DE GRUYTER Journal of Causal Inference 2021

Research Article Open Access

Andrew Forney* and Scott Mueller

Causal Inference in AI Education: A Primer

Abstract: The study of causal inference has seen recent momentum in machine learning and artificial intelligence (AI), particularly in the domains of transfer learning, reinforcement learning, automated diagnostics, and explainability (among others). Yet, despite its increasing application to address many of the boundaries in modern AI, causal topics remain absent in most AI curricula. This work seeks to bridge this gap by providing classroom-ready introductions that integrate into traditional topics in AI, suggests intuitive graphical tools for application to both new and traditional lessons in probabilistic and causal reasoning, and presents avenues for instructors to impress the merit of climbing the "causal hierarchy" to address problems at the levels of associational, interventional, and counterfactual inference. Lastly, this study shares instructor experiences, successes, and challenges integrating these lessons at multiple levels of education.

Keywords: causal inference education, artificial intelligence education, machine learning

MSC: 97Q60,68T01

1 Introduction

The study of causality seeks to model and reason about systems using a formal language of cause and effect, and has undertaken a number of important endeavors across a diverse set of disciplines, including: causal diagrams to inform empirical research [1], structural equation models for econometric analysis [2], systems-thinking in the philosophy of science [3–5], modeling elements of human cognition and learning [6–8], and many others [9].

Yet, for its long history in other disciplines, causal inference has only recently begun to penetrate traditional topics in machine learning and the design of artificial agents. Although perhaps overshadowed by the impressive advances from deep learning, the AI community is turning to causality to address many of its boundaries, such as to avoid overfitting and to transfer learning [10, 11], reasoning beyond observed examples as through counterfactual inference [12], providing meta-cognitive avenues for reinforcement learners in confounded decision-making scenarios [13, 14], improving medical diagnostics beyond mere association of symptoms [15, 16], reducing bias in machine learning models through formalizations of fairness [17, 18], among others [19, 20].

Despite these clarion calls for causality from many prominent researchers and practitioners, it remains a missing topic in the vast majority of traditional AI curricula. This lag can be explained by a number of factors, including the recency of causal developments in the domain, the lack of a bridge between the topics of causality that statisticians and empirical scientists care about and those that computer scientists do, and the lack of templated lesson plans for integration into such curricula; even causality textbooks oriented at undergraduate introduction lack direct examples relating to AI [21]. This work endeavors to address each of these explanations by providing a primer for educators to bring causality into the AI classroom. Specifically, it provides motivated, detailed, and numerical examples of causal topics as they apply to AI, discusses common pitfalls in the course of student learning experiences, and a number of other tools ready to be deployed by instructors teaching topics in AI and ML at the high-school and college levels.

***Corresponding Author: Andrew Forney:** Department of Computer Science, Loyola Marymount University, Los Angeles, CA 90045; E-mail: andrew.forney@lmu.edu

Scott Mueller: Department of Computer Science, University of California, Los Angeles, CA 90095; E-mail: scott@cs.ucla.edu

As such, the main contributions of the present work are as follows:

1. Provides brief, classroom-ready introductions to the three tiers of data and queries that compose the causal hierarchy: associations, interventions, and counterfactuals.
2. Suggests intuitive graphical depictions of core lessons in probabilistic and causal reasoning that enable multi-modal instruction.
3. Demonstrates and motivate examples wherein causal concepts can be easily integrated into typical lessons in AI, alongside novel, interactive learning tools to help concrete select topics.
4. Shares empirical successes and challenges from causally-motivated lessons deployed at both undergraduate and high-school levels.

2 Background

Etiology is core to scientific discovery and philosophical concerns since humans first started asking why things are the way they are. Humans possess a natural ability to learn cause and effect that allows us to understand, deduce, and reason about the data we take in through our senses [22]. Modern tools for inferring causes allow us to systematically interpret these causal connections at a more fundamental level with increased confidence, less data, and fewer assumptions. With this deeper causal knowledge, causal inference serves to make accurate predictions, estimate effects of interventions, and decipher imagined scenarios.

The distinction between these tasks, their underlying types of data, and the inferences possible given assumptions about the system are delineated in the Pearlian Causal Hierarchy (PCH) [23]. The PCH is organized into three tiers / layers of information, each building upon the expressiveness of the last:

$\mathcal{L}_1$ *Associations*: Observing evidence and assessing changes in belief about some variable, e.g., determining the likelihood of having some disease given presentation of certain symptoms.

$\mathcal{L}_2$ *Interventions*: Assessing the likelihood of some causal effect under a manipulation, e.g., determining the efficacy of a drug in treating some condition.

$\mathcal{L}_3$ *Counterfactuals*: Determining the likelihood of some outcome under hypothetical manipulation that is contrary to what happened in reality, e.g., determining whether a headache would have persisted *had one not* taken aspirin.

The ability to traverse the different layers of the PCH often demands causal assumptions to be stated in a mathematical language that clearly disambiguates between them. As will be demonstrated in the following sections, certain interventional ($\mathcal{L}_2$) and counterfactual ($\mathcal{L}_3$) queries of interest cannot be answered using data and traditional statistics alone, but can be enabled by an explanation of the system under scrutiny as through a Structural Causal Model (SCM).

Definition 2.1. ***(Structural Causal Model)*** *[9, pp. 203-207] A Structural Causal Model is a 4-tuple, $M = \langle U, V, F, P(u) \rangle$ where:*

1. *U is a set of background variables (also called exogenous), whose values are determined by factors outside the model.*
2. *V is a set $\{V_1, V_2, ..., V_n\}$ of endogenous variables whose values are each determined by other variables in $U \cup V$.*
3. *F is a set of functions $\{f_1, f_2, ..., f_n\}$ such that each f_i is a mapping from (the respective domains of) $U_i \cup PA_i$ to V_i where $U_i \subseteq U$ and $PA_i \subseteq V \setminus V_i$ and the entire set F forms a mapping from U to V. In other words, each f_i in $v_i = f_i(pa_i, u_i), i = 1, ..., n$ assigns a value to V_i that depends on (the values of) a select set of variables.*
4. *$P(u)$ is a probability density defined on the domain of U.*

The inputs to the functions in F within an SCM induce a causal diagram in the form of a directed acyclic graph (DAG) as in Figure 1. A DAG alone is therefore a partial causal model in itself. This nonparametric

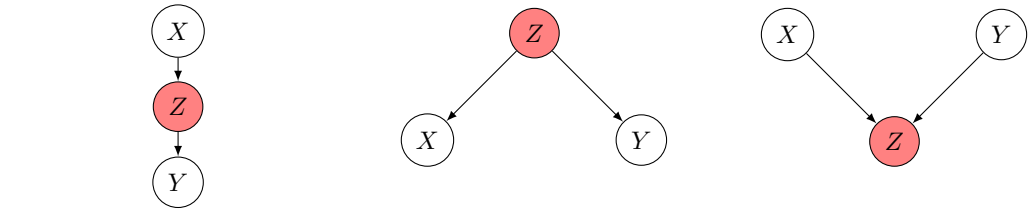
causal model can come from expert knowledge and is often the only portion of the SCM to which we have access.

Definition 2.2. ***(Causal Diagram)*** *Given any SCM M, its associated causal diagram G is a directed, acyclic graph (DAG) that encodes:*

1. *The set of endogenous variables V, represented as solid nodes (vertices).*
2. *The set of exogenous variables U, represented as hollow nodes (sometimes omitted for brevity).*
3. *The functional relationships, F, between variables, represented by directed edges that connect two variables $V_c \rightarrow V_e$ for $V_c, V_e \in V$ if V_c appears as a parameter in $f_{V_e}(V_c, ...)$ (i.e. if V_c has a causal influence on V_e).*
4. *Spurious correlations between variables, represented by a bidirected, dashed edge connects two variables $V_a \leftarrow - \rightarrow V_b$ if their corresponding exogenous parents U_a, U_b are dependent, or if f_{V_a}, f_{V_b} share an exogenous variable U_i as a parameter to their functions.*

Fortunately, a DAG provides enough extra-data information to answer many causal queries, even with the data generating process hidden. DAGs can also allow causal effects, real or counterfactual, to be computed despite an absence of experimental data. *Causal inference* is thus the umbrella label for tools used to compute queries from an existing causal model, while the focus of *causal discovery* is in constructing or learning the model from data.

Causal discovery supports the assembly of DAGs, or parts of DAGs, largely by examining independence relations among variables (potentially conditioned on other variables), to offer a mechanism to uncover their causal relationships. In this sense, data alone is sometimes enough for causal inference, but when it is not, a partial DAG (also known as a pattern or equivalence class) can inform practitioners of what else is required to disambiguate. Children instinctively comprehend this and employ playful manipulation to better grasp their environment when information from their senses is insufficient [7]; adults and scientists also perform experiments to confirm their causal hypotheses. Causal discovery with DAGs provides a systematic way for machines to better *understand* causal situations beyond the traditional machine learning task of prediction.



(a) Chain **(b)** Fork **(c)** Collider

Figure 1. Causal "triplets" demonstrating the rules of conditional independence from the *d*-separation criterion.

All DAGs, regardless of complexity, can be constructed from paths of the three basic structures depicted in Figure 1. The chain in Figure 1a consists of X causing Z, followed by Z causing Y. The fork in Figure 1b consists of Z having a causal influence on both X and Y. In this case, even though X has no causal effect on Y, knowing the value of X does help predict the value of Y, quintessential correlation without causation. In both the chain and the fork, X is independent of Y if and only if conditioning on Z ($X \perp\!\!\!\perp Y|Z$): $P(Y = y|Z = z, X = x) = P(Y = y|Z = z)$ and $P(Y = y|X = x) \neq P(Y = y)$[1].

Colliders, as illustrated in Figure 1c, behave the opposite to chains and forks in regards to independence. Specifically, $X \perp\!\!\!\perp Y$ without conditioning on Z: $P(Y = y|X = x) = P(Y = y)$. However, X and Y notably become *dependent* when conditioning on Z: $P(Y = y|Z = z, X = x) \neq P(Y = y|Z = z)$. By holding the common effect Z to a particular value, any change to X would be compensated by a change to Y.

