Application of Causal Inference to Interpretable Machine Learning

Yonghan Jung

Purdue University

yonghanjung.me

2022.07.12

University of Seoul 2022

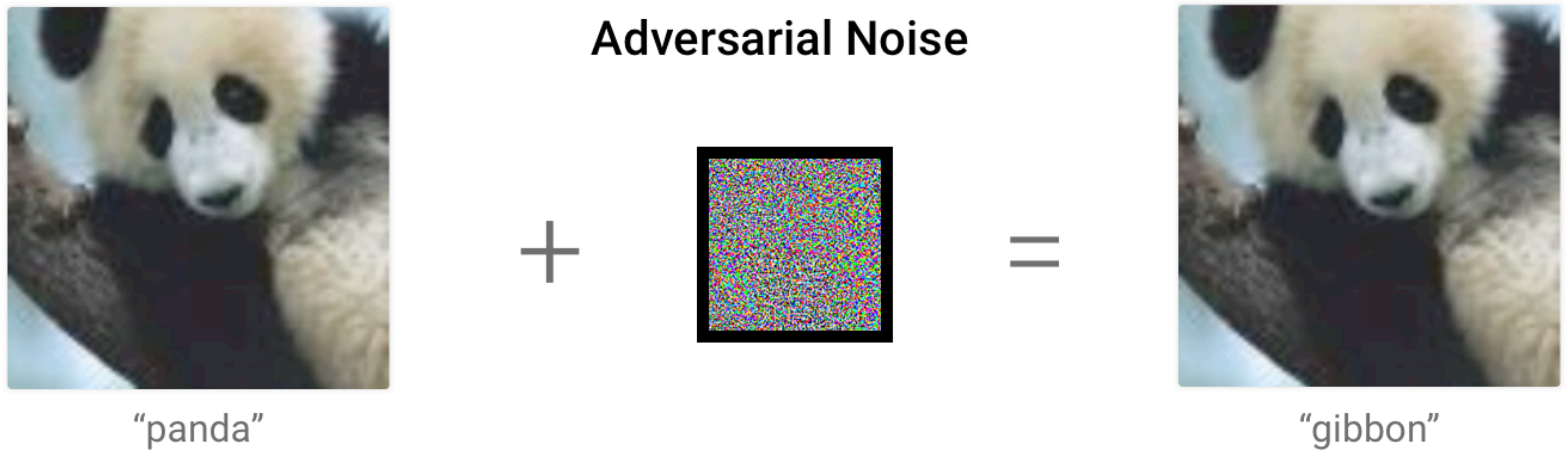
Corresponding Paper

On Measuring Causal Contributions via do-interventions

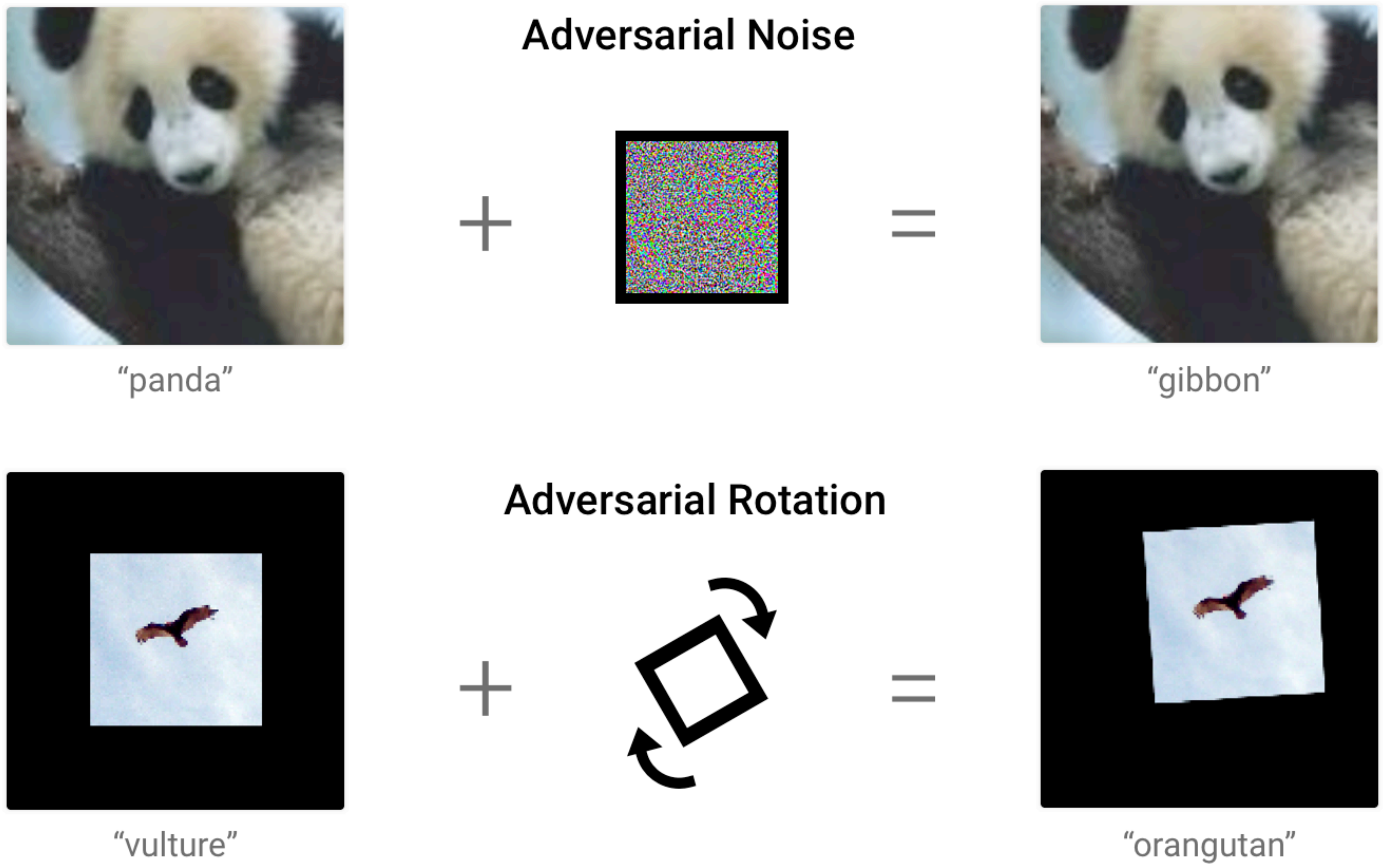
Yonghan Jung, Shiva Kasiviswanathan, Jin Tian, Dominik Janzing, Patrick Bloebaum, Elias Bareinboim *Proceedings of the 39th International Conference on Machine Learning*, PMLR 162:10476-10501, 2022.

Challenges of interpretation in ML

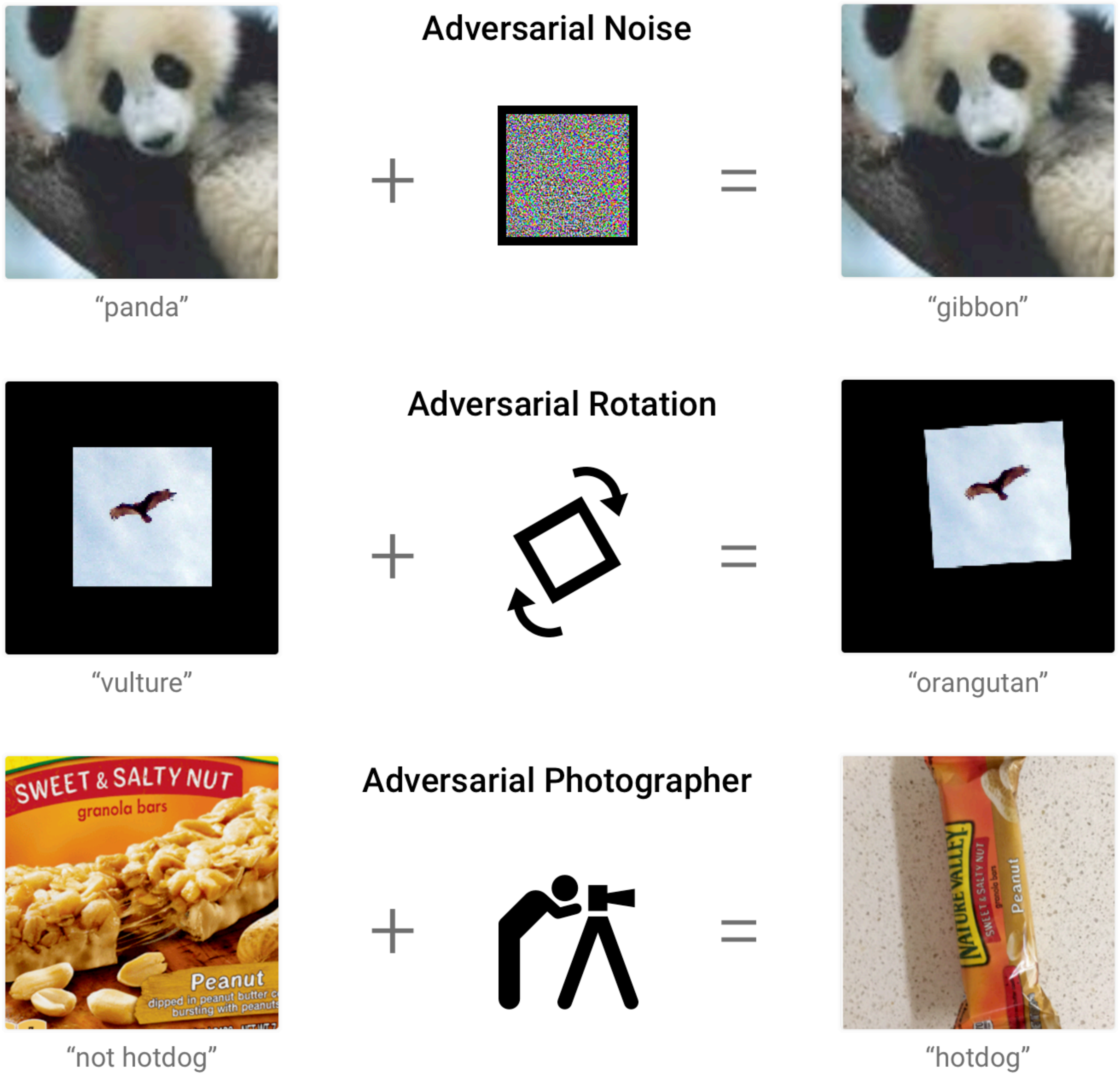
Challenges of interpretation in ML



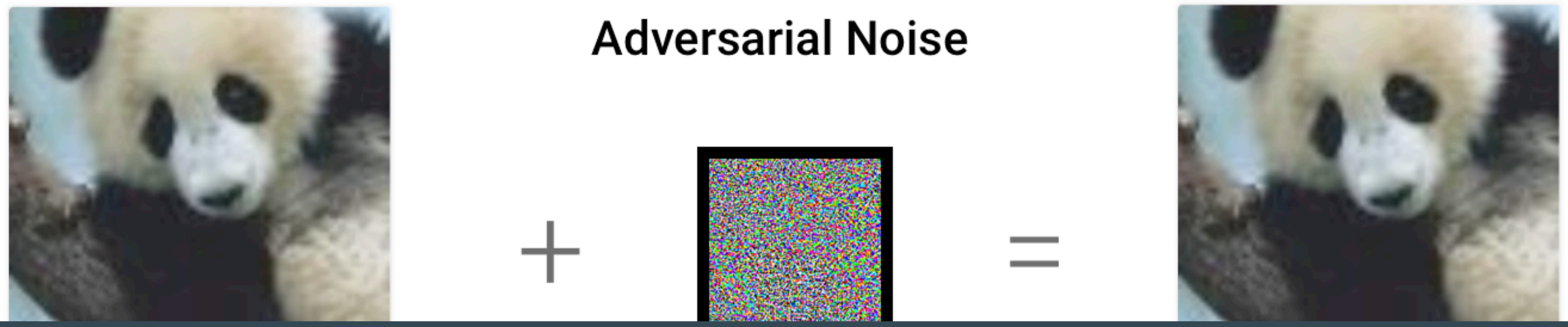
Challenges of interpretation in ML



Challenges of interpretation in ML



Challenges of interpretation in ML



Interpreting behaviors of ML result is important!

