a causal graph to encode assumptions.
2. **Identification:** Formulate what to estimate.
3. **Estimation:** Compute the estimate.
4. **Refutation:** Validate the assumptions.

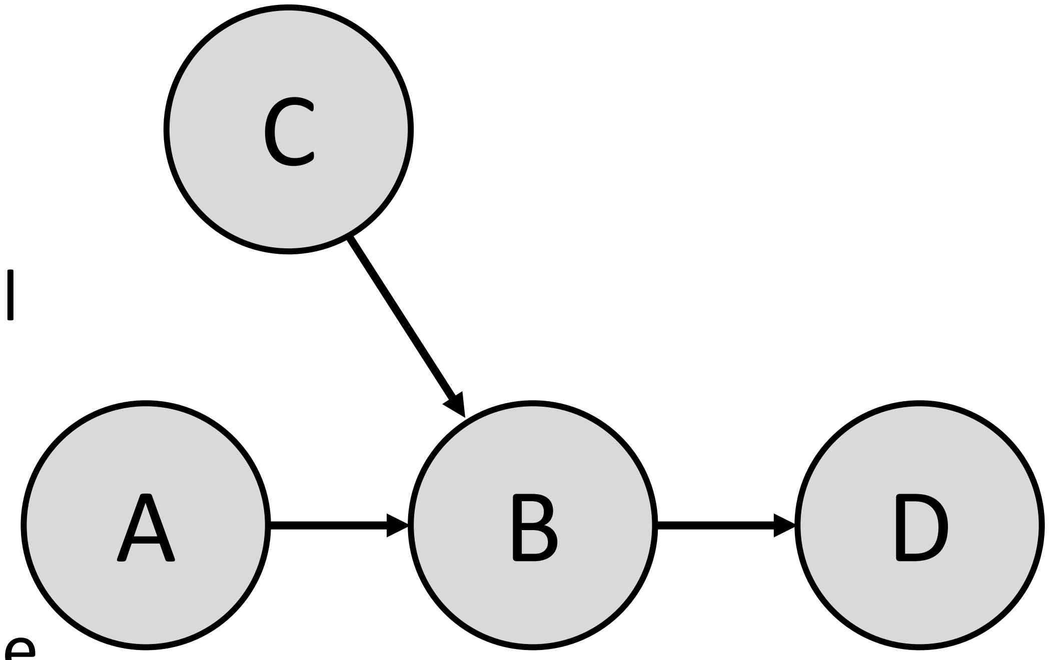
We'll discuss the four steps and show a code example using DoWhy.

I. Model the assumptions using a causal graph

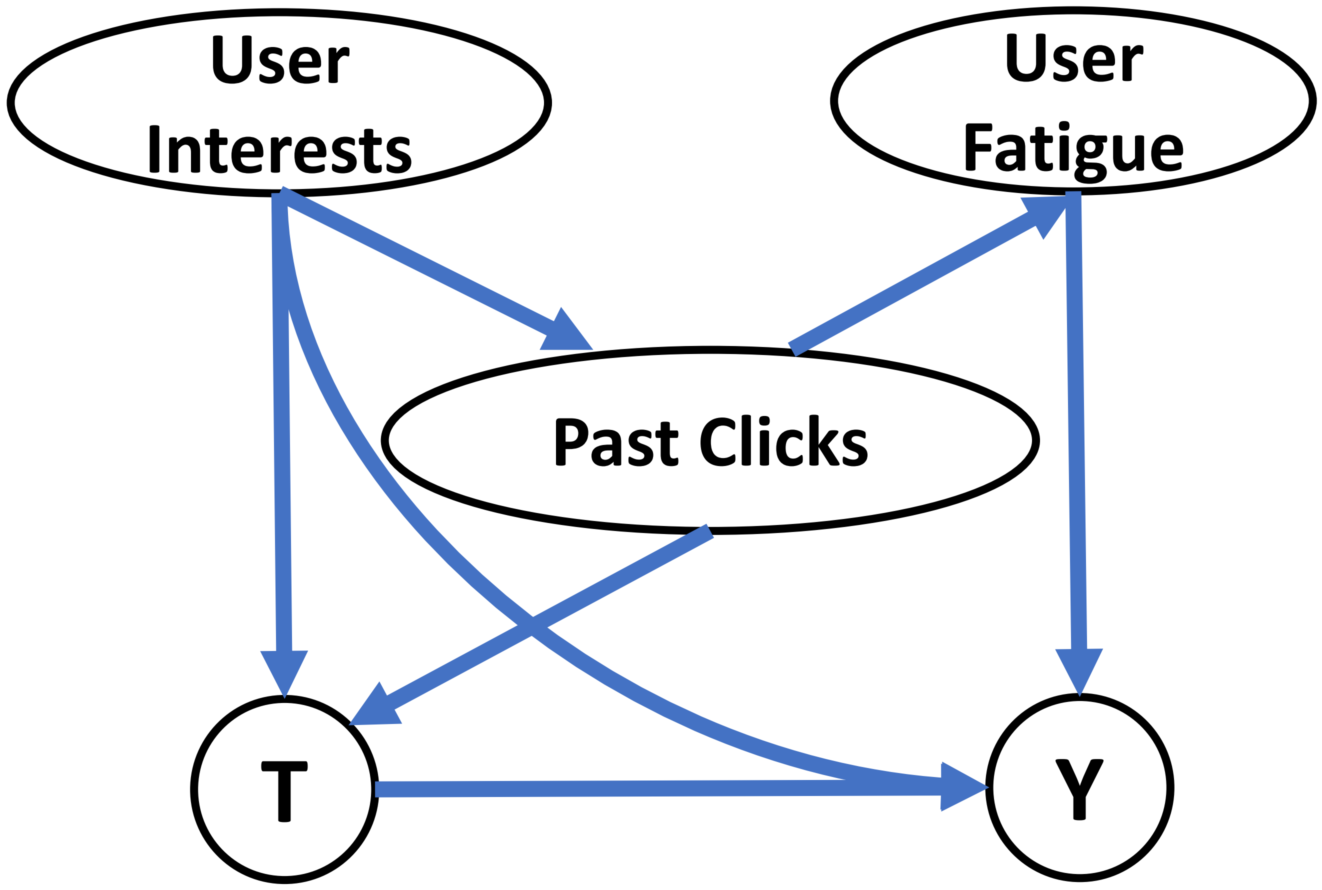
Convert domain knowledge to a formal model of causal assumptions

- $A \rightarrow B$ or $B \rightarrow A$?