1 In rare *intransitive cases* $P(Y = y|X = x) = P(Y = y)$

Students will appreciate causal stories to explain these rules of dependence in a causal graph. For each of the following examples, a fruitful exercise can be to have students provide a graphical explanation for the story, which then motivates the rules of independence expected of any graphs with the same patterns.

Example 2.1. ***Confounding: Heat, Crime, and Ice Cream.*** *Data reveals that sales of ice cream, X, are positively correlated with crime rates, Y, yielding the amusing possibilities that criminals enjoy a post-crime ice cream or that ice cream leads people to commit crime. However, the two become independent after controlling for a confounder, temperature, Z, that is responsible for both (and could not be affected by either). Z is known as a confounder that "explains away" the non-causal relationship between X, Y, making the causal structure a fork, $X \leftarrow Z \rightarrow Y$.*

Example 2.2. ***Mediation: Smoking, Tar, and Lung Cancer.*** *In medical records, smoking cigarettes, X, has been shown to be positively correlated with incidence of lung cancer, Y. It is known that smoking causes deposits of tar, Z, in the lungs, which leads to cancer Y. However, knowing whether or not a patient has lung tar Z makes its source (e.g., whether or not they smoked, X) independent from their propensity for lung cancer, Y. Z is thus known as a mediator between X and Y, making the causal structure a chain, $X \rightarrow Z \rightarrow Y$.*

Example 2.3. ***Colliders: Coin Flips and Coffee.*** *You and your roommates have a game that decides when you will break for coffee: if two of you flip fair coins X, Y, and they both come up heads or both tails, then you will ring a bell Z to summon your dorm to get coffee, C. Alone, the coin flip outcomes X, Y are independent from one another; however, if you hear a bell ring, and know that $X =$ heads, you know also that $Y =$ heads. The same is true if, instead of hearing the bell, you witness your dorm leave to get coffee. This relationship is thus a collider structure with $X \rightarrow Z \leftarrow Y$, and to demonstrate the effects of conditioning upon the descendant of a collider, $Z \rightarrow C$.*

The graphical nature of these types of exercises engender high engagement among students compared to typical probability syntax alone. Causal intuition and probabilistic understanding in this puzzle-like context is thus concerted and enhanced. Building upon these intuitions, we can establish independence or isolate effects in more complex graphs by blocking paths from one node to another through a structural criteria called d-separation (directional separation) [24]; d-separation is already taught alongside traditional AI coverage of Bayesian Networks, and succinctly stated as follows.

Definition 2.3. ***(d-separation)*** *[21, pp. 46-47] A path p between X, Y is blocked by a set of nodes Z if and only if*

1. *p contains a chain of nodes $A \rightarrow B \rightarrow C$ or a fork $A \leftarrow B \rightarrow C$ such that the middle node B is in Z (i.e., B is conditioned on), or*
2. *p contains a collider $A \rightarrow B \leftarrow C$ such that the collision node B is not in Z, and no descendant of B is in Z.*

If all paths between X, Y are blocked given Z, they are said to be "d-separated" and thus $X \perp\!\!\!\perp Y \mid Z$.

With causal models being core to causal inference, d-separation provides us with an important testing mechanism. Because a DAG demonstrates which variables are independent from each other given a subset of the remaining variables to condition on, probabilities can be estimated from data to confirm these conditional independencies. The fitness of a causal model can therefore be verified, and debugging is simplified through d-separation's ability to pinpoint error localities. Unfortunately, it is not possible to test every causal relationship between nodes in a DAG, meaning that causal discovery does not always yield the complete DAG.

Still, certain structural hints provide hope of recovering causal localities. For instance, a v-structure is defined as a pair of nonadjacent nodes, such as X and Y in Figure 1c, with a common effect (Z in the same

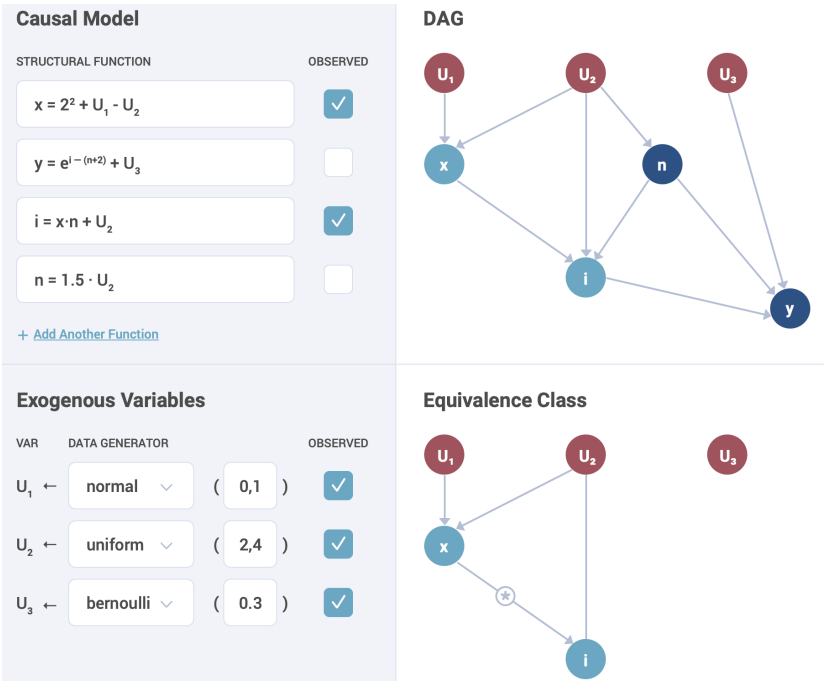


Figure 2. Causal discovery exercise editor.

figure). These v-structures are often embedded throughout larger causal graphs. An example of a testable implication is to verify that Z is not included in the set of nodes that render $X \perp\!\!\!\perp Y$.

A naive approach to causal discovery is to assume every possible DAG compatible with a set of variables and their independence relationships in a dataset. This set of compatible DAGs is called an *equivalence class*, which, for some causal queries, can be sufficient for identifying causal effects even with partial structures. If further experimentation is necessary, an equivalence class can help target those variables on which experiments need be performed to discover the true structure [25, 26].

The Inductive Causation (IC) algorithm[2] [9, pp. 204] is a simple approach to causal discovery:

1. For each pair of variables a and b in V, search for a set S_{ab} such that $(a \perp\!\!\!\perp b|S_{ab})$ holds in $\hat{P}$ (stable distribution of V). Construct an undirected graph G such that vertices a and b are connected with an edge if and only if no set S_{ab} can be found.
2. For each pair of nonadjacent variables a and b with a common neighbor c, check if $c \in S_{ab}$. If it is not, then add arrowheads $a \rightarrow c \leftarrow b$.
3. In the partially directed graph that results, orient as many of the undirected edges as possible subject to two conditions: (i) Any alternative orientation would yield a new v-structure; or (ii) Any alternative orientation would yield a directed cycle.

The first step constructs a complete skeleton. While not all arrowheads in the second and third steps can always be discovered from data alone, systems can also prompt humans for clarity on parts of nonparametric causal models to resolve ambiguity. Robotic algorithms can even perform necessary experiments to disambiguate certain localities of the causal structure.

This work introduces a companion causal inference learning system[3] to help students practice and absorb concepts in causal discovery. Depicted in Figure 2, a teacher simply writes the structural functions and data generating processes of the exogenous variables, and students are presented with the resulting probability distribution and nodes of the equivalence class to connect. Causal discovery exercises such as

2 More advanced algorithms increase efficiency and accommodate latent structures [9, pp. 50-54].
3 CI learning tools found at: https://learn.ci

these provide engaging exploration into the etiology of data generation missing in many statistically-focused curricula.

The remainder of this work focuses on the potential of causal inference to both elucidate traditional topics in AI and ML and to inspire new avenues for students to explore. Using the preliminaries outlined in this section, students will be equipped to understand the challenges and opportunities at each tier of the PCH.

Instructor Reflections

Students treat exercises involving the design and interpretation of compounded conditional independence graphs as puzzles rather than monotonous calculations. This has elicited enjoyment, which feeds graph modifications. The discussions and debates that ensue develops intuition through active engagement.

Instructors may find it useful to generate mock datasets to help students to understand crucial lessons in causal discovery, d-separation, and challenges like observational equivalence and unobserved confounding. Various software packages exist for this endeavor, though Tetrad and Causal Fusion have been popular choices that students can pick up without large amounts of tutorial.[4]

3 Associations

SCMs are capable of answering a wide swath of queries, the most fundamental being associational. Queries at this first tier, or layer $\mathcal{L}_1$, consist of predictions based on what has been *observed*. For instance, after observing many labeled CT scans with and without tumors, an ML algorithm can predict the presence of a tumor in a previously unseen scan. Traditional supervised learning algorithms have excelled in their ability to answer $\mathcal{L}_1$ queries, typically trained on data consisting of large feature vectors along with their associated label. If $\boldsymbol{X}$ is an n-dimensional feature vector with $X_1, X_2, \ldots, X_n$ as the individual features, and Y is the output variable, a model such as a trained neural network will calculate $P(Y|X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n)$. However, this predictive capacity can be stretched thin when faced with important queries that are not associational; indeed, many pains of modern ML techniques can be blamed on their inability to move beyond this tier, as demonstrated over the following examples.

3.1 Simpson's Paradox

Example 3.1. ***AdBot*** *Consider an online advertising agent attempting to maximizing clickthroughs, with* $X \in \{0, 1\}$ *representing two ads,* $Y \in \{0, 1\}$ *whether or not it was clicked upon, and* $Z \in \{0, 1\}$ *the sex of the viewer. A marketing team collects the following data on purchases following ads shown to focus groups to be used by AdBot:*

	Ad 0	**Ad 1**
Male	108/120 (90%)	340/400 (85%)
Female	266/380 (70%)	65/100 (65%)
Total	374/500 (75%)	405/500 (81%)

Table 1. Clickthroughs in the AdBot setting striated by the ad shown to participants in a focus group, and the sex of the viewer.