"not hotdog" "hotdog"

What is interpretability?

Even if there are no technical definitions due to the perceived subjectivity, common *consensus* is the following:

What is interpretability?

Even if there are no technical definitions due to the perceived subjectivity, common *consensus* is the following:

Interpretability is the degree to which a human can

What is interpretability?

Even if there are no technical definitions due to the perceived subjectivity, common *consensus* is the following:

Interpretability is the degree to which a human can

1. consistently predict the model's result [Kim et al., 2016]

What is interpretability?

Even if there are no technical definitions due to the perceived subjectivity, common *consensus* is the following:

Interpretability is the degree to which a human can

1. consistently predict the model's result [Kim et al., 2016]
2. understand the cause of a prediction [Miller, 2019]

What is interpretability?

Even if there are no technical definitions due to the perceived subjectivity, common *consensus* is the following:

Interpretability **is the degree to which a human can**

1. consistently predict the model's result [Kim et al., 2016]
2. understand the cause of a prediction [Miller, 2019]

This leads "Feature attribution task" taking account of Causality!

Task of Interpretable Machine Learning

Feature attribution given $(\mathbf{v}, f(\mathbf{v}))$

Task of Interpretable Machine Learning

Feature attribution given $(\mathbf{v}, f(\mathbf{v}))$

- **Input**: A pair of $(\mathbf{v}, f(\mathbf{v}))$, where $f(\mathbf{v})$ is a black-box machine learning model prediction for some input $\mathbf{v} = \{v_1, v_2, \cdots, v_n\}$ (where x_i means the ith feature).

Task of Interpretable Machine Learning

Feature attribution given $(\mathbf{v}, f(\mathbf{v}))$

- **Input**: A pair of $(\mathbf{v}, f(\mathbf{v}))$, where $f(\mathbf{v})$ is a black-box machine learning model prediction for some input $\mathbf{v} = \{v_1, v_2, \cdots, v_n\}$ (where x_i means the ith feature).

- **Output**: A vector $attr(f, \mathbf{v}) \equiv \{\phi_{v_1}, \cdots, \phi_{v_n}\}$ where ϕ_i is an importance of v_i on $f(\mathbf{v})$.

Task of Interpretable Machine Learning

Feature attribution given $(\mathbf{v}, f(\mathbf{v}))$

- **Input**: A pair of $(\mathbf{v}, f(\mathbf{v}))$, where $f(\mathbf{v})$ is a black-box machine learning model prediction for some input $\mathbf{v} = \{v_1, v_2, \cdots, v_n\}$ (where x_i means the ith feature).
- **Output**: A vector $attr(f, \mathbf{v}) \equiv \{\phi_{v_1}, \cdots, \phi_{v_n}\}$ where ϕ_i is an importance of v_i on $f(\mathbf{v})$.

💡 This task is called ***local*** (or 'unit') explanation since it only consider an individual input-output pair $(\mathbf{v}, f(\mathbf{v}))$.

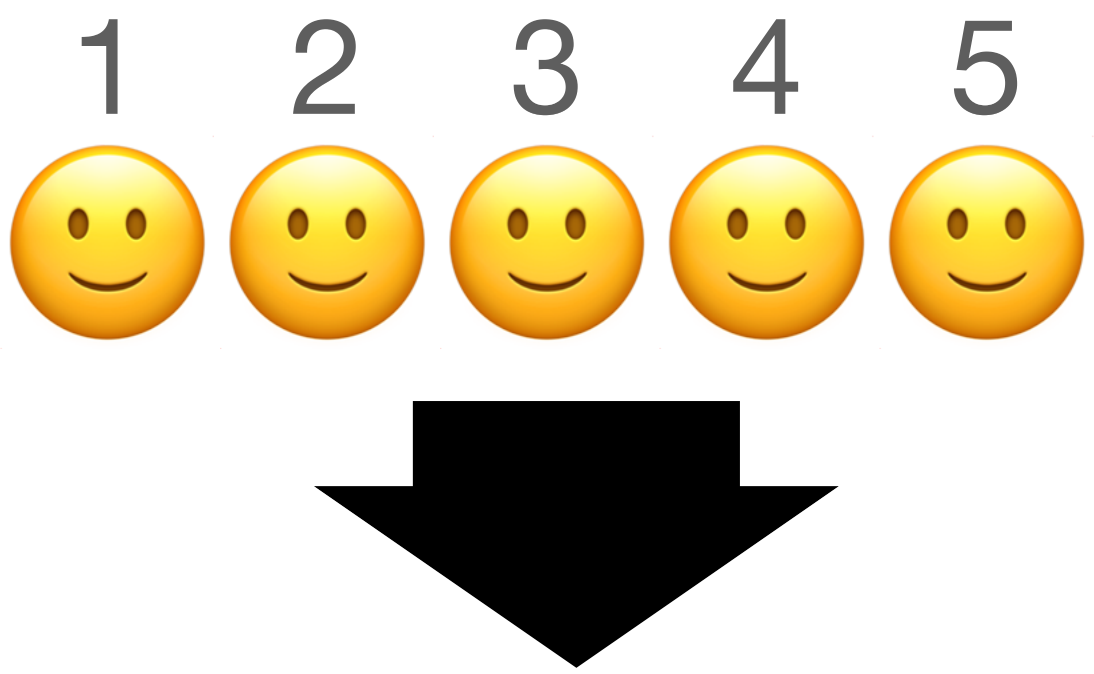
Attribution task in a coalition game - 1

Attribution task in a coalition game - 1

- **Coalition game** — The prediction result with n features (called coalition function $v([n])$) made by those n players, how do we attribute the payoff to each individual players?

Attribution task in a coalition game - 1

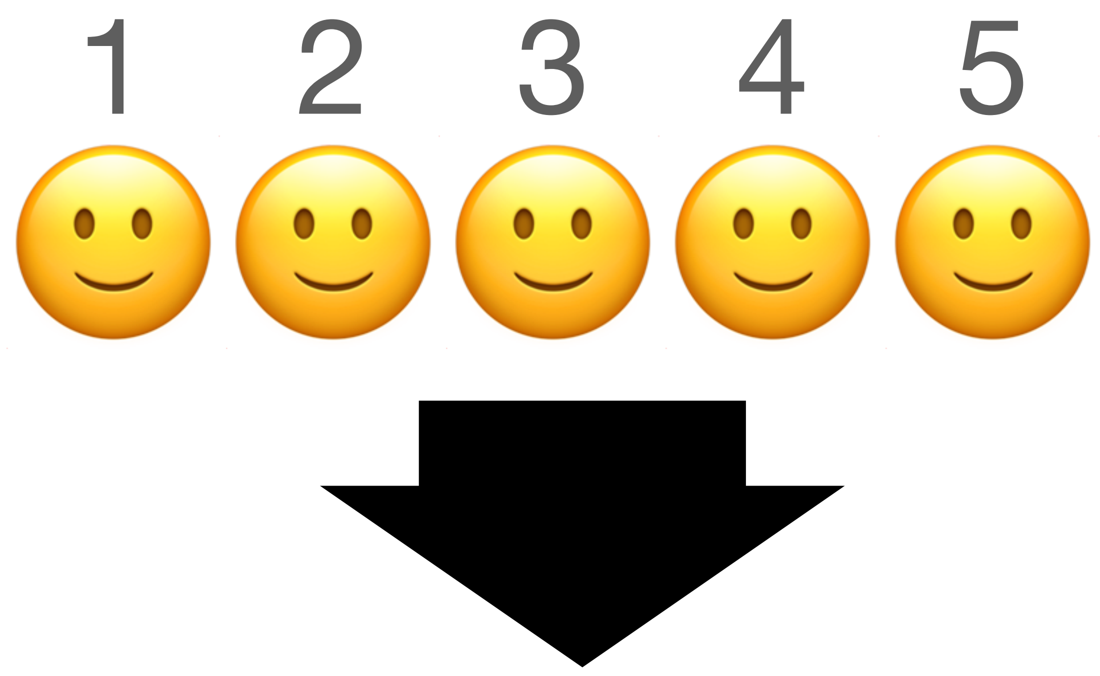
- **Coalition game** — The prediction result with n features (called coalition function $v([n])$) made by those n players, how do we attribute the payoff to each individual players?



$$v(\{1,2,3,4,5\}) = 1000$$

Attribution task in a coalition game - 1

- **Coalition game** — The prediction result with n features (called coalition function $v([n])$) made by those n players, how do we attribute the payoff to each individual players?

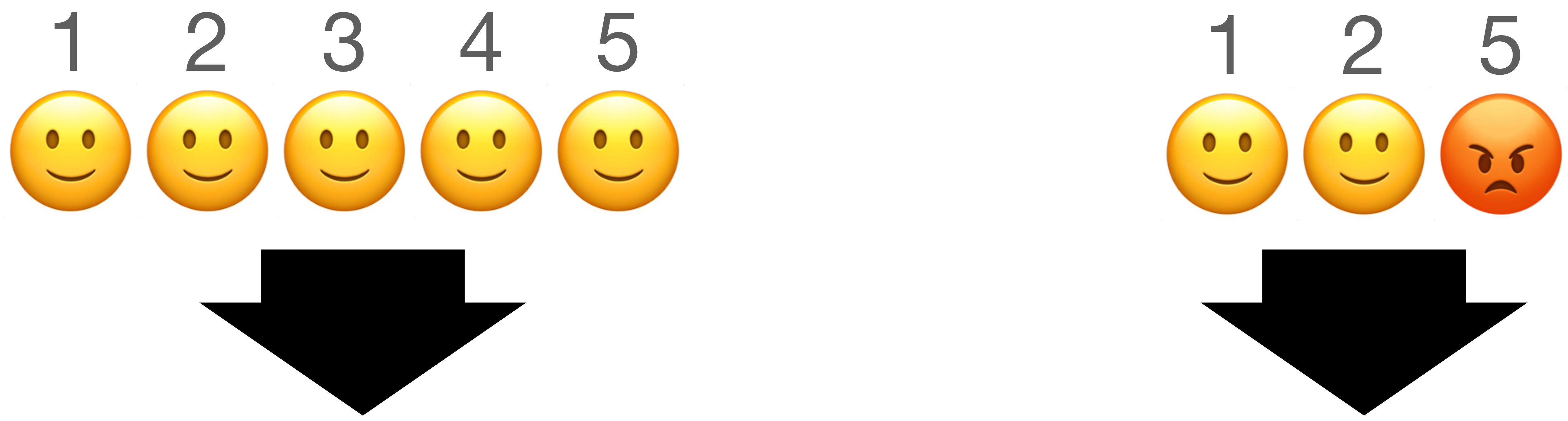


$$v(\{1,2,3,4,5\}) = 1000$$

$v(S)$ for $S \subseteq [n] = \{1,\cdots,n\}$ is a "coalition function", a payoff of a coalition S.