- Causal graph implies conditional statistical independences
  - E.g., $A \perp\!\!\!\perp C,\ D \perp\!\!\!\perp \mathrm{A} \mid \mathrm{B}, \ldots$
  - Identified by *d-separation* rules [Pearl 2009]
- These assumptions significantly impact the causal estimate we'll obtain.



Example Graph



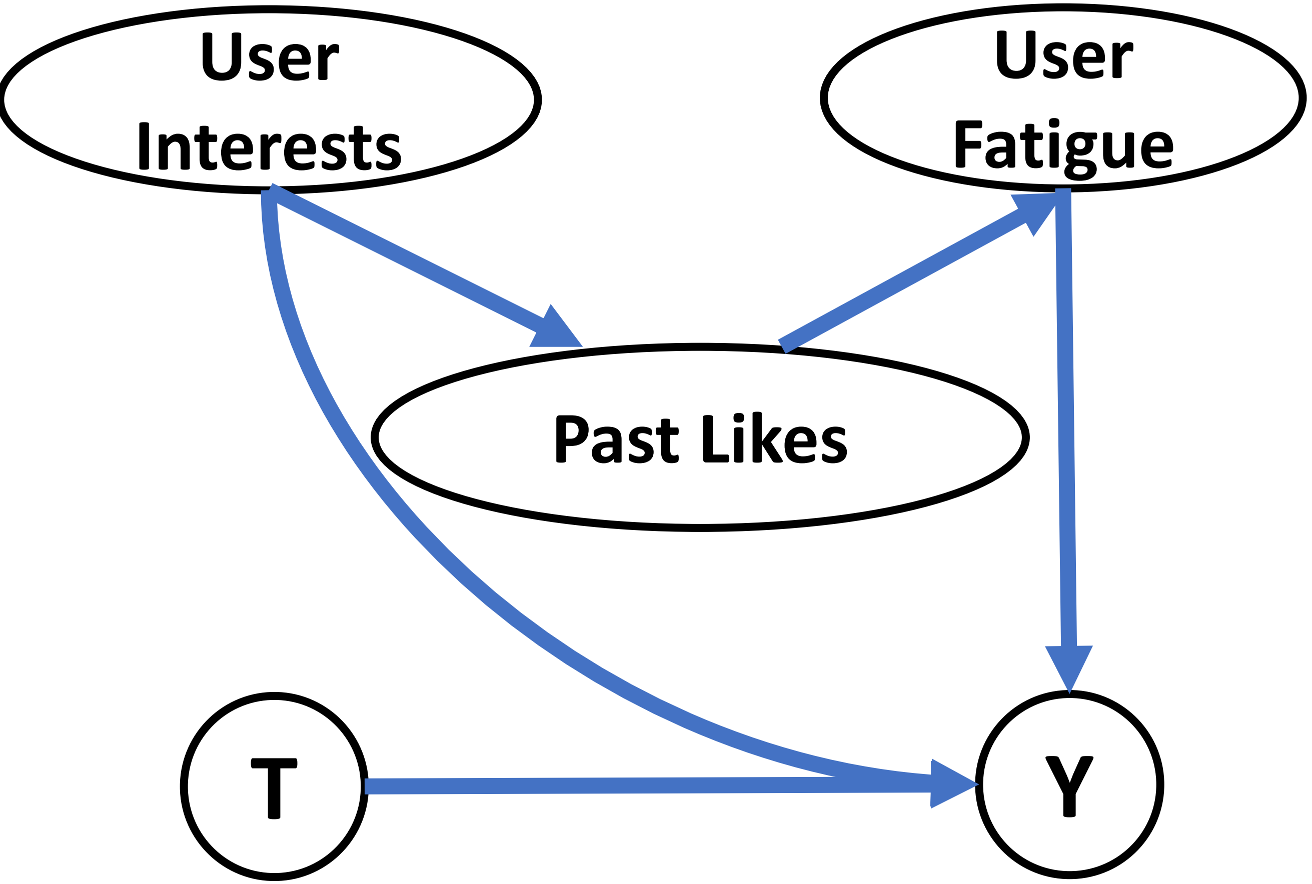
Assumption 1: User fatigue does not affect user interests

Assumption 2: Past clicks do not directly affect outcome

Assumption 3: Treatment does not affect user fatigue.

..and so on.

