Introduction to Causal Inference

A Machine Learning Perspective

Silin DU

Department of Management Science and Engineering
School of Economics and Management
Tsinghua University
`dsl21@mails.tsinghua.edu.cn`

SEM
TSINGHUA UNIVERSITY
清华经管学院

September 4, 2023

Lecture Overview

In this slide, we first introduce the basic ideas in causal inference, including

- Potential outcome framework.
- Causal graph.
- Structural causal model.

Then, we focus on Causal Machine Learning, including

- Causal supervised learning.
- Causal generative modeling.
- Causal explanations.
- Causal fairness.
- Causal reinforcement learning.

Contents

Potential Outcome Framework

Causal Graph

Causal Models

Causal Supervised Learning

- Invariant Feature Learning
  - Deconfound Data
  - Deconfound Intermediate Representations
  - Deconfound Models during Training
- Invariant Mechanism Learning

Causal Generative Modeling

- Structural Assignment Learning
- Causal Disentanglement

Causal Explanations

- Feature Attributions
- Contrastive Explanations

Causal Fariness

Causal Reinforcement Learning

Potential Outcomes

- T: the random variable for treatment and is binary in most cases.
- Y: the random variable for the outcome of interest.
- X: covariates.

Let's consider the scenario where you are unhappy. And you are considering whether or not to get a dog to help make you happy.

- You will still be happy if you get a dog.
- If you don't get a dog, you will remain unhappy.

In this scenario, your outcome Y is happiness

$$Y = 1 : \text{happy}, \quad Y = 0 : \text{unhappy}$$

The treatment T is whether or not to get a dog.

- We denote by $Y(1)$ the *potential outcome* of happiness you would observe if you were to get a dog ($T = 1$).
- Similarly, we can define $Y(0)$.
- In this scenario, $Y(1) = 1, Y(0) = 0$.

Individual Treatment Effects

- More generally, the *potential outcome* $Y(t)$ denotes the outcome would be if the treatment is t.
- A potential outcome $Y(t)$ is distinct from the observed outcome Y in that not all potential outcomes are observed.
- In the previous scenarios, there was only a single individual (unit) in the whole population.
- Generally, there are many units in the population of interest.
- The treatment, covariates, and outcome of the i_{th} unit: T_i, X_i, Y_i.
- We can define the *individual treatment effect* (ITE) for unit i:

$$\tau_i \triangleq Y_i(1) - Y_i(0)$$

- $Y(t)$ is a random variable because different units will have different potential outcomes.
- $Y_i(t)$ is usually treated as non-random.

The Fundamental Problem of Causal Inference

Holland (1986)

It is impossible to observe all potential outcomes of the same unit.

- Same object or person at a different time is a different unit.
- We cannot observe both $Y_i(0)$ and $Y_i(1)$.
- This is known as the *fundamental problem of causal inference*.
- This problem is unique to causal inference because, in causal inference, we care about making causal claims, which are defined in terms of potential outcomes.
- In machine learning, we often only care about predicting the observed outcome Y, so there is no need for potential outcomes.
- The potential outcomes that you do not (and cannot) observe are known as *counterfactuals* because they are counter to fact (reality).

Average Treatment Effect I

- We get the *average treatment effect* (ATE) by taking an average over the ITEs:

$$\tau \triangleq \mathbb{E}[Y_i(1) - Y_i(0)] = \mathbb{E}[Y(1) - Y(0)]$$

where the average is over the units i.

- How to compute the ATE?

Consider the following table as the whole population of interest

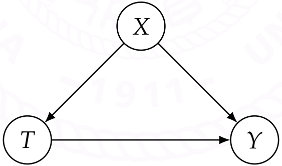
i	T	Y	$Y(1)$	$Y(0)$	$Y(1) - Y(0)$
1	0	0	?	0	?
2	1	1	1	?	?
3	1	0	0	?	?
4	0	0	?	0	?
5	0	1	?	1	?
6	1	1	1	?	?

Average Treatment Effect II

- The fundamental problem of causal inference is actually a missing data problem.
- Association difference: $\mathbb{E}[Y \mid T = 1] - \mathbb{E}[Y \mid T = 0]$.

$$\text{ATE} = \mathbb{E}[Y(1) - Y(0)] = \mathbb{E}[Y(1)] - \mathbb{E}[Y(0)] \overset{?}{=} \mathbb{E}[Y \mid T = 1] - \mathbb{E}[Y \mid T = 0]$$

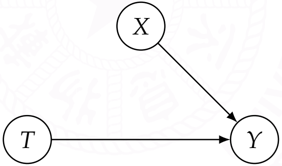
- Unfortunately, this is not true in general.
- If it were, that would mean that causation is simply association.
- They are not equal due to confounding.



Ignorability and Exchangeability I

What assumption(s) would make it so that the ATE is simply the associational difference?

- What makes it valid to calculate the ATE by taking the average of the $Y(0)$ column, ignoring the question marks, and subtracting that from the average of the $Y(1)$ column, ignoring the question marks?
- This ignoring of the question marks (missing data) is known as ignorability.
- Assuming ignorability is like assuming units were randomly assigned to different treatment.



Assumption 1.1 (Ignorability / Exchangeability)

$$(Y(0), Y(1)) \perp T$$

Ignorability and Exchangeability II

- This assumption allows us to reduce the ATE to the associational difference:

$$\begin{aligned} \mathbb{E}[Y(1)] - \mathbb{E}[Y(0)] &= \mathbb{E}[Y(1) \mid T = 1] - \mathbb{E}[Y(0) \mid T = 0] \quad (1.1) \\ &= \mathbb{E}[Y \mid T = 1] - \mathbb{E}[Y \mid T = 0] \quad (1.2) \end{aligned}$$

  The ignorability assumption is used in Equation (1.1).

- Another perspective on this assumption is that of exchangeability.
- Exchangeability means that the treatment groups are exchangeable in the sense that if they were swapped, they would observe the same outcomes.
- Formally, this assumption means

$$\mathbb{E}[Y(1) \mid T = 0] = \mathbb{E}[Y(1) \mid T = 1], \quad \mathbb{E}[Y(0) \mid T = 0] = \mathbb{E}[Y(0) \mid T = 1]$$

  which implies that

$$\mathbb{E}[Y(1) \mid T = t] = \mathbb{E}[Y(1)], \quad \mathbb{E}[Y(0) \mid T = t] = \mathbb{E}[Y(0)]$$

Identifiability

- We have leveraged Assumption 1.1 to identify causal effects.
- To identify a causal effect is to reduce a causal expression (potential outcome notations) to a purely statistical expression (such as T, X, and expectations).
- This means that we can calculate the causal effect from just the observational distribution $P(X, T, Y)$.

Definition 1.1 (Identifiability)

A causal quantity (e.g. $\mathbb{E}[Y(t)]$) is identifiable if we can compute it from a purely statistical quantity (e.g. $\mathbb{E}[Y \mid t]$).

Conditional Exchangeability and Unconfoundedness I

- In general, it is completely unrealistic to assume that the treatment groups are exchangeable.
- However, if we control for relevant variables by conditioning, then maybe the subgroups will be exchangeable.

Definition 1.2 (Conditional Exchangeability / Unconfoundedness)

$$(Y(0), Y(1)) \perp T \mid X$$

Unconfoundedness is the main assumption necessary for causal inference.

$$\begin{aligned}
\mathbb{E}[Y(1) - Y(0) \mid X] &= \mathbb{E}[Y(1) \mid X] - \mathbb{E}[Y(0) \mid X] && (1.3) \\
&= \mathbb{E}[Y(1) \mid T = 1, X] - \mathbb{E}[Y(0) \mid T = 0, X] && (1.4) \\
&= \mathbb{E}[Y \mid T = 1, X] - \mathbb{E}[Y \mid T = 0, X] && (1.5)
\end{aligned}$$

Conditional Exchangeability and Unconfoundedness II

We get Equation (1.4) by conditional exchangeability.

$$\begin{aligned}\mathbb{E}[Y(1) - Y(0)] &= \mathbb{E}_X \mathbb{E}[Y(1) - Y(0) \mid X] && (1.6)\\ &= \mathbb{E}_X[\mathbb{E}[Y \mid T = 1, X] - \mathbb{E}[Y \mid T = 0, X]] && (1.7)\end{aligned}$$

Theorem 1.1 (Adjustment Formula)

Given the assumptions of unconfoundedness, overlap, consistency, and no interference, we can identify the average treatment effect:

$$\mathbb{E}[Y(1) - Y(0)] = \mathbb{E}_X[\mathbb{E}[Y \mid T = 1, X] - \mathbb{E}[Y \mid T = 0, X]]$$

Positivity / Overlap I

Positivity is the condition that all subgroups of the data with different covariates have some probability of receiving any value of treatment.

Assumption 1.2 (Positivity / Overlap / Common Support)

For all values of covariates x present in the population of interest (i.e. x such that $P(X = x) > 0$),

$$0 < P(T = 1 \mid X = x) < 1$$

- If we have a positivity violation, then we will be conditioning on a zero probability event.
- There will be some value of x with with non-zero probability for which $P(T = 1 \mid X = x) = 0$ or $P(T = 0 \mid X = x) = 0$.

Positivity/Overlap II

- For discrete covariates and outcome, we have

$$
\begin{aligned}
&\mathbb{E}[Y(1) - Y(0) \mid X] \\
=&\mathbb{E}_X[\mathbb{E}[Y \mid T = 1, X] - \mathbb{E}[Y \mid T = 0, X]] \\
=&\sum_x P(X = x) \left(\sum_y y P(Y = y \mid T = 1, X = x) - \sum_y y P(Y = y \mid T = 0, X = x) \right) \\
=&\sum_x P(X = x) \left(\sum_y y \frac{P(Y = y, T = 1, X = x)}{P(T = 1 \mid X = x) P(X = x)} - \sum_y y \frac{P(Y = y, T = 0, X = x)}{P(T = 0 \mid X = x) P(X = x)} \right) && \text{(Bayes' rule)}
\end{aligned}
$$

- If $P(T = 1 \mid X = x) = 0$ for any level of covariates x with non-zero probability, then there is division by zero in the first term, so $\mathbb{E}_X[\mathbb{E}[Y \mid T = 1, X] - \mathbb{E}[Y \mid T = 0, X]$ is undefined.

Positivity/Overlap III

- Another name for positivity is *overlap*. The intuition for this name is that we want the covariate distribution of the treatment group to overlap with the covariate distribution of the control group.
- **Positivity-Unconfoundedness Tradeoff**. Although conditioning on more covariates could lead to a higher chance of satisfying unconfoundedness, it can lead to a higher chance of violating positivity.

No Interference, Consistency, and SUTVA I

- No interference means that my outcome is unaffected by anyone else's treatment. Rather, my outcome is only a function of my own treatment.

Assumption 1.3 (No Interference)

$$Y_i(t_1, ..., t_{i-1}, t_i, t_{i+1}, ..., t_n) = Y_i(t_i)$$

- Consistency is the assumption that the outcome we observe Y is actually the potential outcome under the observed treatment T.
- Consistency encompasses the assumption that is sometimes referred to as no multiple versions of treatment.

No Interference, Consistency, and SUTVA II

Assumption 1.4 (Consistency)

If the treatment is T, then the observed outcome Y is the potential outcome under treatment T. Formally,

$$T = t \Rightarrow Y = Y(t)$$

equivalentlt,

$$Y = Y(T)$$

- It's commonly to see the *stable unit-treatment value assumption* (SUTVA) in the literature.
- SUTVA is satisfied if unit (individual) i's outcome is simply a function of unit i's treatment.
- Therefore, SUTVA is a combination of consistency and no interference.

Revisit the Adjustment Formula

All assumptions are needed

- Unconfoundedness (Assumption 1.2)
- Positivity (Assumption 1.2)
- No interference (Assumption 1.3)
- Consistency (Assumption 1.4)

Proof of Theorem 1.1.

$$\begin{aligned}
\mathbb{E}[Y(1) - Y(0)] &= \mathbb{E}[Y(1)] - \mathbb{E}[Y(0)] && \text{(linearity of expectation)} \\
&= \mathbb{E}_X[\mathbb{E}[Y(1) \mid X] - \mathbb{E}[Y(0) \mid X]] && \text{(law of iterated expectations)} \\
&= \mathbb{E}_X[\mathbb{E}[Y(1) \mid T = 1, X] - \mathbb{E}[Y(0) \mid T = 0, X]] && \text{(unconfoundedness and positivity)} \\
&= \mathbb{E}_X[\mathbb{E}[Y \mid T = 1, X] - \mathbb{E}[Y \mid T = 0, X]] && \text{(consistency)}
\end{aligned}$$

Statistics Terminology

- An *estimand* is the quantity that we want to estimate. For example, $\mathbb{E}_X[\mathbb{E}[Y \mid T = 1, X] - \mathbb{E}[Y \mid T = 0, X]]$ is the estimand we care about for estimating the ATE.
- An *estimate (none)* is an approximation of some estimand, which we get using data.
- An *estimator* is a function that maps a dataset to an estimate of the estimand.
- To *estimate (verb)* is to feed data into an estimator to get an estimate.
- The process that we will use to go from data + estimand to a concrete number is known as *estimation*.
- *Causal estimand* refers to any estimand that contains a potential outcome.
- Identification refers to the process of moving from a causal estimand to an equivalent statistical estimand.
- Estimation refers to the process of moving from a statistical estimand to an estimate.

Contents

Potential Outcome Framework

Causal Graph

Causal Models

Causal Supervised Learning

- Invariant Feature Learning
  - Deconfound Data
  - Deconfound Intermediate Representations
  - Deconfound Models during Training
- Invariant Mechanism Learning

Causal Generative Modeling

- Structural Assignment Learning
- Causal Disentanglement

Causal Explanations

- Feature Attributions
- Contrastive Explanations

Causal Fariness

Causal Reinforcement Learning

Graph Terminology I

- A *graph* is a collection of nodes (also called vertices) and edges that connect the nodes.
- We will denote the parents of a node X with $\text{pa}(X)$.
- A *path* in a graph is any sequence of adjacent nodes, regardless of the direction of the edges that join them.
- A *directed path* is a path that consists of directed edges that are all directed in the same direction.
- If there is a directed path that starts at node X and ends at node Y, then X is an *ancestor* of Y, and Y is a *descendant* of X.
- We will denote descendants of X by $\text{de}(X)$.
- A directed path from some node X back to itself is known as a *cycle*.
- If there are no cycles in a directed graph, the graph is known as a *directed acyclic graph* (DAG).
- We mostly focus on DAGs.

Bayesian Network I

- Bayesian networks are the main probabilistic graphical model that causal graphical models (causal Bayesian networks) inherit most of their properties from.
- In general, we can use the chain rule of probability to factorize any distribution:

$$P(x_1, ..., x_n) = P(x_1) \prod_i P\left(x_i \mid x_{i-1}, ..., x_1\right)$$

  However, it would take an exponential number of parameters to model the distribution.
- Only model *local dependencies*.
- Given a probability distribution and a corresponding DAG, we can formalize the specification of independencies with the local Markov assumption:

Assumption 2.1 (Local Markov Assumption)

Given its parents in the DAG, a node X is independent of all its non-descendants.

Bayesian Network II

- Consider an example with 4 variables.
- We can factorize any P such that

$$P(x_1, x_2, x_3, x_4) = P(x_1)P(x_2 \mid x_1)P(x_3 \mid x_2, x_1)P(x_4 \mid x_3, x_2, x_1)$$

- If P is Markov with respect to the graph in Figure 2.1, then we can simplify

$$P(x_1, x_2, x_3, x_4) = P(x_1)P(x_2 \mid x_1)P(x_3 \mid x_2, x_1)P(x_4 \mid x_3)$$

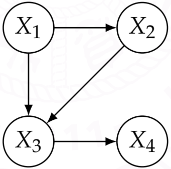


Fig. 2.1: Four node DAG

Bayesian Network III

- The main consequences of the local Markov assumption:

Definition 2.1 (Bayesian Network Factorization)

Given a probability distribution P and a DAG G, P factorizes according to G if

$$P(x_1, ..., x_n) = \prod_i P(x_i \mid \mathrm{pa}_i)$$

- The Bayesian network factorization is also known as the *chain rule for Bayesian networks* or *Markov compatibility*.
- The local Markov assumption does not even tell us that if X and Y are adjacent in the DAG, then X and Y are dependent.
- We will generally assume a slightly stronger assumption than the local Markov assumption.

Bayesian Network IV

Assumption 2.2 (Minimality Assumption)

1. *Given its parents in the DAG, a node X is independent of all its non-descendants (Assumption 2.1)*
2. *Adjacent nodes in the DAG are dependent.*

The minimality assumption is equivalent to saying that we can't remove any more edges from the graph. In a sense, every edge is active.

Causal Graphs

Definition 2.2 (What is a cause?)

A variable X is said to be a cause of a variable Y if Y can change in response to changes in X.

Assumption 2.3 ((Strict) Causal Edges Assumption)

In a directed graph, every parent is a direct cause of all its children.

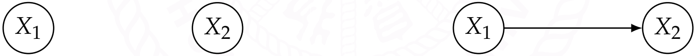
- If we fix all of the direct causes of Y, then changing any other cause of Y won't induce any changes in Y.
- This assumption is strict in the sense that every edge is active, just like in DAGs that satisfy minimality.
- When we add the causal edges assumption, directed paths in the DAG take on a very special meaning; they correspond to causation.

Two Nodes

- Flow of association: whether any two nodes in a graph are associated (statistically dependent) or not associated (statistically independent).
- Two unconnected nodes.

$$P(x_1, x_2) = P(x_1)P(x_2)$$

- In contrast, if there is an edge between the two nodes then the two nodes are associated.