Attribution task in a coalition game - 1

- **Coalition game** — The prediction result with n features (called coalition function $v([n])$) made by those n players, how do we attribute the payoff to each individual players?

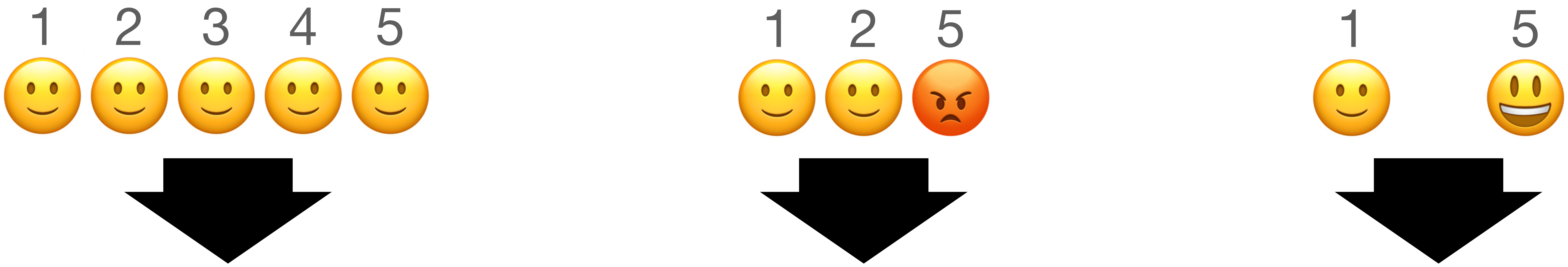


$v(\{1,2,3,4,5\}) = 1000$ $v(\{1,2,5\}) = 500$

$v(S)$ for $S \subseteq [n] = \{1,\cdots,n\}$ is a "coalition function", a payoff of a coalition S.

Attribution task in a coalition game - 1

- **Coalition game** — The prediction result with n features (called coalition function $v([n])$) made by those n players, how do we attribute the payoff to each individual players?

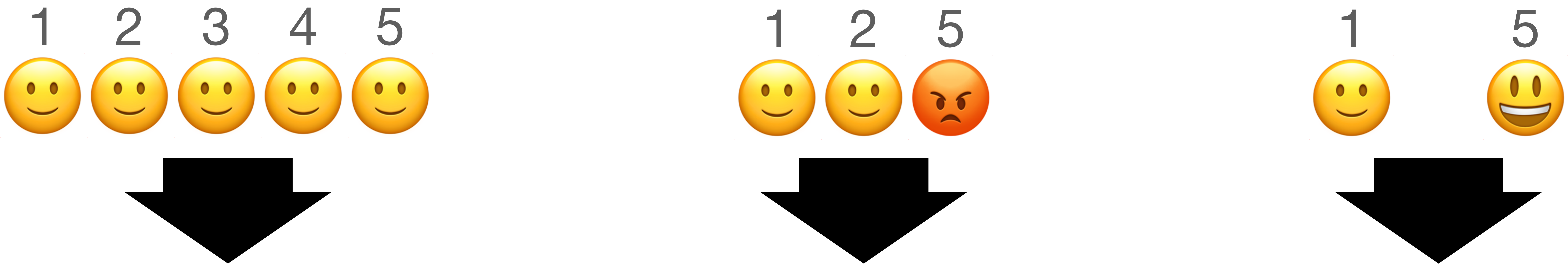


$v(\{1,2,3,4,5\}) = 1000$ $v(\{1,2,5\}) = 500$ $v(\{1,5\}) = 600$

$v(S)$ for $S \subseteq [n] = \{1,\cdots,n\}$ is a "coalition function", a payoff of a coalition S.

Attribution task in a coalition game - 1

- **Coalition game** — The prediction result with n features (called coalition function $v([n])$) made by those n players, how do we attribute the payoff to each individual players?



$v(\{1,2,3,4,5\}) = 1000$ $v(\{1,2,5\}) = 500$ $v(\{1,5\}) = 600$

$v(S)$ for $S \subseteq [n] = \{1, \cdots, n\}$ is a "coalition function", a payoff of a coalition S.

How do we attribute the total payoff (e.g., $v(\{1,2,3,4,5\})$ to individual players, taking account of interaction between players?

Attribution task in a coalition game - 2

- Let $\nu([n])$ denote the total payoff made by $[n] := \{1,2,\cdots, n\}$ players.

- Let $\nu(S)$ for $S \subseteq [n]$ denote the payoff made by a set of players S.

Attribution task in a coalition game - 2

- Let $\nu([n])$ denote the total payoff made by $[n] := \{1,2,\cdots,n\}$ players.

- Let $\nu(S)$ for $S \subseteq [n]$ denote the payoff made by a set of players S.

- **Shapley value**: The contribution of the player i is given by the *average of the marginal contribution of the player i*

$$\phi_i \equiv \frac{1}{n} \sum_{S \subseteq [n] \setminus \{i\}} \binom{n-1}{|S|}^{-1} \{v(S \cup \{i\}) - v(S)\}.$$

Attribution task in a coalition game - 2

- Let $\nu([n])$ denote the total payoff made by $[n] := \{1,2,\cdots,n\}$ players.
- Let $\nu(S)$ for $S \subseteq [n]$ denote the payoff made by a set of players S.
- **Shapley value**: The contribution of the player i is given by the *average of the marginal contribution of the player i*

$$\phi_i \equiv \frac{1}{n} \sum_{S \subseteq [n] \setminus \{i\}} \binom{n-1}{|S|}^{-1} \underbrace{\{v(S \cup \{i\}) - v(S)\}}_{\text{Marginal contribution of } x_i \text{ given } \mathbf{x}_S}.$$

Axiomatic characterization of feature attribution

Desirable properties for $\{\phi_1, \cdots, \phi_n\}$ is the following:

Axiomatic characterization of feature attribution

Desirable properties for $\{\phi_1, \cdots, \phi_n\}$ is the following:

- **Efficiency**: $\sum_{i=1}^{n} \phi_i = \nu([n]) - \nu(\varnothing) = \nu([n])$;

  The Shapley value perfectly assigned the contribution $\nu([n])$.

Axiomatic characterization of feature attribution

Desirable properties for $\{\phi_1, \cdots, \phi_n\}$ is the following:

- **Efficiency**: $\sum_{i=1}^{n} \phi_i = \nu([n]) - \nu(\emptyset) = \nu([n])$;

  The Shapley value perfectly assigned the contribution $\nu([n])$.

- **Dummy**: If $v(S \cup \{i\}) - v(S) = 0$ for all $S \subseteq [n] \backslash \{i\}$, then $\phi_i = 0$.

  If the ***marginal contribution of the player*** i ***in the team*** S, $v(S \cup \{i\}) - v(S)$, *is zero for all team* S, *then* $\phi_i = 0$

Axiomatic characterization of feature attribution

Desirable properties for $\{\phi_1, \cdots, \phi_n\}$ is the following:

- **Efficiency**: $\sum_{i=1}^{n} \phi_i = \nu([n]) - \nu(\emptyset) = \nu([n])$;

  The Shapley value perfectly assigned the contribution $\nu([n])$.

- **Dummy**: If $v(S \cup \{i\}) - v(S) = 0$ for all $S \subseteq [n] \backslash \{i\}$, then $\phi_i = 0$.

  If the ***marginal contribution of the player*** i ***in the team*** S, $v(S \cup \{i\}) - v(S)$, *is zero for all team* S, *then* $\phi_i = 0$

- **Symmetry** : If $v(S \cup \{i\}) = v(S \cup \{j\})$ for all $S \subseteq [n] \backslash \{i, j\}$, then $\phi_i = \phi_j$.

  If the ***marginal contribution of the player*** i, j ***in the team*** S *are the same for all team, then* $\phi_i = \phi_j$.

Axiomatic characterization of feature attribution

Desirable properties for $\{\phi_1, \cdots, \phi_n\}$ is the following:

- **Efficiency**: $\sum_{i=1}^{n} \phi_i = \nu([n]) - \nu(\emptyset) = \nu([n])$;

  The Shapley value perfectly assigned the contribution $\nu([n])$.

- **Dummy**: If $v(S \cup \{i\}) - v(S) = 0$ for all $S \subseteq [n] \backslash \{i\}$, then $\phi_i = 0$.

  If the ***marginal contribution of the player*** i ***in the team*** S, $v(S \cup \{i\}) - v(S)$, *is zero for all team* S, *then* $\phi_i = 0$

- **Symmetry** : If $v(S \cup \{i\}) = v(S \cup \{j\})$ for all $S \subseteq [n] \backslash \{i, j\}$, then $\phi_i = \phi_j$.

  If the ***marginal contribution of the player*** i, j ***in the team*** S *are the same for all team, then* $\phi_i = \phi_j$.

- **Linearity** : ϕ_i is a linear function of $\nu(S)\ \forall S \subseteq [n]$.

Application of Shapley to ML

Application of Shapley to ML

- Set the ML output $f(\mathbf{v})$ as $\nu([n])$. Then the problem is to assign the contribution of individual features $v_1, \cdots, v_n$ for explaining $f(\mathbf{v})$.

Application of Shapley to ML

- Set the ML output $f(\mathbf{v})$ as $\nu([n])$. Then the problem is to assign the contribution of individual features $v_1, \cdots, v_n$ for explaining $f(\mathbf{v})$.

- [Lundberg & Lee, 2017] proposed to use the Shapley value to explain the ML output $f(\mathbf{v})$ —"SHAP" or "Conditional Shapley". For $Y := f(\mathbf{V})$,

$$\phi_{v_i} \equiv \frac{1}{n} \sum_{S \subseteq [n] \setminus \{i\}} \binom{n-1}{|S|}^{-1} \{\mathbb{E}[Y | \mathbf{v}_S] - \mathbb{E}[Y | \mathbf{v}_S, v_i]\}$$

.

Application of Shapley to ML

- Set the ML output $f(\mathbf{v})$ as $\nu([n])$. Then the problem is to assign the contribution of individual features $v_1, \cdots, v_n$ for explaining $f(\mathbf{v})$.

- [Lundberg & Lee, 2017] proposed to use the Shapley value to explain the ML output $f(\mathbf{v})$ — "SHAP" or "Conditional Shapley". For $Y := f(\mathbf{V})$,

$$\phi_{v_i} \equiv \frac{1}{n} \sum_{S \subseteq [n] \setminus \{i\}} \binom{n-1}{|S|}^{-1} \underbrace{\{\mathbb{E}[Y \mid \mathbf{v}_S] - \mathbb{E}[Y \mid \mathbf{v}_S, v_i]\}}_{\text{Marginal contribution of } v_i}$$

.

Application of Shapley to ML

- Set the ML output $f(\mathbf{v})$ as $\nu([n])$. Then the problem is to assign the contribution of individual features $v_1, \cdots, v_n$ for explaining $f(\mathbf{v})$.