Intervention is represented by a new graph

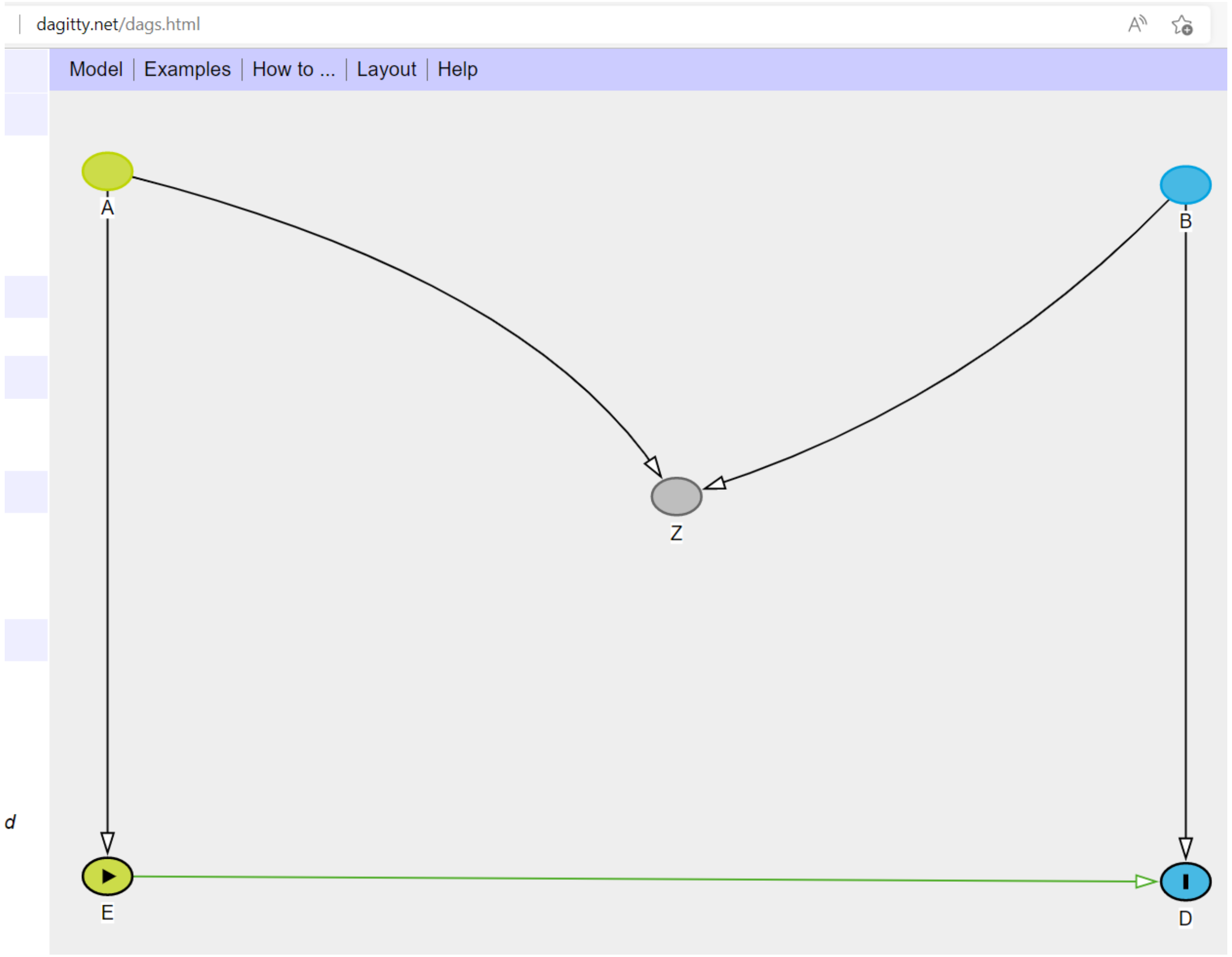


Interventional graph:

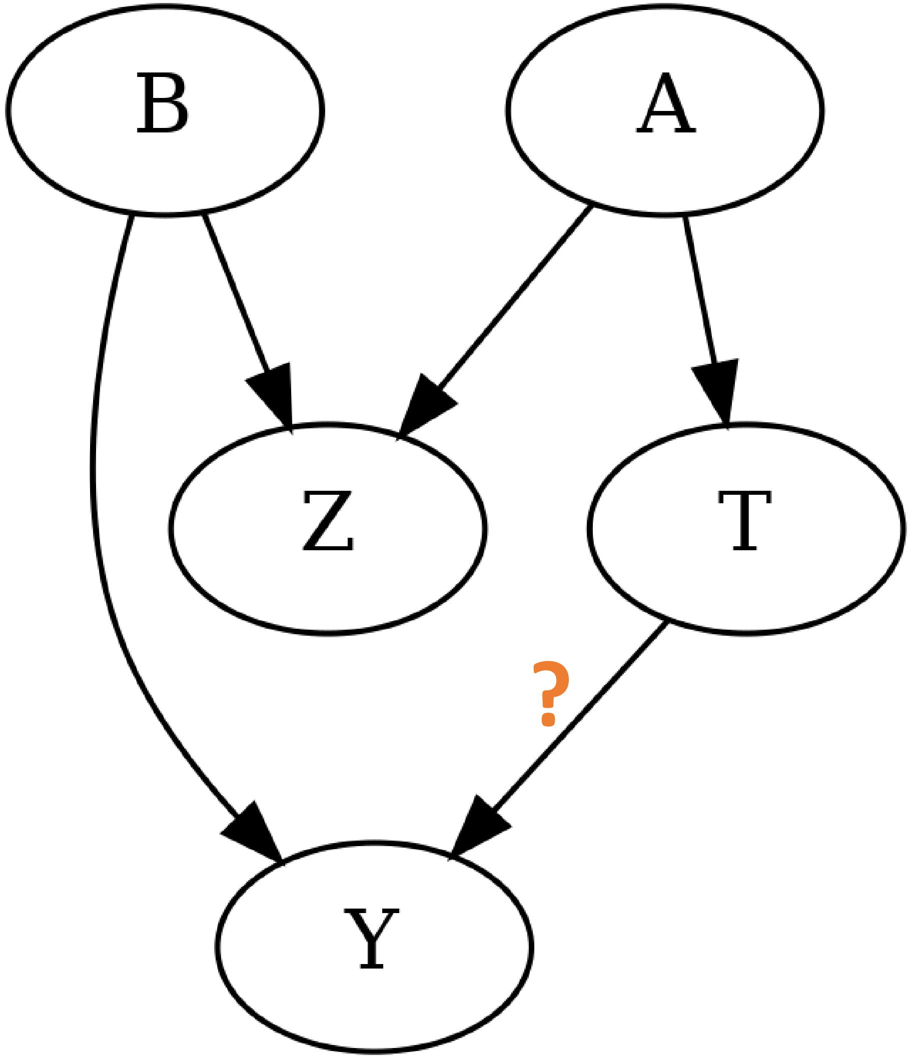
All edges to Treatment T removed, *keeping everything else the same*.