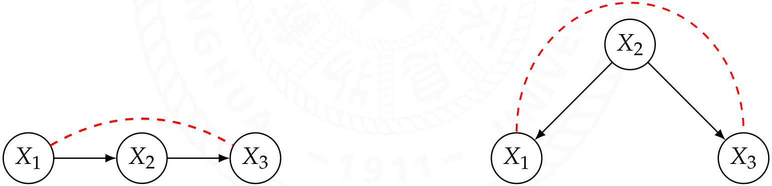
(a) Two unconnected nodes

(b) Two connected nodes

Fig. 2.2: Two nodes in a graph

Chain and Fork I

- Chain and forks share the same set of dependencies.
- In both structures, X_1 and X_2 are dependent, and X_2 and X_3 are dependent.
- X_1 and X_3 are associated in both chains and forks.
- In the chain, association flows from X_1 to X_3 along the path $X_1 \to X_2 \to X_3$.
- In the fork, association flows from X_1 to X_3 along the path $X_1 \leftarrow X_2 \to X_3$.
- In general, the flow of association is symmetric.



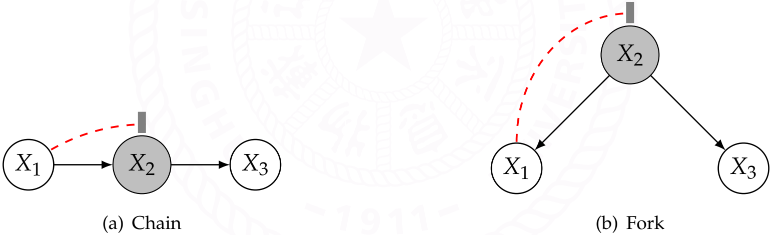
(a) Chain

(b) Fork

Fig. 2.3: Chain and fork with flow of association drawn as a dashed red arc.

Chain and Fork II

- When we condition on X_2 in both graphs, it blocks the flow of association from X_1 to X_3., i.e., $X_1 \perp X_3 \mid X_2$.
- This is because of the local Markov assumption; each variable can locally depend on only its parents.



(a) Chain

(b) Fork

Fig. 2.4: Chain and fork with association blocked by conditioning on X_2.

Chain and Fork III

- For chains, we can factorize $P(x_1, x_2, x_3)$ as follows:

$$P(x_1, x_2, x_3) = P(x_1)P(x_2 \mid x_1)P(x_3 \mid x_2)$$

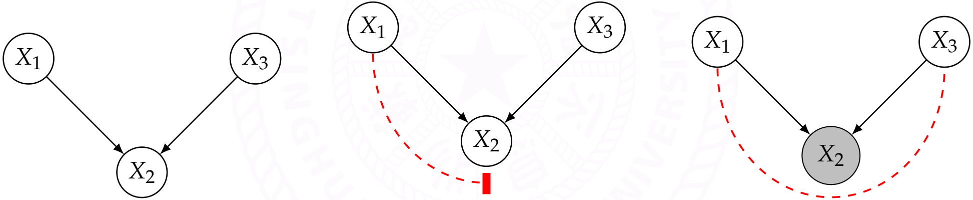
Then by the Bayes' rule, we have

$$\begin{aligned} P(x_1, x_3 \mid x_2) &= \frac{P(x_1, x_2, x_3)}{P(x_2)} \\ &= \frac{P(x_1)P(x_2 \mid x_1)P(x_3 \mid x_2)}{P(x_2)} \\ &= \frac{P(x_1, x_2)}{P(x_2)} P(x_3 \mid x_2) \\ &= P(x_1 \mid x_2)P(x_3 \mid x_2) \end{aligned}$$

- The flow of association is symmetric, whereas the ow of causation is not.
- Under the causal edges assumption (Assumption 2.3), causation only flows in a single direction. Causation only flows along directed paths.

Immorality and Colliders I

- We have an immorality when we have a child whose two parents do not have an edge connecting them.
- The child is known as a *collider*.



(a) Immorality

(b) Immorality with association blocked by a collider.

(c) Immorality with association unblocked by conditioning on the collider.

Fig. 2.5: Immorality

Immorality and Colliders II

- In an immorality, $X_1 \perp X_3$.

$$
\begin{aligned}
P(x_1, x_3) &= \sum_{x_2} P(x_1, x_2, x_3) \\
&= \sum_{x_2} P(x_1)P(x_3)P(x_2 \mid x_1, x_3) \\
&= P(x_1)P(x_3) \sum_{x_2} P(x_2 \mid x_1, x_3) \\
&= P(x_1)P(x_3)
\end{aligned}
$$

- Conditioning on a collider can turn a blocked path into an unblocked path.
- This is sometimes referred to as *selection bias*.
- Conditioning on descendants of a collider also induces association in between the parents of the collider.

d-separation I

Definition 2.3 (Blocked Path)

A path between nodes X and Y is blocked by a (potentially empty) conditioning set Z if either of the following is true:

1. *Along the path, there is a chain $\cdots \to W \to \cdots$ or a fork $\cdots \leftarrow W \to \cdots$, where W is conditioned on ($W \in Z$).*
2. *There is a collider W, on the path that is not conditioned on ($W \notin Z$) and none of its descendants are conditioned on ($\mathrm{de}(W) \not\subseteq Z$).*

Definition 2.4 (d-Separation)

Two (sets of) nodes X and Y are d-separated by a set of nodes Z if all of the paths between (any node in) X and (any node in) Y are blocked by Z.

d-separation II

- Similarly, if there exists at least one path between X and Y that is unblocked, then we say that X and Y are *d-connected*.
- d-separation is such an important concept because it implies conditional independence.
- $X \perp_G Y \mid Z$ ($X \perp_P Y \mid Z$) denotes that X and Y are d-separated in the graph G (the distribution P) when conditioning on Z.

Theorem 2.1

Given that P is Markov with respect to G (satisfies the local Markov assumption, Assumption 2.1), if X and Y are d-separated in G conditioned on Z , then X and Y are independent in P conditioned on Z. We can write this succinctly as follows:

$$X \perp_G Y \mid Z \Longrightarrow X \perp_P Y \mid Z \tag{2.1}$$

d-separation III

- We call Equation (2.1) the *global Markov assumption*.
- Theorem 2.1 tells us that the local Markov assumption implies the global Markov assumption.
- The local Markov assumption, global Markov assumption, and the Bayesian network factorization are all equivalent.
- We will use *Markov assumption* to refer to these concepts as a group, or we will simply say P is Markov with respect to G.

Flow of Association and Causation

- We refer to the flow of association along directed paths as *causal association*.
- A common type of non-causal association that makes total association not causation is *confounding association*.
- d-separation implies association is causation.

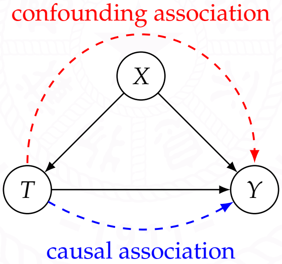


Fig. 2.6: Causal graph depicting an example of how confounding association and causal association flow.

Contents

- Potential Outcome Framework
- Causal Graph
- **Causal Models**
- Causal Supervised Learning
  - Invariant Feature Learning
    - Deconfound Data
    - Deconfound Intermediate Representations
    - Deconfound Models during Training
  - Invariant Mechanism Learning
- Causal Generative Modeling
  - Structural Assignment Learning
  - Causal Disentanglement
- Causal Explanations
  - Feature Attributions
  - Contrastive Explanations
- Causal Fariness
- Causal Reinforcement Learning

do-operator I

- In the regular notation for probability, we have conditioning, but that isn't the same as intervening.
- Conditioning on $T = t$ just means that we are restricting our focus to the subset of the population to those who received treatment t.
- In contrast, an intervention would be to take the whole population and give everyone treatment t.
- We will denote intervention with the do-operator: $do(T = t)$.

do-operator II

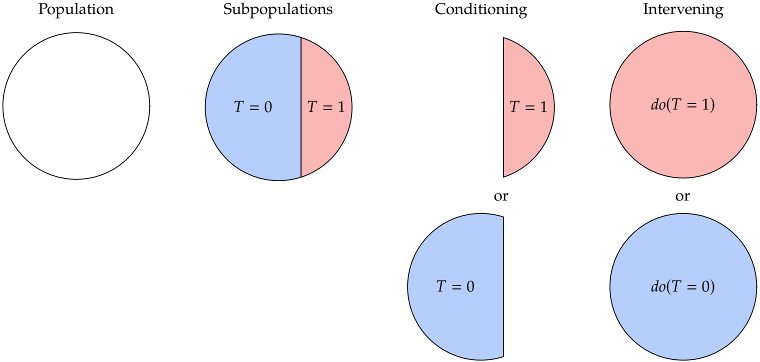


Fig. 3.1: Illustration of the difference between conditioning and intervening

do-operator III

- For example, we can write the distribution of the potential outcome $Y(t)$ as follows:

$$P(Y(t) = y) \triangleq P(Y = y \mid do(T = t)) \triangleq P(y \mid do(t))$$

  Also, we can similarly write the ATE (average treatment effect) when the treatment is binary as follows:

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

- We will commonly refer to $P(y \mid do(t))$ and other other expressions with the *do*-operator in them as *interventional distributions*.
- If we can reduce an expression Q with *do* in it (an interventional expression) to one without *do* in it (an observational expression), then Q is said to be identifiable.
- we will refer to an estimand as a causal estimand when it contains a *do*-operator, and we refer to an estimand as a statistical estimand when it doesn't contain a *do*-operator.

Modularity I

- We refer to the *causal mechanism* that generates X_i as the conditional distribution of X_i given all of its causes: $P(x_i \mid \mathrm{pa}_i)$.
- In order to get many causal identification results, the main assumption we will make is that interventions are local.
- More specifically, we will assume that intervening on a variable X_i only changes the causal mechanism for X_i; it does not change the causal mechanisms that generate any other variables.

Assumption 3.1 (Modularity / Independent Mechanisms / Invariance)

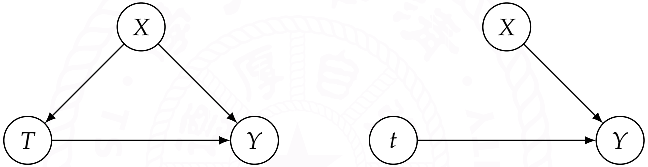
If we intervene on a set of nodes $S \subseteq \{1, ..., n\}$, setting them to constants, then for all i , we have the following:

1. *If $i \notin S$, then $P(x_i \mid \mathrm{pa}_i)$ remains unchanged.*
2. *If $i \in S$, then $P(x_i \mid \mathrm{pa}_i) = 1$ if x_i is the value that X_i was set to by the intervention; otherwise, $P(x_i \mid \mathrm{pa}_i) = 0$.*

Modularity II

- In the second part, we could have alternatively said $P(x_i \mid \mathrm{pa}_i) = 1$ if x_i is *consistent* with the intervention and 0 otherwise.
- The modularity assumption is what allows us to encode many different interventional distributions all in a single graph.
- The causal graph for interventional distributions is simply the same graph that was used for the observational joint distribution, but with *all of the edges to the intervened node(s) removed*.
- The graph with edges removed is known as the *manipulated graph*.

Modularity III



(a) Causal graph in the observational setting.

(b) Manipulated graph when intervening T to t.

Fig. 3.2: Causal graph and manipulated graph.

- Taking the modularity assumption (Assumption 3.1) and the Markov assumption (the other key principle) together gives us *causal Bayesian networks*.

Truncated Factorization

- Bayesian network factorization (Definition 2.1)

$$P(x_1, \ldots, x_n) = \prod_i P(x_i \mid \mathrm{pa}_i)$$

- If we intervene on some set of nodes S and assume modularity, then all of the factors should remain the same except the factors for $X_i \in S$.
- Those factors should change to 1 because those variables have been intervened on.

Proposition 3.1 (Truncated Factorization)

We assume that P and G satisfy the Markov assumption and modularity. Given, a set of intervention nodes S, if x is consistent with the intervention, then

$$P(x_1, \ldots, x_n \mid do(S = s)) = \prod_{i \notin S} P(x_i \mid \mathrm{pa}_i)$$

Otherwise, $P(x_1, \ldots, x_n \mid do(S = s)) = 0.$

Revisit "Association is Not Causation" I

- To identify the causal quantity $P(y \mid do(t))$.
- The distribution P is Markov with respect to the graph in Figure 3.3.

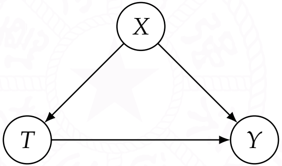


Fig. 3.3: Simple causal structure where X confounds the effect of T on Y and X is the only confounder.

- The Bayesian network factorization gives us

$$P(y, t, x) = P(x)P(t \mid x)P(y \mid t, x)$$

Revisit "Association is Not Causation" II

- When we intervene on the treatment, the truncated factorization gives us

$$P(y, x \mid do(t)) = P(x)P(y \mid t, x)$$

- We simply need to marginalize out X to get

$$P(y \mid do(t)) = \sum_x P(y \mid t, x)P(x) \tag{3.1}$$

- Replacing $P(x)$ by $P(x \mid t)$ in Equation (3.1), we have

$$\sum_x P(y \mid t, x)P(x \mid t) = \sum_x P(y, x \mid t) = P(y \mid t)$$

- In this example, the difference between $P(y \mid do(t))$ and $P(y \mid t)$ is the difference between $P(x)$ and $P(x \mid t)$.

Revisit "Association is Not Causation" III

- Assume that T is a binary variable.
- $P(y \mid do(1))$ is the distribution for $Y(1)$. Then we can write the ATE as follows:

$$\mathbb{E}[Y(0) - Y(1)] = \sum_y yP(y \mid do(1)) - \sum_y yP(y \mid do(0))$$

- Then plugging in Equation 3.1 yields a fully identified ATE.

Backdoor Adjustment I

- We want to turn the causal estimand $P(y \mid do(t))$ into a statistical estimand.
- We'll start with assuming we have a set of variables W that satisfy the *backdoor criterion*.

Definition 3.1 (Backdoor Criterion)

A set of variables W satisfies the backdoor criterion relative to T and Y if the following are true:

1. *W blocks all backdoor paths from T to Y.*
   - *there is a chain $\cdots \to X \to \cdots$ or a fork $\cdots \leftarrow X \to \cdots$ and $X \in W$.*
   - *there is a collider on the path that is not in W and none of its descendants are in W.*
2. *W dose not contain any descendants of T.*

- Satisfying the backdoor criterion makes W a *sufficient adjustment set*.

Backdoor Adjustment II

Theorem 3.1 (Backdoor Adjustment)

Given the modularity assumption (Assumption 3.1), that W satisfies the backdoor criterion (Definition 3.1), and positivity (Assumption 1.2), we can identify the causal effect of T on Y:

$$P(y \mid do(t)) = \sum_{w} P(y \mid t, w) P(w)$$

- Use the usual trick of conditioning on variables and marginalizing them out:

$$P(y \mid do(t)) = \sum_{w} P(y \mid do(t), w) P(w \mid do(t))$$

Backdoor Adjustment III

- Given that W satisfies the backdoor criterion, we can write

$$\sum_w P(y \mid do(t), w) P(w \mid do(t)) = \sum_w P(y \mid t, w) P(w \mid do(t))$$

  This follows from the modularity assumption.

- It can't be through any path that has an edge into T because T doesn't have any incoming edges in the manipulated graph. Thus, $P(w \mid do(t)) = P(w)$

$$\sum_w P(y \mid t, w) P(w \mid do(t)) = \sum_w P(y \mid t, w) P(w)$$

- *Relation to d-separation*. We can use the backdoor adjustment if W d-separates T from Y in the manipulated graph.

Backdoor Adjustment and Potential Outcomes I

- Recall Theorem 1.1

$$\mathbb{E}[Y(1) - Y(0)] = \mathbb{E}_X[\mathbb{E}[Y \mid T = 1, X] - \mathbb{E}[Y \mid T = 0, X]]$$

- We can derive this from the more general backdoor adjustment in a few steps.
- First, we take an expectation over Y:

$$\begin{aligned}\mathbb{E}[Y \mid do(t)] &= \sum_y yP(y \mid do(t)) \\ &= \sum_y \sum_w yP(y \mid t, w)P(w) \\ &= \sum_w \mathbb{E}[Y \mid t, w]P(w) \\ &= \mathbb{E}_W \mathbb{E}[Y \mid t, W]\end{aligned}$$

Backdoor Adjustment and Potential Outcomes II

- Then we look at the difference between $T = 0$ and $T = 1$:

$$\mathbb{E}[Y \mid do(1)] - \mathbb{E}[Y \mid do(0)] = \mathbb{E}_W[\mathbb{E}[Y \mid T = 1, W] - \mathbb{E}[Y \mid T = 0, W]]$$

The *do*-notation $\mathbb{E}[Y \mid do(t)]$ is just another notation for the potential outcomes $\mathbb{E}[Y(t)]$.

- Recall the conditional exchangeability (Assumption 1.2)

$$(Y(1), Y(0)) \perp T \mid W$$

However, we had no way of knowing how to choose W. Using graphical causal models, we know how to choose a valid W: we simply choose W so that it satisfies the backdoor criterion.

Structural Equations I

- The equals sign in mathematics does not convey any causal information.
- We need something *asymmetric*.
- A is a cause of B, meaning that changing A results in changes in B, but changing B does not result in changes in A.
- Then we can write the following *structural equation*:

$$B := f(A)$$

where f is some function that maps A to B. The mapping between A and B is deterministic.

- Ideally, we'd like to allow it to be probabilistic, which allows room for some unknown causes of B that factor into this mapping.

$$B := f(A, U)$$

where U is some unobserved random variable.