- [Lundberg & Lee, 2017] proposed to use the Shapley value to explain the ML output $f(\mathbf{v})$ — "SHAP" or "Conditional Shapley". For $Y := f(\mathbf{V})$,

$$\phi_{v_i} \equiv \frac{1}{n} \sum_{S \subseteq [n] \setminus \{i\}} \binom{n-1}{|S|}^{-1} \underbrace{\{\mathbb{E}[Y \mid \mathbf{v}_S] - \mathbb{E}[Y \mid \mathbf{v}_S, v_i]\}}_{\text{Marginal contribution of } v_i}$$

.

- The conditional Shapley measures the importance by its association / predictive power for the output $f(\mathbf{V})$.

Limitation of Conditional Shapley - (1)

https://towardsdatascience.com/be-careful-when-interpreting-predictive-models-in-search-of-causal-insights-e68626e664b6

Limitation of Conditional Shapley - (1)

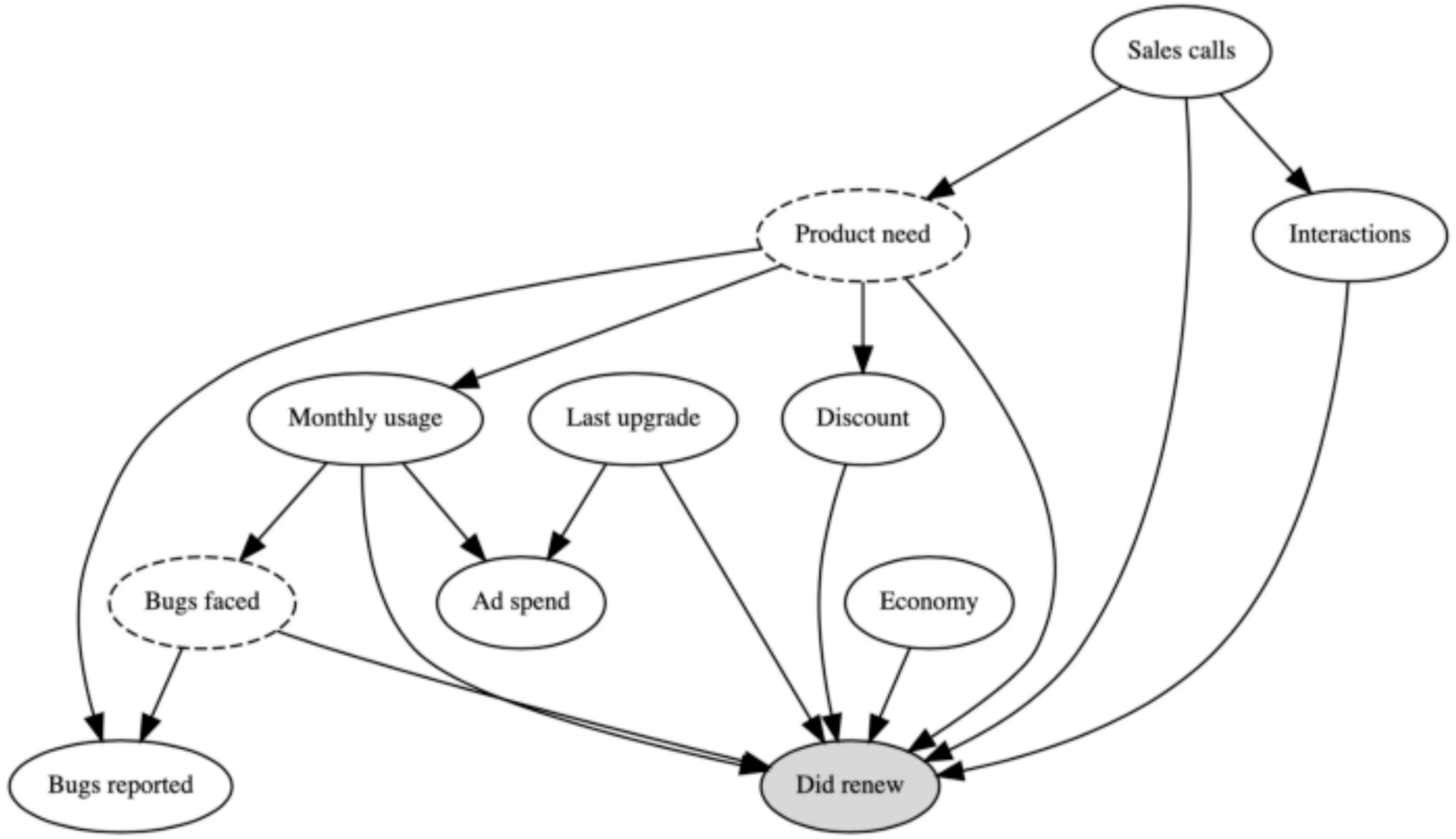
Scenario: Predict customers’ retention rate.

https://towardsdatascience.com/be-careful-when-interpreting-predictive-models-in-search-of-causal-insights-e68626e664b6

Limitation of Conditional Shapley - (1)

Scenario: Predict customers' retention rate.

The data-generating process is here:

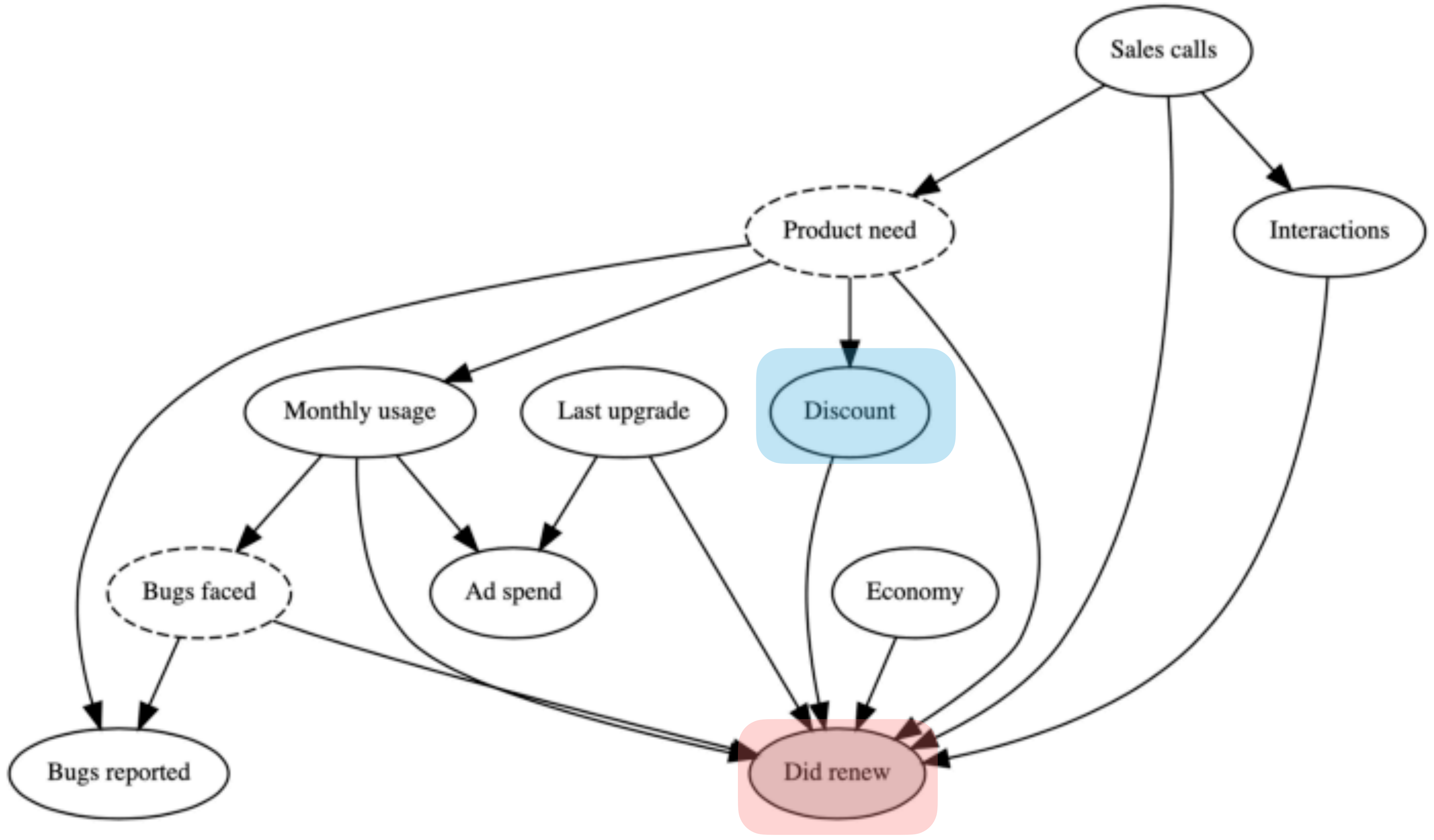


https://towardsdatascience.com/be-careful-when-interpreting-predictive-models-in-search-of-causal-insights-e68626e664b6

Limitation of Conditional Shapley - (1)

Scenario: Predict customers' retention rate.

The data-generating process is here:



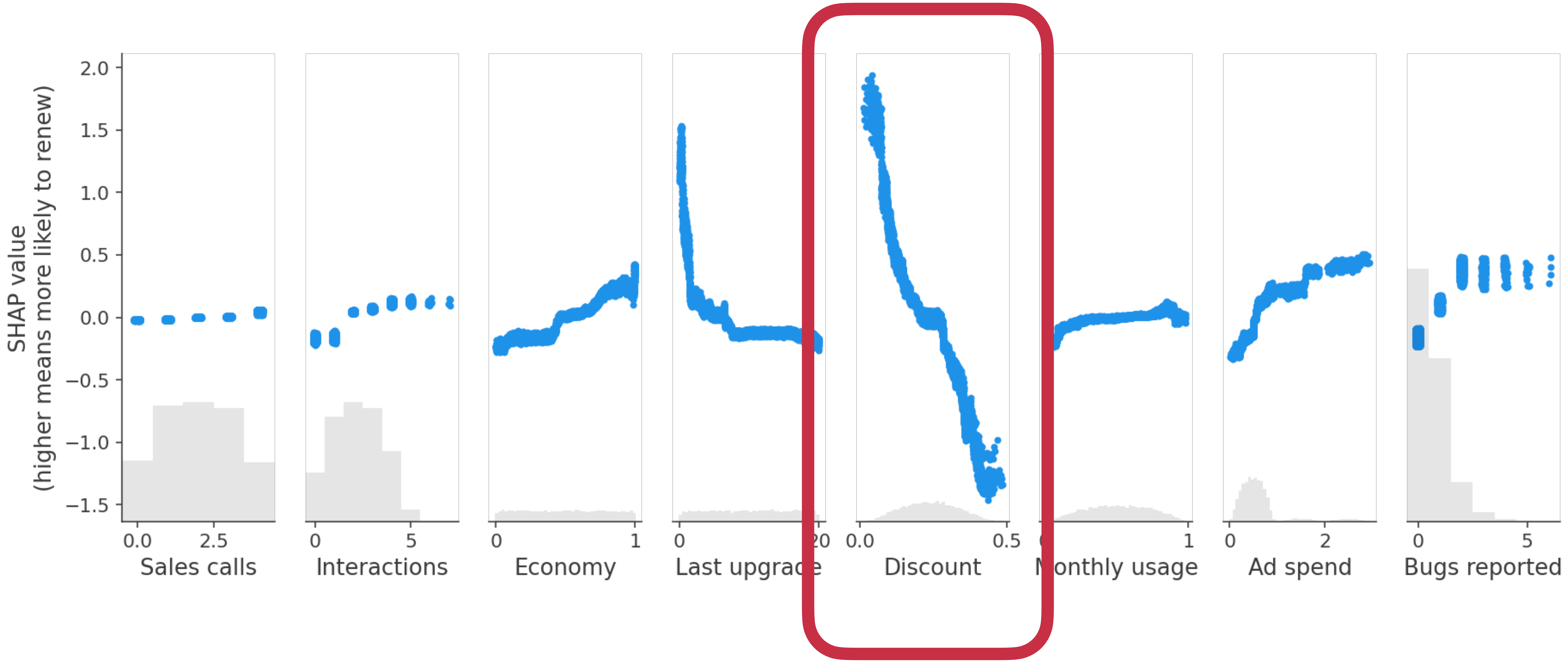
We measure the feature importance of "Discount" to explain Retention.