Represents new data distribution, referred as $do(T)$

Causal effect: $P(Y|do(T))$



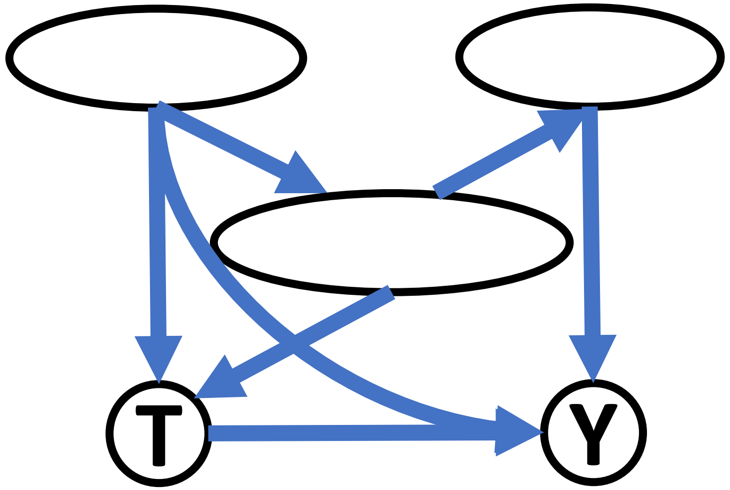
DAGitty.net



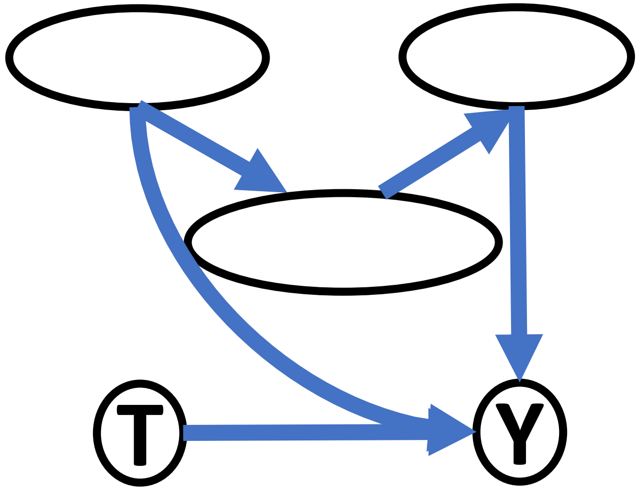
DoWhy

II. **Identification:** Formulate desired quantity and check if it is estimable from given data

Observed data generated by this graph



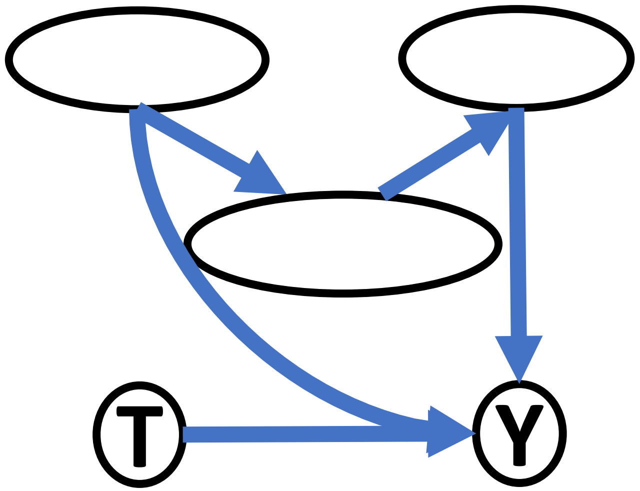
Want to answer questions about data that *will* be generated by intervention graph



How to represent quantities from right hand graph (e.g., $P(Y|do(T))$ using only statistical observations from data generated from left hand graph?

Randomized Experiments and Backdoor criterion

- Observed graph is same as intervention graph in randomized experiment!
  - Treatment T is already generated independent of all other features
  - ➔ $P(Y|do(T)) = P(Y|T)$
- **Backdoor Intuition:** Generalize by simulating randomized experiment
  - When treatment T is caused by other features, Z, adjust for their influence to simulate a randomized experiment



Backdoor Adjustment formula

$$p(Y|do(T)) = \sum_{Z} p(Y|T,Z)p(Z)$$

Many kinds of identification methods

Graphical constraint-based methods

- Randomized and natural experiments
- Adjustment Sets
  - Backdoor, “towards necessity”
- Front-door criterion
- Mediation formula

Identification under additional non-graphical constraints

- Instrumental variables
- Regression discontinuity
- Difference-in-differences

Many of these methods can be used through DoWhy.

III. **Estimation:** Compute the causal effect

Estimation uses observed data to compute the target probability expression from the Identification step.

For common identification strategies using adjustment sets,

$$E[Y|do(T=t), W=w] = E[Y|T=t, W=w]$$

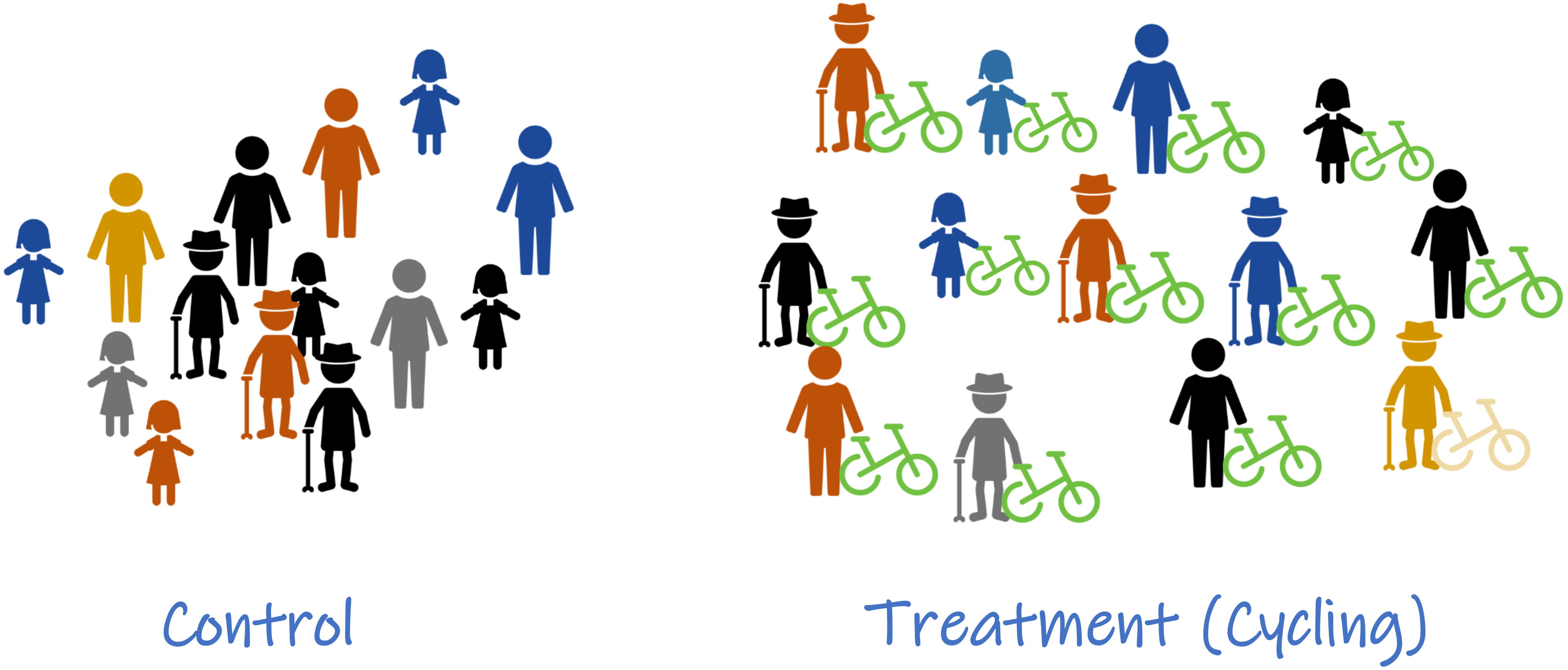
assuming W is a valid adjustment set.

- For binary treatment,

Causal Effect = $E[Y|T=1, W=w] - E[Y|T=0, W=w]$

Goal: Estimating conditional probability Y|T=t when all confounders W are kept constant.