4 Tetrad can be found at https://www.ccd.pitt.edu/tools/ and Causal Fusion at causalfusion.net.

Table 1 shows $P(Y = 1|X = 1) = 0.81 > P(Y = 1|X = 0) = 0.75$, which may lead AdBot to conclude that Ad 1 is always more effective. However, the same data also shows within sex-specific strata that $P(Y = 1|X = 1, Z = 0) = 0.85 < P(Y = 1|X = 0, Z = 0) = 0.9$ and $P(Y = 1|X = 1, Z = 1) = 0.65 < P(Y = 1|X = 0, Z = 1) = 0.7$, indicating that Ad 0 is better. AdBot thus faces a dilemma: if the sex of a viewer is *not* known, which ad is the best choice? This conflict is known as Simpson's paradox, which long haunted practitioners using only $\mathcal{L}_1$ tools without causal considerations. Its solution, and those to many other problems, can be found in the next tier.

3.2 Linear Regression

Linear regression is a common topic in introductory statistics and machine learning (ML) courses. ML models evolve naturally from linear regression since basic neural networks, even with many layers, are essentially linear models with multivariate linear regression if their activation functions are linear. Although this simplicity will seldom yield highly predictive algorithms with real-world data, linear regression can clearly illustrate the value of causal constructs through coding exercises. Student discussion can be fostered through debate about linearity assumptions among exercises and examples.

Other work has provided examples for inferring causal effects from associational multivariate linear regression ([27]), but which we adapt herein as useful exercises for ML students to start examining problems from different tiers of the causal hierarchy. A first exercise corresponds to the chain DAG of figure 1a.

Example 3.2. ***Athletic Performance*** *Consider an athletic sport where the goal is to predict an athlete's performance. An ML model uses features X and Z, corresponding to training intensity and skill level, respectively. The outcome, Y, is level of athletic performance. The following PyTorch code*[5] *generates example data:*

```
x = torch.randn(n, 1) # training intensity for n individuals
z = 2 * x + torch.randn(n, 1) # skill level for n individuals
features = torch.cat([x, z], 1) # feature vector with training intensity and skill level
y = 3 * z + torch.randn(n, 1) # athletic performance for n individuals
```

The next step is to train an ML model that lacks non-linear activation functions. The weights of the model can then be analyzed:

```
model = train_model(features, y) # train 1-layer model on features {X,Z}, and outcome Y
weights, bias = model.parameters() # retrieve weights and bias for the neural network
print(weights.tolist()) # print the weights for X and Z to the console
# [[-0.00918455421924591, 2.9990761280059814]]
print(bias.item()) # print the bias to the console
# -0.004577863961458206
```

The weight on X has a negligible[6] impact on the result. This makes sense as the model was trained on both X and Z, while Y only "*listens* to" Z (i.e., $f_y(z, u_y)$). Looking only at the weights, it would seem that training intensity is irrelevant to athletic performance. If an analyst wanted to predict the performance of someone with increased training intensity, using this model they would observe no difference in performance. On the other hand, if the model had been trained only on X:

5 Full source code is at: https://github.com/CausalEd/exercises.

6 A traditional linear model can provide a confidence interval that will very likely contain 0.

```
model = train_model(x, y) # train model only on X instead of both X and Z
weights, bias = model.parameters()
print(weights.tolist())
# [[6.0043745040893555]]
print(bias.item())
# 0.0020016487687826157
```

Here, X clearly plays a major role in predicting performance. This time, making a prediction using this model with increased training intensity will yield increased athletic performance.

Which feature vector do we use for our ML model? This decision isn't clear because predicting athletic performance when changing only training intensity is an intervention. Thus, this is a causal question requiring tools from $\mathcal{L}_2$ covered in the following section.

Example 3.3. ***Competitiveness*** *How an athlete fares in a competition against others depends, among other things, on their athletic ability and preparation. Unfortunately, The Tortoise and the Hare taught us that high performers often suffer from overconfidence, which reduces their preparation time and effort. In order to predict an athlete's level of competitiveness, Y, an ML model uses features X and Z, corresponding to preparation and athletic performance. The following PyTorch code generates example data accordingly:*

```
z = torch.randn(n, 1) # athletic performance for n individuals
x = -2 * z + torch.randn(n, 1) # preparation level for n individuals
features = torch.cat([x, z], 1) # feature vector with preparation and performance
y = x + 3 * z + torch.randn(n, 1) # competitiveness level for n individuals
```

Similar to example 3.2, an ML model can be trained on features X and Z or just on X. First, a feature vector consisting of both X and Z produces the following weights and bias:

```
model = train_model(features, y)
weights, bias = model.parameters()
print(weights.tolist())
# [[1.0000419616699219, 3.0182747840881348]]
print(bias.item())
# -0.0009028307977132499
```

The weight on X indicates a positive impact on the outcome. Predicting the level of competitiveness of someone with increased preparation time would yield an increased level of competitiveness. This makes sense as the example data was generated where Y was calculated with a positive multiple of X (1 to be precise). Next, a singleton feature vector of X produces the following weights and bias:

```
model = train_model(x, y)
weights, bias = model.parameters()
print(weights.tolist())
# [[-0.21623165905475616]]
print(bias.item())
# -0.023961037397384644
```

This time, the weight on X is negative, indicating a negative impact on the outcome. It would seem that increasing preparation in this model *decreases* competitiveness.

These two models have very different weights on X. Which model is correct? The answer depends on the quantity of interest. A causal question, such as, "what is the effect of preparation on competitiveness?" requires an analysis in $\mathcal{L}_2$.

Example 3.4. ***Money*** *How much money does an athlete earn? This depends, among other things, on their previous athletic performance and their ability to negotiate. Can an ML model predict an athlete's negotiating skill based on their performance? The following PyTorch code generates example data for athletic performance, X, negotiating skill, Y, and salary, Z:*

```
x = torch.randn(n, 1) # athletic performance for n individuals
y = torch.randn(n, 1) # negotiating skill for n individuals
z = 2 * x + y + torch.randn(n, 1) # salary for n individuals
features = torch.cat([x, z], 1) # feature vector with athletic performance and salary
```

An ML model trained with a feature vector consisting of both X and Z produces the following weights and bias:

```
model = train_model(features, y)
weights, bias = model.parameters()
print(weights.tolist())
# [[-1.0020248889923096, 0.5002512335777283]]
print(bias.item())
# 0.011319190263748169
```

The weight on X indicates an inverse relationship between athletic performance and negotiating skill. Are better athletes worse negotiators? Using a singleton feature vector of X paints a different picture:

```
model = train_model(x, y)
weights, bias = model.parameters()
print(weights.tolist())
# [[0.0004336435522418469]]
print(bias.item())
# 0.021031389012932777
```

This time, the weight on X is negligible. We know, from the code that generated the example data, that negotiating skill and athletic performance are uncorrelated. So, this appears to be a better model for understanding the causal effect of negotiating skill on athletic performance (a null causal effect). In addition to the two previous examples, this is another example where $\mathcal{L}_2$ tools are necessary to know which variables to include in the feature vector.

Instructor Reflections

Students are usually surprised when first exposed to Simpson's paradox. This revelation is their first hint that the story behind the data is crucial for thorough and valid interpretations of the results. This is a prime opportunity for active learning. Using DAGs as a discussion source [28], students review and debate both the diagrams and the need to be careful about which features to train their ML models on and how to utilize their results.

For many students, learning the mathematics of probability and statistics may feel mechanical, thus missing the forest (the ability to use these as tools to inform decisions, automated or otherwise) for the trees (the rote computation). Examples such as the above break the mold of this script and ask students to make a defendable *choice* with the data and assumptions at-hand because such *acts* are causal questions often unanswerable by the data alone.

The causal "solutions" to these problems have intuitive, graphical criteria that students tend to find more appealing than reasoning over the symbolic or numerical parameters of each system alone. What follows is an overview of these approaches that can both enhance student understanding of traditional

tools in $\mathcal{L}_1$, as well as understanding their limits: both when and how to seek solutions to questions at higher tiers of the causal hierarchy.

4 Interventions

The second tier in the causal hierarchy is the interventional layer, $\mathcal{L}_2$. Queries of this nature ask what happens when we intervene and *change* an input as opposed to *seeing* the input of the associational layer. Analyzing Table 1 in the AdBot example, the question of what outcome we can predict based on which ad was shown is answered by *seeing* that Ad 1 received more clicks. However, the causal question of which ad causes more clicks is a different question, predicated on determining the effect of *changing* the ad that was seen despite its natural causes.

To isolate these causal effects, the randomized controlled trial (RCT) was invented [29], free of so called "confounding bias" that can make spurious correlation masquerade as causal effect. Unfortunately, experiments are not always feasible, affordable, nor ethical: if we consider an example experiment to discern the effects of smoking on lung cancer, and confess that while there are valuable techniques for dealing with imperfect compliance [9, 30], a study that forced certain groups to smoke and others to abstain would not be ethically sound.

4.1 Resolving Simpson's Paradox

As such, practitioners are often left with causal questions but only observational data, like in Example 3.1. Herein, we witness an instance of Simpson's paradox, when a better outcome is predicted for one treatment versus another, but the reverse is true when calculating treatment effects for each subgroup.

(a) Observed confounder Z between X and Y.

(b) M-graph with unobserved confounders U_1, U_2 between X, Z and Z, Y, respectively.

Figure 3. Potential models explaining Simpson's paradox.

Resolving Simpson's paradox demands that we understand the underlying data-generating causal system, which in general may cause confusion through only the associational lens. Examining Figure 3, these two observationally equivalent causal models of the data in Example 3.1 tell two different interventional stories. In (a), Z is a confounder whose influence in the observational data must be controlled to isolate the causal effect of $X \to Y$. In (b), Z is only spuriously correlated with $\{X, Y\}$, and so controlling for Z in this setting will actually enable confounding bias (by the rules of d-separation, since $U_1 \to Z \leftarrow U_2$ forms a collider). Practically, this means that if (a) is our explanation of the observed data, then AdBot should consult the sex-specific clickthrough rates and display Ad 0; if (b) is our explanation, then we consult the aggregate data and display Ad 1. In this specific scenario, model (a) seems the more defendable since it is unlikely to have latent confounders on sex.