$$\mathbb{E}[\text{Rentention} \mid \text{Discount}, \mathbf{v}_S]$$

https://towardsdatascience.com/be-careful-when-interpreting-predictive-models-in-search-of-causal-insights-e68626e664b6

Limitation of Conditional Shapley - (2)

The results state that providing more discount leads to less retention. 🤷🏻‍♀️



https://towardsdatascience.com/be-careful-when-interpreting-predictive-models-in-search-of-causal-insights-e68626e664b6

Limitation of Conditional Shapley - (3)

https://towardsdatascience.com/be-careful-when-interpreting-predictive-models-in-search-of-causal-insights-e68626e664b6

Limitation of Conditional Shapley - (3)

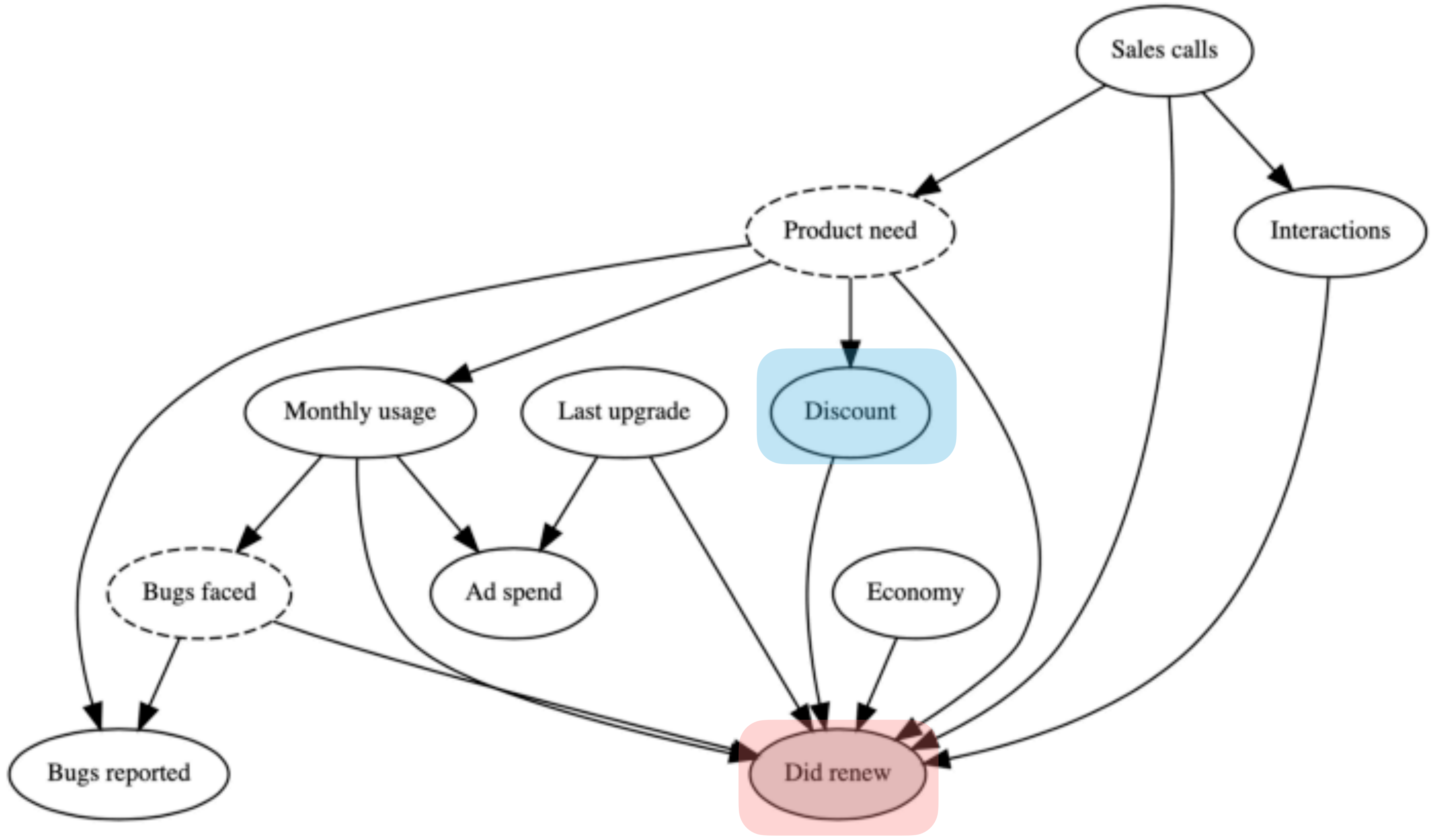
Lundberg, who developed SHAP, diagnosed this model fails due to the lack of considering causality.

https://towardsdatascience.com/be-careful-when-interpreting-predictive-models-in-search-of-causal-insights-e68626e664b6

Limitation of Conditional Shapley - (3)

Lundberg, who developed SHAP, diagnosed this model fails due to the lack of considering causality.

“interpreting a normal predictive model as causal are often unrealistic.”



https://towardsdatascience.com/be-careful-when-interpreting-predictive-models-in-search-of-causal-insights-e68626e664b6

Limitation of Conditional Shapley - (3)

Lundberg, who developed SHAP, diagnosed this model fails due to the lack of considering causality.

“interpr

Feature attribution method must take account of causality!

Outline

Outline

Motivated by the previous example, we develop ***causally*** interpretable ***feature attribution method***.

Outline

Motivated by the previous example, we develop ***causally* interpretable *feature attribution method***.

1. We *axiomatize and characterize* a causally interpretable feature attribution method, and propose do-Shapley values.

Outline

Motivated by the previous example, we develop ***causally*** interpretable ***feature attribution method***.

1. We *axiomatize and characterize* a causally interpretable feature attribution method, and propose do-Shapley values.

2. We provide *identifiability* condition where the do-Shapley values can be inferred from the observational data.

Outline

Motivated by the previous example, we develop ***causally*** interpretable ***feature attribution method***.

1. We *axiomatize and characterize* a causally interpretable feature attribution method, and propose do-Shapley values.

2. We provide *identifiability* condition where the do-Shapley values can be inferred from the observational data.

3. We construct a *double/debiased machine learning (DML)* [Chernozhukov et al., 2018] based do-Shapley estimator for practical settings.

Outline

Motivated by the previous example, we develop ***causally*** interpretable ***feature attribution method***.

1. We *axiomatize and characterize* a causally interpretable feature attribution method, and propose do-Shapley values.

2. We provide *identifiability* condition where the do-Shapley values can be inferred from the observational data.

3. We construct a *double/debiased machine learning (DML)* [Chernozhukov et al., 2018] based do-Shapley estimator for practical settings.

Structural Causal Model

Structural Causal Model $\mathscr{M} = \langle \mathbf{V}, \mathbf{U}, \mathbf{F}, P(\mathbf{u}) \rangle$

Structural Causal Model

Structural Causal Model $\mathscr{M} = \langle \mathbf{V}, \mathbf{U}, \mathbf{F}, P(\mathbf{u}) \rangle$

- $\mathbf{V}$: A set of endogenous (observable) variables.

Structural Causal Model

Structural Causal Model $\mathscr{M} = \langle \mathbf{V}, \mathbf{U}, \mathbf{F}, P(\mathbf{u}) \rangle$

- $\mathbf{V}$: A set of endogenous (observable) variables.
- $\mathbf{U}$: A set of exogenous (latent) variables.

Structural Causal Model

Structural Causal Model $\mathscr{M} = \langle \mathbf{V}, \mathbf{U}, \mathbf{F}, P(\mathbf{u}) \rangle$

- $\mathbf{V}$: A set of endogenous (observable) variables.
- $\mathbf{U}$: A set of exogenous (latent) variables.
- $\mathbf{F}$: A set of structural equations $\{f_{V_i}\}_{V_i \in \mathbf{V}}$ determining the value of $V_i \in \mathbf{V}$, where $V_i \leftarrow f_{v_i}(PA_{V_i}, U_{V_i})$ for some $PA_{V_i} \subseteq \mathbf{V}$ and $U_{V_i} \subseteq \mathbf{U}$.

Structural Causal Model

Structural Causal Model $\mathscr{M} = \langle \mathbf{V}, \mathbf{U}, \mathbf{F}, P(\mathbf{u}) \rangle$

- $\mathbf{V}$: A set of endogenous (observable) variables.
- $\mathbf{U}$: A set of exogenous (latent) variables.
- $\mathbf{F}$: A set of structural equations $\{f_{V_i}\}_{V_i \in \mathbf{V}}$ determining the value of $V_i \in \mathbf{V}$, where $V_i \leftarrow f_{v_i}(PA_{V_i}, U_{V_i})$ for some $PA_{V_i} \subseteq \mathbf{V}$ and $U_{V_i} \subseteq \mathbf{U}$.
- $P(\mathbf{u})$: A probability measure for $\mathbf{U}$.

Structural Causal Model

Structural Causal Model $\mathcal{M} = \langle \mathbf{V}, \mathbf{U}, \mathbf{F}, P(\mathbf{u}) \rangle$

- $\mathbf{V}$: A set of endogenous (observable) variables.
- $\mathbf{U}$: A set of exogenous (latent) variables.
- $\mathbf{F}$: A set of structural equations $\{f_{V_i}\}_{V_i \in \mathbf{V}}$ determining the value of $V_i \in \mathbf{V}$, where $V_i \leftarrow f_{v_i}(PA_{V_i}, U_{V_i})$ for some $PA_{V_i} \subseteq \mathbf{V}$ and $U_{V_i} \subseteq \mathbf{U}$.
- $P(\mathbf{u})$: A probability measure for $\mathbf{U}$.

💡 An SCM induced a qualitative description in the form of a "**causal graph**" $G \equiv G(\mathcal{M})$.

Interpretability Tasks w.r.t. SCM

Task — Interpretability task w.r.t. SCM

Interpretability Tasks w.r.t. SCM

Task — Interpretability task w.r.t. SCM

Input: A causal graph G; an input-output $(\mathbf{v}, f(\mathbf{v}))$; and a dataset for G. Note G is induced by the SCM.