Structural Equations II

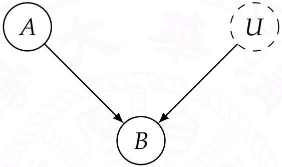


Fig. 3.4: Graph for simple structural equation. The dashed node U means that U is unobserved.

- The unobserved U is analogous to the randomness that we would see by sampling units (individuals).
- There are analogs to every part of the potential outcome $Y_i(t)$: B is the analog of Y, $A = a$ is the analog of $T = t$, and U is the analog of i.
- Although the mapping is deterministic, because it takes a random variable U as input, it can represent any stochastic mapping, so structural equations generalize the probabilistic factors $P(x_i \mid \text{pa}_i)$.

Structural Causal Model I

- A causal mechanism that generates a variable is the structural equation that corresponds to that variable.
- We write structural equations for Figure 3.5 below:

$$M : \quad \begin{aligned} B &:= f_B(A, U_B) \\ C &:= f_C(A, B, U_C) \\ D &:= f_D(A, C, U_D) \end{aligned} \tag{3.2}$$

- The variables that we write structural equations for are known as *endogenous* variables.
- In contrast, *exogenous* variables are variables who do not have any parents in the causal graph

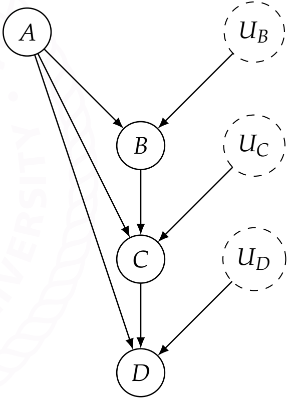


Fig. 3.5: Graph for the structural equations in Equation (3.2).

Structural Causal Model II

Definition 3.2 (Structural Causal Model (SCM))

A structural causal model is a tuple of the following sets:

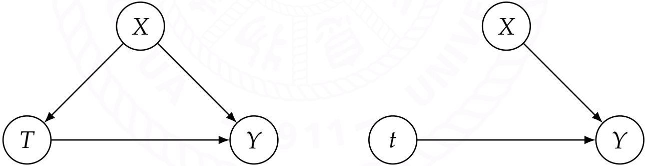
1. *A set of endogenous variables V.*
2. *A set of exogenous variables U.*
3. *A set of functions f, one to generate each endogenous variable as a function of other variables.*

- If the causal graph contains no cycles (is a DAG) and the noise variables U are independent, then the causal model is *Markovian*.
- If the causal graph doesnt contain cycles but the noise terms are dependent, then the model is *semi-Markovian*.
- The graphs of *non-Markovian* models contain cycles.

Interventions I

- The intervention $do(T = t)$ simply corresponds to replacing the structural equation for T with $T := t$.
- For example, consider the following causal model M with corresponding causal graph in Figure 3.6.

$$M : \quad \begin{aligned} T &:= f_T(X, U_T) \\ Y &:= f_Y(X, T, U_Y) \end{aligned}$$



(a) Basic causal graph. (b) $do(T = t)$.

Fig. 3.6: Causal graph and manipulated graph.

Interventions II

- If we then intervene on T to set it to t, we get the *interventional SCM* M_t

$$M: \quad \begin{aligned} T &:= t \\ Y &:= f_Y(X, T, U_Y) \end{aligned}$$

- The fact that $do(T = t)$ only changes the equation for T and no other variables is a consequence of the modularity assumption.

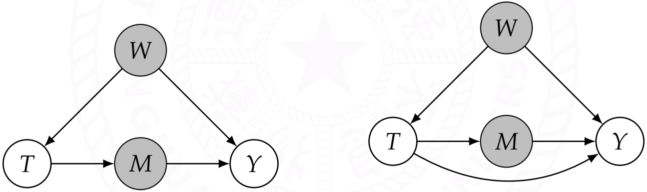
Assumption 3.2 (Modularity Assumption for SCMs)

Consider an SCM M and an interventional SCM M_t that we get by performing the intervention $do(T = t)$. The modularity assumption states that M and M_t share all of their structural equations except the structural equation for T , which is $T := t$ in M_t.

Collider Bias I

- In Theorem 3.1, we specify that W dose not contain any descendants of T.
- There are two categories of things that could go wrong if we condition on descendants of T.

Case 1: We block the flow of causation from T to Y.



(a) Causal graph where all causation is blocked by conditioning on M.

(b) Causal graph where part of the causation is blocked by conditioning on M.

Fig. 3.7: M blocks the flow of causation from T to Y.

Collider Bias II

- If we condition on a node that is on a directed path from T to Y, then we block the flow of causation along that causal path (Figure 3.7 (a)).
- We will refer to a node on a directed path from T to Y as a *mediator*, as it mediates the effect of T on Y.
- In Figure 3.7 (b), only a portion of the causal flow is blocked by M. This is because causation can still flow along the $T \to Y$ edge.
- In this case, we will get a non-zero estimate of the causal effect, but it will still be *biased*, due to the causal ow that M blocks.

Collider Bias III

Case 2: We induce non-causal association between T and Y.

- If we condition on a descendant of T that isn't a mediator, it could unblock a path from T to Y that was blocked by a collider.
- In Figure 3.8, conditioning on Z , or any descendant of Z in a path like this, will induce *collider bias*.

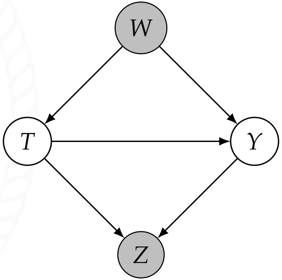
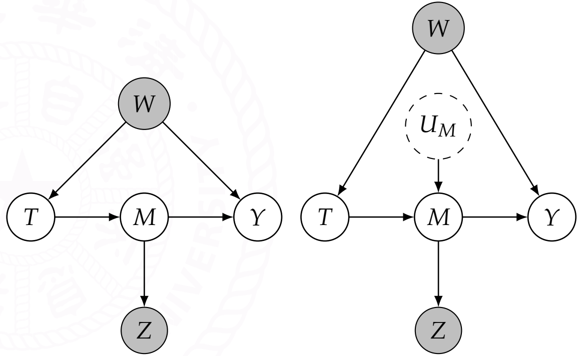


Fig. 3.8: Causal graph where conditioning on the collider Z induces bias..

Collider Bias IV

- Conditioning on Z in Figure 3.9 (a)?
- Graphs are frequently drawn without explicitly drawing the noise variables.
- Making M's noise variable explicit, we get Figure 3.9 (b).
- $T \to M \leftarrow U_M$ forms an immorality.
- There is now induced association flowing between T and U_M through the edge $T \to M$.
- Two types of association getting tangled up along the $T \to M$ edge, making the observed association between T and Y not purely causal.



(a) Causal graph where the child of a mediator is conditioned on.

(b) Magnified causal graph where the child of a mediator is conditioned on.

Fig. 3.9: M blocks the flow of causation from T to Y.

Collider Bias V

- Note that we actually can condition on some descendants of T without inducing non-causal associations between T and Y.
- However, this can get a bit tricky, so it is safest to just not condition on any descendants of T, as the backdoor criterion prescribes.
- This rule is usually described as not conditioning on any *post-treatment covariates*.

M-Bias.

- Unfortunately, even if we only condition on pretreatment covariates, we can still induce collider bias.
- Conditioning on the collider Z_2 in Figure 3.10 will open up a backdoor path, along which non-causal association can flow.
- This is known as *M-bias*.

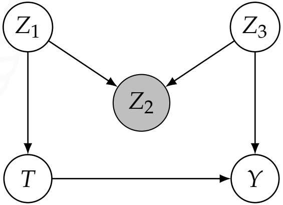


Fig. 3.10: Causal graph depicting M-bias.

Contents

Potential Outcome Framework

Causal Graph

Causal Models

Causal Supervised Learning

- Invariant Feature Learning
  - Deconfound Data
  - Deconfound Intermediate Representations
  - Deconfound Models during Training
- Invariant Mechanism Learning

Causal Generative Modeling

- Structural Assignment Learning
- Causal Disentanglement

Causal Explanations

- Feature Attributions
- Contrastive Explanations

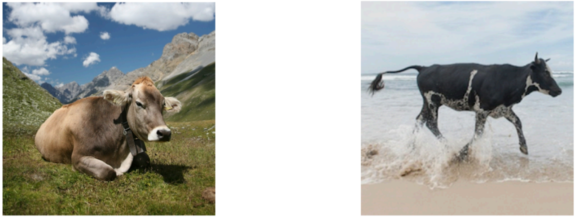
Causal Fariness

Causal Reinforcement Learning

Causal Supervised Learning I

- One of the most fundamental principles in supervised learning is to assume that our data $\mathcal{D}$ is *independent and identically distributed* (i.i.d.).
- It implies that unseen inputs occurring when the model is in production follow the same distribution as the training set.
- As an alternative to the i.i.d. assumption, we can assume that our data is sampled from interventional distributions governed by an SCM.
- For a given dataset generated across a set of environments $\mathcal{E}$, $\left\{\left(x_i^e, y_i^e\right)_{i=1}^N\right\}_{e\in\mathcal{E}}$, we view each environment $e \in \mathcal{E}$ as being sampled from a separate interventional distribution.
- In this section, we will discuss two classes of methods that aim to learn domain-robust, transferable *features* or *mechanisms*Invariant Feature Learning and Invariant Mechanism Learning.

Causal Supervised Learning II



(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

Fig. 4.1: Cows in ’common’ contexts (e.g., Alpine pastures) are detected and classified correctly (A), while cows in uncommon contexts (beach, waves, and boat) are not detected (B).

Invariant Feature Learning I

- Invariant feature learning (IFL) is the task of identifying features of our data X that are predictive of Y across a range of environments $\mathcal{E}$.
- From a causal perspective, the causal parents $\mathrm{pa}(Y)$ are always predictive of Y under any interventional distribution except where Y itself has been intervened upon.
- We can abstract a complex SCM into a simple SCM by collecting the causal parents of Y into one variable, while the other variables are collected into another.
- The most general abstraction is the *Style and Content Decomposition* (SCD).

Invariant Feature Learning II

Definition 4.1 (Style and Content Decomposition)

The style and content decomposition (SCD) is a causal graph of a data generating process (DGP) for X and Y . We call S the style variables and C the content variables, where both are assumed to be latent. The content variables group all of the causal parents of Y, $\mathrm{pa}(Y)$, while the style variables group the rest of the variables. The generations of X and Y follow the distributions

$$X \sim p(x \mid s, c), \quad Y \sim p(y \mid c)$$

Definition 4.2 (Invariant Feature Learning)

Invariant Feature Learning (IFL) aims to identify the content features C that cause both X and Y , and a mapping $p(y \mid c)$, such that

$$C = \Phi(X), \quad \text{s.t.} \quad Y \sim p(y \mid c)$$

Invariant Feature Learning III

We will introduce IFL through the following aspects

- Deconfound data (data augmentation) [link]
- Deconfound intermediate representations [link]
- Deconfound models during training [link]

We select one representative work for each category.

Deconfound Data

- *T*itle: Explaining the Efficiency of *Counterfactually Augmentation Data*
- *A*uthor: Divyansh Kaushik, Amrith Setlur, Eduard Hovy, Zachary C. Lipton (CMU)
- *P*ublished: International Conference on Learning Representations, ICLR 2021
- Counterfactually Augmentation Data (CAD): obtained via a human-in-the-loop process in which given some documents and their (initial) labels, humans must revise the text to make a *counterfactual label* applicable.
- Models trained on the augmented (original and revised) data appear, empirically, to rely less on semantically irrelevant words and to generalize better out of domain.
- Provide some insights that help to explain the efficacy of CAD.

Introduction I

- Recently in NLP, Kaushik et al. (2020) proposed *Counterfactually Augmented Data (CAD)*, injecting causal thinking into real world settings by leveraging human-in-the-loop feedback.
- Human editors are presented with document-label pairs and tasked with editing documents to render counterfactual labels applicable.
- The instructions restrict editors to only make modifications that are *necessary* to flip the label's applicability.
- The process can be viewed as the identification of *casually* relevant features (versus *spurious* features).
- Models trained on CAD enjoyed *out-of-domain* performance benefits.

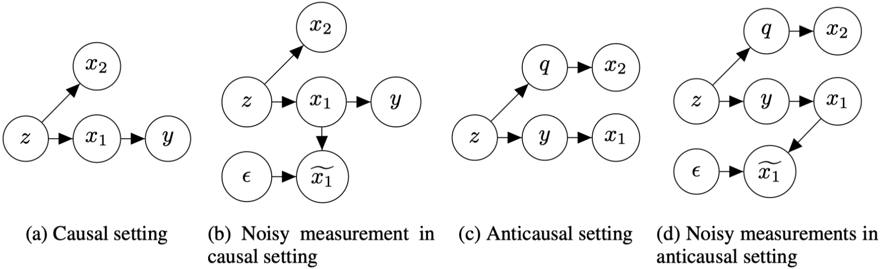
Research Questions:

1. What is the assumed causal structure underlying settings where CAD might be effective?
2. What are the principles underlying its out-of-domain benefits?

Introduction II

3. Must humans really intervene, or could automatic feature attribution methods, e.g., attention, or cheaper feedback mechanisms, e.g., feature feedback, produce similar results?

Consider linear Guassian model: causal setting and anti-causal setting (Figure 4.2).



(a) Causal setting (b) Noisy measurement in causal setting (c) Anticausal setting (d) Noisy measurements in anticausal setting

Fig. 4.2: Toy causal models with one hidden confounder. In (a) and (c), the observed covariates are x_1, x_2. In (b) and (d), the observed covariates are $\widetilde{x}_1, x_2$. In all cases, y denotes the label.

Causal Setting I

Consider the linear model

$$Y = X\beta + \epsilon$$

where $Y \in \mathbb{R}^n, X \in \mathbb{R}^{n\times p}, \beta \in \mathbb{R}^p$, and $\epsilon \in \mathcal{N}\left(0, \sigma_\epsilon^2 \mathbf{I}_n\right)$ an i.i.d. noise term. The OLS estimate β^{ols} is given by

$$\beta^{ols} = \frac{\text{Cov}(X,Y)}{\text{Cov}(X,X)}$$

If we observe only two covariates $(p = 2)$, then:

$$\beta_1^{ols} = \frac{\sigma_{x_2}^2 \sigma_{x_1,y} - \sigma_{x_1,x_2}\sigma_{x_2,y}}{\sigma_{x_1}^2\sigma_{x_2}^2 - \sigma_{x_1,x_2}^2}, \qquad \beta_2^{\text{ols}} = \frac{\sigma_{x_1}^2 \sigma_{x_2,y} - \sigma_{x_1,x_2}\sigma_{x_1,y}}{\sigma_{x_1}^2\sigma_{x_2}^2 - \sigma_{x_1,x_2}^2} \tag{4.1}$$

Causal Setting II

- We know focus on the casual setting (Figure 4.2(a), (b)).

$$\begin{aligned} z &= u_z, & u_z &\sim \mathcal{N}\left(0, \sigma^2_{u_z}\right) \\ x_1 &= bz + u_{x_1}, & u_{x_1} &\sim \mathcal{N}\left(0, \sigma^2_{u_{x_1}}\right) \\ x_2 &= cz + u_{x_2}, & u_{x_2} &\sim \mathcal{N}\left(0, \sigma^2_{u_{x_2}}\right) \\ y &= ax_1 + u_y, & u_y &\sim \mathcal{N}\left(0, \sigma^2_{u_y}\right). \end{aligned}$$

  Applying OLS, we obtain $\beta_1^{ols} = a, \beta_2^{ols} = 0$.

- If we only observe x_1 via a noisy proxy $\widetilde{x}_1 \sim \mathcal{N}\left(x_1, \sigma^2_{u_{x_1}} + \sigma^2_{\epsilon_{x_1}}\right)$ (Figure 4.2(b)).

Causal Setting III

- Assuming $\epsilon_{x_1} \perp (x_1, x_2, y)$, from Equation (4.1), we get

$$\begin{aligned}\widehat{\beta_1^{ols}} &= \frac{a\left(\sigma_{u_z}^2\left(b^2\sigma_{u_{x_2}}^2 + c^2\sigma_{u_{x_1}}^2\right) + \sigma_{u_{x_1}}^2\sigma_{u_{x_2}}^2\right)}{\sigma_{u_z}^2\left(b^2\sigma_{u_{x_2}}^2 + c^2\sigma_{u_{x_1}}^2\right) + \sigma_{u_{x_1}}^2\sigma_{u_{x_2}}^2 + \sigma_{\epsilon_{x_1}}^2\left(c^2\sigma_{u_z}^2 + \sigma_{u_{x_2}}^2\right)} \\ \widehat{\beta_2^{ols}} &= \frac{acb\sigma_{\epsilon_{x_1}}^2\sigma_{u_z}^2}{\sigma_{u_z}^2\left(b^2\sigma_{u_{x_2}}^2 + c^2\sigma_{u_{x_1}}^2\right) + \sigma_{u_{x_1}}^2\sigma_{u_{x_2}}^2 + \sigma_{\epsilon_{x_1}}^2\left(c^2\sigma_{u_z}^2 + \sigma_{u_{x_2}}^2\right)}\end{aligned} \tag{4.2}$$

- $\widehat{\beta_1^{ols}} \propto \frac{1}{\sigma_{\epsilon_{x_1}}^2}$. As $\sigma_{\epsilon_{x_1}}^2$ increases, $|\widehat{\beta_1^{ols}}|$ decreases and $|\widehat{\beta_2^{ols}}|$ increases.
- $\lim_{\sigma_{\epsilon_{x_1}}^2 \to \infty} \widehat{\beta_1^{ols}} = 0$, whereas $\widehat{\beta_2^{ols}}$ converges to a finite non-zero value.
- Only observing a noisy version of x_2 will not affect our OLS estimates.
- Under perfect measurement, the causal variable d-separates the non-causal variable from the label.