Generalizing the intuitions above, the foundational tool from the interventional tier is known as *do*-calculus [31], which allows analysts to take both observational data and a causal model, and answer interventional queries.

Definition 4.1. ***(Intervention)*** *An intervention represents an external force that fixes a variable to a constant value (akin to random assignment if an experiment), and is denoted $do(X = x)$, meaning that X is fixed to the value x. This amounts to replacing the structural equation for the intervened variable with its fixed constant such that $f_X = x$ (eliciting the "mutilated submodel" M_x). This operation is also represented graphically by severing all inbound edges to X in G, resulting in an "interventional subgraph" G_x.*

To compare quantities at associational ($\mathcal{L}_1$) and interventional ($\mathcal{L}_2$) tiers, the probability of event Y happening given that variable X was observed to be x is denoted by $P(Y|X = x)$. The probability of event Y happening given that variable X was *intervened upon* and made to be x is denoted by $P(Y|\text{do}(X = x))$. For instance, in Figure 3a, the effect of intervention $do(X = x)$ would be to sever the edge $Z \to X$.

To assess the average causal effect of an ad on clickthroughs in Example 3.1, and assuming our setting conforms to the model in Figure 3a, we must compute X's influence on Y in homogeneous conditions of Z, weighted by the likelihood of each condition $Z = z$. This adjustment is accomplished through the Backdoor Criterion:

Definition 4.2. ***(Backdoor Criterion)*** *[21, pp. 61] Given an ordered pair of variables (X, Y) in a directed acyclic graph G, a set of variables Z satisfies the backdoor criterion relative to (X, Y) if:*

1. *No node in Z is a descendant of X*
2. *Z blocks every path between X and Y that contains an arrow into X*

The Backdoor Adjustment formula for computing causal effects ($\mathcal{L}_2$) from observational data ($\mathcal{L}_1$) is thus:

$$P(Y|do(X)) = \sum_{z \in Z} P(Y|X, Z = z) \cdot P(Z = z)$$

By employing the backdoor criterion, we control for the spurious correlative pathway $X \leftarrow Z \to Y$ to isolate the desired causal pathway $X \to Y$ in estimation of $P(Y|do(X))$. Numerically applied to the AdBot Example 3.1 (with $Z = \{\text{female}, \text{male}\} = \{0, 1\}$), and assuming the model in Figure 3a, we would find:

$$\begin{aligned}
P(Y = 1|do(X = 0)) &= P(Y = 1|X = 0, Z = 0)P(Z = 0) + P(Y = 1|X = 0, Z = 1)P(Z = 1) \\
&= 0.70 * 0.48 + 0.90 * 0.52 \\
&\approx 0.80 \\
P(Y = 1|do(X = 1)) &= P(Y = 1|X = 1, Z = 0)P(Z = 0) + P(Y = 1|X = 1, Z = 1)P(Z = 1) \\
&= 0.65 * 0.48 + 0.85 * 0.52 \\
&\approx 0.75
\end{aligned}$$

From this adjustment, we confirm that displaying Ad $X = 0$ has the highest average causal effect on clickthrough rates.

4.2 Causal Recipes for Feature Selection

The power of *do*-calculus means ML algorithms can understand causal effects without having to perform experiments or be trained on experimental data.[7] This has implications for ML feature-selection: unless the causal structure is consulted, a collider might be conditioned on without conditioning on non-colliders along the path from action X to outcome Y, introducing bias. Consider the M-graph of Figure 3b: variables U_1 and U_2 cannot be included in the feature vector of an ML model because they are unobserved, and if Z is included in the feature vector, this model will produce correlative, but not causal, predictions.

Revisiting the three linear regression examples of section 3.2, Example 3.2 poses a decision to use a feature vector consisting of $\langle X, Z\rangle$ or just $\langle X\rangle$. Since the data generating process makes Z a function of X, and Y a function of Z, the DAG of figure 1a corresponds to this model. The DAG makes it clear that by including Z in the feature vector, we are conditioning on a mediator, thus blocking X's influence on Y and preventing the correct calculation of the causal effect of X on Y. This can be seen from the fact that $Y \perp\!\!\!\perp X \mid Z$, therefore, $E(Y|do(X), Z) = E(Y|Z)$:

Since there are no backdoor paths from X to Y, the causal effect can be predicted by not including Z in the feature vector. Students are then left to debate the linearity assumption. Does every additional level of training intensity, within a reasonable range, yield the same increase in athletic performance? This application of $\mathcal{L}_2$ tools to get the causal effect of interest by including only X in the feature vector doesn't depend on linearity. So, the linearity discussions can aid intuition and lead to the generalization of dropping the linearity assumption.

Example 3.3 showcases the same feature vector decision, $\langle X, Z\rangle$ or $\langle X\rangle$. This time the corresponding DAG is Figure 3a, which was used to explain Simpson's paradox. The backdoor path $X \leftarrow Z \rightarrow Y$ must be blocked in order to have a model that predicts the causal effect of X, preparation, on Y, competitiveness. Blocking this backdoor between X, Y is accomplished by including Z in the feature vector.

Example 3.4 is a collider scenario depicted in the DAG of Figure 1c. Here, attention must be paid to including the collider Z in the feature vector. By including Z predictions will be far more accurate (in fact, excluding Z will make predictions simply the mean of Y). However, then a spurious correlation between X and Y will be created. The causal effect of X and Y will be non-zero, but the DAG makes it clear that the causal effect should be null. Therefore, we have to exclude Z from the feature vector if the ML model is to determine the causal effect of X, athletic performance, on Y, negotiating skill.

Students can extend the insights gained from the above (which are useful in eliciting insights distinguishing $\mathcal{L}_1$ and $\mathcal{L}_2$ in simple settings) in more complex models like the following that demands a synthesis of these modular lessons.[8]

Example 4.1. ***Feature Selection Playground*** *Consider the SCM in Figure 4 with treatment X, outcome Y, and covariates $\{R, T, W, V\}$. Determine which of the covariates should be included in addition to X in the feature vector Z to provide: (1) the most precise observational estimate of Y, $P(Y|X, Z)$, and (2) an unbiased estimates of the causal effect of X on Y, $P(Y|do(X), Z)$.*

From Figure 4, conventional wisdom allows for the inclusion of all covariates $Z = \{R, T, W, V\}$ to maximize precise prediction of Y for the $\mathcal{L}_1$ quantity $P(Y|X, R, W, T, V)$, but the causal quantity requires more selectivity. To control for all non-causal pathways requires that $Z = \{R\}$ alone, because (1) controlling for T opens the M-graph backdoor path from $X \leftrightarrow T \leftrightarrow Y$, (2) controlling for W blocks the direct path from $X \rightarrow W \rightarrow Y$, and (3) controlling for V opens a spurious pathway at collider $X \rightarrow V \leftarrow Y$. Thus, $Z = \{R\}$ serves as a back-door admissible set to allow for estimation of $P(Y|do(X))$ via adjustment like in Example 3.1.

7 The Backdoor Criterion is a special case of the *do*-calculus ruleset, which is proven complete: If its rules are insufficient for identifying a causal effect from observational data, then it is not possible to identify that causal effect.

8 Additional modular examples of "good and bad controls" using regression and SCMs can be found in [32].

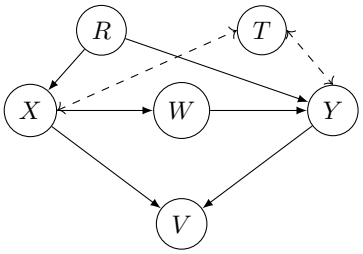


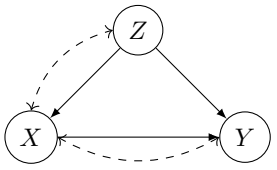
Figure 4. Feature selection playground depicting the causal diagram associated with treatment X, outcome Y, and other covariates.

4.3 Transportability & Data Fusion

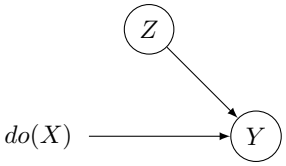
Although much of traditional ML education focuses on the ability or suitability of models to fit a particular dataset, there are several adjacent discussions that are commonly omitted, including: the qualitative differences between observational ($\mathcal{L}_1$) and experimental ($\mathcal{L}_2$) data, how these datasets can often be "fused" to support certain inference tasks, and how to take data collected at some tier in one environment / population and *transport* it to another. This *transportability problem* [33, 34] has long been studied in the empirical sciences under the heading of *external validity* [35, 36], and has received attention from the AI and ML communities under a variety of related tasks like transfer learning [37, 38] and model generalization [39–41]. Many modern techniques have focused on the ability to take a model trained in one environment and then to adapt it to a new setting that may differ in key respects. This capability is particularly palatable to fields that train agents in simulation settings to be later deployed in the real world, often because it is too risky, expensive, or otherwise impractical to perform the bulk of training in reality [42–44]. In general, when the training domain differs from the deployment domain (even slightly), predictions are biased, sometimes with significant model degradation. This often occurs when data from the deployment environment is limited, otherwise the ML model could have been trained on deployment data. To illustrate the utility of causal tools for this task, we provide a simple example in the domain of recommender systems that motivates distinctions in environments with heterogeneous data.