Interpretability Tasks w.r.t. SCM

Task — Interpretability task w.r.t. SCM

Input: A causal graph G; an input-output $(\mathbf{v}, f(\mathbf{v}))$; and a dataset for G. Note G is induced by the SCM.

1. A causal graph G is on $\mathbf{V}, Y$ for $Y := f(\mathbf{V})$.

Interpretability Tasks w.r.t. SCM

Task — Interpretability task w.r.t. SCM

Input: A causal graph G; an input-output $(\mathbf{v}, f(\mathbf{v}))$; and a dataset for G. Note G is induced by the SCM.

1. A causal graph G is on $\mathbf{V}, Y$ for $Y := f(\mathbf{V})$.

2. A pair of an individual input-output: $(\mathbf{v}, f(\mathbf{v}))$,

Interpretability Tasks w.r.t. SCM

Task — Interpretability task w.r.t. SCM

Input: A causal graph G; an input-output $(\mathbf{v}, f(\mathbf{v}))$; and a dataset for G. Note G is induced by the SCM.

1. A causal graph G is on $\mathbf{V}, Y$ for $Y := f(\mathbf{V})$.
2. A pair of an individual input-output: $(\mathbf{v}, f(\mathbf{v}))$,
3. A dataset D of $\{\mathbf{V}_i, Y_i = f(\mathbf{V}_i)\}_{i=1}^{N}$.

Interpretability Tasks w.r.t. SCM

Task — Interpretability task w.r.t. SCM

Input: A causal graph G; an input-output $(\mathbf{v}, f(\mathbf{v}))$; and a dataset for G. Note G is induced by the SCM.

1. A causal graph G is on $\mathbf{V}, Y$ for $Y := f(\mathbf{V})$.
2. A pair of an individual input-output: $(\mathbf{v}, f(\mathbf{v}))$,
3. A dataset D of $\{\mathbf{V}_i, Y_i = f(\mathbf{V}_i)\}_{i=1}^{N}$.

Output: A vector $attr(f, \mathbf{v}) \equiv \{\phi_{v_1}, \cdots, \phi_{v_n}\}$ where ϕ_{v_i} is an importance of a node v_i.

Interpretability Tasks w.r.t. SCM

Ta

I

i

C

What properties a *desirable causally interpretable feature attribution method* should satisfy?

Task: Application to ML Interpretation

When the outcome is a ML model output $Y := f(\mathbf{V})$, the target is reduced to $Q := f(\mathbf{v}) = \mathbb{E}[Y \mid do(\mathbf{v})]$, and the problem reduces to measuring the importance of inputs.

Causal Graph
$G = G(\mathbf{V})$

Samples
$D \sim P(\mathbf{V})$

Output of the ML model $f(\cdot)$
$Q := f(\mathbf{v})$

Feature Attribution

Contribution of inputs $v_i \in \mathbf{v}$ to the ML output $f(\mathbf{v})$

Desideratum for causally IML methods

“Causal IML Axiom”: Desideratum for causally interpretable ML

Desideratum for causally IML methods

"Causal IML Axiom": Desideratum for causally interpretable ML

- **Perfect assignment**: $\sum_{v_i \in \mathbf{v}} \phi_{v_i} = f(\mathbf{v})$.

Desideratum for causally IML methods

"Causal IML Axiom": Desideratum for causally interpretable ML

- **Perfect assignment**: $\sum_{v_i \in \mathbf{v}} \phi_{v_i} = f(\mathbf{v})$.

- **Causal Irrelevance**: If V_i is *causally irrelevant* to $Y = f(\mathbf{V})$, then $\phi_{v_i} = 0$.

  $P(y \mid do(v_i)) = P(y)\ \forall y, v_i$ for $V_i \in \mathbf{V}$.

Desideratum for causally IML methods

"Causal IML Axiom": Desideratum for causally interpretable ML

- **Perfect assignment**: $\sum_{v_i \in \mathbf{v}} \phi_{v_i} = f(\mathbf{v})$.

- **Causal Irrelevance**: If V_i is *causally irrelevant* to $Y = f(\mathbf{V})$, then $\phi_{v_i} = 0$.

  $P(y \mid do(v_i)) = P(y)\ \forall y, v_i$ for $V_i \in \mathbf{V}$.

- **Causal Symmetry**: If $v_i, v_j \in \mathbf{v}$ have the same *causal explanatory power* to Y, then $\phi_{v_i} = \phi_{v_j}$.

  $P(y \mid do(v_i), do(\mathbf{w})) = P(y \mid do(v_j), do(\mathbf{w}))\ \forall y$ and $\mathbf{W} \subseteq \mathbf{V} \backslash \{V_i, V_j\}$.

Desideratum for causally IML methods

"Causal IML Axiom": Desideratum for causally interpretable ML

- **Perfect assignment**: $\sum_{v_i \in \mathbf{v}} \phi_{v_i} = f(\mathbf{v})$.

- **Causal Irrelevance**: If V_i is *causally irrelevant* to $Y = f(\mathbf{V})$, then $\phi_{v_i} = 0$.

  $P(y \mid do(v_i)) = P(y)\ \forall y, v_i$ for $V_i \in \mathbf{V}$.

- **Causal Symmetry**: If $v_i, v_j \in \mathbf{v}$ have the same *causal explanatory power* to Y, then $\phi_{v_i} = \phi_{v_j}$.

  $P(y \mid do(v_i), do(\mathbf{w})) = P(y \mid do(v_j), do(\mathbf{w}))\ \forall y$ and $\mathbf{W} \subseteq \mathbf{V} \backslash \{V_i, V_j\}$.

- **Linearity** : ϕ_{v_i} must be a linear function of $\mathbb{E}[Y \mid do(\mathbf{v}_S)]$

do-Shapley as a desirable causal IML method

Thm. 1. Axiomatic characterization of do-Shapley

do-Shapley as a desirable causal IML method

Thm. 1. Axiomatic characterization of do-Shapley

A following attribution method $attr(f, \mathbf{v}) = \{\phi_{v_i}\}_{v_i \in \mathbf{v}}$, named do-Shapley, is **uniquely** satisfying the Causal IML Axioms.

do-Shapley as a desirable causal IML method

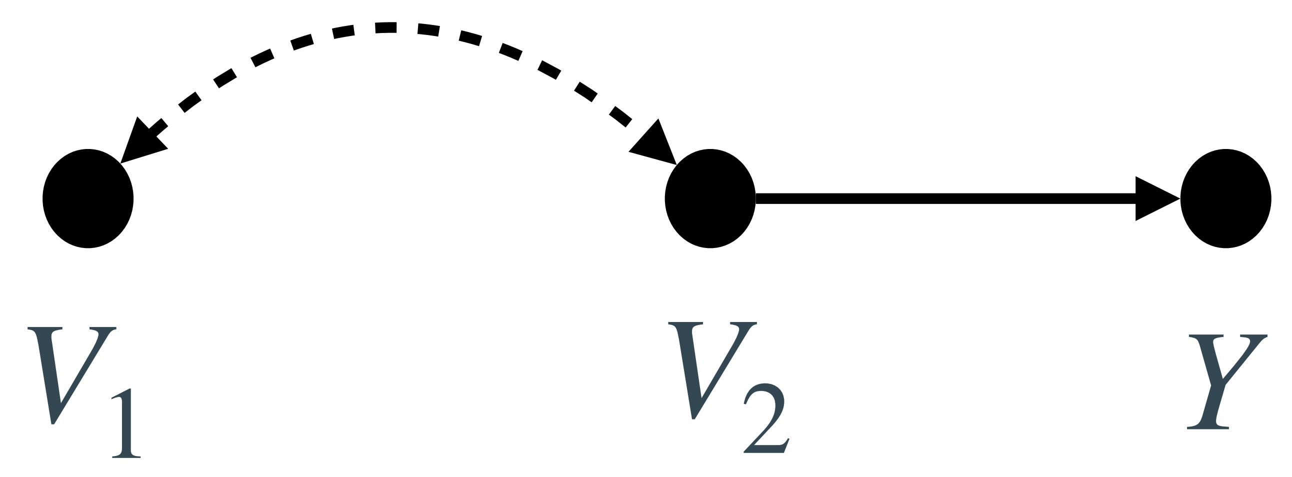
Thm. 1. Axiomatic characterization of do-Shapley

A following attribution method $attr(f, \mathbf{v}) = \{\phi_{v_i}\}_{v_i \in \mathbf{v}}$, named do-Shapley, is **uniquely** satisfying the Causal IML Axioms.

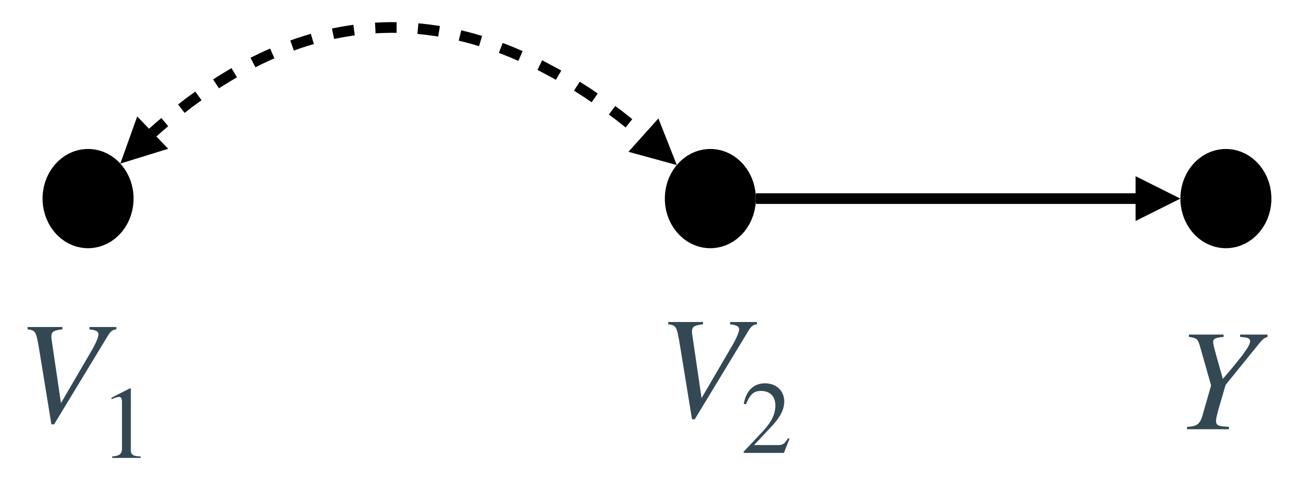
$$\phi_{v_i} = (1/n) \sum_{S \subseteq [n] \setminus \{i\}} \binom{n-1}{|S|}^{-1} \mathbb{E}[Y | do(\mathbf{v}_S, v_i)] - \mathbb{E}[Y | do(\mathbf{v}_S)]\},$$

vs. Conditional Shapley

vs. Conditional Shapley

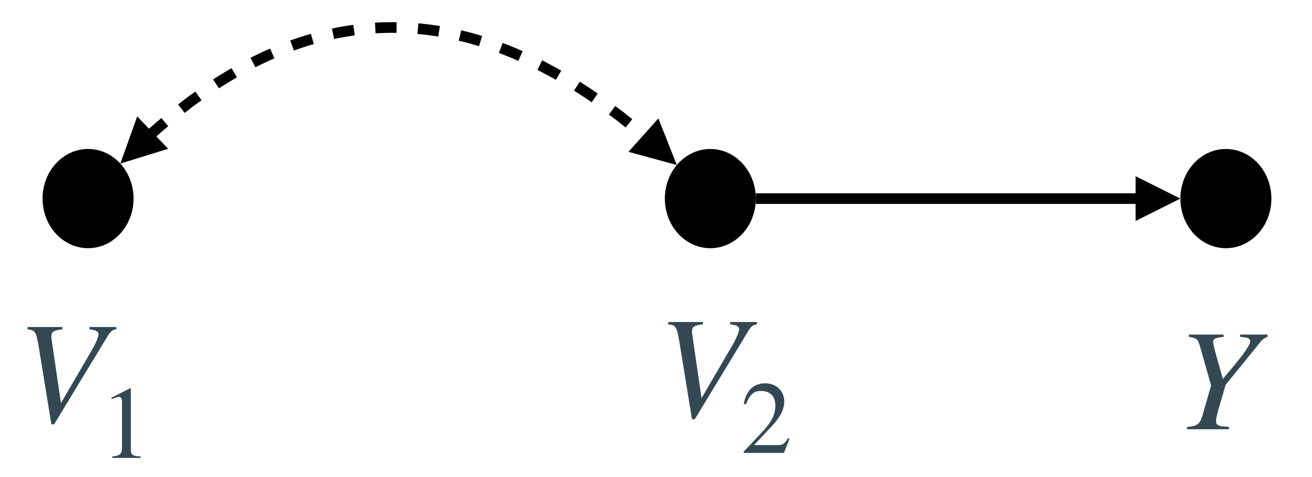


vs. Conditional Shapley



V_1 is causally irrelevant to Y (i.e., $P(y \mid do(v_1)) = P(y)$).