Causal Setting IV

- Under observation noise, a predictor will rely on the non-causal variable (Equation (4.2)).
- Moreover, when the causal feature x_1 is noisily observed, additional observation noise on non-causal features x_2 yields models that are more reliant on causal features. (Cannot find evidence in the paper)
- In a qualitative sense, the process of generating CAD is the intervention on the casual features.
- For each example, we produce two sets of values of x_1, one such that the label is applicable and one such that it is not applicable. One is given in the dataset, and the other is produced via the revision.

Anticausal Setting I

- Now consider the anticausal setting (Figure 4.2(c), (d)).

$$
\begin{aligned}
z &= u_z, & u_z &\sim \mathcal{N}\left(0, \sigma_{u_z}^2\right) \\
q &= az + u_q, & u_q &\sim \mathcal{N}\left(0, \sigma_{u_q}^2\right) \\
y &= bz + u_y, & u_y &\sim \mathcal{N}\left(0, \sigma_{u_y}^2\right) \\
x_2 &= cq + u_{x_2}, & u_{x_1} &\sim \mathcal{N}\left(0, \sigma_{u_{x_1}}^2\right) \\
x_1 &= dy + u_{x_1}, & u_{x_2} &\sim \mathcal{N}\left(0, \sigma_{u_{x_2}}^2\right)
\end{aligned}
$$

Anticausal Setting II

If we were to solve the linear regression problem $y = x_1\beta_1 + x_2\beta_2 + \beta_0$, we get

$$\begin{aligned}
\beta_1^{ols} &= \frac{d\left(a^2c^2\sigma_{u_z}^2\sigma_{u_y}^2 + \left(c^2\sigma_{u_q}^2 + \sigma_{u_{x2}}^2\right)\left(b^2\sigma_{u_z}^2 + \sigma_{u_y}^2\right)\right)}{\left(d^2b^2\sigma_{u_z}^2 + \sigma_{u_{x1}}^2 + d^2\sigma_{u_y}^2\right)\left(\sigma_{u_{x2}}^2 + c^2\sigma_{u_q}^2\right) + \left(\sigma_{u_{x1}}^2 + d^2\sigma_{u_y}^2\right)c^2a^2\sigma_{u_z}^2} \\
\beta_2^{ols} &= \frac{abc\sigma_{u_z}^2\sigma_{u_{x1}}^2}{\left(d^2b^2\sigma_{u_z}^2 + \sigma_{u_{x1}}^2 + d^2\sigma_{u_y}^2\right)\left(\sigma_{u_{x2}}^2 + c^2\sigma_{u_q}^2\right) + \left(\sigma_{u_{x1}}^2 + d^2\sigma_{u_y}^2\right)c^2a^2\sigma_{u_z}^2}
\end{aligned} \tag{4.3}$$

- Similarly, we observe a noisy version $\widetilde{x_1}$:

$$\widetilde{x_1} = x_1 + \epsilon_{x_1}, \quad \epsilon_{x_1} \sim \mathcal{N}\left(0, \sigma_{\epsilon_{x_1}}^2\right), \quad \epsilon \perp x_2, y$$

Then we need to replace $\sigma_{u_{x_1}}^2$ with $\sigma_{u_{\widetilde{x_1}}}^2$ in Equation (4.3):

$$\sigma_{u_{\widetilde{x_1}}}^2 = \sigma_{u_{x_1}}^2 + \sigma_{\epsilon_{x_1}}^2$$

Anticausal Setting III

- Finally we get

$$\widehat{\beta_1^{ols}} = \frac{\beta_1^{ols}}{1+\lambda_{ac}^{x_1}} \quad \widehat{\beta_2^{ols}} = \frac{\beta_2^{ols}}{1+\lambda_{ac}^{x_1}}\left[1+\frac{\sigma_{\epsilon_{x_1}}^2}{\sigma_{u_{x_1}}^2}\right] \tag{4.4}$$

$$\lambda_{ac}^{x_1} = \frac{\sigma_{\epsilon_{x_1}}^2\left(c^2a^2\sigma_{u_z}^2 + c^2\sigma_{u_q}^2 + \sigma_{u_{x_2}}^2\right)}{\left(d^2b^2\sigma_{u_z}^2 + \sigma_{u_{x_1}}^2 + d^2\sigma_{u_y}^2\right)\left(\sigma_{u_{x_2}}^2 + c^2\sigma_{u_q}^2\right) + \left(\sigma_{u_{x_1}}^2 + d^2\sigma_{u_y}^2\right)c^2a^2\sigma_{u_z}^2} \tag{4.5}$$

As $\sigma_{\epsilon_{x_1}}^2$ increases, $|\widehat{\beta_1^{ols}}|$ decreases and $|\widehat{\beta_2^{ols}}|$ increases.

- If we observe a noisy version of x_2, we find that as $\sigma_{\epsilon_{x_2}}^2$ increases, $|\widehat{\beta_1^{ols}}|$ increases and $|\widehat{\beta_2^{ols}}|$ decreases.
- As observation noise on the non-causal feature x_2 increases, we expect the learned predictor to rely more on the causal feature.

Anticausal Setting IV

- In this interpretation, we think of CAD as a process by which we (the designers of the experiment) *intervene on the label* itself and the human editors, play the role of a simulator that we imagine to be capable of generating a counterfactual example, holding all other latent variables constant.
- Note that by intervening on the label, we d-separate it from the spurious correlate x_2.

Experiment Setting I

Hypotheses

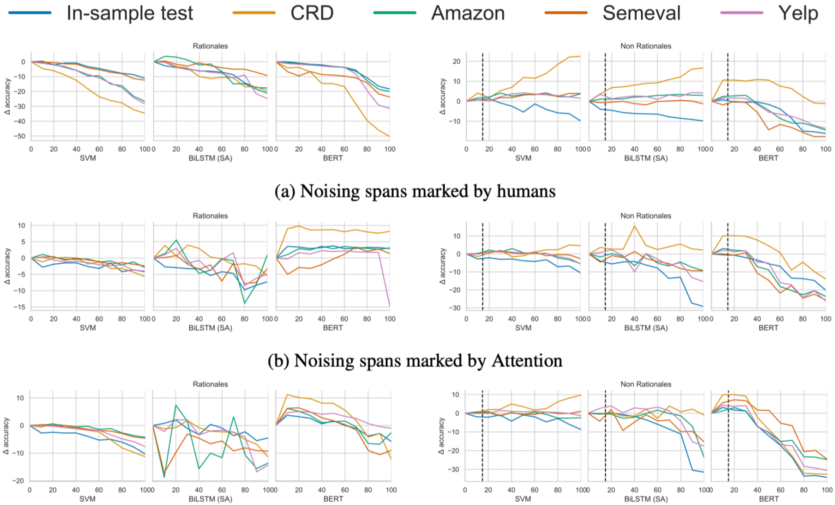
1. If spans edited to generate counterfactually revised data (CRD) are analogous to the causal (or anticausal) variables, then noising those spans (e.g. by random word replacement) should lead to models that *rely more on noncausal features* and perform worse on out of domain data.
2. Noising unedited spans should have the opposite behavior, leading to degraded in-domain performance, but comparatively *better out-of-domain performance*.
3. Whether the feedback from human workers is yielding anything qualitatively different from what might be seen with spans marked by *automated feature attribution methods* such as attention and saliency.
4. Is CAD better than automatic sentiment flipping methods (e.g., text style transfer algorithm)?

Settings

- Tasks: sentiment analysis and NLI.

Experiment Setting II

- All datasets are accompanied with human feedback (tokens deemed relevant to the label's applicability) which we refer to as *rationales*.
- In each document, we replace a fraction of rationale (or non-rationale) tokens with random tokens sampled from the vocabulary.
- In the first set of experiments, we inject noise into rationales and non-rationales marked by human and automated feature attribution methods.
- In the second set of experiments, we train models on original, CAD, and original & sentiment flipped reviews, which are produced by text style transfer methods.



(a) Noising spans marked by humans

(b) Noising spans marked by Attention

(c) Noising spans marked via gradient based feature attribution

Fig. 4.3: Change in classifier accuracy as noise is injected on rationales/non-rationales for IMDb reviews from Kaushik et al. (2020). The vertical dashed line indicates the fraction of median length of non-rationales equal to the median length of rationales.

Experiment Results I

In Figure 4.3,

- All classifiers are trained on the original 1.7*k* IMDb reviews from Kaushik et al. (2020).
- In-sample test: models are tested on the IMDb test set.
- CRD: models are tested on counterfactually revised data.

Figure 4.3 (a).

- The SVM classifier experiences a drop of ≈ 11% by the time all rationale tokens are replaced with noise. However, it experiences an 28.7% drop in accuracy on Yelp reviews.
- However, as more *non-rationales* are replaced with noise, in-sample accuracy for SVM goes down by ≈ 10% but *increases* by ≈ 1.5% on Yelp.
- For BERT, in-sample accuracy decreases by only 16.1% and only 13.6% on Yelp.

Experiment Results II

Figure 4.3 (b) and (c).

- We obtain different results using rationales identified via feature feedback and gradient based feature attribution.
- While we might not expect spurious signals to be as reliable out of domain, that does not mean that they will always fail.
- In such settings, even though noising non-causal features would lead to models relying more on causal features, this may not result in better out-of-domain performance.

Experiment Results III

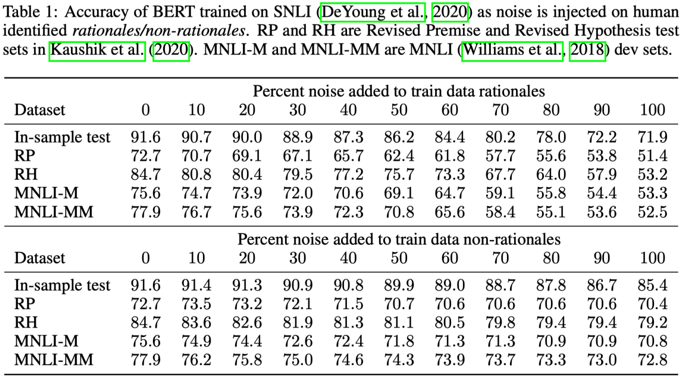


Table 1: Accuracy of BERT trained on SNLI (DeYoung et al., 2020) as noise is injected on human identified *rationales/non-rationales*. RP and RH are Revised Premise and Revised Hypothesis test sets in Kaushik et al. (2020). MNLI-M and MNLI-MM are MNLI (Williams et al., 2018) dev sets.

	Percent noise added to train data rationales										
Dataset	0	10	20	30	40	50	60	70	80	90	100
In-sample test	91.6	90.7	90.0	88.9	87.3	86.2	84.4	80.2	78.0	72.2	71.9
RP	72.7	70.7	69.1	67.1	65.7	62.4	61.8	57.7	55.6	53.8	51.4
RH	84.7	80.8	80.4	79.5	77.2	75.7	73.3	67.7	64.0	57.9	53.2
MNLI-M	75.6	74.7	73.9	72.0	70.6	69.1	64.7	59.1	55.8	54.4	53.3
MNLI-MM	77.9	76.7	75.6	73.9	72.3	70.8	65.6	58.4	55.1	53.6	52.5
	Percent noise added to train data non-rationales										
Dataset	0	10	20	30	40	50	60	70	80	90	100
In-sample test	91.6	91.4	91.3	90.9	90.8	89.9	89.0	88.7	87.8	86.7	85.4
RP	72.7	73.5	73.2	72.1	71.5	70.7	70.6	70.6	70.6	70.6	70.4
RH	84.7	83.6	82.6	81.9	81.3	81.1	80.5	79.8	79.4	79.4	79.2
MNLI-M	75.6	74.9	74.4	72.6	72.4	71.8	71.3	71.3	70.9	70.9	70.8
MNLI-MM	77.9	76.2	75.8	75.0	74.6	74.3	73.9	73.7	73.3	73.0	72.8

We can observe similar patterns in NLI tasks. But, for various models the drops in both in-sample and out-of-domain accuracy are greater in magnitude when noise is injected in rationales versus when it is injected in non-rationales. This is opposite to what we observe in sentiment analysis.

Experiment Results IV

- We use SOTA transfer methods to convert *Positive* reviews into *Negative* and vice versa.
- Ideally, we would expect these methods to preserve a document's content while modifying the attributes that relate to sentiment.
- Sentiment classifiers trained on original and *sentiment-flipped reviews* often give better out-of-domain performance.
- However, models trained on CAD perform even better across all datasets, hinting at the value of human feedback.

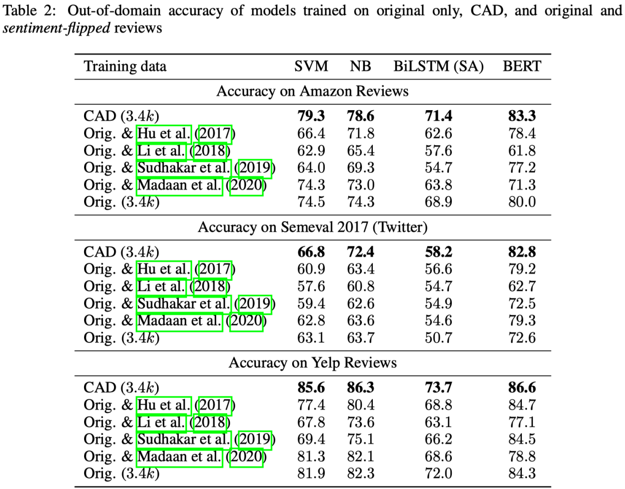


Table 2: Out-of-domain accuracy of models trained on original only, CAD, and original and *sentiment-flipped* reviews

Training data	SVM	NB	BiLSTM (SA)	BERT
Accuracy on Amazon Reviews				
CAD (3.4k)	**79.3**	**78.6**	**71.4**	**83.3**
Orig. & Hu et al. (2017)	66.4	71.8	62.6	78.4
Orig. & Li et al. (2018)	62.9	65.4	57.6	61.8
Orig. & Sudhakar et al. (2019)	64.0	69.3	54.7	77.2
Orig. & Madaan et al. (2020)	74.3	73.0	63.8	71.3
Orig. (3.4k)	74.5	74.3	68.9	80.0
Accuracy on Semeval 2017 (Twitter)				
CAD (3.4k)	**66.8**	**72.4**	**58.2**	**82.8**
Orig. & Hu et al. (2017)	60.9	63.4	56.6	79.2
Orig. & Li et al. (2018)	57.6	60.8	54.7	62.7
Orig. & Sudhakar et al. (2019)	59.4	62.6	54.9	72.5
Orig. & Madaan et al. (2020)	62.8	63.6	54.6	79.3
Orig. (3.4k)	63.1	63.7	50.7	72.6
Accuracy on Yelp Reviews				
CAD (3.4k)	**85.6**	**86.3**	**73.7**	**86.6**
Orig. & Hu et al. (2017)	77.4	80.4	68.8	84.7
Orig. & Li et al. (2018)	67.8	73.6	63.1	77.1
Orig. & Sudhakar et al. (2019)	69.4	75.1	66.2	84.5
Orig. & Madaan et al. (2020)	81.3	82.1	68.6	78.8
Orig. (3.4k)	81.9	82.3	72.0	84.3

Conclusion

- Simple analysis on toy linear Gaussian models + a large-scale empirical investigation on sentiment analysis and NLI tasks → to understand the efficacy of CAD.
- Data corrupted by adding noise to *rationale* spans (analogous to adding noise to causal features) will degrade out-of-domain performance, while noise added to non-causal features *may* make models more robust out-of-domain.
- Models trained on the augmentation of original data and revised data generated by *style transfer methods* had better out-of-domain generalization in some cases compared to models trained on original data alone, but performed worse than models trained on CAD.

Deconfound Intermediate Representations

- Title: Causal Transportability for Visual Recognition
- Author: Chengzhi Mao[1], Kevin Xia[1], James Wang[1], Hao Wang[2], Junfeng Yang[1], Elias Bareinboim[1], Carl Vondrick[1] ([1]Columbia University, [2]Rutgers University)
- Published: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022.
- Visual representations often contain *robust* and *non-robust* features.
- Image classifiers may perform poorly on *out-of-distribution* samples because spurious correlations between non-robust features and labels can be changed in a new *environment*.
- Standard classifiers fail because the association between images and labels is not transportable across settings.
- The causal effect, which severs all sources of confounding, remains invariant across domains.

Introduction

- In this paper, we investigate visual representations for object recognition through the lenses of causality.
- First, we will show that the association between image and label is not in generalizable (in causal language, *transportable*) across domains.
- We then note that the *causal* effect from the input to the output, which severs any spurious correlations, is invariant when the environment changes with respect to the features' distributions.
- Getting the causal effect for natural images is challenging because there are *innumerable unobserved confounding factors* within realistic data.
- Under some relatively mild assumptions, we will be able to extract the robust features from observational data through both causal and deep representations, and then use the representations as *proxies* for identifying the causal effect without requiring observations of the confounding factors.

Probelm Formulation – Structural Modeling I

- X, Y: random variables related to images and labels. x, y: the specific instantiations of the pixels and label.
- Consider a structural causal model (SCM) M that encodes a 4-tuple

$$\langle V = \{X, Y\}, U = \{U_X, U_{XY}\}, \mathcal{F} = \{f_X, f_Y\}, P(U)\rangle$$

- V is the set of observed variables (the image X and its label Y).
- U represents unobserved variables encoding external sources of variation not captured in the image and the label themselves.
- $\mathcal{F}$ is the set of mechanisms $\{f_X, f_Y\}$, which determine the generative processes of X and Y such that $X \leftarrow f_X(U_X, U_{XY})$ and $Y \leftarrow f_Y(X, U_{XY})$.
- $P(U)$ represents a probability distribution over the unobserved variables.