Example 4.2. ***DietBot.*** *You are designing an app that recommends diets $X \in \{0, 1\}$ (starting with only 2 for simplicity) that have been shown to interact with two strata of age $Z \in \{< 65, \geq 65\} = \{0, 1\}$ in how they predict heart health $Y \in \{unhealthy, healthy\} = \{0, 1\}$. The challenge: your model has been trained on experimental data from randomized diet assignment in a source environment, π (yielding the $\mathcal{L}_2$ distribution $P(Y, Z|do(X))$) that differs in its population's age distribution compared to a target environment, π^* in which you wish to deploy your app. From this target environment, you have only observations from surveys (yielding the $\mathcal{L}_1$ distribution $P^*(X, Y, Z)$), and (due to your budget) cannot conduct an experiment in this domain to determine the best diets to recommend to its population. Your task: without having to collect more data, determine the best policy your agent should adopt in π^* for maximizing the likelihood of users' health, i.e., find: $x^* = \underset{x}{\text{argmax}}\ P^*(Y = 1|do(X = x))$.*

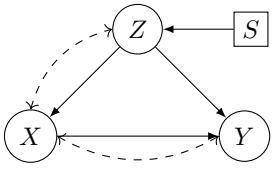
The training and deployment causal diagrams of Example 4.2 are depicted in Figure 5. Notably, because we conducted an experiment (i.e., performed an intervention) in environment π (Figure 5b, represented by the interventional subgraph G_x), the intervention $do(X)$ severs any of the would-be inbound edges to X in the observational setting that we see in the target environment π^* (Figure 5b, representing the unintervened graph G). Graphically, the challenge in the target environment becomes clear: we wish to estimate $P^*(Y = 1|do(X = x))$, but this causal effect is not identifiable because it is impossible to control for all backdoor paths between X, Y due to the presence of unobserved confounders indicated by the bidirected arcs. Yet, by assumption, the only difference between the two *environments* is the difference in age distributions such that $P(Z) \neq P^*(Z)$, so insights from the experiment conducted in π (in which the direct effect of $X \to Y$ has been isolated) may yet transport into π^*. To encode these assumptions of where



(a) Target/Deployment environment π^*, causal graph G, eliciting $P^*(X, Y, Z)$.



(b) Source/Training environment π, sub-model G_x, eliciting $P(Z, Y|do(X))$.



(c) Selection diagram D constructed from shared graph G.

Figure 5. Causal and selection diagrams for data collected in different environments, but employing the same underlying causal graph G.

structural differences occur between environments, and thus to determine if and how to transport we can make use of another graphical tool known as a *selection diagram*.

Definition 4.3. ***(Selection Diagram)****[45] Let $\langle M, M^* \rangle$ be two SCMs relative to environments $\langle \pi, \pi^* \rangle$ sharing a causal diagram G. By introducing selection nodes, boxed variables representing causes of variables that differ between source and target environment, $\langle M, M^* \rangle$ is said to induce a selection diagram D if D is constructed as follows:*

1. *Every edge in G is also an edge in D.*
2. *D contains an extra edge $S_i \rightarrow V_i$ (i.e., between a selection node and some other variable) whenever there might exist a discrepancy $f_i \neq f_i^*$ or $P(U_i) \neq P^*(U_i)$ between M and M^*.*

Importantly, selection diagrams encode *both* the differences in causal mechanisms between environments (via the presence of a selection node) *and* the similarities, with the assumption that any absence of a selection node represents the same local causal mechanisms between environments at that variable. In Example 4.2, the selection diagram requires only a single addition to G (Figure 5a): a selection node S representing the difference in age distributions at Z. Notationally, this also allows us to represent distributions in terms of the S variable, such that $S = s^*$ indicates that the population under consideration is the target π^*. Similarly, we can re-write distributions that are sensitive to selection like $P^*(Z) = P(Z|S = s*)$ and our target query from Example 4.2, $P^*(Y = 1|do(X)) = P(Y = 1|do(X), S = s^*)$. Doing so provides us a starting point for adjustment, similar to the backdoor adjustment from Example 3.1, wherein (using the rules of do-calculus) if we are able to find a sequence of rules to transform the target causal effect into an expression where the do-operator is independent from the selection variables, transportability is possible [46]. In the present DietBot Example, the goal is thus to phrase $P(Y = 1|do(X), S = s^*)$ in terms of our available data, $P(Z, Y|do(X))$ and $P^*(X, Y, Z)$. Such a derivation is as follows:

$$P(Y = 1|do(X = x), S = s^*) = \sum_z P(Y = 1, Z = z|do(X = x), S = s^*) \tag{1}$$

$$= \sum_z P(Y = 1|do(X = x), Z = z, S = s^*) P(Z = z|do(X = x), S = s^*) \tag{2}$$

$$= \sum_z P(Y = 1|do(X = x), Z = z) P(Z = z|do(X = x), S = s^*) \tag{3}$$

$$= \sum_z P(Y = 1|do(X = x), Z = z) P(Z = z|S = s^*) \tag{4}$$

$$= \sum_z P(Y = 1|do(X = x), Z = z) P^*(Z = z) \tag{5}$$

$$\tag{6}$$

Eqn. (1) follows from the law of total probability, (2) from the product rule, (3) from d-separation (because $Y \perp\!\!\!\perp S \mid Z, do(X = x)$), (4) from do-calculus (because, examining G_x, $Z \perp\!\!\!\perp do(X = x)$)[9], and (5) is simply a notational equivalence for the distribution of Z belonging to π^*.

$P(Y=1\|Z,do(X))$	$Z=0$	$Z=1$
$X=0$	**0.3**	**0.7**
$X=1$	**0.4**	**0.6**

	$P(Z)$	$P^*(Z)$
$Z=0$	**0.2**	**0.9**
$Z=1$	**0.8**	**0.1**

Table 2. Select distributions from environments π, π^* in Example 4.2.

While many theoretical lessons may end at the derivation of the transport formula concluding in Eqn. 5, including the numerical walkthrough using the parameters of Table 2 serves as an effective dramatization for why transportability has important implications for heterogeneous data and policy formation. Consider the scenario wherein the agent designer *did not* perform a transport adjustment between source and target domains, using only the model that would have been fit during training. In this risky setting, the agent would maximize the source environment's $P(Y = 1|do(X = x))$:

$$
\begin{aligned}
P(Y=1|do(X=x)) &= \sum_z P(Y=1|do(X=x), Z=z)P(Z=z) \\
P(Y=1|do(X=0)) &= P(Y=1|do(X=0), Z=0)P(Z=0) + P(Y=1|do(X=0), Z=1)P(Z=1) \\
&= 0.3 * 0.2 + 0.7 * 0.8 \\
&= 0.62 \\
P(Y=1|do(X=1)) &= P(Y=1|do(X=1), Z=0)P(Z=0) + P(Y=1|do(X=1), Z=1)P(Z=1) \\
&= 0.4 * 0.2 + 0.6 * 0.8 \\
&= 0.56
\end{aligned}
$$

Above, $P(Y = 1|do(X = 0)) > P(Y = 1|do(X = 1))$, meaning that the optimal choice in the training environment is $X = 0$. However, by properly applying the transport formula, we find that the opposite is true in the deployment environment:

$$
\begin{aligned}
P^*(Y=1|do(X=x)) &= \sum_z P(Y=1|do(X=x), Z=z)P^*(Z=z) \\
P^*(Y=1|do(X=0)) &= 0.3 * 0.9 + 0.7 * 0.1 \\
&= 0.34 \\
P^*(Y=1|do(X=1)) &= 0.4 * 0.9 + 0.6 * 0.1 \\
&= 0.42
\end{aligned}
$$

The DietBot Example provides a host of important lessons at $\mathcal{L}_2$ of the causal hierarchy, juxtaposing different causal inferences that would be obtained in different environments, demonstrating the utility of graphical models and do-calculus, and the dangers of unobserved confounding. Though these theoretical premises are typically taught in the study of causality in the empirical sciences, its practical utility in AI and ML can be driven home by casting transportability in terms of "training and deployment" environments, and by showing the surprise of opposite inferences that would be drawn with and without adjustment.

9 This rule is more formally stated in the specific rules of do-calculus as Rule 3: Deletion of Actions, whose full coverage may be a diversion from topics in traditional AI courses, though would feature prominently on a course with focus in causal inference. See [1].

As learning data scientists, students also obtain insights into the risks and opportunities of heterogeneous data, and how their fusion can overcome an otherwise difficult task of training and deployment environment differences. Plainly, in practice, adjustment formulae would not necessarily be computed by hand like in the above, but the experience of the demonstration is valuable for students; a fuller treatment of automated tools used in transportability can be found in [10, 34, 46].

Instructor Reflections

The surprise experienced in associational exercises and questions of section 3 continues with the interventional exercises for feature selection, transportability, and data fusion. More than just discussions arising from the revelations $\mathcal{L}_2$ brings, high school students have shown a keen interest in immediately using $\mathcal{L}_2$ tools to explain everyday experiences, and then learning how to encode those using a formal vocabulary.

Instructors of introductory courses in artificial intelligence have expressed frustrations discussing probabilistic models like Bayesian networks as ad-hoc or supporting topics that lack an impactful conclusion. However, examining these graphical models through the causal lens yields a fruitful experience for students to move beyond the probability calculus and the mantra that "correlation does not equal causation." Though this mantra is indeed true in general, there is a lesson to be learned in its dual: causation *does* bestow some structure to observed correlations, and this structure can be harnessed in support of many tasks that lead beyond the data alone.

By using the intuitions of d-separation as the structure of independence relationships in Bayesian networks, this strictly graphical explanation of the data serves as an effective stepping stone into Causal Bayesian Networks (CBNs) and SCMs; by completing this transition, instructors can more fully develop students' understanding of how probability leads to policy. This insight is clearly illustrated by use of graphical models in which observations and interventions can disagree (as in Example 3.1, $P(Y|X) \neq P(Y|do(X))$), how the environments and circumstances of data collection powerfully matter (as in Example 4.2, $\text{argmax}_x P(Y = 1|do(X = x)) \neq \text{argmax}_x P^*(Y = 1|do(X = x))$), and in causal discovery exercises for which an equivalence class of observationally equivalent models may explain some dataset, only some of which may follow a defendable causal explanation.

Along this path, students may struggle to understand the notion of latent variables and unobserved confounding unless the following are explained in unison: (1) the graphical depiction provides a causal explanation for *where* latent, outside influences may be present, and (2) *how* these influences outside of the model yield differences in causal $\mathcal{L}_2$ and non-causal $\mathcal{L}_1$ inferences that the data can provide.