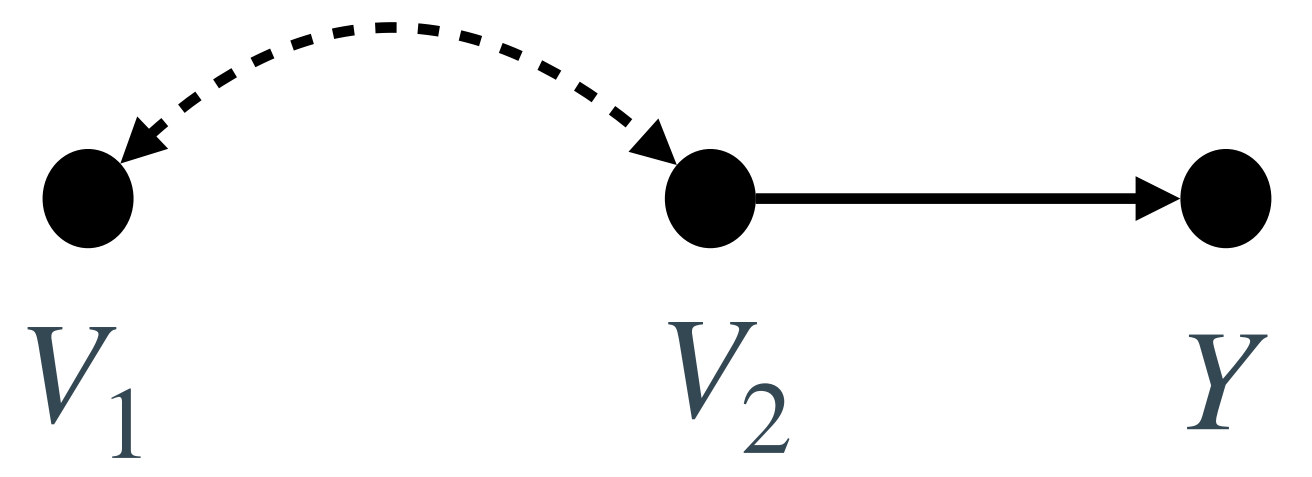
vs. Conditional Shapley



V_1 is causally irrelevant to Y (i.e., $P(y \mid do(v_1)) = P(y)$).

- $\phi_{V_1}(\nu_{do}) = 0$, because $\nu_{do}(\{1\}) - \nu_{do}(\{\}) = \nu_{do}(\{1{,}2\}) - \nu_{do}(\{2\}) = 0$,

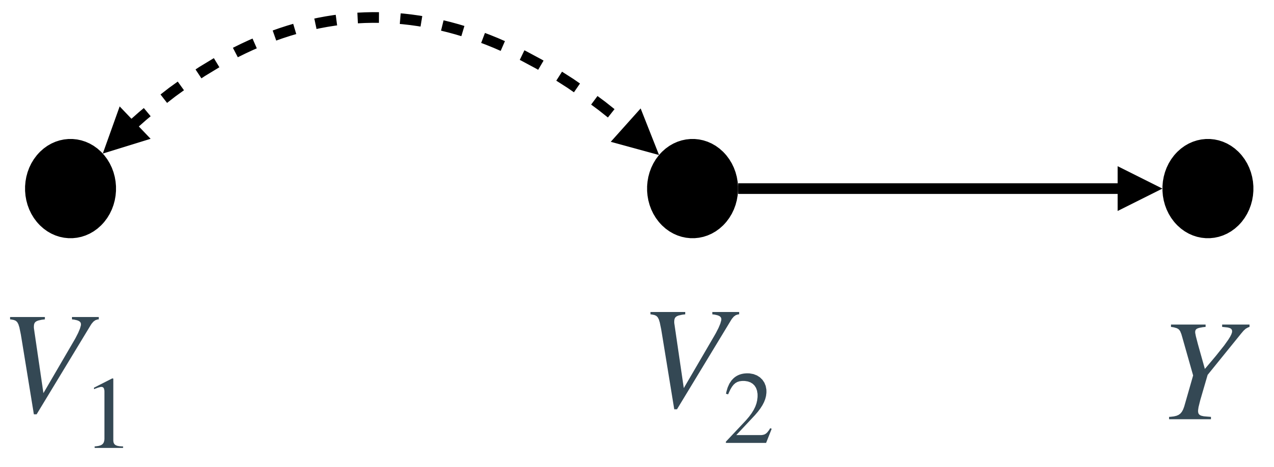
vs. Conditional Shapley



V_1 is causally irrelevant to Y (i.e., $P(y \mid do(v_1)) = P(y)$).

- $\phi_{V_1}(\nu_{do}) = 0$, because $\nu_{do}(\{1\}) - \nu_{do}(\{\}) = \nu_{do}(\{1,2\}) - \nu_{do}(\{2\}) = 0$,
- However, it's possible that $\phi_{V_1}(\nu_{cond}) \neq 0$

vs. Conditional Shapley



V_1 is causally irrelevant to Y (i.e., $P(y \mid do(v_1)) = P(y)$).

- $\phi_{V_1}(\nu_{do}) = 0$, because $\nu_{do}(\{1\}) - \nu_{do}(\{\}) = \nu_{do}(\{1,2\}) - \nu_{do}(\{2\}) = 0$,
- However, it's possible that $\phi_{V_1}(\nu_{cond}) \neq 0$

Causal Irrelevance axiom does not hold in Conditional Shapley.

Outline

We develop ***causally*** interpretable ***feature attribution method***.

1. We *axiomatize and characterize* a causally interpretable feature attribution method, and propose do-Shapley values.

2. We provide *identifiability* condition where the do-Shapley values can be inferred from the observational data.

3. We construct a *double/debiased machine learning (DML)* [Chernozhukov et al., 2018] based do-Shapley estimator for practical settings.

Outline

We develop ***causally*** interpretable ***feature attribution method***.

1. We *axiomatize and characterize* a causally interpretable feature attribution method, and propose do-Shapley values.

2. We provide *identifiability* condition where the do-Shapley values can be inferred from the observational data.

3. We construct a *double/debiased machine learning (DML)* [Chernozhukov et al., 2018] based do-Shapley estimator for practical settings.

do-Shapley Identifiability - Challenge

do-Shapley Identifiability - Challenge

$$\phi_{v_i} := \frac{1}{n} \sum_{S \subseteq [n]} \binom{n-1}{|S|}^{-1} \left\{ \mathbb{E}[Y \mid do(\mathbf{v}_S, v_i)] - \mathbb{E}[Y \mid do(\mathbf{v}_S)] \right\}$$

do-Shapley Identifiability - Challenge

$$\phi_{v_i} := \frac{1}{n} \sum_{S \subseteq [n]} \binom{n-1}{|S|}^{-1} \left\{ \mathbb{E}[Y \mid do(\mathbf{v}_S, v_i)] - \mathbb{E}[Y \mid do(\mathbf{v}_S)] \right\}$$

- We have to determine the identifiably of $\mathbb{E}[Y \mid do(\mathbf{v}_S)]$ for all $\mathbf{V}_S \subseteq \mathbf{V}$.

do-Shapley Identifiability - Challenge

$$\phi_{v_i} := \frac{1}{n} \sum_{S \subseteq [n]} \binom{n-1}{|S|}^{-1} \left\{ \mathbb{E}[Y \mid do(\mathbf{v}_S, v_i)] - \mathbb{E}[Y \mid do(\mathbf{v}_S)] \right\}$$

- We have to determine the identifiably of $\mathbb{E}[Y \mid do(\mathbf{v}_S)]$ for all $\mathbf{V}_S \subseteq \mathbf{V}$.
- This might take exponential computational time.

do-Shapley Identifiability - Challenge

do-Shapley Identifiability - Challenge

Identification of do-Shapley

Assume Y is not connected by bidirected paths. If any variables are not connected to its children by bidirected paths (i.e., V_i and $Ch(V_i)$ are not in the same C-component), then the do-Shapley is identifiable (i.e., $\mathbb{E}[Y | do(\mathbf{v}_S)]$ for all $\mathbf{V}_S \subseteq \mathbf{V}$ is identifiable).