Probelm Formulation – Structural Modeling II

More explanation on U:

- U_{XY} is called *concept vector*, as it represents all underlying factors that produce both the core features of the object in image x and its label, y.
- For example, one instantiation of $U_{XY} = u_{XY}$ may encode the concepts of "flippers" and "wing", which are translated into an image of a "waterbird" when passed into f_X.
- U_X represents *nuisance factors*, such as the background, that affect the generation process of the image.
- f_Y may represent someone who is labeling image x and will have a conceptual understanding of waterbird through u_{XY}.
- Together, the underlying distribution over $P\left(U_{XY}, U_X\right)$ combined with functions f_X and f_Y induce a distribution over $P(X, Y)$, which is how data is generated.
- It is *impossible* to recover the structural functions ($\mathcal{F}$) and probability over the exogenous variables ($P(U)$) from observational data alone ($P(V)$).

Modeling In vs. Out-of-Distribution Generalization I

- Out-of-distribution case (transportability problem): training data may come from a domain π that differs from the test domain π^*.
- Assume that the *labeling process and underlying concepts* are consistent across domains (i.e., f_Y and $P(U_{XY})$ remain the same in both settings), but the *generative process* of the image X may change (i.e., f_X^* and $P^*(U_X)$ may differ from f_X and $P(U_X)$, respectively).
- In general, we do not know the true underlying mechanisms f_X, f_X^*, and f_Y, nor can we observe the immeasurably large space of $P(U_X, U_{XY})$.
- We can represent the structural invariances across domains by leveraging a graphical representation shown in Figure 4.4. The disparities across domains π and π^* are usually modeled by a *transportability node* called S, which can be interpreted as a switch across domains; i.e., f_X will be active if $S = 0$, and f_X^* otherwise.

Modeling In vs. Out-of-Distribution Generalization II

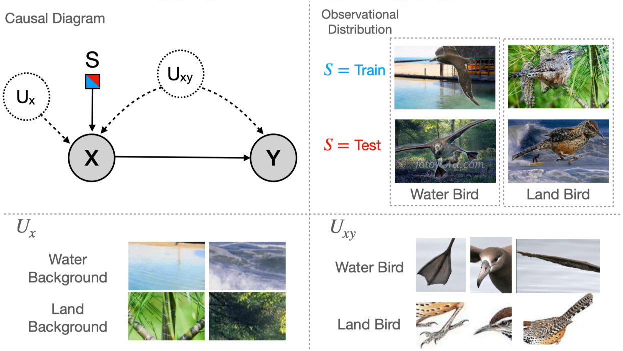


Fig. 4.4: Causal graph for out-of-distribution image classification (top left). S, the transportability node, points to nodes with changes between domains, where X combines "waterbird" with "water background" during the training $(S = 0)$ and "water bird" with "land background" at testing $(S = 1)$ (top right).

Modeling In vs. Out-of-Distribution Generalization III

- In-distribution case: to learn $P(Y \mid X)$, which leverages all possible information to maximize the chance of predicting the correct label.
- However, given the way the data generation process is modeled, it is easy to see why this same strategy fails in the out-of-distribution case.
- Since only data from domain π is given, we can only train a model on $P(Y \mid X)$, which does not adequately model $P^*(Y \mid X)$.

Proposition 4.1

Let M and M^* be the two underlying SCMs representing the source and target domains, π and π^*, and compatible with the assumptions represented in the causal graph in Figure 4.4. Then, $P^*(Y \mid X) \neq P(Y \mid X)$.

- In words, the classifier represented by the quantity $P(Y \mid X)$, in π, is not *transportable* across settings and cannot be used to make statements about $P^*(Y \mid X)$.

Modeling In vs. Out-of-Distribution Generalization IV

- By conditioning on X, the variables Y and S become *d-connected* via the path through U_{XY}, i.e., $P(Y \mid X, S=0) \neq P(Y \mid X, S=1)$.

Proposition 4.2

Let M and M^* be the two underlying SCMs representing the source and target domains, π and π^*, and compatible with the assumptions represented in the causal graph in Figure 4.4. Then, $P^*(Y \mid do(X)) = P(Y \mid do(X))$.

- $P(Y \mid do(X))$: remove all arrows towards X, including the S-node, by forcing X to take a certain value, say x.
- Regardless of the change in the mechanism of f_X^* and $P^*(U_X)$, it is guaranteed that the causal effect of X on Y will remain invariant across π and π^*. In causal language, $P^*(Y \mid do(X))$ is *transportable* across settings.

Identifiability

Proposition 4.3

Let M be the SCM representing domain π and described through the causal diagram G in Figure 4.4. The interventional distribution $P(Y \mid do(X))$ is not identifiable from G and the observational distribution $P(X, Y)$.

- Non-identifiability suggests that there are multiple SCMs that are consistent with $P(X, Y)$ and that induce different distributions $P(Y \mid do(X))$.
- Some prior work has assumed that *all back-door variables* can be observed, which means that all the variations represented originally in the unobserved confounder U_{XY} are, in some sense, captured by the model.
- In most image datasets that contain only images and their labels, the assumption that all back-door variables (and sources of co-variation) are observable is overly stringent.
- Our goal now is to identify the effect of X on Y without having knowledge of the back-door variables.

Neural Representation Approach to Derive a Causal Estimand I

Assumption 4.1 (Decomposition)

Each image X can be decomposed into causal factors Z and spurious factors W (i.e. $X = (Z, W)$), and the generative process follows the causal graph in Figure 4.5.

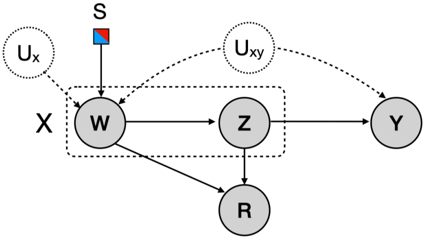


Fig. 4.5: Expanded causal model with decomposition of image X and representation R. Gray nodes denote observed variables.

Neural Representation Approach to Derive a Causal Estimand II

- We build two neural network models: $\hat{P}(R \mid X)$, which generates visual representations R from images X, and $\hat{P}(Y \mid R, X)$, which uses both R and X to classify Y.
- W contains all of the *lower level* signals or patches of the image, which may contain concepts confounding with Y.
- Z refines these patches into *interpretable* factors, which is what is visually used by the labeler. Since Z is a direct function of W, these factors are not confounded.

Assumption 4.2 (Sufficient representation)

The neural representations $R \sim \hat{P}(R \mid Z, W)$ are learned such that they do not lose information w.r.t. Z. In words, for two samples r_1 and r_2 from $\hat{P}(R \mid z_1, w_1)$ and $\hat{P}(R \mid z_2, w_2)$, respectively, $r_1 \neq r_2$ if $z_1 \neq z_2$.

- The neural representation has enough capacity to represent unambiguously the causal factors.
- This assumption should hold in general given a proper choice of model for $\hat{P}(R \mid X)$.

Neural Representation Approach to Derive a Causal Estimand III

Assumption 4.3 (Selective prediction)

Consider two images of X, $x = (z, w)$ *and* $x' = (z', w')$, *with neural output* $\hat{P}$, *and the true labeling probability* P. *Let* $R = r$ *be a representation of* x, *sampled from* $\hat{P}(R \mid x)$. *Then,* $\hat{P}(Y = y \mid R = r, X = x') = P(y \mid z, w')$.

- Once inputted with two images x and x' (x in its representation form, r), the network will make the *same* prediction y as if it were the true labeler when inputted with the causal feature z, from the first image, and the spurious feature w', from the second image.
- $\hat{P}(Y \mid R, X)$ can learn causal features from R.

Causal Identification I

Theorem 4.1 (Causal Identification)

Given the assumptions about the generative process encoded in the causal graph in Figure 4.5 together with Assumption 4.1, 4.2, 4.3, the causal effect can be computed using neural representation R via

$$P(Y = y \mid do(X = x)) = \sum_{r} \hat{P}(r \mid x) \sum_{x'} \hat{P}(y \mid r, x') P(x') \tag{4.6}$$

$$\begin{aligned}
P(y \mid do(x)) &= P(y \mid do(z, w)) && \text{(Assumption 4.1)} \\
&= P(y \mid do(z)) && \text{(Do-Calculus Rule 3)} \\
&= \sum_{w'} P\left(y \mid z, w'\right) P\left(w'\right) && \text{(Backdoor Criterion)} \\
&= \sum_{z', w'} P\left(y \mid z, w'\right) P\left(z', w'\right) && \text{(Marginalization)}
\end{aligned}$$

Causal Identification II

By Assumption 4.2 and 4.3, the last expression can be rewritten as

$$= \sum_{x'} \hat{P}(y \mid r, x' = (z', w'))P(x')$$

where r is sampled from $\hat{P}(R \mid x)$. Since Assumption 4.3 applies for any sampled value of R, we can average across samples of R,

$$= \sum_{r} \hat{P}(r \mid x) \sum_{x'} \hat{P}(y \mid r, x' = (z', w'))P(x')$$

- The intuition behind this derivation is that if the image x can be decomposed into causal factors (z) and spurious factors (w), as shown in Figure 4.5, then the causal effect is isolated in z, and w can be ignored.
- By conditioning on $W = w'$, using another image, all the backdoor paths from Z to Y are blocked, which leads to an identifiable result.

Causal Identification III

- We need to construct the neural models to satisfy the three assumptions and properly estimate $\mathrm{P}(X)$, $P(R \mid X)$, and $P(Y \mid X, R)$.
- The term $P(X)$ is straightforward to calculate because we can assume it is sampled from a uniform distribution.

Constructing $P(R \mid X)$ and $P(Y \mid R, X)$ I

Some classes of models that are valid ways of estimating $\hat{P}(R \mid X)$ while satisfying Assumption 4.2.

- Variational Auto-Encoder (VAE).
- Constrastive Learning: given enough negative examples, contrastive learning will produce representations that are invariant under data augmentation, which still maintains all causal information from the input images.
- Pretrained models from larger dataset.

To properly estimate Equation (4.6), we also need to estimate a $P(Y \mid R, X)$ such that Assumption 4.3 is satisfied.

- In addition to the representation R, we use as input *a bag of patches*, which are subsampled from input image X into the branch that takes the input X.
- A bag of image patches corrupts the global shape information and often contains local features that are spurious, such as color, texture and background.

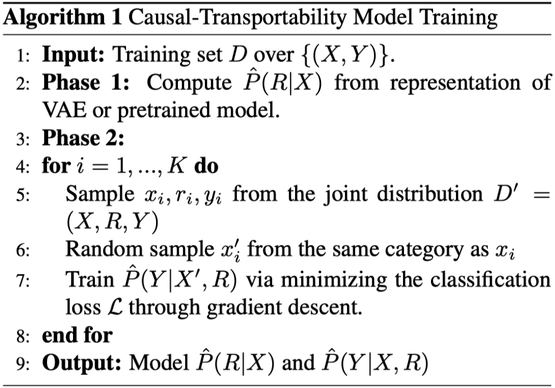
$$\hat{P}(Y \mid R \sim \hat{P}(R \mid Z, W), X = (Z, W)) = \hat{P}(Y \mid R \sim \hat{P}(R \mid Z, W), W)$$

Constructing $P(R \mid X)$ and $P(Y \mid R, X)$ II

- During training, the image X and the representation R are sampled from the same instance. During testing, the image X can be sampled from an arbitrary instance. (*Seems inconsistent with the pseudocode*)
- The model $\hat{P}(Y \mid R, X)$ has *limited capacity*. Given that the model has learned the information about W, learning W from R again will not further decrease the empirical loss. Thus, the model will learn Z from the representation R and ignore the W from the representation.
- By limiting the capacity of $\hat{P}(Y \mid R, X)$, the model tends to use *low-level* features from the input images X while using high-level deep features from the latent representation R.

Algorithm I

- Line 6: sample random images X from the same category as the representation R.

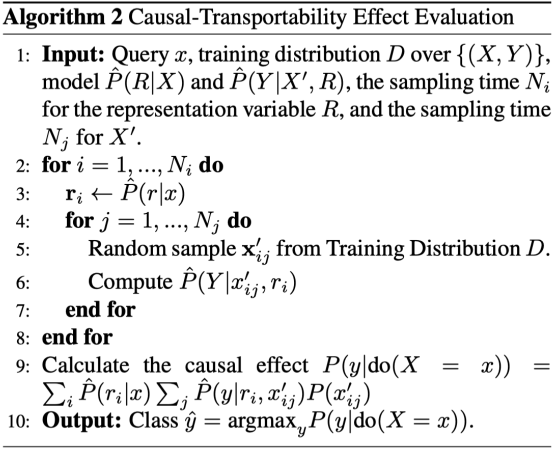


Algorithm 1 Causal-Transportability Model Training

```
1: Input: Training set $D$ over $\{(X, Y)\}$.
2: Phase 1: Compute $\hat{P}(R|X)$ from representation of VAE or pretrained model.
3: Phase 2:
4: for $i = 1, ..., K$ do
5:     Sample $x_i, r_i, y_i$ from the joint distribution $D' = (X, R, Y)$
6:     Random sample $x_i'$ from the same category as $x_i$
7:     Train $\hat{P}(Y|X', R)$ via minimizing the classification loss $\mathcal{L}$ through gradient descent.
8: end for
9: Output: Model $\hat{P}(R|X)$ and $\hat{P}(Y|X, R)$
```

Algorithm II

- First randomly sample R (Line 3).
- Then, for each R, we sample images X from random categories (Line 5).
- Make prediction through Theorem 4.1.



Algorithm 2 Causal-Transportability Effect Evaluation

```
1:  Input: Query $x$, training distribution $D$ over $\{(X, Y)\}$,
    model $\hat{P}(R|X)$ and $\hat{P}(Y|X', R)$, the sampling time $N_i$
    for the representation variable $R$, and the sampling time
    $N_j$ for $X'$.
2:  for $i = 1, \ldots, N_i$ do
3:      $\mathbf{r}_i \leftarrow \hat{P}(r|x)$
4:      for $j = 1, \ldots, N_j$ do
5:          Random sample $\mathbf{x}'_{ij}$ from Training Distribution $D$.
6:          Compute $\hat{P}(Y|x'_{ij}, r_i)$
7:      end for
8:  end for
9:  Calculate the causal effect $P(y|\mathrm{do}(X = x)) =$
    $\sum_i \hat{P}(r_i|x) \sum_j \hat{P}(y|r_i, x'_{ij}) P(x'_{ij})$
10: Output: Class $\hat{y} = \mathrm{argmax}_y P(y|\mathrm{do}(X = x))$.
```

Experiment Settings I

Datasets

- **CMNIST**: combine digits with different background colors from the training domain, creating an out-of-distribution (OOD) dataset.
- **WaterBird**: contains two classes of foreground birds, the waterbird and the landbird, and two types of backgrounds: water and land.
- **ImageNet-Rendition**: has renditions of 200 ImageNet classes, including art, cartoons, etc, which is an OOD test set for ImageNet.
- **ImageNet-Sketch**: contains sketch of 1000 ImageNet classes without texture and color.
- **ImageNet-9 Backgrounds Challenge**: backgrounds are adversarially chosen on ImageNet.

Baselines

- ERM: Empirical Risk Minimization.
- GenInt: learns a causal classifier by steering the generative models to simulate interventions.

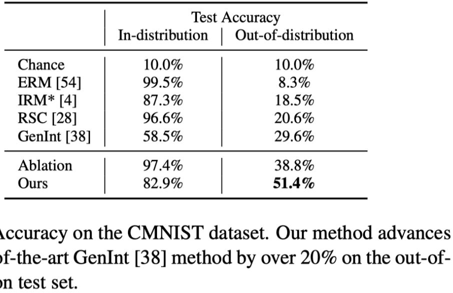
Experiment Settings II

- RSC: uses representation self-challenging to improve generation to the OOD data.
- IRM (Invariant Risk Minimization): use domain index information.

Settings

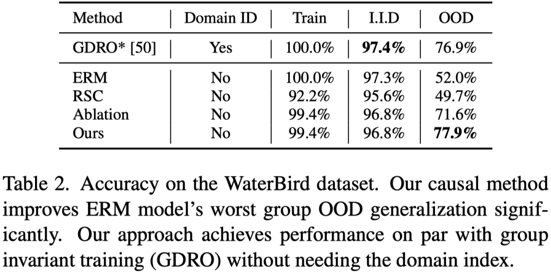
- $\hat{P}(Y \mid X, R)$: 3 convolutional layers applied to X, concatenating the obtained feature with R, and then using 2-layer fully connected network to predict Y.
- **Ours**: $N_j = 256, N_i = 10$.
- **Ablation**: $N_j = 1$ and $N_i = 1$.

Results



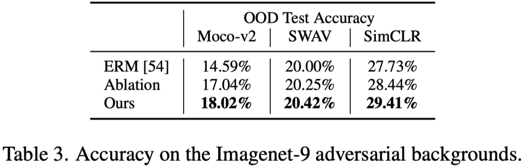
	Test Accuracy	
	In-distribution	Out-of-distribution
Chance	10.0%	10.0%
ERM [54]	99.5%	8.3%
IRM* [4]	87.3%	18.5%
RSC [28]	96.6%	20.6%
GenInt [38]	58.5%	29.6%
Ablation	97.4%	38.8%
Ours	82.9%	**51.4%**

Table 1. Accuracy on the CMNIST dataset. Our method advances the state-of-the-art GenInt [38] method by over 20% on the out-of-distribution test set.



Method	Domain ID	Train	I.I.D	OOD
GDRO* [50]	Yes	100.0%	**97.4%**	76.9%
ERM	No	100.0%	97.3%	52.0%
RSC	No	92.2%	95.6%	49.7%
Ablation	No	99.4%	96.8%	71.6%
Ours	No	99.4%	96.8%	**77.9%**

Table 2. Accuracy on the WaterBird dataset. Our causal method improves ERM model's worst group OOD generalization significantly. Our approach achieves performance on par with group invariant training (GDRO) without needing the domain index.