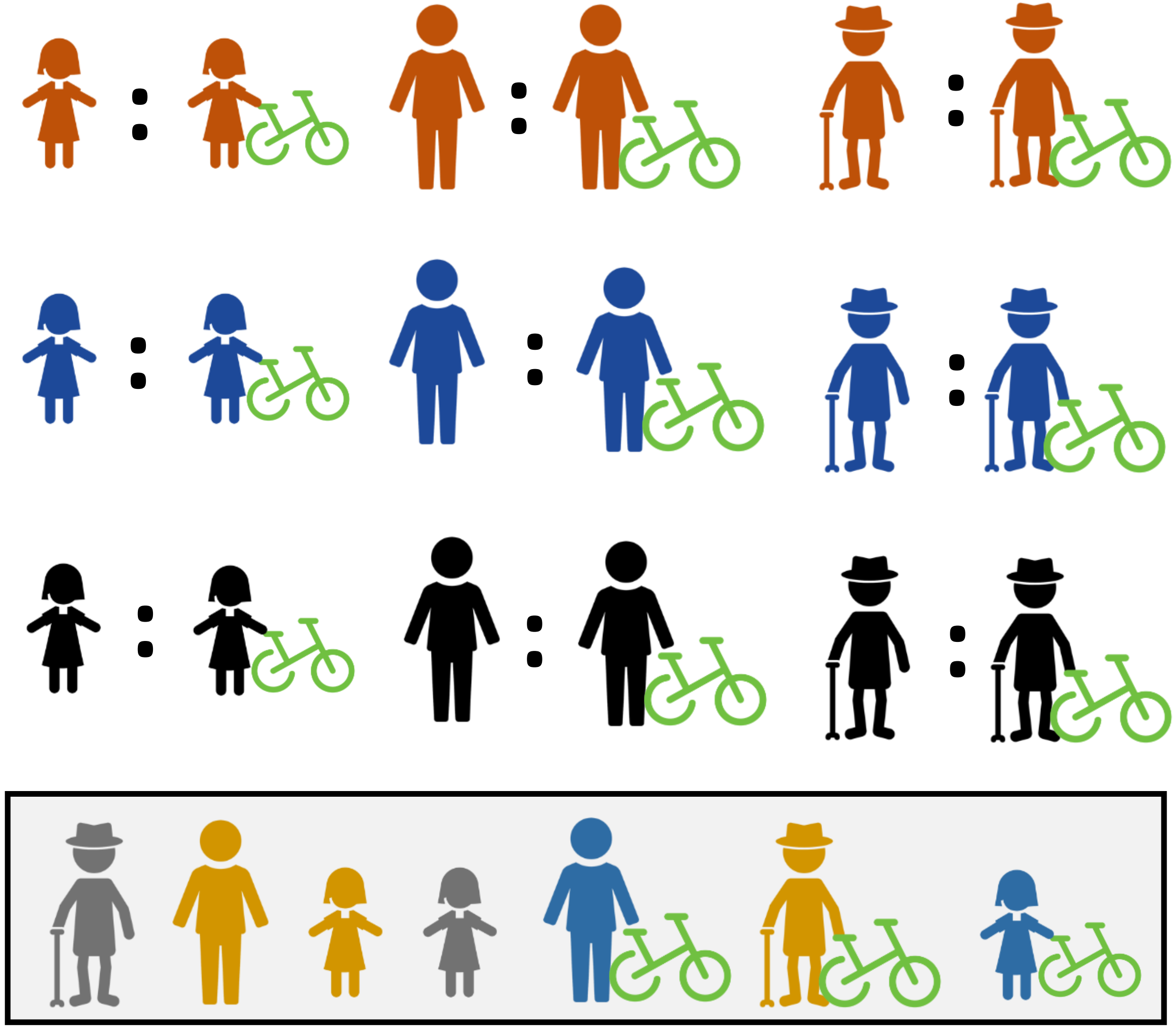
Simple Matching: Match data points with the same confounders and then compare their outcomes



Simple Matching: Match data points with the same confounders and then compare their outcomes

Identify pairs of treated (j) and untreated individuals (k) who are similar or identical to each other.

Match := $Distance(W_j, W_k) < \epsilon$



- Paired individuals have almost the same confounders.

Causal Effect =

$$\sum_{(j,k)\in Match}(y_j - y_k)$$

Challenges of building a good estimator

- **Variance:** If we have a stringent matching criterion, we may obtain very few matches and the estimate will be unreliable.

- **Bias:** If we relax the matching criterion, we obtain many more matches but now the estimate does not capture the target estimand.

- **Uneven treatment assignment:** If very few people have treatment, leads to both high bias and variance.

Need better methods to navigate the bias-variance tradeoff.

Depending on the dataset properties, different estimation methods can be used

Simple Conditioning

- Matching
- Stratification

Propensity Score-Based [Rubin 1983]

- Propensity Matching
- Inverse Propensity Weighting

Synthetic Control [Abadie et al.]

Outcome-based

- Double ML [Chernozhukov et al. 2016]
- T-learner
- X-learner [Kunzel et al. 2017]

Loss-Based

- R-learner [Nie & Wager 2017]

Threshold-based

- Difference-in-differences

All these methods can be called through DoWhy.
(directly or through the Microsoft EconML library)

IV. Robustness Checks: Test robustness of obtained estimate to violation of assumptions

Obtained estimate depends on many (untestable) assumptions.

Model:

Did we miss any unobserved variables in the assumed graph?

Did we miss any edge between two variables in the assumed graph?

Identify:

Did we make any parametric assumption for deriving the estimand?

Estimate:

Is the assumed functional form sufficient for capturing the variation in data?

Do the estimator assumptions lead to high variance?

Best practice: Do refutation/robustness tests for as many assumptions as possible

UNIT TESTS

Model:

- Conditional Independence Test

Identify:

- D-separation Test

Estimate:

- Bootstrap Refuter
- Data Subset Refuter

INTEGRATION TESTS

Test all steps at once.