5 Counterfactuals

The counterfactual layer of the hierarchy, $\mathcal{L}_3$, both subsumes and expands upon the previous two, newly allowing for an expression of queries akin to asking: "What if an event had happened differently than it did in reality?" Humans compute such queries often and with ease (as can be elicited from a classroom), especially through the experience of regret, which envisions a better outcome to an unchosen action. Regret is of great utility for dynamic agents, as it informs policy changes for future actions made in similar circumstances (e.g., the utterance of "Had I only exited the freeway earlier, I would not have gotten stuck in traffic" may bias future trips along the route to take side streets instead).

Counterfactual expressions are valuable to reasoning agents for a number of reasons, including that: (1) they allow for insights beyond the observed data, as it is not possible to rewind time and observe the outcome of a different event than what happened; (2) they can be used to establish precedent of necessary and sufficient causes, important for agents needing to understand how actions affect their environment (e.g., "Would the patient have recovered *had they not* taken the drug?"); and (3) they can be used to quantify an

agent's regret, which can be used for specific kinds of policy iteration in even confounded decision-making scenarios.

5.1 Structural Counterfactuals

Despite the expressive and creative potential of counterfactuals, the common student's initial exposure to them risks being overly formal and notationally-heavy, often beginning with the following definition:

Definition 5.1. ***(Counterfactual)*** *[9, pp. 204] In a SCM M, Let X and Y be two subsets of endogenous variables such that $\{X, Y\} \in V$. The counterfactual sentence "Y would be y (in situation / instantiation of exogenous variables $U = u$), had X been x" is interpreted as the equality with $Y_x(u) = y$, where $Y_x(u)$ encodes the solution for Y in the mutilated structural system M_x where for every $V_i \in X$, the equation f_i is replaced with the constant x. Alternatively, we can write:*

$$Y_{M_x}(u) = Y_x(u) = Y_x$$

Ostensibly, a counterfactual appears similar to the definition of an intervention. However, whereas the *do*-operator expresses a population-level intervention across *all* possible situations $u \in U \; \forall \; u$, a counterfactual computes an intervention for a *particular* unit / individual / situation $U = u$. This new syntax allows us to write queries of the format $P(Y_{X=x} = y | X = x')$, which computes the likelihood that the query Y attains value y in the world where $X = x$ (the hypothetical *antecedent*), given that $X = x'$ was observed in reality. The clash between the observed evidence $X = x'$ and hypothetical antecedent $X = x$ motivates the need for the new subscript syntax, and demonstrates how the previous tiers of the hierarchy cannot express such a query.

These expressions are often a source of syntactic and semantic confusion for beginners; an anecdotally better strategy is to instead begin with a discrete, largely-deterministic, simple motivating example with a plain-English counterfactual query, and then to work backwards to the formalisms.

$$M = \begin{cases} S & \leftarrow f_s(A) = A \\ X & \leftarrow f_x(S) = S \\ W & \leftarrow f_w(S) = S \\ R & \leftarrow f_r(X, W) = X \vee W \\ A & \sim P(A) \end{cases}$$

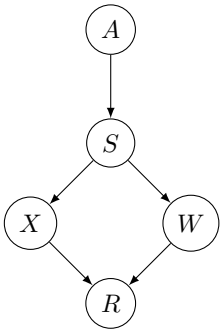


Figure 6. Structural Causal Model M and its associated graph G pertaining to Example 5.1.

Example 5.1. ***MediBot*** *An automated medical assistant, MediBot, is used to prescribe treatments for simple ailments, one of which has a policy designed around the following SCM containing Boolean variables to represent the presence of an ailment A, its symptom S, prescription of treatments X, W, and the recovery status of the patient R. The system abides by the SCM in Figure 6.*

Additionally, we are aware that the ailment's prevalence in the population is $P(A = 1) = 0.1$. Suppose we observe that MediBot prescribed treatment X (i.e., $X = 1$) to a particular patient u; determine the likelihood that the patient would recover from their ailment had it not prescribed this treatment (i.e., hypothesizing $X = 0$).

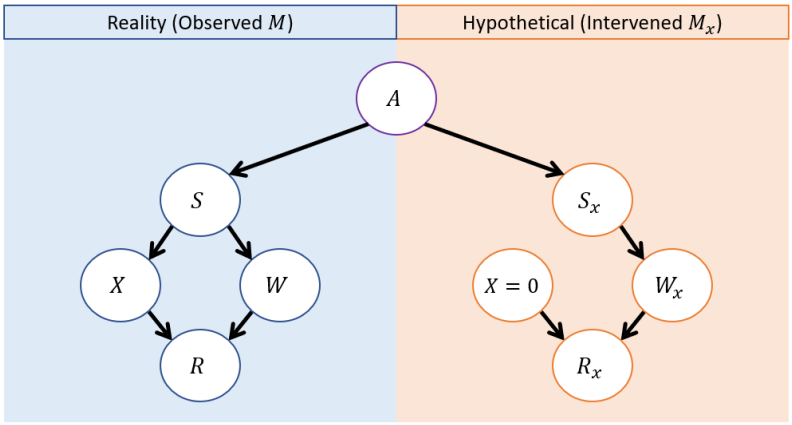


Figure 7. Twin network M^* for the SCM in Example 5.1 and counterfactual query $P(R_{X=0}|X=1)$.

To address this counterfactual query, intuitions best begin with the causal graph, whose observational state is depicted in Figure 6. Secondly, it is instructive to show how the previous layers' notations break down with the query of interest, as we cannot make sense of the contrasting evidence and hypothesis using the do-operator alone (i.e., the expression $P(R|do(X=0), X=1)$ is syntactically invalid, which is an instructive exercise for students to arrive at as well). Instead, the query of interest focuses upon the recovery state in the world where $X=0$, though in reality, $X=1$ was observed. This can be expressed via the counterfactual query $P(R_{X=0}|X=1)$, which can be teased in the lesson either before or after the computation mechanics that follow.

Before performing this computation, it is useful for students to visualize its steps. Intuitively, we expect that some information about our observed evidence $X=1$ may change our beliefs about the counterfactual query $R_{X=0}$; this information thus flows between the *observed* ($\mathcal{L}_1$, associational) and *hypothetical* ($\mathcal{L}_2$, interventional) worlds through the only source of variance in the system: the exogenous variables (in Example 5.1, A). Depicting this bridge can be accomplished through a technique known as the Twin Network Model [47].

Definition 5.2. ***(Twin Network Model)*** *For SCM M, arbitrary counterfactual query of the format $P(Y_x|x')$, and interventional submodel of the counterfactual antecedent M_x, the Twin Network Model M^* is also an SCM defined as a combination of M and M_x with the following traits:*

1. *The structures of M and M_x are identical (including the same structural equations), except that all inbound edges to X in M_x are severed.*
2. *All exogenous variables are shared between M and M_x in M^*, since these remain invariant under modification.*
3. *All endogenous variables in the hypothetical M_x are labeled with the same subscript to distinguish them from their un-intervened counterparts, as they may obtain different values.*

The Twin Network of the SCM in Example 5.1 is depicted in Figure 7. This model is not only an intuitive depiction of the means of computing the query at-hand, but also serves the practical purposes of being a model through which standard evidence propagation techniques can be used to update beliefs from evidence to antecedent, and through which the standard rules of d-separation can be used to determine independence relations between variables in counterfactual queries. It is also useful to examine some axioms of counterfactual notation at this point, noting the equivalence of certain $\mathcal{L}_3$ expressions with previous tiers, like $P(R_{X=x}) = P(R|do(X=x))$ (the "potential outcomes" subscripted format for writing the $\mathcal{L}_2$ intervention) and $P(R_{X=x}|X=x) = P(R|X=x)$ (the *consistency axiom* in which antecedent and observed evidence are the same, making it an observational quantity from $\mathcal{L}_1$).

Returning to our example, the actual computation of $P(R_{X=0}|X=1)$ follows a 3-step process motivated by the twin-network representation.

Step 1: Abduction. The abduction step updates beliefs about the shared exogenous variable distributions based on observed evidence, meaning we effectively replace $P(u) \leftarrow P(u|e)$. In the current, largely deterministic example, this amounts to propagating the evidence that $X=1$ through the rest of $M \in M^*$, which can be trivially shown to indicate that all variables attain a value of 1 with certainty. However, for the abduction step, we need only update beliefs about the exogenous variable, A:

$$X = 1, f_x(S) = S \Rightarrow S = 1$$
$$S = 1, f_s(A) = A \Rightarrow A = 1$$

Step 2: Action. With $P(u) \leftarrow P(u|e)$ (viz., $P(A=1|X=1)=1$), we can effectively discard / ignore the observational model M and shift to the hypothetical twin M_x, forcing $X=x$ per the counterfactual antecedent, which in our example, means severing all inbound edges to X in M_x and forcing its value to $X=0$. Let M^* be the modified model following steps 1 and 2.

$$M^* = \begin{cases} S & \leftarrow f_s(A) = A \\ X & \leftarrow 0 \\ W & \leftarrow f_w(S) = S \\ R & \leftarrow f_r(X,W) = X \vee W \\ A & \sim P(A|X=1) \end{cases}$$

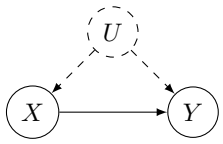
Step 3: Prediction. Finally, we perform standard belief propagation within the modified M^* to solve for our query variable, R_x, and find that the patient would indeed still have recovered (i.e., $P(R_{X=0}=1)=1$) because MediBot would still have also administered the other effective treatment, $W_x=1$.

$$P(A=1|X=1)=1, S \leftarrow f_s(A)=A \Rightarrow S_x = 1$$
$$S_x = 1, W \leftarrow f_w(S) = S \Rightarrow W_x = 1$$
$$W_x = 1, R \leftarrow f_r(X,W) = X \vee W \Rightarrow R_x = 1$$

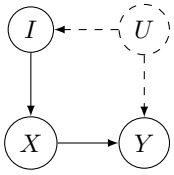
This simple example not only demonstrates the mechanics and potential of structural counterfactuals, but also serves as a launchpad for more intricate and challenging applications. Worthwhile follow-on exercises include the addition of noisy exogenous variables to the system in Example 5.1 (e.g., non-deterministic patient recovery), and analogies to linear SCMs in which the 3-step process is repeated through application of conditional expectation. Moreover, the example leads into questions of necessity and sufficiency [48] of the medical treatments, which can segue into other, more applied and data-driven, counterfactuals.