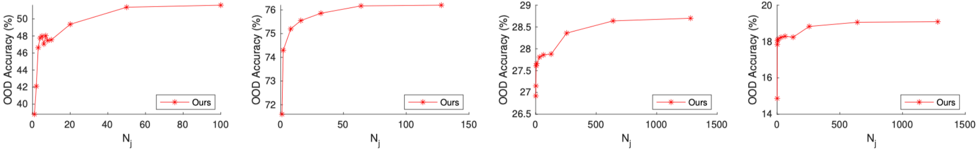


	OOD Test Accuracy		
	Moco-v2	SWAV	SimCLR
ERM [54]	14.59%	20.00%	27.73%
Ablation	17.04%	20.25%	28.44%
Ours	**18.02%**	**20.42%**	**29.41%**

Table 3. Accuracy on the Imagenet-9 adversarial backgrounds.

Algorithm	ImageNet Rendition ERM	RSC	Ablation	Ours	ImageNet Sketch ERM	RSC	Ablation	Ours
Moco-v2	26.92%	26.14%	25.96%	**28.70%**	17.29%	16.43%	14.11%	**19.09%**
SWAV	31.77%	30.47%	30.32%	**33.32%**	21.51%	21.03%	17.26%	**22.48%**
SimCLR	37.82%	34.06%	35.74%	**38.25%**	27.43%	19.26%	24.90%	**29.51%**
ResNet50	25.02%	**33.34%**	30.96%	32.22%	14.45%	22.54%	19.19%	**22.57%**
ResNet152	30.53%	**37.86%**	34.94%	36.07%	18.53%	26.60%	24.61%	**27.07%**
ResNet101-2x	31.44%	35.50%	35.82%	**36.70%**	19.92%	26.38%	25.07%	**27.41%**

Table 4. Robust accuracy on ImageNet-Rendition and ImageNet-Sketch. For contrastive learning based representations, our model achieves improved robustness than standard ERM and the state-of-the-art RSC approach. On superivsed learning representations, the representation may fail to capture all the causal information, where RSC method out-performs ours on two variants on ImageNet Rendition. Overall, our method improves robustness by estimating the causal effect from the representation.



(a) CMNIST, $K = 10$

(b) WaterBird, $K = 2$

(c) Rendition, $K = 1000$

(d) Sketch, $K = 1000$

Figure 4. OOD generalization accuracy under different number of N_j. At inference time, by increasing N_j that samples more images X', OOD generalization improve because the spurious correlation is better removed through our approach.

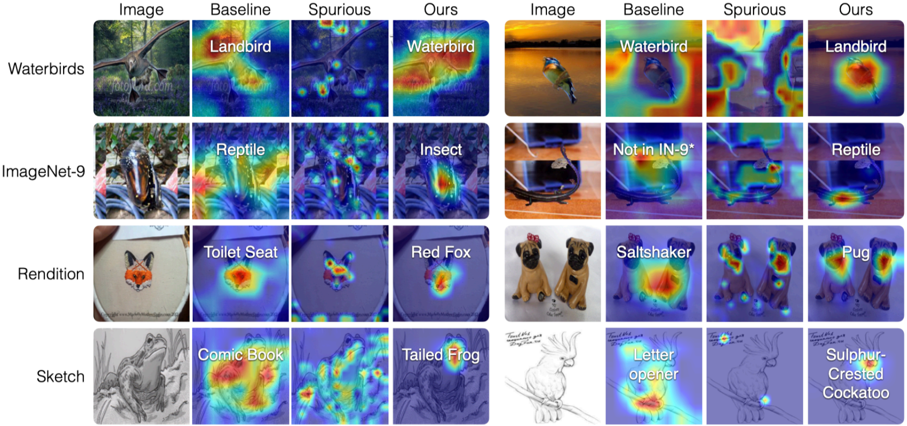


Figure 5. We visualize the input regions that the models use for prediction. We use GradCAM [51] and highlight the the discriminative regions that the model relies on with red. The white text shows the model's prediction. The correlation based ERM method often attends to spurious background context. By marginalizing over the spurious features (visualized in the Spurious column), our model captures the right, causal features, which predict the right thing for the right reason.

Spurious: the branch that conditions on the variable X of model $P(Y \mid R, X)$.

Conclusion

- This paper focuses on the visual transportability problem.
- The association between image and label is not in generalizable across domains.
- The causal effect $P(Y \mid do(X))$ is invariant when the environment changes.
- This paper develops an algorithm which uses the deep representations R as proxies for identifying the causal effect without requiring observations of the confounding factors.

Deconfound Models

- Title: *Counterfactual Invariance* to Spurious Correlations: Why and How to Pass *Stress Tests*
- Author: Victor Veitch[1,2], Alexander D'Amour[1], Steve Yadlowsky[1], and Jacob Eisenstein[1] ([1]Google Research, [2]University of Chicago)
- Published: 35th Conference on Neural Information Processing Systems, NIPS 2021
- *Stress test*: check for spurious correlations by perturbing irrelevant parts of input data and see if the model predictions change.
- *Counterfactual invariance*: a formalization of the requirement that changing irrelevant parts of the input shouldn't change model predictions.
- Provide practical schemes for learning (approximately) counterfactual invariant predictors (without access to counterfactual examples).

Introduction

- Stress test example: we might test a sentiment analysis tool by changing one proper noun for another (tasty Mexican food to tasty Indian food).
- What is the connection between passing stress tests and model performance on prediction?
- How should we develop models that pass stress tests when our ability to generate perturbed examples is limited?
- We will formalize passing stress tests as *counterfactual invariance*, a condition on how a predictor should behave when given certain (unobserved) counterfactual input data.
- We will then derive *implications* of counterfactual invariance that *can be measured in the observed data*.
- *Regularizing* predictors to satisfy these observable implications provides a means for achieving (partial) counterfactual invariance.

Counterfactual Invariance

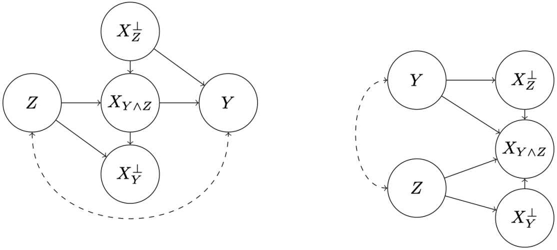
- Consider the problem of learning a predictor f that predicts a label Y from covariates X.
- Predictions of f should be invariant to certain perturbations on X.
- Assume that there is an additional variable Z that captures information that should not influence predictions. However, Z may*causally influence* the covariates X.
- Potential outcomes notation: $X(z)$ is the conterfactual X when Z is set to z, leaving all else fixed.

Definition 4.3 (Counterfactual Invariance)

A predictor f is counterfactually invariant to Z if $f(X(z)) = f(X(z'))$ almost everywhere, for all z, z' in the sample space of Z. When Z is clear from context, we'll just say the predictor is counterfactually invariant.

The *true causal structure* fundamentally affects both the implications of counterfactual invariance, and the techniques we use to achieve it.

Consider two common causal structures in this paper.



(a) Causal direction

(b) Anticausal direction

Fig. 4.6: Causal models for the data generating process. We decompose the observed covariate X into latent parts defined by their causal relationships with Z and Y. Solid arrows denote causal relationships, while dashed lines denote non-causal associations.

Causal Direction

Consider the following example where X is a cause of Y (Figure 4.6 (a)).

- Goal: classify the quality of product reviews using the text of the product review X.
- Each review has a number of helpful votes Y (from site users).
- Interventions on the sentiment Z of the text change our prediction (e.g., "good shoes!" to "bad shoes!").

Usually, the causal relationship between the text and Y and Z will be complex.

- Decompose the observed X into two parts defined by their causal relationships with Y and Z.
- $X_{Z}^{\perp}$: the part of X that is not causally influenced by Z (but may influence Y)
- $X_{Y}^{\perp}$: the part that does not causally influence Y (but may be influenced by Z)
- $X_{Y \wedge Z}$: the remaining part that is both influenced by Z and that influences Y

Anti-Causal Direction

Consider the following example where Y causes X (Figure 4.6 (b)).

- Goal: predict the star rating Y of movie reviews from the text X.
- Predictions are influenced by the movie genre Z.
- Again decompose the observed X into two parts defined by their causal relationships with Y and Z.

Z can be associated with Y through two paths.

- Conditioning on $X_{Y \wedge Z}$ causes a dependence between Z and Y (collider). For example, if Adam Sandler tends to appear in good comedy movies but bad movies of other genres then seeing Sandler in the text induces a dependency between sentiment and genre.
- Z and Y may be associated due to a common cause (the dashed line). For example, fans of romantic comedies may tend to give higher reviews (to all films) than fans of horror movies.

Non-Causal Associations

- A predictor trained to predict Y from X will rely on $X_Y^{\perp}$, even though there is *no causal connection* between Y and $X_Y^{\perp}$, and therefore will fail counterfactual invariance.
- $X_Y^{\perp}$ serves as a proxy for Z, and Z is predictive of Y due to non-causal association.
- There are two mechanisms that can induce such associations: confoundedness and selection bias.
- There is also dependency induced by between Y and Z by $X_{Y \wedge Z}$. Whether or not each of these dependencies is spurious is a *problem-specific judgement* that must be made by each analyst based on their particular use case.

Definition 4.4 (Purely Spurious)

We say that the association between Y and Z is purely spurious if $Y \perp X \mid X_Z^{\perp}, Z$.

That is, if the dashed-line association did not exist (removed by conditioning on Z) then the part of X that is not influenced by Z would suffice to estimate Y.

Observable Signatures of Counterfactually Invariant Predictors I

- Counterfactual invariance is defined by the behavior of the predictor on counterfactual data that is *never actually observed*.
- Instead, we'll derive a signature of counterfactual invariance that actually can be measured using ordinary datasets where Z (or a proxy) is measured.
- Intuitively, a predictor f is counterfactually invariant if it depends only on $X_Z^\perp$.
- The following lemma ensures that $X_Z^\perp$ is well-defined.

Lemma 4.1

Let $X_Z^\perp$ be a X-measurable random variable such that, for all measurable functions f, we have that f is counterfactually invariant if and only if $f(X)$ is $X_Z^\perp$-measurable. If Z is discrete then such a $X_Z^\perp$ exists.

Observable Signatures of Counterfactually Invariant Predictors II

Theorem 4.2 (Signatures of Counterfactual Invariance)

If f is a counterfactually invariant predictor:

1. *Under the anti-causal graph, $f(X) \perp Z \mid Y$.*
2. *Under the causal-direction graph, if Y and Z are not subject to selection (but possibly confounded), $f(X) \perp Z$.*
3. *Under the causal-direction graph, if the association is purely spurious, $Y \perp X \mid X_Z^{\perp}, Z$, and Y and Z are not confounded (but possibly selected), $f(X) \perp Z \mid Y$.*

- In the fairness setting, counterfactual invariance is equivalent to counterfactual fairness.
- $f(X) \perp Z$: demographic parity.
- $f(X) \perp Z \mid Y$: equalized odds.

Causal Regularization I

- We cannot directly enforce counterfactual invariance without access to counterfactual examples.
- However, we can require a trained model to satisfy the counterfactual invariance signature of Theorem 4.2.
- Y and Z are binary. The regularization terms are

$$\begin{aligned}
\text{marginal regularization} =& \mathrm{MMD}(\mathrm{P}(f(X) \mid Z=0), \mathrm{P}(f(X) \mid Z=1)) && (4.7)\\
\text{conditional regularization} =& \mathrm{MMD}(\mathrm{P}(f(X) \mid Z=0, Y=0), \mathrm{P}(f(X) \mid Z=1, Y=0)) && (4.8)\\
&+ \mathrm{MMD}(\mathrm{P}(f(X) \mid Z=0, Y=1), \mathrm{P}(f(X) \mid Z=1, Y=1))
\end{aligned}$$

  where MMD (Maximum Mean Discrepancy) is a metric on probability measures.
- $f(X) \perp Z$ is equal to Equation (4.7) $= 0$.
- $f(X) \perp Z \mid Y$ is equal to Equation (4.8) $= 0$.
- We can estimate the MMD with finite data samples.

Causal Regularization II

- A key point is that the regularizer we must use depends on the *true causal structure*.
- The conditional independence signature of Theorem 4.2 is necessary but not sufficient for counterfactual invariance.
- Unfortunately, the gap between the signature and counterfactual invariance is a fundamental restriction of using observational data.

Experiment

To verify the following claims

1. Stress test violations can be reduced by suitable conditional independence regularization.
2. This reduction will improve out-of-domain prediction performance.
3. To get the full effect, the imposed penalty must match the causal structure of the data.

Setting

- For each experiment, we use BERT finetuned to predict a label Y from the text as our base model.
- Marginal regularization for causal-confounded structure.
- Conditional regularization for anti-causal structure.

Robustness to Stress Test I

In this part, we build experimental datasets using Amazon reviews from the product category Clothing, Shoes, and Jewelry.

Synthetic

- To study the relationship between counterfactual invariance and the distributional signature of Theorem 4.2, we construct a synthetic confound.
- For each review, we draw a Bernoulli random Z, and then perturb the text X so that the common words ’the’ and ’a’ carry information about Z: for example, we replace ’the’ with the token ’thexxxxx’ when $Z = 1$.
- Y: the review score.
- This data has *anti-causal structure*: the text X is written to explain the score Y.
- We expect that the Y, Z association is purely spurious, because ’the’ and ’a’ carry little information about the label.
- Model is trained on $P(Y = Z) = 0.3$.

Robustness to Stress Test II

- We then create perturbed stress-test datasets by changing each example $X_i(z)$ to the counterfactual $X_i(1-z)$
- Measurement: checklist failures, which is measured by the frequency that the predicted label changes due to perturbation as well as the mean absolute difference in predictive probabilities that is induced by perturbation.

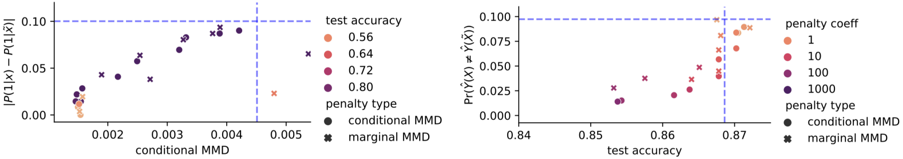


Fig. 4.7: Regularizing conditional MMD improves counterfactual invariance on synthetic anti-causal data.

Robustness to Stress Test III

From Figure 4.7, we observe

- **Left**: lower conditional MMD implies that predictive probabilities are invariant to perturbation. Although *marginal* MMD penalization can result in low conditional MMD and good stress test performance, this comes at the cost of very low in-domain accuracy.
- **Right**: MMD regularization reduces the rate of predicted label flips on perturbed data, with little affect on indomain accuracy.

Natural

- We now take Z to be the score, binarized as $Z \in \{1 \text{ or } 2 \text{ stars}, 4 \text{ or } 5 \text{ stars}\}$.
- Z is a proxy for sentiment and we consider problems where sentiments should not have a causal effect on Y.
- Y: the helpfulness score of the review.
- This data has *causal structure*: readers decide whether the review is helpful based on the text.

Robustness to Stress Test IV

- For the anti-causal problem, we take Y to be whether 'Clothing' is included as a category tag for the product under review. This is *anti-causal* because the product category affects the text.
- Stress test: randomly changing adjectives in the examples (positive → negative).

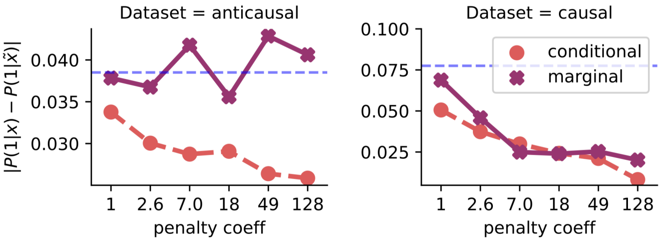


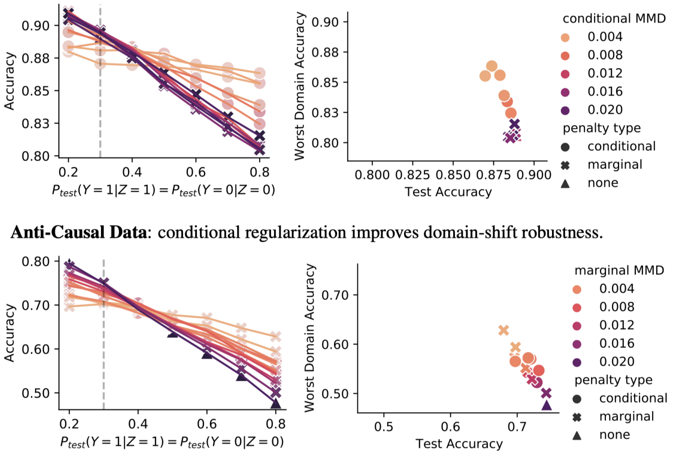
Fig. 4.8: Penalizing the MMD matching the causal structure improves stress test performance on natural product review data.

Robustness to Stress Test V

- Regularizing to reduce the MMD that *matches* the causal structure does indeed reduce sensitivity to perturbations.
- Note that penalizing the + *wrong* MMD may not help: the marginal MMD hurts on the anticausal dataset.

Domain Shift

- Amazon review data described above.
- We control the strength of the spurious association between Y and Z.
- Anti-causal: randomly subset the data to enforce a target level of dependence between Y and Z.
- Causal: $Y = 1[V > T_Z]$ where T_Z is a Z-dependent threshold and V is the number of helpfulness votes.
- Choose $P(Y = 0 \mid Z = 0) = P(Y = 1 \mid Z = 1) = \gamma$.
- Models are trained on $\gamma = 0.3$.