- Placebo Treatment Refuter
- Dummy Outcome Refuter
- Random Common Cause Refuter
- Sensitivity Analysis
- Simulated Outcome Refuter /Synth-validation [Schuler et al. 2017]

All these refutation methods are implemented in Do Why.
Caveat: They can refute a given analysis, *but cannot prove its correctness.*

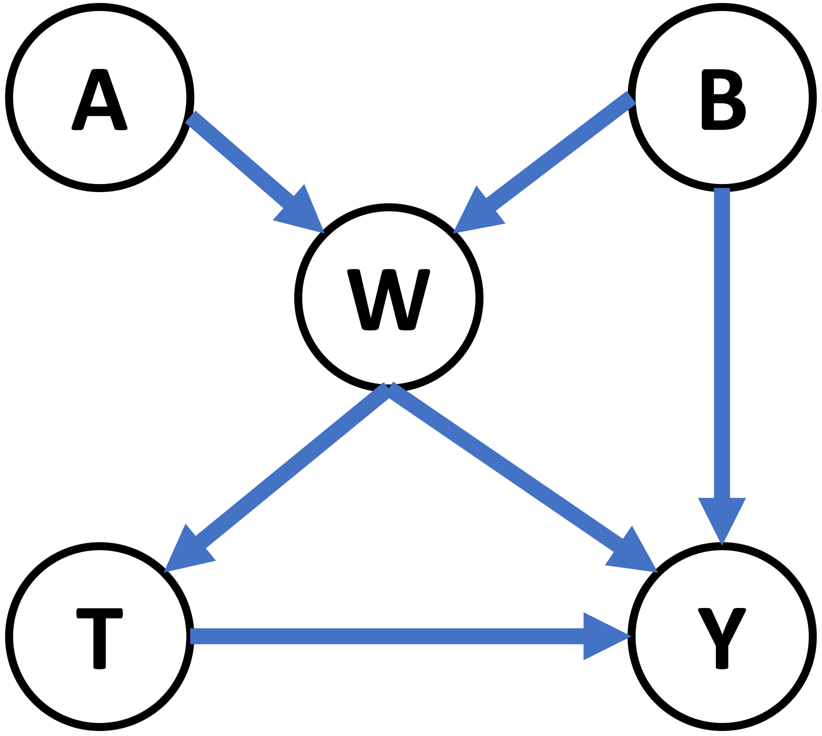
Example 1: Conditional Independence Refuter

Through its edges, each causal graph implies certain conditional independence constraints on its nodes. *[d-separation, Pearl 2009]*

Model refutation: Check if the observed data satisfies the assumed model's independence constraints.

- Use an appropriate statistical test for independence *[Heinze-Demel et al. 2018].*

- If not, the model is incorrect.



Conditional Independencies:

$A \perp\!\!\!\perp B$ $A \perp\!\!\!\perp \mathrm{T}|\mathrm{W}$ $B \perp\!\!\!\perp \mathrm{T}|\mathrm{W}$

Example 2: Placebo Treatment ("A/A") Refuter

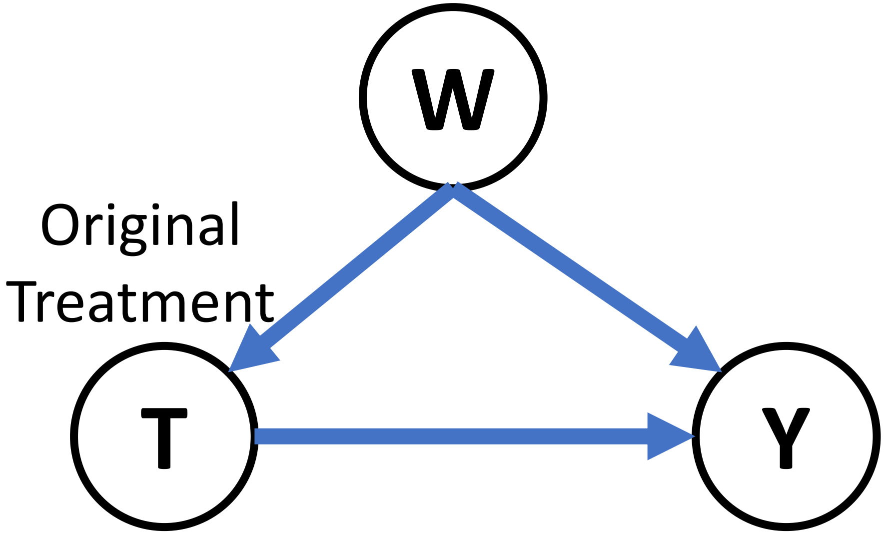
Q: *What if we can generate a dataset where the treatment does not cause the outcome?*

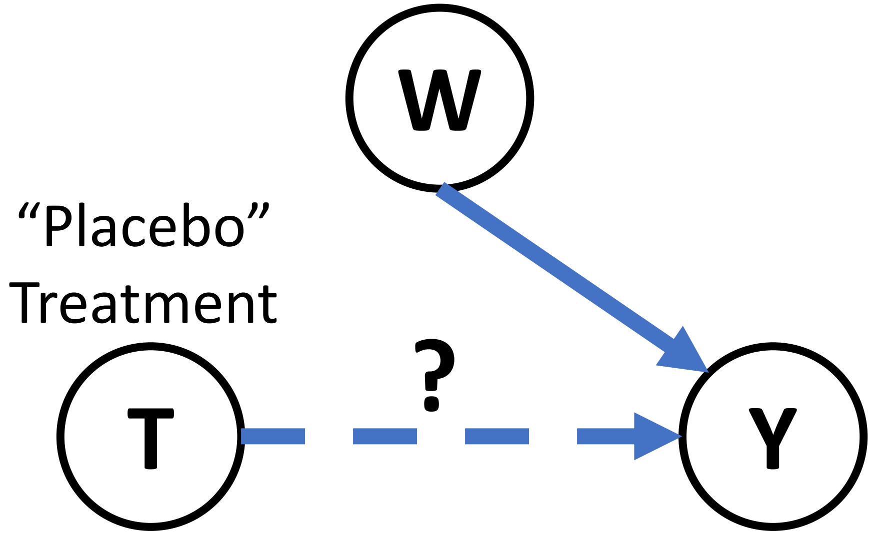
Then a correct causal inference method should return an estimate of zero.

Placebo Treatment Refuter:

Replace treatment variable T by a randomly generated variable (e.g., Gaussian).

- Rerun the causal inference analysis.
- If the estimate is significantly away from zero, then analysis is incorrect.