5.2 Counterfactuals for Metacognitive Agents

In some more adventurous explorations in AI oriented at crafting self-improving and reflective artificial agents, counterfactuals in $\mathcal{L}_3$ may prove to be a useful tool for *metacognitive* agents [49, 50]. Related to the transportability problem with DietBot in Example 4.2, agents may find the need to evolve their policies learned earlier in their lifespan or in environments that change over time in order to optimize their performance. This need complements a growing area of reinforcement learning that incorporates causal concepts, especially with respect to meta-learning [13, 51, 52]. To demonstrate such a scenario, we reconsider MediBot in a setting wherein its current policy's decisions are confounded, damaging its performance and requiring it to perform some measure of metacognition to improve that is analogous to the human experience of regret.



(a) Observational model.



(b) The same system, but with intent explicitly modeled.

Figure 8. SCM associated with Example 5.2 with treatment X, recovery Y, UC U, and intent I.

Example 5.2. ***Confounded MediBot***[10]. *MediBot is back assigning treatment for a separate condition in which two treatments $X \in \{0, 1\}$ have been shown to be equally effective remedies by an FDA randomized clinical trial (i.e., $P(Y = 1|do(X = 0)) = P(Y = 1|do(X = 1))$ where $Y = 1$ indicates recovery). As such, patients are given the option to choose between the two treatments for the final prescription given. Seemingly innocuous, this patient-choice is actually problematic given the following wrinkles:*

1. *The patient's treatment request is actually affected by an unobserved confounder (UC), linking the treatment and recovery through an uncontrolled back-door path (Fig. 8a). This unobserved, exogenous variable U is unrecorded in the data, and could potentially be anything, like the influence of direct-to-consumer advertising of drug treatments that are primarily observed by different treatment-sensitive subpopulations (like a drug that is only advertised on sports-radio with a primarily exercise-friendly audience).*
2. *Because of this confounding influence, MediBot's observed recovery rates are actually less than the FDA's reported ones (Table 3). Worse is that the observed ($\mathcal{L}_1$) and experimental ($\mathcal{L}_2$) recovery rates look equivalent within each respective tier, making it a challenge to determine whether a superior, individualized treatment exists.*

	$P(Y = 1\|X)$	$P(Y = 1\|do(X))$
$X = 0$	**0.50**	**0.70**
$X = 1$	**0.50**	**0.70**

Table 3. MediBot's observed treatment recovery rates vs. those reported by the FDA's randomized clinical trials.

The data in Table 3 demonstrates the tell-tale sign of unobserved confounding wherein the observed and experimental treatment effects differ ($P(Y|x) \neq P(Y|do(x))\ \exists x \in X$), implicating an uncontrolled latent factor that explains the difference. Surprisingly, despite the unknown identity of the confounder, a better treatment policy than MediBot's current one does indeed exist in this context, and is derived from a counterfactual quantity known as the Effect of Treatment on the Treated (ETT). The ETT traditionally computes the difference between the effect of an alternate treatment $X = x$ than the one actually given to an individual $X = x'$, which can be expressed in this context as $P(Y_{X=x} = 1|X = x'), x \neq x'$.

With only the partially specified model, and the observational and experimental recovery rates, it is possible to compute the ETT for binary treatments (assuming, in this setting, that the patient requested treatments are observed in equal proportion, $P(X = 0) = P(X = 1) = 0.5$), as in the following derivation that is true for any treatment $X = x$ and its alternative $X = x'$.

10 The simplicity of Example 5.2 should not undermine the prevalence of confounded decision-making scenarios that are found in many adversarial settings with traditional machine learning [53] and a myriad of human-decider-AI-recommender scenarios [15]

$$
\begin{aligned}
P(Y_x) &= P(Y_x|x')P(x') + P(Y_x|x)P(x) \\
&= P(Y_x|x')P(x') + P(Y|x)P(x) \\
P(Y_x = 1|x') &= \frac{P(Y_x = 1) - P(Y = 1|x)P(x)}{P(x')} \\
&= \frac{0.7 - 0.5 * 0.5}{0.5} \\
&= 0.9
\end{aligned}
$$

This algebraic trick (using only the law of total probability) allows us to derive $\mathcal{L}_3$ quantities of interest from a combination of $\mathcal{L}_1$ and $\mathcal{L}_2$ data (though only for binary treatment), and tells an important tale about the system: MediBot is presently in a state of *inevitable regret* [54] in which the likelihood of recovery for those given treatment under its policy ($P(Y = 1|X = x') = 0.5 \; \forall \; x' \in X$) is 40% less than had those same patients been treated differently ($P(Y_{X=x} = 1|X = x') = 0.9 \; \forall \; x \neq x' \in X$).

Ostensibly, this computation yields only bleak retrospect, but also leads to a surprising remedy for online agents. Two insights contribute to the solution, known as *intent-specific decision-making* [13]: (1) the formation of the confounded agent's observational / naturally decided action (i.e., its *intent*) can be separated from the ultimately chosen one, and (2) this intended action choice serves as a back-door admissible proxy for the state of the UC (see Figure 8(b) with agent intent I).

Definition 5.3. ***(Intent)*** *[13] In a confounded decision-making scenario with desired outcome $Y = 1$, final agent choice X, unobserved confounder(s) U_c, and structural equation $X \leftarrow f_x(U_c)$, SCMs modeling the agent's intent I represent its pre-choice $\mathcal{L}_1$ response to $U_c = u_c$ such that I adopts the structural equation of X with $I \leftarrow f_i(U_c) = f_x(U_c)$, and the structural equation for X indicates that, observationally, the final choice always follows the intended, $X \leftarrow f_x(I) = I$.*

When intent is explicitly modeled in a confounded decision-making scenario (Figure 8b), the ETT (previously, a retrospective $\mathcal{L}_3$ quantity) can be measured empirically *before* a decision is made by using *do*-calculus conversions to a $\mathcal{L}_2$ quantity through a process known as Intent-Specific Decision-Making.

Definition 5.4. ***(Intent-Specific Decision-Making (ISDM))*** *[14, 15, 55] In the context of a confounded decision-making scenario with decision X, intent of that decision I, and desired outcome $Y = 1$, the counterfactual $\mathcal{L}_3$ expression $P(Y_{X=x} = 1|X = x')$, $x, x' \in X$ may be measured empirically via the intent-specific $\mathcal{L}_2$ expression $P(Y = 1|do(X = x), I = x')$, namely:*

$$
P(Y_{X=x} = 1|X = x') = P(Y = 1|do(X = x), I = x'), \; x, x' \in X, I \tag{7}
$$

The confounded agent can thus choose the action that maximizes the counterfactual ETT to develop a meta-policy that will always be equally, or more, effective than actions chosen by maximizing either the observational or experimental recovery rates. This technique is known as the *regret decision criteria (RDC)* [13] and can be expressed (for action X, intent I, and desired outcome $Y = 1$) as:

$$
x^* = \underset{x}{\text{argmax}} \; P(Y_{X=x} = 1|I = x') = \underset{x}{\text{argmax}} \; P(Y = 1|do(X = x), I = x')
$$

For Example 5.2, students could find (either analytically through Table 3 or experientially through a contextual multi-armed bandit assignment) that $P(Y_{X=1} = 1|X = 0) = 0.9 > P(Y_{X=0} = 1|X = 0) = 0.5$, meaning that in settings wherein MediBot intends to treat with $X = 0$, it is better off choosing $X = 1$. The full intent-specific distribution of expected recovery rates is shown in Table 4.

The RDC is useful because (1) it allows a confounded agent to make strictly better decisions as a function of a confounding-sensitive existing policy (in Ex. 5.2, by prescribing the treatment opposite its first intended), even in complete naivety of the confounding factors, (2) it provides an *empirical* means of sampling a counterfactual datapoint (surprising given the mechanics of counterfactuals) [15], and (3) it

$P(Y_{X=x} = 1 \mid I = x')$	$I = 0$	$I = 1$
$X = 0$	0.5	0.9
$X = 1$	0.9	0.5

Table 4. Intent-specific recovery rates for the confounded MediBot Example 5.2.

can be intuitively rooted for students in the familiar experience of beginning to do something once regretted, stopping, and then choosing differently. Example 5.2 thus addresses a number of learning outcomes, including the clear distinctions of quantities at all three tiers of the causal hierarchy, how UCs can account for these differences, and how to design agents that either exploit or are resilient to them.

Instructor Reflections

Motivating the utility of counterfactual inference can begin with active learning through a Socratic dialogue, rooting the capacity for human counterfactual reasoning in experiences like regret. "Why do we not return to restaurants that give us food poisoning? How do we place this blame? Would we have gotten food poisoning *had we not* eaten there?" Transitioning from these intuitions to why artificial agents can benefit from the ability to answer similar questions can make for an enjoyable classroom discussion. In classes or levels with more room for debate, discussions on counterfactuals as the origins of human creativity may also yield fruitful explorations. More broadly, the ability of counterfactuals to "escape from the data" can offer inspiration; students enjoy mention of the Lion Man of Ur (an ice-age sculpture depicting a humanoid figure that is half-lion), which demonstrates the ability of humans to conceive of ideas without a bearing in reality [56].

More formally, situating counterfactuals in the PCH can provide a bridge to other courses or contexts in which the term is used, such as in Rubin's potential-outcomes framework [57] or in philosophical and logical discourse [58]. By proceduralizing counterfactual computation in the structural 3-step approach, students not only appreciate the reasoning mechanics underlying these others approaches, but receive hints on future applications in the domain of AI.