Anti-Causal Data: conditional regularization improves domain-shift robustness.

Causal-Direction Data: marginal regularization improves domain-shift robustness.

Fig. 4.9: The best domain-shift robustness is obtained by using the regularizer that matches the underlying causal structure of the data.

Domain Shift

- First, the unregularized predictors do indeed learn to rely on the spurious association between sentiment and the label.
- The regularization that matches the underlying causal structure yields a predictor that is (approximately) counterfactually invariant.
- Such models have somewhat worse in-domain performance, because they no longer exploit the spurious correlation.

Conclusion

- We use the tools of causal inference to formalize and study perturbative stress tests.
- A main insight of the paper is that counterfactual desiderata can be linked to observationally-testable conditional independence criteria.
- This requires consideration of the *true underlying causal structure* of the data.
- Done correctly, the link yields a simple procedure for enforcing the counterfactual desiderata, and mitigating the effects of domain shift.
- The main limitation of the paper is the restrictive causal structures we consider.

Contents

Potential Outcome Framework

Causal Graph

Causal Models

Causal Supervised Learning

- Invariant Feature Learning
  - Deconfound Data
  - Deconfound Intermediate Representations
  - Deconfound Models during Training
- Invariant Mechanism Learning

Causal Generative Modeling

- Structural Assignment Learning
- Causal Disentanglement

Causal Explanations

- Feature Attributions
- Contrastive Explanations

Causal Fariness

Causal Reinforcement Learning

Causal Generative Modeling I

- Generative Modeling: to produce samples that mimic the characteristics of the training data.
- *Controllable* Generation: techniques that allow us to enforce a set of attributes that novel samples should satisfy.
- *Causal Generative Modeling* (CGM) offers a causal perspective on controllable generation and sample editing by estimating an interventional or counterfactual distribution, respectively.
- Structural assignment learning: techniques demanding some domain knowledge of the *underlying causal graph*. Note that these methods rely on the absence of any hidden confounders.
- Causal disentanglement: methods exist that relax this requirement.

We select one representative work for each category.

Structural Assignment Learning I

- $\mathcal{T}$itle: Counterfactual Generative Networks
- $\mathcal{A}$uthor: Axel Sauer[1,2], Andreas Geiger[1,2] ([1]Max Planck Institute for Intelligent Systems, Tübingen, [2]University of Tübingen)
- $\mathcal{P}$ublished: International Conference on Learning Representations, ICLR 2021
- Deep classifiers tend to exploit spurious correlations with low-level texture or the background for solving the image classification task.
- We propose to decompose the image generation process into *independent causal mechanisms* that we train without direct supervision.
- By exploiting appropriate inductive biases, these mechanisms disentangle object shape, object texture, and background; hence, they allow for generating *counterfactual images*.

Introduction

- *Independent Mechanisms* (IM): a causal generative process is composed of autonomous modules that do not influence each other.
- In the context of image classification (e.g., on ImageNet), we can interpret the generation of an image as a causal process.
- We decompose this process into separate IMs, each controlling one factor of variation (FoV) of the image.
- Concretely, we consider three IMs: one generates the object's shape, the second generates the object's texture, and the third generates the background.
- With access to these IMs, we can produce *counterfactual images*, which is helpful for the out-of-domain robustness.

Problem Setting

- $\mathbf{x}$: high-dimensional observations (e.g. images); y: corresponding labels.
- An SCM $\mathfrak{C}$ is defined as a collection of d (structural) assignments

$$S_j := f_j\left(\mathbf{PA}_j, U_j\right), \quad j = 1, \ldots, d$$

  where each random variable S_j is a function of its parents $\mathbf{PA}_j \subseteq \{S_1, \ldots, S_d\} \setminus \{S_j\}$ and a noise variable U_j.
- The functions f_i are independent mechanisms, intervening on one mechanism f_j does not change the other mechanisms.
- Our goal is to represent the image generation process with an SCM.
- If we learn a sensible set of IMs, we can intervene on a subset of them and generate interventional images x_{IV}.
- To generate a set of counterfactual images $x_C F$, we fix the noise W and randomly draw a, which corresponds to a class label that we provide as input, denoted as y_{CF}.

Counterfactual Generative Network I

- We assume the *causal structure to be known*, and consider three learned IMs for generating shape, texture, and background, respectively.
- Here, we consider MNISTs and ImageNet. The two SCM's are as follows:

$$
\begin{array}{rl@{\qquad}rl}
\multicolumn{2}{c}{\textbf{MNISTs}} & \multicolumn{2}{c}{\textbf{ImageNet}} \\
\mathbf{M} & := f_{shape}\left(Y_1, U_1\right) & \mathbf{M} & := f_{shape}\left(Y_1, U_1\right) \\
\mathbf{F} & := f_{text,1}\left(Y_2, U_2\right) & \mathbf{F} & := f_{text}\left(Y_2, U_2\right) \\
\mathbf{B} & := f_{text,2}\left(Y_3, U_3\right) & \mathbf{B} & := f_{bg}\left(Y_3, U_3\right) \\
\mathbf{X}_{gen} & := C(\mathbf{M}, \mathbf{F}, \mathbf{B}) & \mathbf{X}_{gen} & := C(\mathbf{M}, \mathbf{F}, \mathbf{B})
\end{array}
$$

  where $\mathbf{M}$ is the mask, $\mathbf{F}$ is the foreground, $\mathbf{B}$ is the background, U_j is the exogenous noise, Y_j is the class label, $\mathbf{X}_{\text{gen}}$ is the generated image, and f_j and C are the independent mechanisms.

- In both cases, the learned IMs feed into another, fixed, IM: *the composer*.

Counterfactual Generative Network II

- An overview of our CGN is shown in Figure 5.1.
- For the experiments on ImageNet, we initialize each IM backbone with weights from a pre-trained BigGAN-deep-256.
- BigGAN-deep-256 is the current SOTA for conditional image generation, but it cannot generate images of only texture or only background.

Counterfactual Generative Network III

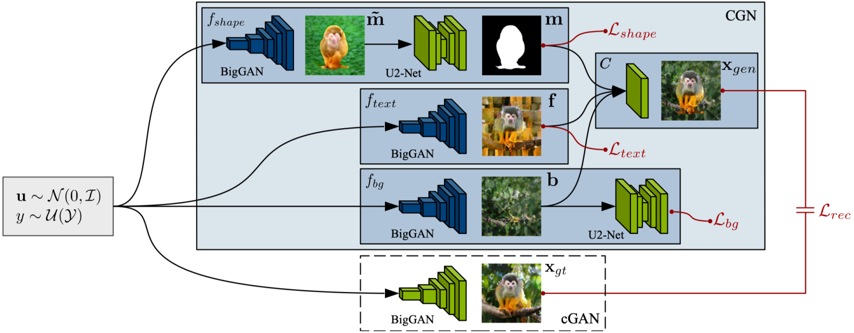


Fig. 5.1: Counterfactual Generative Network (CGN). Here, we illustrate the architecture used for the ImageNet experiments. The CGN is split into four mechanisms, the shape mechanism f_{shape}, the texture mechanism f_{text}, the background mechanism f_{bg}, and the composer C. Components with trainable parameters are blue, components with fixed parameters are green.

Composition Mechanism

- The function of the composer is not learned but defined analytically.
- Given the generated masks, textures and backgrounds, we composite the image $\mathbf{x}_{gen}$ using alpha blending, denoted as C:

  $$\mathbf{x}_{gen} = C(\mathbf{m}, \mathbf{f}, \mathbf{b}) = \mathbf{m} \odot \mathbf{f} + (1 - \mathbf{m}) \odot \mathbf{b}$$

  where $\mathbf{m}$ is the mask, $\mathbf{f}$ is the foreground, and $\mathbf{b}$ is the background. The operator $\odot$ denotes elementwise multiplication.
- This fixed composition is a strong inductive bias in itself – the generator needs to generate realistic images through this bottleneck.
- To get a stronger and more stable supervisory signal, we, therefore, use an unconstrained, conditional GAN (cGAN) to generate pseudo-ground-truth images $\mathbf{x}_{gt}$ from noise $\mathbf{u}$ and label y.
- We feed the same $\mathbf{u}$ and y into the IMs to generate $\mathbf{x}_{gen}$ and minimize a reconstruction loss $\mathcal{L}_{rec}(\mathbf{x}_{gt}, \mathbf{x}_{gen})$.

Shape Mechanism

- We model the shape using a binary mask predicted by shape IM f_{shape}, where 0 corresponds to the background and 1 to the object.
- This mechanism implements *foreground segmentation*.
- The loss is comprised of two terms: $\mathcal{L}_{binary}$ and $\mathcal{L}_{mask}$. $\mathcal{L}_{binary}$ is the pixel-wise binary entropy of the mask. $\mathcal{L}_{mask}$ prohibits trivial solutions.
- We add a pre-trained U2-Net as a head on top of the BigGAN backbone.
- The U2-Net was trained for salient object detection on DUTS-TR.
- By *fine-tuning* the BigGAN backbone, we learn to generate images of the relevant part with exaggerated features to increase saliency. We refer to these as pre-masks $\tilde{\mathbf{m}}$.

Texture Mechanism

- The texture mechanism f_{text} is responsible for generating the foreground object's appearance, while not capturing any object shape or background cues.
- We, therefore, sample patches from the full composite image and concatenate them into a grid.
- We denote this patch grid as **pg**. The patches are sampled from regions where the mask values are highest (hence, the object is likely located).
- We then minimize a perceptual loss between the foreground **f** (the output of f_{text}) and the patch-grid: $\mathcal{L}_{text}(\mathbf{f}, \mathbf{pg})$.

Background Mechanism

- The background mechanism f_{bg} needs to capture the background's global structure while the object must be removed and inpainted realistically.
- we exploit the same U2-Net as used for the shape mechanism f_{shape}.
- Again, we feed the output of the BigGAN backbone through the U2-Net with fixed weights. However, this time, we *minimize the predicted saliency*.
- Over the progress of training, this leads to the object shrinking and finally disappearing, while the model learns to inpaint the object region.
- We refer to this loss as $\mathcal{L}_{bg}$.

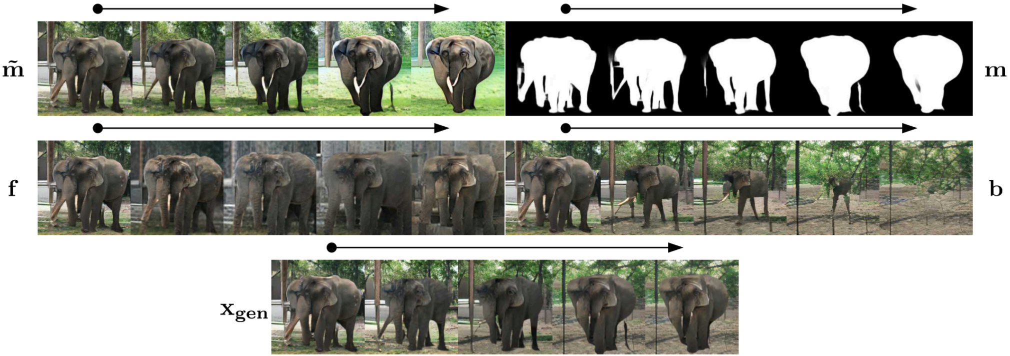


Fig. 5.2: Individual IM Outputs over Training. The arrows indicate the beginning and end of the training. The initial output of the pre-trained models is gradually transformed while the composite image only marginally changes.

Experiment I

1. Does our approach reliably learn the disentangled IMs on datasets of different complexity? (Qualitative Results)
2. Which inductive biases are necessary to achieve this?
3. Do counterfactual images enable training invariant classifiers?

We first apply our approach to different versions of MNIST: colored-, double-colored- and Wildlife-MNIST, then scale our approach to ImageNet.

Experiment II

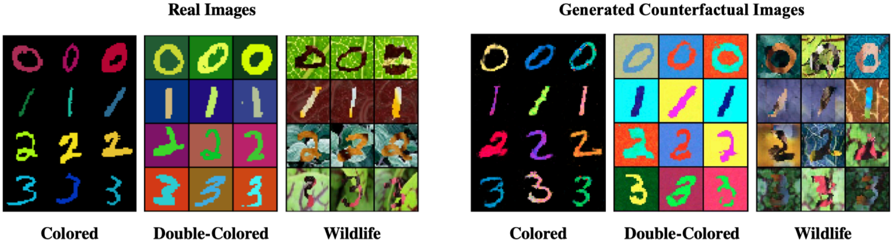
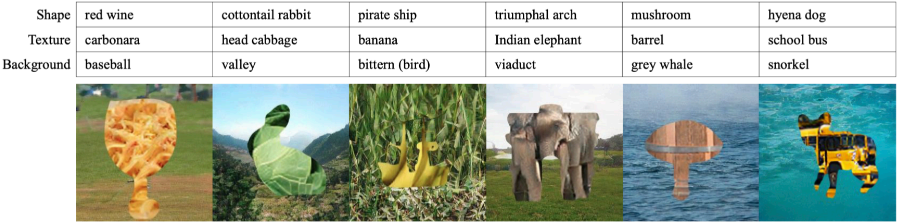


Fig. 5.3: MINISTs.

- The results on Wildlife MNIST are surprisingly good, considering that the object texture is only observable on the relatively thin digits.
- All experiments on MNIST are done without pre-training any network.

Experiment III



Shape	red wine	cottontail rabbit	pirate ship	triumphal arch	mushroom	hyena dog
Texture	carbonara	head cabbage	banana	Indian elephant	barrel	school bus
Background	baseball	valley	bittern (bird)	viaduct	grey whale	snorkel

Fig. 5.4: ImageNet Counterfactuals.

The CGN can fail to produce high-quality texture maps for very small objects, e.g., for a bird high up in the sky.

Loss Ablation

- Disable one loss at a time.
- The composite images are on the image manifold, hence, we can calculate their Inception score (IS).
- To measure if the CGN collapsed during training, we monitor the mean value of the generated mask μ_{mask}. A μ_{mask} close to 1 means that f_{text} is not training. The same is true for μ_{mask} close to 0 and f_{bg}.

$\mathcal{L}_{shape}$	$\mathcal{L}_{text}$	$\mathcal{L}_{bg}$	$\mathcal{L}_{rec}$	IS $\Uparrow$	μ_{mask}
✗	✓	✓	✓	85.9	0.2 ± 0.2 %
✓	✗	✓	✓	198.4	0.9 ± 0.1 %
✓	✓	✗	✓	195.6	0.1 ± 0.1 %
✓	✓	✓	✗	38.39	0.3 ± 0.2 %
✓	✓	✓	✓	130.2	0.3 ± 0.2%
BigGAN (Upper Bound)				202.9	-

Fig. 5.5: Loss Ablation Study.

Counterfactual Images for Training of Invariant Classifiers I

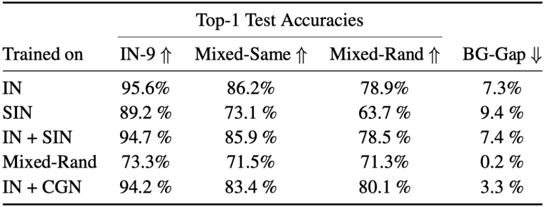
MNIST

	colored MNIST		double-colored MNIST		Wildlife MNIST	
	Train Acc ⇑	Test Acc ⇑	Train Acc ⇑	Test Acc ⇑	Train Acc ⇑	Test Acc ⇑
Original	99.5 %	35.9 %	**100.0** %	10.3 %	**100.0** %	10.1 %
IRM (2 Envs)	99.6 %	59.8 %	**100.0** %	67.7 %	99.9 %	11.3 %
IRM (5 Envs)	-	-	99.9 %	78.9 %	99.8 %	76.8 %
LNTL	99.3 %	81.8 %	98.7 %	69.9 %	99.9 %	11.5 %
Original + GAN	**99.8** %	40.7 %	**100.0** %	10.8 %	**100.0** %	10.4 %
Original + CGN	99.7 %	**95.1** %	97.4 %	**89.0** %	99.2 %	**85.7** %

Fig. 5.6: MNISTs Classification. In the test set, colors and textures are randomized, only the digit's shape corresponds to the class label. Random performance is at 10%.

IN-9

- BG-Gap: measure a classifier's dependence on the background signal.
- SIN: stylized ImageNet.
- For Mixed-Rand, the backgrounds are randomized, while the object remains unchanged.
- For Mixed-Same they sample class-consistent backgrounds.
- Directly training on Mixed-Rand leads to a drop in performance on the original data which might be due to the smaller training dataset.



	Top-1 Test Accuracies			
Trained on	IN-9 ⇑	Mixed-Same ⇑	Mixed-Rand ⇑	BG-Gap ⇓
IN	95.6%	86.2%	78.9%	7.3%
SIN	89.2 %	73.1 %	63.7 %	9.4 %
IN + SIN	94.7 %	85.9 %	78.5 %	7.4 %
Mixed-Rand	73.3%	71.5%	71.3%	0.2 %
IN + CGN	94.2 %	83.4 %	80.1 %	3.3 %

Fig. 5.7: Accuracies on IN-9. The reported accuracies are all obtained using a Resnet-50.

Conclusion

- Three independent mechanisms: shape, textual and background.
- We structure a generative network into independent mechanisms to generate counterfactual images useful for training classifiers.
- We demonstrate our approach on various MNIST variants as well as ImageNet.
- This paper links two previously distinct domains: disentangled generative models and robust classification.