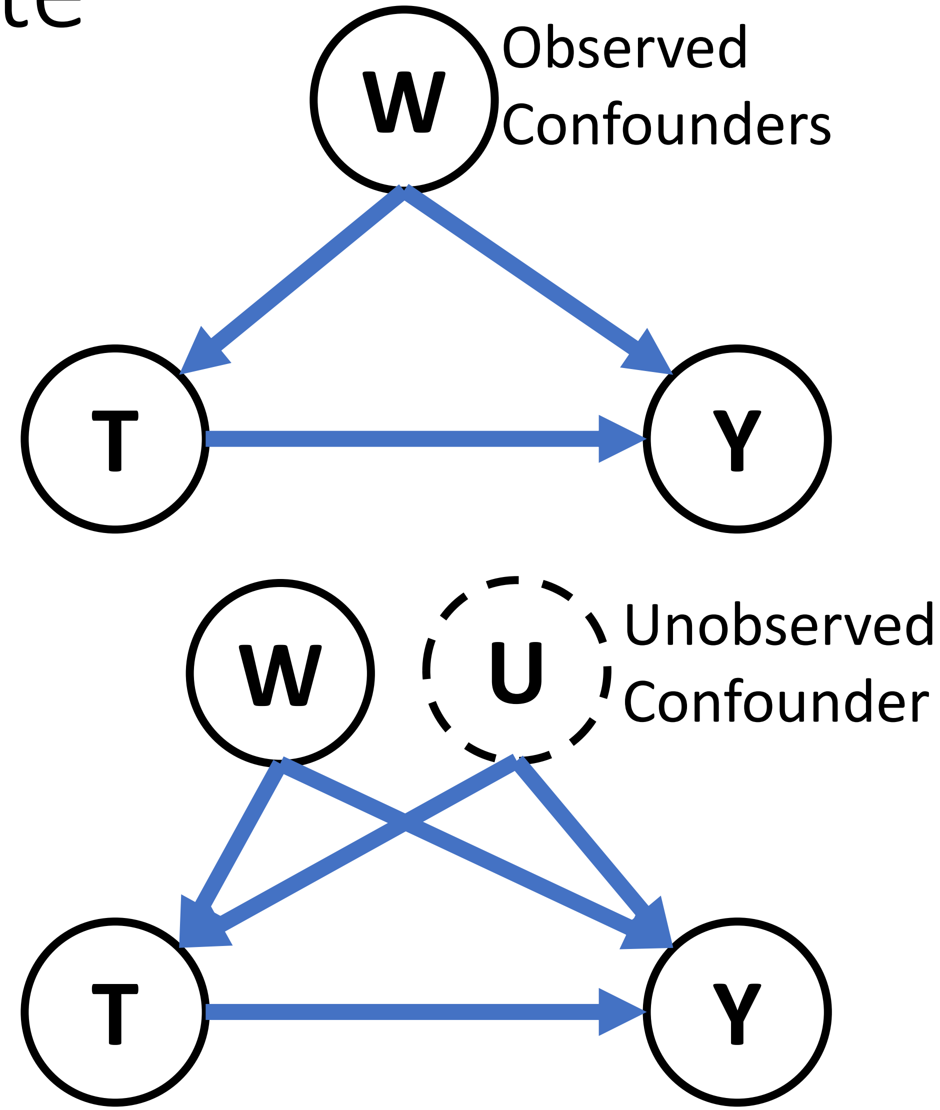




Example 3: Add Unobserved Confounder to check sensitivity of an estimate

Q: *What if there was an unobserved confounder that was not included in the causal model?*

Check how sensitive the obtained estimate is after introducing a new confounder.



Unobserved Confounder Refuter:

- Simulate a confounder based on a given correlation ρ with both treatment and outcome.
  - Maximum Correlation ρ is based on the maximum correlation of any observed confounder.
- Re-run the analysis and check if the sign/direction of estimate flips.

Walk-through of the 4 steps using the DoWhy Python library

Problem: Estimating the effect of a customer loyalty rewards program

What is the impact of offering the customer loyalty program on total sales?

If the current members ***had not signed up*** for the program, how much less would they have spent?

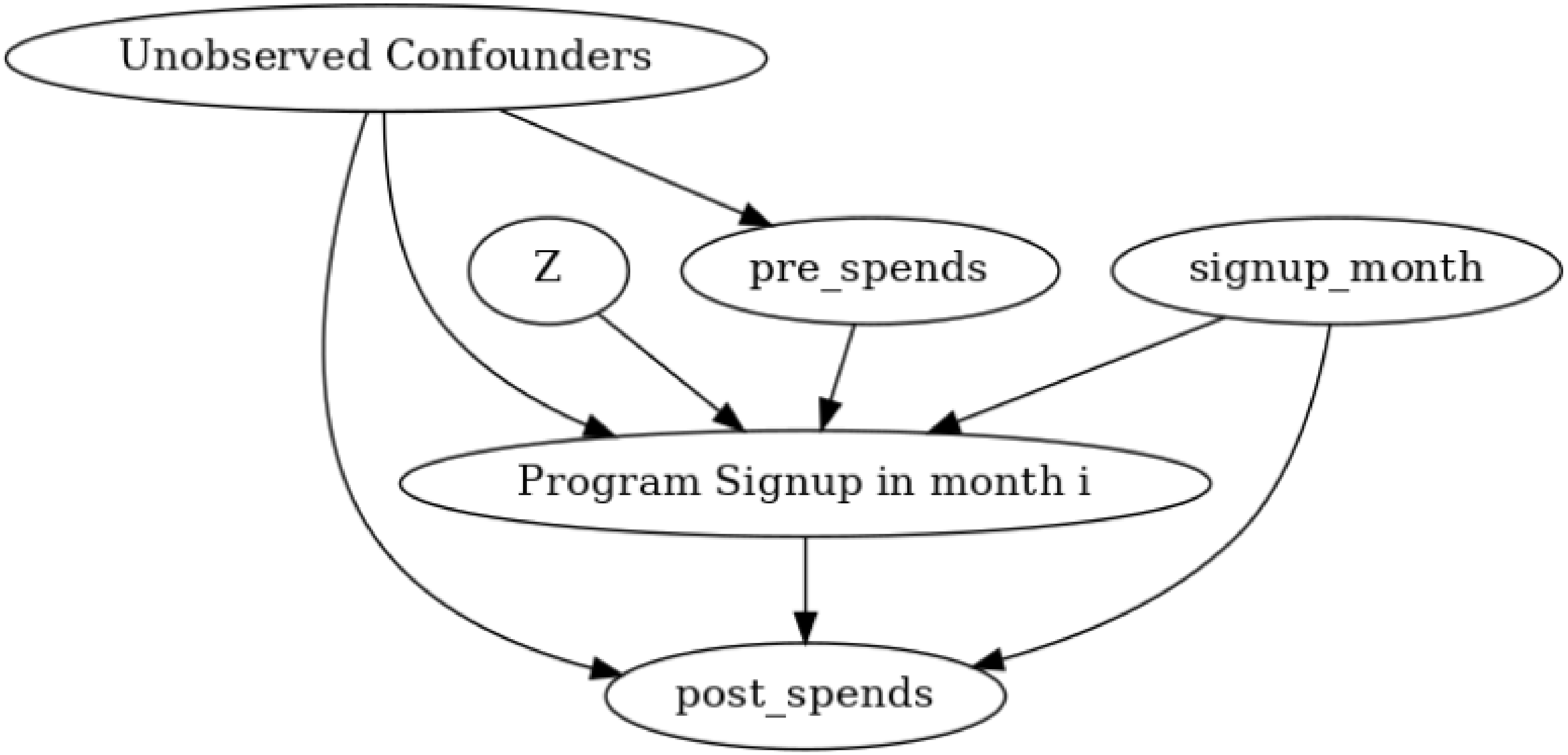
ATT: *Average treatment effect on the treated* (customers who signed up for the program)

	user_id	signup_month	month	spend	treatment
0	0	6	1	507	True
1	0	6	2	506	True
2	0	6	3	490	True
3	0	6	4	464	True
4	0	6	5	475	True
...	...	...	...	...	...
119995	9999	0	8	396	False
119996	9999	0	9	387	False
119997	9999	0	10	367	False
119998	9999	0	11	436	False