6 Conclusion

In this work, we have endeavored to not only impress the importance of causal topics to the future of AI and ML, but have also provided instructor-ready content to supplement existing AI curricula. Through this earlier exposure to causal concepts, we invite a new generation of data scientists, machine learning practitioners, and designers of autonomous agents to employ and extend these tools to address problems beyond the empirical sciences. Though this work provides only a cursory exposure to the many possible avenues of synthesis for causality and AI, students familiar with its contents will more deeply understand their data, models, and the types of questions that each are capable of answering. As the demands of artificial agents continue to extend beyond associations, practitioners familiar with causal concepts will be equipped to address the needs of tomorrow apart from only the data of today.

Funding: This research was supported in parts by grants from the National Science Foundation [#IIS-2106908], Office of Naval Research [#N00014-17-S-12091 and #N00014-21-1-2351], and Toyota Research Institute of North America [#PO-000897].

References

[1] Pearl J. Causal diagrams for empirical research. Biometrika. 1995;82(4):669–688.
[2] Fisher FM. A correspondence principle for simultaneous equation models. Econometrica: Journal of the Econometric Society. 1970:73–92.
[3] Machamer P, Darden L, Craver CF. Thinking about mechanisms. Philosophy of science. 2000;67(1):1–25.
[4] Mackie JL. The cement of the universe: A study of causation. Oxford: Clarendon Press; 1974.
[5] Glymour C, Scheines R, Spirtes P. Discovering causal structure: Artificial intelligence, philosophy of science, and statistical modeling. Academic Press; 2014.
[6] Danks D. Unifying the mind: Cognitive representations as graphical models. Mit Press; 2014.
[7] Gopnik A. Scientific thinking in young children: Theoretical advances, empirical research, and policy implications. Science. 2012;337(6102):1623–1627.
[8] Penn DC, Povinelli DJ. Causal cognition in human and nonhuman animals: A comparative, critical review. Annual review of psychology. 2007;58.
[9] Pearl J. Causality: Models, Reasoning, and Inference. 2nd ed. New York: Cambridge University Press; 2009.
[10] Bareinboim E, Pearl J. Causal inference and the data-fusion problem. Proceedings of the National Academy of Sciences. 2016;113(27):7345–7352.
[11] Bengio Y, Deleu T, Rahaman N, Ke R, Lachapelle S, Bilaniuk O, et al. A meta-transfer objective for learning to disentangle causal mechanisms. arXiv preprint arXiv:190110912. 2019.
[12] Pearl J. Causal and counterfactual inference. The Handbook of Rationality. 2019:1–41.
[13] Bareinboim E, Forney A, Pearl J. Bandits with unobserved confounders: A causal approach. In: Advances in Neural Information Processing Systems; 2015. p. 1342–1350.
[14] Forney A, Pearl J, Bareinboim E. Counterfactual Data-Fusion for Online Reinforcement Learners. In: International Conference on Machine Learning; 2017. p. 1156–1164.
[15] Forney A, Bareinboim E. Counterfactual Randomization: Rescuing Experimental Studies from Obscured Confounding. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33; 2019. p. 2454–2461.
[16] Richens JG, Lee CM, Johri S. Improving the accuracy of medical diagnosis with causal machine learning. Nature Communications. 2020 Aug;11(1). Available from: https://doi.org/10.1038/s41467-020-17419-7.
[17] Yan JN, Gu Z, Lin H, Rzeszotarski JM. Silva: Interactively Assessing Machine Learning Fairness Using Causality. In: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems; 2020. p. 1–13.
[18] Makhlouf K, Zhioua S, Palamidessi C. Survey on Causal-based Machine Learning Fairness Notions. arXiv preprint arXiv:201009553. 2020.
[19] Vlontzos A, Kainz B, Gilligan-Lee CM. Estimating the probabilities of causation via deep monotonic twin networks. arXiv preprint arXiv:210901904. 2021.
[20] Pearl J. Theoretical impediments to machine learning with seven sparks from the causal revolution. arXiv preprint arXiv:180104016. 2018.
[21] Pearl J, Glymour M, Jewell NP. Causal Inference in Statistics: A Primer. Wiley; 2016.
[22] Gopnik A, Wellman HM. Reconstructing constructivism: Causal models, Bayesian learning mechanisms, and the theory theory. Psychological bulletin. 2012;138(6):1085.
[23] Bareinboim E, Correa JD, Ibeling D, Icard T. On Pearl's hierarchy and the foundations of causal inference. ACM Special Volume in Honor of Judea Pearl (provisional title). 2020;2(3):4.
[24] Geiger D, Verma T, Pearl J. d-separation: From theorems to algorithms. In: Machine Intelligence and Pattern Recognition. vol. 10. Elsevier; 1990. p. 139–148.
[25] Hyttinen A, Eberhardt F, Hoyer PO. Experiment selection for causal discovery. Journal of Machine Learning Research. 2013;14:3041–3071.
[26] Claassen T, Heskes T. Causal discovery in multiple models from different experiments. In: Advances in Neural Information Processing Systems; 2010. p. 415–423.
[27] Lübke K, Gehrke M, Horst J, Szepannek G. Why We Should Teach Causal Inference: Examples in Linear Regression With Simulated Data. Journal of Statistics Education. 2020;28(2):133–139. Available from: https://doi.org/10.1080/10691898.2020.1752859.
[28] Cummiskey K, Adams B, Pleuss J, Turner D, Clark N, Watts K. Causal Inference in Introductory Statistics Courses. Journal of Statistics Education. 2020;28(1):2–8. Available from: https://doi.org/10.1080/10691898.2020.1713936.
[29] Fisher R. The Design of Experiments. 6th ed. Edinburgh: Oliver and Boyd; 1951.
[30] Balke A, Pearl J. Bounds on treatment effects from studies with imperfect compliance. Journal of the American Statistical Association. 1997 September;92(439):1172–1176.
[31] Pearl J. Causal diagrams for empirical research. Biometrika. 1995;82(4):669–710.
[32] Cinelli C, Forney A, Pearl J. A crash course in good and bad controls. Available at SSRN. 2020;3689437.

[33] Bareinboim E, Pearl J. Causal Transportability with Limited Experiments. In: desJardins M, Littman M, editors. Proceedings of the Twenty-Seventh National Conference on Artificial Intelligence (AAAI 2013). Menlo Park, CA: AAAI Press; 2013. p. 95–101.
[34] Subbaswamy A, Schulam P, Saria S. Learning predictive models that transport. arXiv preprint arXiv:181204597. 2018.
[35] Pearl J, Bareinboim E. External validity: From do-calculus to transportability across populations. Statistical Science. 2014;29(4):579–595.
[36] Manski CF. Identification for prediction and decision. Harvard University Press; 2009.
[37] Torrey L, Shavlik J. Transfer learning. In: Handbook of research on machine learning applications and trends: algorithms, methods, and techniques. IGI global; 2010. p. 242–264.
[38] Weiss K, Khoshgoftaar TM, Wang D. A survey of transfer learning. Journal of Big data. 2016;3(1):1–40.
[39] Chung Y, Haas PJ, Upfal E, Kraska T. Unknown examples & machine learning model generalization. arXiv preprint arXiv:180808294. 2018.
[40] Bousquet O, Elisseeff A. Stability and generalization. The Journal of Machine Learning Research. 2002;2:499–526.
[41] Kawaguchi K, Kaelbling LP, Bengio Y. Generalization in deep learning. arXiv preprint arXiv:171005468. 2017.
[42] Talpaert V, Sobh I, Kiran BR, Mannion P, Yogamani S, El-Sallab A, et al. Exploring applications of deep reinforcement learning for real-world autonomous driving systems. arXiv preprint arXiv:190101536. 2019.
[43] Paleyes A, Urma RG, Lawrence ND. Challenges in deploying machine learning: a survey of case studies. arXiv preprint arXiv:201109926. 2020.
[44] Lwakatare LE, Raj A, Crnkovic I, Bosch J, Olsson HH. Large-scale machine learning systems in real-world industrial settings: A review of challenges and solutions. Information and Software Technology. 2020;127:106368.
[45] Bareinboim E, Pearl J. Transportability of causal effects: Completeness results. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 26; 2012. .
[46] Bareinboim E, Pearl J. Transportability from multiple environments with limited experiments: Completeness results. Advances in neural information processing systems. 2014;27:280–288.
[47] Balke A, Pearl J. Probabilistic evaluation of counterfactual queries. In: Proceedings of the twelfth national conference of the Association for the Advancement of Artificial Intelligence. Seattle, Washington: AAAI; 1994. p. 230–237.
[48] Tian J, Pearl J. Probabilities of causation: Bounds and identification. Annals of Mathematics and Artificial Intelligence. 2000;28(1):287–313.
[49] Cox MT. Metacognition in computation: A selected research review. Artificial intelligence. 2005;169(2):104–141.
[50] Savitha R, Suresh S, Sundararajan N. Metacognitive learning in a fully complex-valued radial basis function neural network. Neural computation. 2012;24(5):1297–1328.
[51] Dasgupta I, Wang J, Chiappa S, Mitrovic J, Ortega P, Raposo D, et al. Causal reasoning from meta-reinforcement learning. arXiv preprint arXiv:190108162. 2019.
[52] Zhang J. Designing optimal dynamic treatment regimes: A causal reinforcement learning approach. In: International Conference on Machine Learning. PMLR; 2020. p. 11012–11022.
[53] Biggio B, Roli F. Wild patterns: Ten years after the rise of adversarial machine learning. Pattern Recognition. 2018;84:317–331.
[54] Pearl J. The curse of free-will and the paradox of inevitable regret. Journal of Causal Inference. 2013;1(2):255–257.
[55] Forney A. A Framework for Empirical Counterfactuals, or For All Intents, a Purpose. University of California, Los Angeles; 2018.
[56] Pearl J, Mackenzie D. The book of why: the new science of cause and effect. Basic books; 2018.
[57] Rubin DB. Causal inference using potential outcomes: Design, modeling, decisions. Journal of the American Statistical Association. 2005;100(469):322–331.
[58] Alonso-Ovalle L. Counterfactuals, correlatives, and disjunction. Linguistics and Philosophy. 2009;32(2):207–244.