Causal Disentanglement

- Now we focus on methods that do not require the specification of any underlying causal graph.
- Instead, they identify both the *underlying graph* and the *structural assignments* between the variables, thus learning a set of causally disentangled representations.
- These methods do not require access to the complete causal graph $\mathcal{G}$, instead requiring practitioner knowledge about the generative variables $\mathbf{Z}$ of interest.

Definition 5.1 (Causal Disentanglement)

We say a set of representations $\mathbf{Z}$, *s.t.* $\mathbf{X} = g(\mathbf{Z})$ *for some mapping* g, *are causally disentangled if they permit the factorization*

$$p\left(z_1, .., z_K\right) = \prod_{i=1}^{K} p\left(z_i \mid \mathbf{pa}\left(z_i\right)\right)$$

where $\mathbf{pa}\left(Z_i\right) \subset \left\{Z_j\right\}_{j \neq i} \cup \epsilon_i$ *and* ϵ_i *is the exogenous causal factor of* Z_i.

CausalVAE

- Title: CausalVAE: Disentangled Representation Learning via Neural Structural Causal Models
- Author: Mengyue Yang[1,2], Furui Liu[1,*] Zhitang Chen[1], Xinwei Shen[3], Jianye Hao[1], Jun Wang[2] ([1]Noah's Ark Lab, Huawei, [2]University College London, [3]The Hong Kong University of Science and Technology)
- Published: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2021.
- The framework of variational autoencoder (VAE) is commonly used to disentangle *independent* factors from observations.
- However, in real scenarios, factors with semantics are not necessarily independent. Instead, there might be an underlying causal structure which renders these factors dependent.
- CausalVAE: includes a Causal Layer to transform independent exogenous factors into causal endogenous ones that correspond to causally related concepts in data.

Introduction I

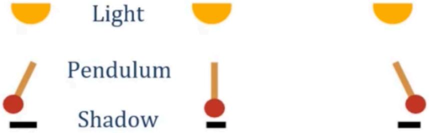
Conventional Disentangled Method

- Observations counld be represented by limit concepts.
- Concepts are totally independent.
- Using unsupervised VAE-based method.

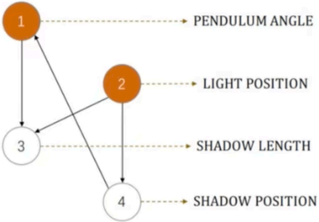
Problems in traditional disentanglement works

- The concepts are causally related.
- Unsupervised process could not guarantee the learned representations is identifiable.

Introduction II



Swing Pendulum



Causal Graph

Motivation

Learning disentanglement representations which align to real world concepts:

- Structural Causal Models.
- To guarantee identifiable representations, we introduce supervision signal.

Achieve do-operation on causal representation

- The changing on representation will be reflected on reconstruct images.
- Allow reasoning with interventions and counterfactuals.

Learning causal graph automatically.

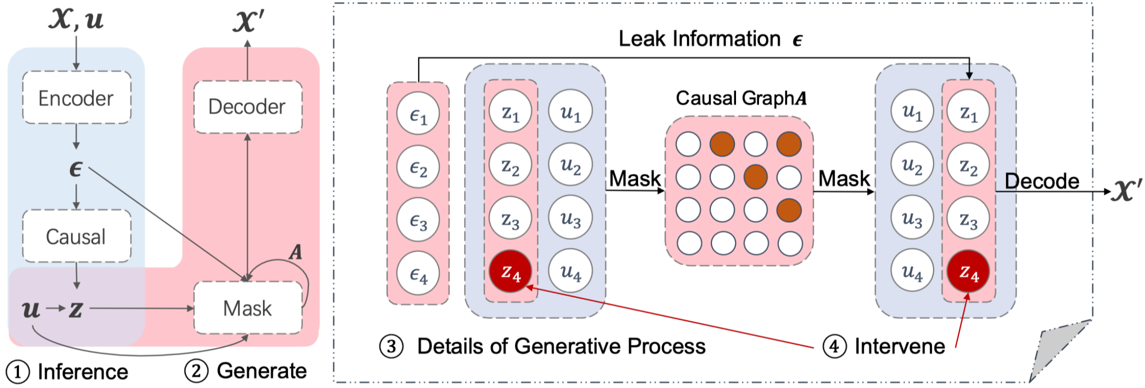


Fig. 5.8: Model structure of CausalVAE.

Transforming Independent Exogenous Factors into Causal Representations I

In addition to the encoder and the decoder structures, we introduce a Structural Causal Model (SCM) layer to learn causal representations.

- Consider n concepts of interest in data which are causally structured by a DAG with *adjacent matrix* $\mathbf{A}$.
- The Causal Layer exactly implements a Linear SCM

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

  where $\mathbf{A}$ is the parameters to be learnt in this layer. $\boldsymbol{\epsilon}$ are independence Gaussian *exogenous factors*. z_i is the lower dimensional representation of i-th concept.
- $\mathbf{A}$ is the causal graph. (e.g. $z_1 \rightarrow z_3$)

$$\begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}^T \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{pmatrix} = \begin{pmatrix} 0 + \epsilon_1 \\ 0 + \epsilon_2 \\ z_1 + \epsilon_3 \end{pmatrix}$$

Transforming Independent Exogenous Factors into Causal Representations II

- Unsupervised learning of the model might be infeasible due to the identiability issue.
- We adopt additional information $\mathbf{u}$ associated with the true causal concepts as supervising signals.
- In our work, we use the labels of the concepts.
- We propose a conditional prior $p(\mathbf{z} \mid \mathbf{u})$ to regularize the learned posterior of $\mathbf{z}$.
- We also leverage $\mathbf{u}$ to learn the causal structure $\mathbf{A}$.

Structural Causal Model Layer

- Once the causal representation $\mathbf{z}$ is obtained, it passes through a Mask Layer to reconstruct itself.
- We have a set of mild nonlinear and invertible functions $[g_1, g_2, ..., g_n]$ that map parental variables to the child variable.

$$z_i = g_i\left(\mathbf{A}_i \circ \mathbf{z}; \boldsymbol{\eta}_i\right) + \epsilon_i \tag{5.1}$$

  where $\circ$ is the element-wise multiplication and $\boldsymbol{\eta}_i$ is the parameter of $g_i(\cdot)$.
- This layer makes intervention or do-operation possible.
- To intervene z_i, we set z_i on the RHS of Equation (5.1) to a fixed value and then its effect is delivered to all its children as well as itself on the LHS of Equation (5.1).

Learning Strategy I

Inference Model $q_\phi(\mathbf{z}, \epsilon \mid \mathbf{x}, \mathbf{u})$

- Encoder transform observations $\boldsymbol{X}$ into $\boldsymbol{\epsilon}$
- A Causal layer generate causal representation

$$z = \left(I - A^T\right)^{-1} \epsilon$$

- Introduce additional observation $\boldsymbol{u}$: $\boldsymbol{z}$ satisfy conditional Gaussian $\boldsymbol{z} \sim \mathcal{N}\left(\lambda_1(\boldsymbol{u}), \lambda_2^2(\boldsymbol{u})\right)$, where $\boldsymbol{u}$ is additional observation.

Generative Model

$$p_\theta(\mathbf{x}, \mathbf{z}, \boldsymbol{\epsilon} \mid \mathbf{u}) = p_\theta(\mathbf{x} \mid \mathbf{z}, \boldsymbol{\epsilon}, \mathbf{u}) p_\theta(\boldsymbol{\epsilon}, \mathbf{z} \mid \mathbf{u})$$

- Introduce a Mask Layer

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

- Achieve do-operation

Learning Strategy II

Given data set $\mathcal{X}$ with the empirical data distribution $q_{\mathcal{X}}(\mathbf{x}, \mathbf{u})$

$$\mathbb{E}_{q_{\mathcal{X}}}[\log p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{u})] \geq \text{ELBO} = \mathbb{E}_{q_{\mathcal{X}}}\left[\mathbb{E}_{\epsilon, \mathbf{z} \sim q_{\phi}}[\log p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z}, \boldsymbol{\epsilon}, \mathbf{u})] - \underbrace{\mathcal{D}\left(q_{\phi}(\mathbf{z}, \epsilon \mid \mathbf{x}, \mathbf{u}) || p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z} \mid \mathbf{u})\right)}_{\text{Intractable}}\right]$$

Thanks to the one-to-one correspondence between $\boldsymbol{\epsilon}$ and $\mathbf{z}$.

- Inference model : $q_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \mathbf{z} \mid \mathbf{x}, \mathbf{u}) = q_{\boldsymbol{\phi}}(\boldsymbol{\epsilon} \mid \mathbf{x}, \mathbf{u})\delta(z = \mathbf{C}\boldsymbol{\epsilon}) = q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x}, \mathbf{u})\delta\left(\boldsymbol{\epsilon} = \mathbf{C}^{-1}z\right)$, where $\mathbf{C} = \left(I - \mathbf{A}^{T}\right)^{-1}$.
- Generative model: $p_{\boldsymbol{\theta}}(\boldsymbol{\epsilon}, \mathbf{z} \mid \mathbf{u}) = p_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon})p_{\boldsymbol{\theta}}(\mathbf{z} \mid \mathbf{u})$

Tractable ELBO

$$\text{ELBO} = \mathbb{E}_{q_{\mathcal{X}}}\left[\mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x},\mathbf{u})}[\log p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z})] - \mathcal{D}\left(q_{\phi}(\boldsymbol{\epsilon} \mid \mathbf{x}, \mathbf{u}) \| p_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon})\right) - \mathcal{D}\left(q_{\phi}(\mathbf{z} \mid \mathbf{x}, \mathbf{u}) \| p_{\boldsymbol{\theta}}(\mathbf{z} \mid \mathbf{u})\right)\right]$$

Addtional Constraints:

Learning Strategy III

- Acyclic constraint: $H(\mathbf{A}) \equiv \operatorname{tr}\left((\mathbf{I}+\mathbf{A}\circ\mathbf{A})^n\right) - n = 0.$
- Constraint in Mask Layer:

$$l_u = \mathbb{E}_{q\mathcal{X}} \left\|\mathbf{u} - \sigma\left(\mathbf{A}^T\mathbf{u}\right)\right\|_2^2 \leq \kappa_1$$

$$l_m = \mathbb{E}_{\mathbf{z}\sim q_\phi} \sum_{i=1}^{n} \left\|z_i - g_i\left(\mathbf{A}_i \circ \mathbf{z}; \boldsymbol{\eta}_i\right)\right\|^2 \leq \kappa_2$$

Experiment I

Synthetic

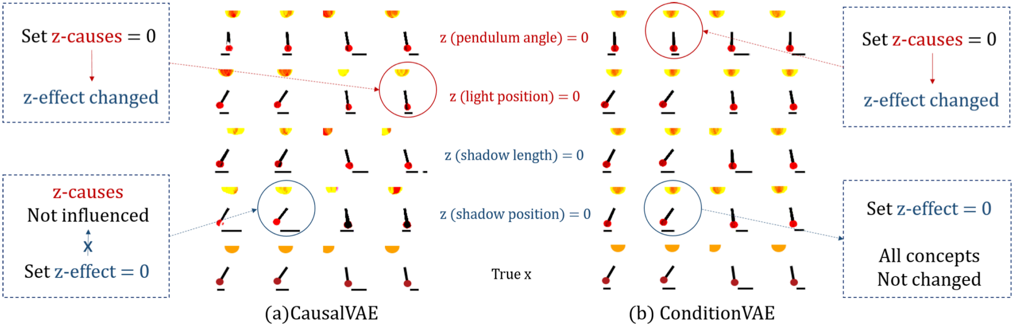


Fig. 5.9: The results of Intervention experiments on the pendulum dataset.

Experiment II

Real world benchmark

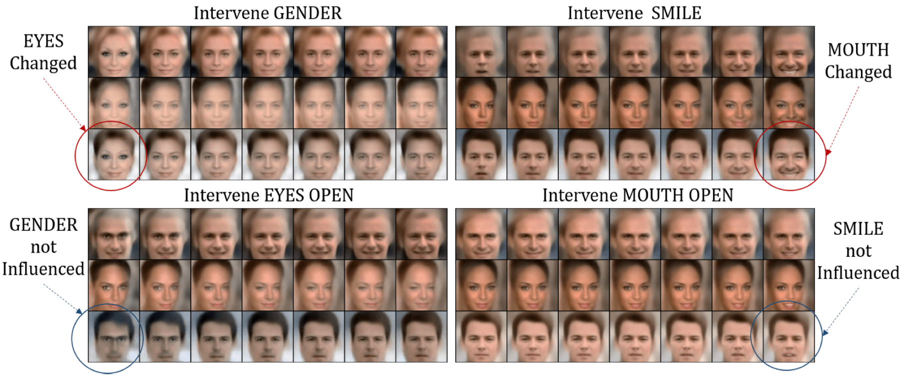


Fig. 5.10: Results of CausalVAE model on CelebA(SMILE).

Experiment III

The mutual information (MIC/TIC) between the learned representation and the ground truth concept labels of all compared methods.

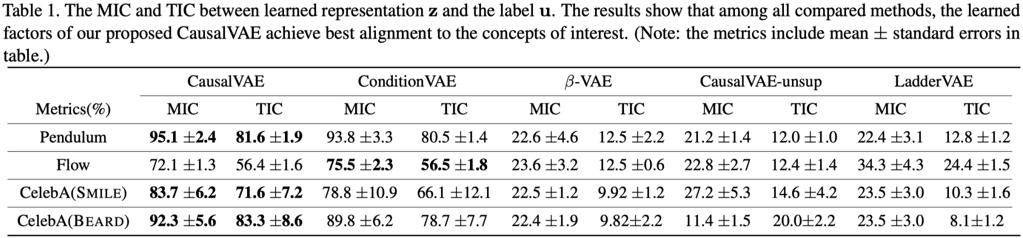
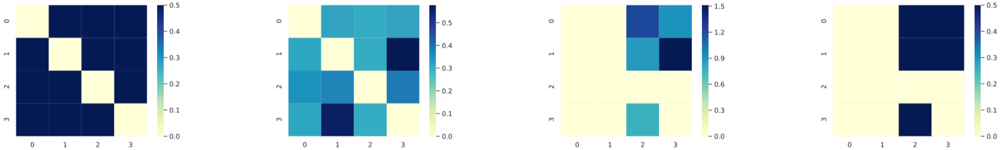


Table 1. The MIC and TIC between learned representation **z** and the label **u**. The results show that among all compared methods, the learned factors of our proposed CausalVAE achieve best alignment to the concepts of interest. (Note: the metrics include mean ± standard errors in table.)

	CausalVAE		ConditionVAE		β-VAE		CausalVAE-unsup		LadderVAE	
Metrics(%)	MIC	TIC	MIC	TIC	MIC	TIC	MIC	TIC	MIC	TIC
Pendulum	**95.1 ±2.4**	**81.6 ±1.9**	93.8 ±3.3	80.5 ±1.4	22.6 ±4.6	12.5 ±2.2	21.2 ±1.4	12.0 ±1.0	22.4 ±3.1	12.8 ±1.2
Flow	72.1 ±1.3	56.4 ±1.6	**75.5 ±2.3**	**56.5 ±1.8**	23.6 ±3.2	12.5 ±0.6	22.8 ±2.7	12.4 ±1.4	34.3 ±4.3	24.4 ±1.5
CelebA(SMILE)	**83.7 ±6.2**	**71.6 ±7.2**	78.8 ±10.9	66.1 ±12.1	22.5 ±1.2	9.92 ±1.2	27.2 ±5.3	14.6 ±4.2	23.5 ±3.0	10.3 ±1.6
CelebA(BEARD)	**92.3 ±5.6**	**83.3 ±8.6**	89.8 ±6.2	78.7 ±7.7	22.4 ±1.9	9.82±2.2	11.4 ±1.5	20.0±2.2	23.5 ±3.0	8.1±1.2

Experiment IV



(a) Initialize (b) After 1 epoch (c) After 5 epochs (d) The true

Fig. 5.11: The learning process of causal matrix **A**. The concepts include: GENDER, SMILE, EYES OPEN, MOUTH OPEN (top-to-bottom and left-to-right order); (c) converged **A**, (d) ground truth (from causal GAN).

Conclusion

- CausalVAE: includes a SCM layer to model the causal generation mechanism of data.
- We prove that the proposed model is fully identifiability given additional supervision signal.
- Experimental results with synthetic and real data show that CausalVAE successfully learns representations of causally related concepts and allows intervention to generate counterfactual outputs as expected.
- How to choose the additional supervised signal in more general situation?

Contents

Potential Outcome Framework

Causal Graph

Causal Models

Causal Supervised Learning

- Invariant Feature Learning
  - Deconfound Data
  - Deconfound Intermediate Representations
  - Deconfound Models during Training
- Invariant Mechanism Learning

Causal Generative Modeling

- Structural Assignment Learning
- Causal Disentanglement

Causal Explanations

- Feature Attributions
- Contrastive Explanations

Causal Fariness

Causal Reinforcement Learning

Contents

Potential Outcome Framework

Causal Graph

Causal Models

Causal Supervised Learning

- Invariant Feature Learning
  - Deconfound Data
  - Deconfound Intermediate Representations
  - Deconfound Models during Training
- Invariant Mechanism Learning

Causal Generative Modeling

- Structural Assignment Learning
- Causal Disentanglement

Causal Explanations

- Feature Attributions
- Contrastive Explanations

Causal Fariness

Causal Reinforcement Learning

Contents

Potential Outcome Framework

Causal Graph

Causal Models

Causal Supervised Learning

- Invariant Feature Learning
  - Deconfound Data
  - Deconfound Intermediate Representations
  - Deconfound Models during Training
- Invariant Mechanism Learning

Causal Generative Modeling

- Structural Assignment Learning
- Causal Disentanglement

Causal Explanations

- Feature Attributions
- Contrastive Explanations

Causal Fariness

Causal Reinforcement Learning

Reference I