You can try out this example on Github:
github.com/microsoft/dowhy/blob/master/docs/source/example_notebooks/dowhy_example_effect_of_memberrewards_program.ipynb

Step 1: Modeling. Create causal graph to encode assumptions.

```
model = dowhy.CausalModel(data=df_i_signupmonth,
                          graph=causal_graph.replace("\n", " "),
                          treatment="treatment",
                          outcome="post_spends")
```



Step 2: Identification. Formulate what to estimate

```
identified_estimand = model.identify_effect(proceed_when_unidentifiable=True)
print(identified_estimand)
```

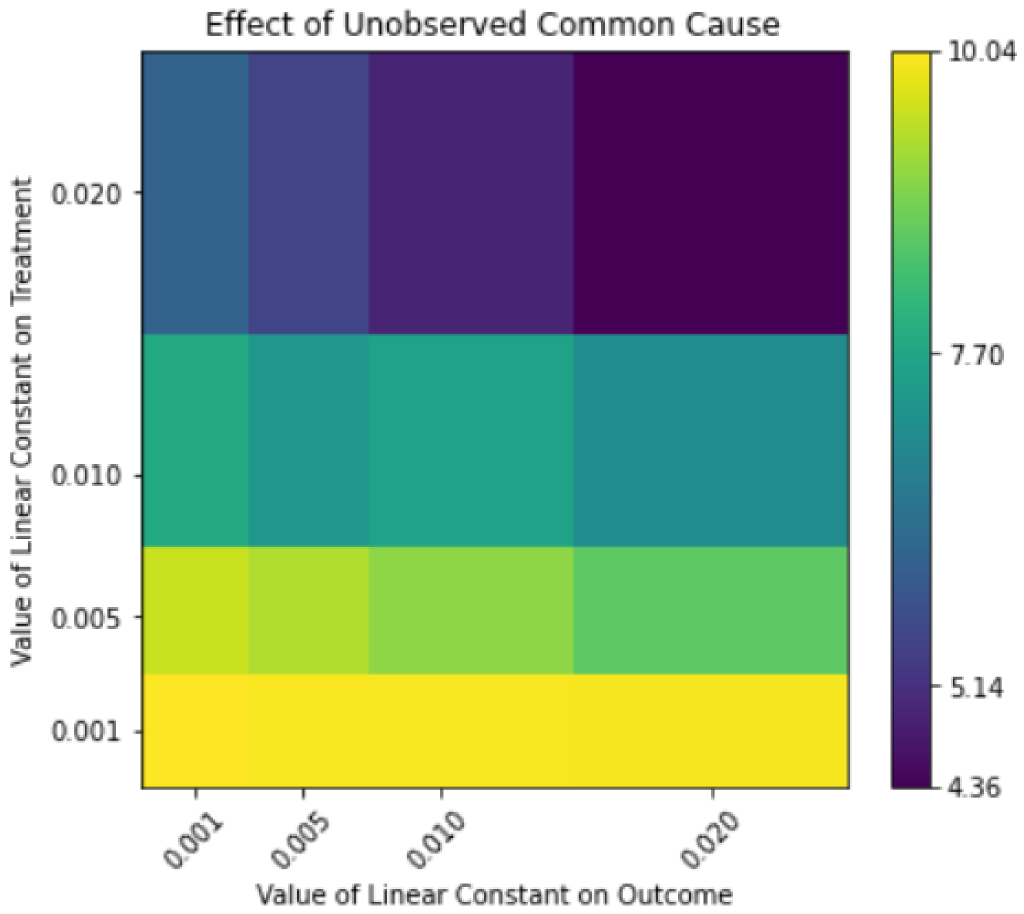
Step 3: Estimation. Compute the estimate

```
estimate = model.estimate_effect(identified_estimand,
                                  method_name="backdoor.propensity_score_matching",
                                 target_units="att")
print(estimate)
```

Step 4: Refutation. Validate the assumptions

```
refutation = model.refute_estimate(identified_estimand, estimate, method_name="placebo_treatment_refuter",
                                   placebo_type="permute", num_simulations=20)
print(refutation)
```

```
Refute: Use a Placebo Treatment
Estimated effect:100.03963044006804
New effect:0.6054947726720156
p value:0.24154316295878647
```



Future: Extending the four-step API to other causal tasks

- A unified, extensible API for causal inference that allows external implementations for the 4 steps
  - Supports invoking estimation methods from external libraries such as EconML and CausalML.

```
dml_estimate = model.estimate_effect(identified_estimand,
                                     method_name="backdoor.econml.dml.DMLCateEstimator",
                                     target_units = lambda df: df["X0"]>1,
                                     confidence_intervals=True,
```

- Extend the same 4-step API for,
  - Graphical causal model inference
  - Learning a causal graph from data (experimental)
  - Causal prediction models (coming soon!)

Summary: DoWhy, a library that focuses on **causal assumptions and their validation**

Goal: A unified API for causal tasks, just like PyTorch or Tensorflow for predictive ML.

Growing open-source community: > 50 contributors

- Roadmap: More powerful refutation tests, counterfactual prediction.
- Please contribute! Join the community on Discord or Github.

Resources

- DoWhy Library: https://github.com/py-why/dowhy
- Arxiv paper on the four steps: https://arxiv.org/abs/2011.04216
- Upcoming book on causality and ML: http://causalinference.gitlab.io/

Conclusion: Causal reasoning is necessary for both prediction and decision-making

- Causal models require assumptions, but not the full graph
- Can achieve superior results by simple, standard assumptions
  - CACM: attributes and their correlation type
  - DoWhy: confounders based on time order
- Big open question: Evaluation of causal models
  - Important to track progress in the field, for widespread adoption

thank you– Amit Sharma
(@amt_shrma)