do-Shapley Identifiability - Challenge

Identification of do-Shapley

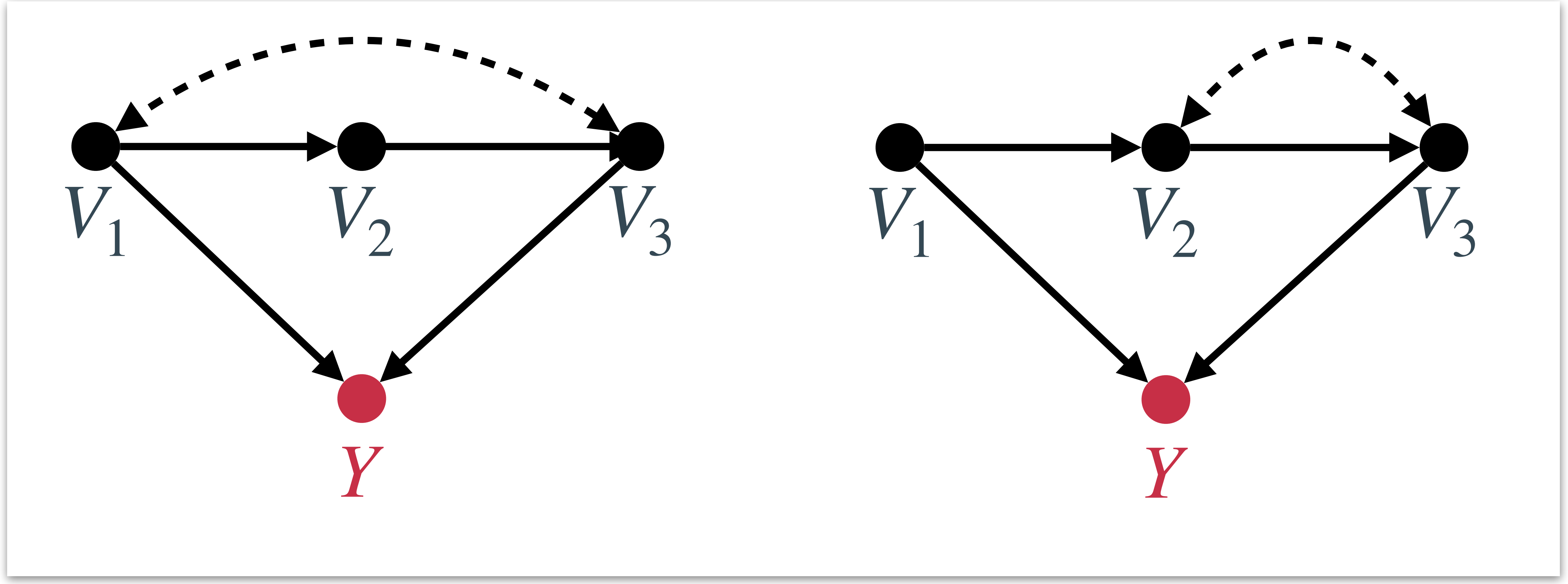
Assume Y is not connected by bidirected paths. If any variables are not connected to its children by bidirected paths (i.e., V_i and $Ch(V_i)$ are not in the same C-component), then the do-Shapley is identifiable (i.e., $\mathbb{E}[Y|do(\mathbf{v}_S)]$ for all $\mathbf{V}_S \subseteq \mathbf{V}$ is identifiable).

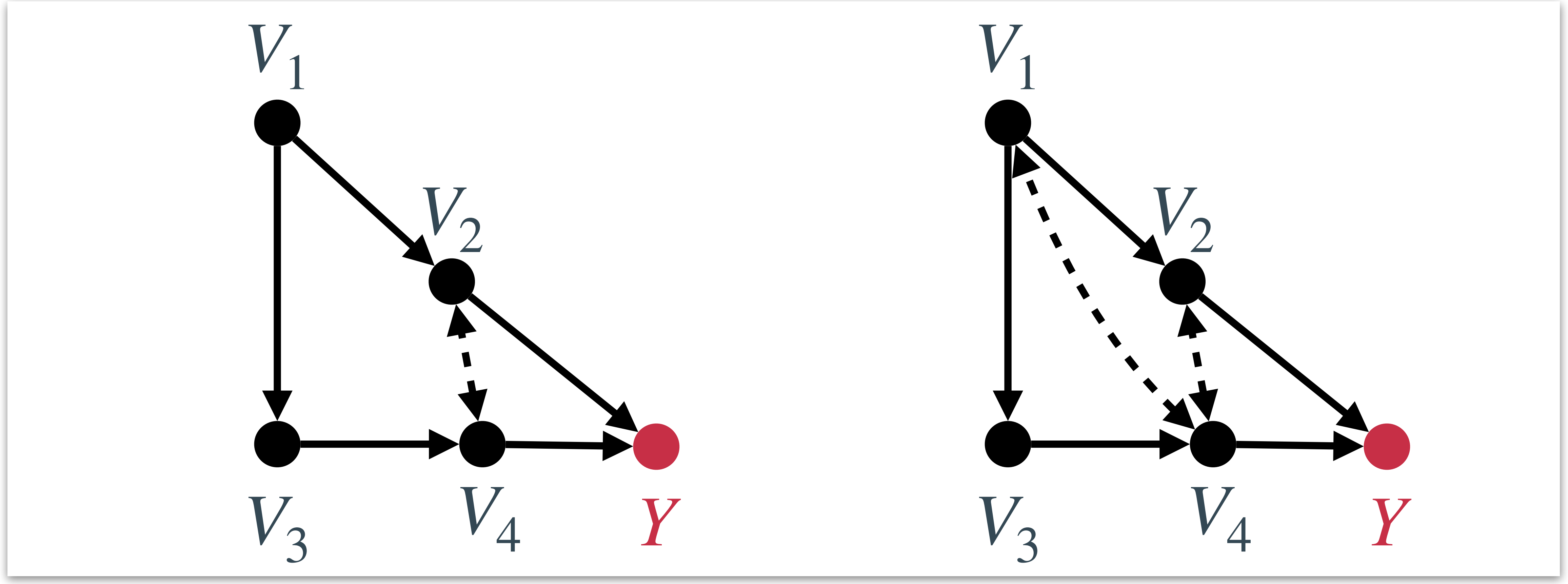
Specifically,

$$\mathbb{E}[Y|do(\mathbf{v}_S)] = \sum_{\mathbf{v}_{\overline{S}}} \mathbb{E}[Y|\mathbf{v}] \frac{P(\mathbf{v})}{\prod_{V_a \in C(\mathbf{V}_S)} P(v_a|pre(v_a))} \prod_{k=1}^{c} \sum_{\mathbf{s}_k} \prod_{V_b \in C(\mathbf{S}_k)} P(v_b|pre(v_b))$$

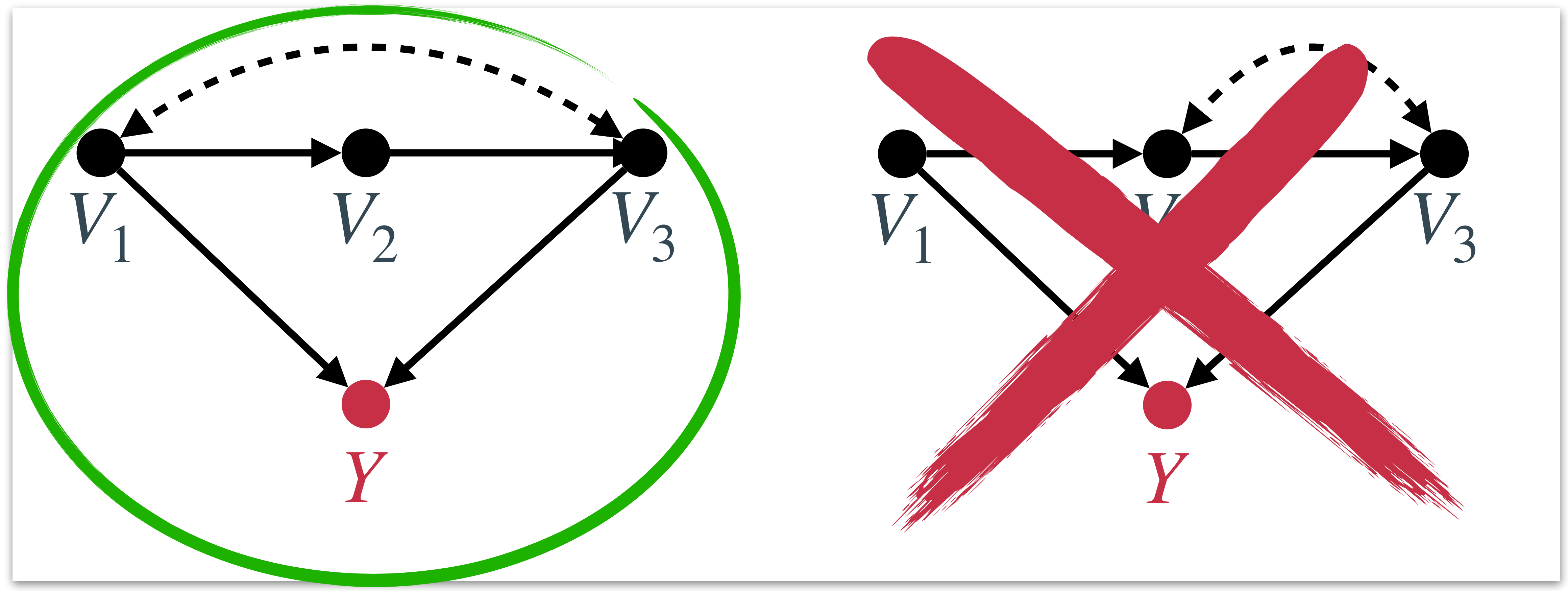
where $\mathbf{S}_k$ is some partition of $\mathbf{V}_S$.

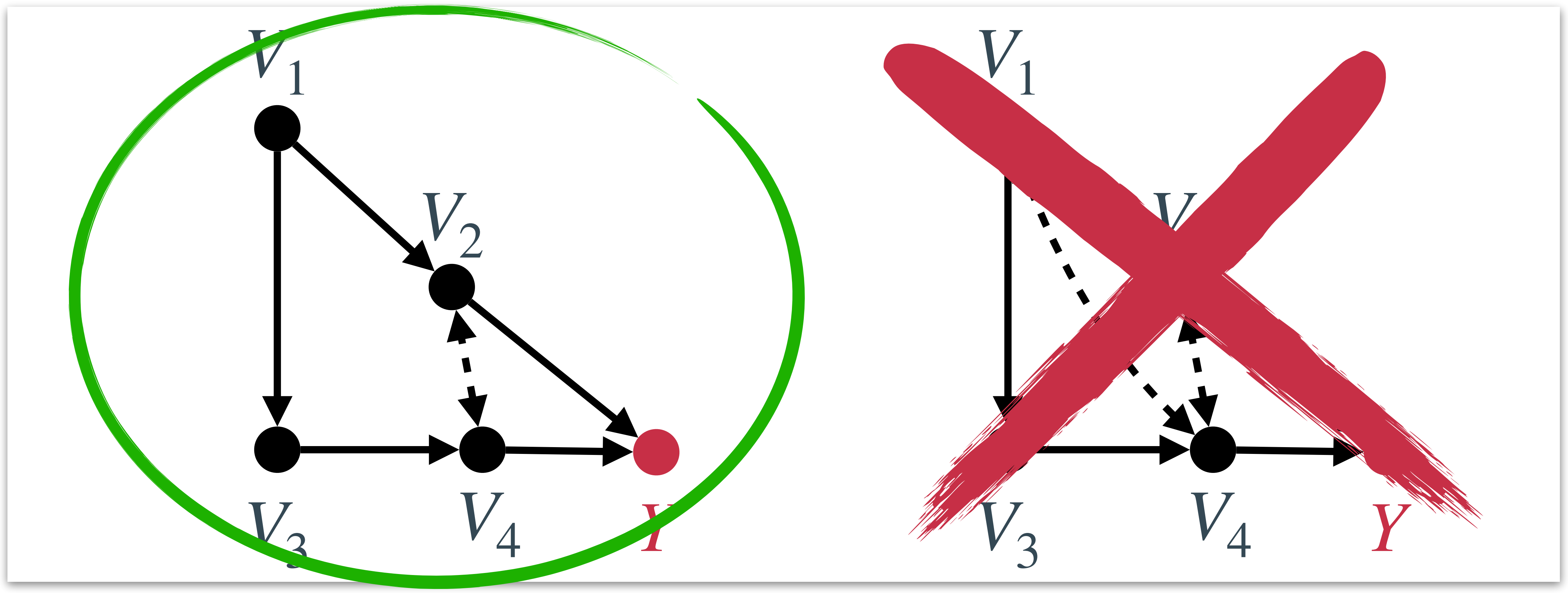
do-Shapley Identifiability: Examples



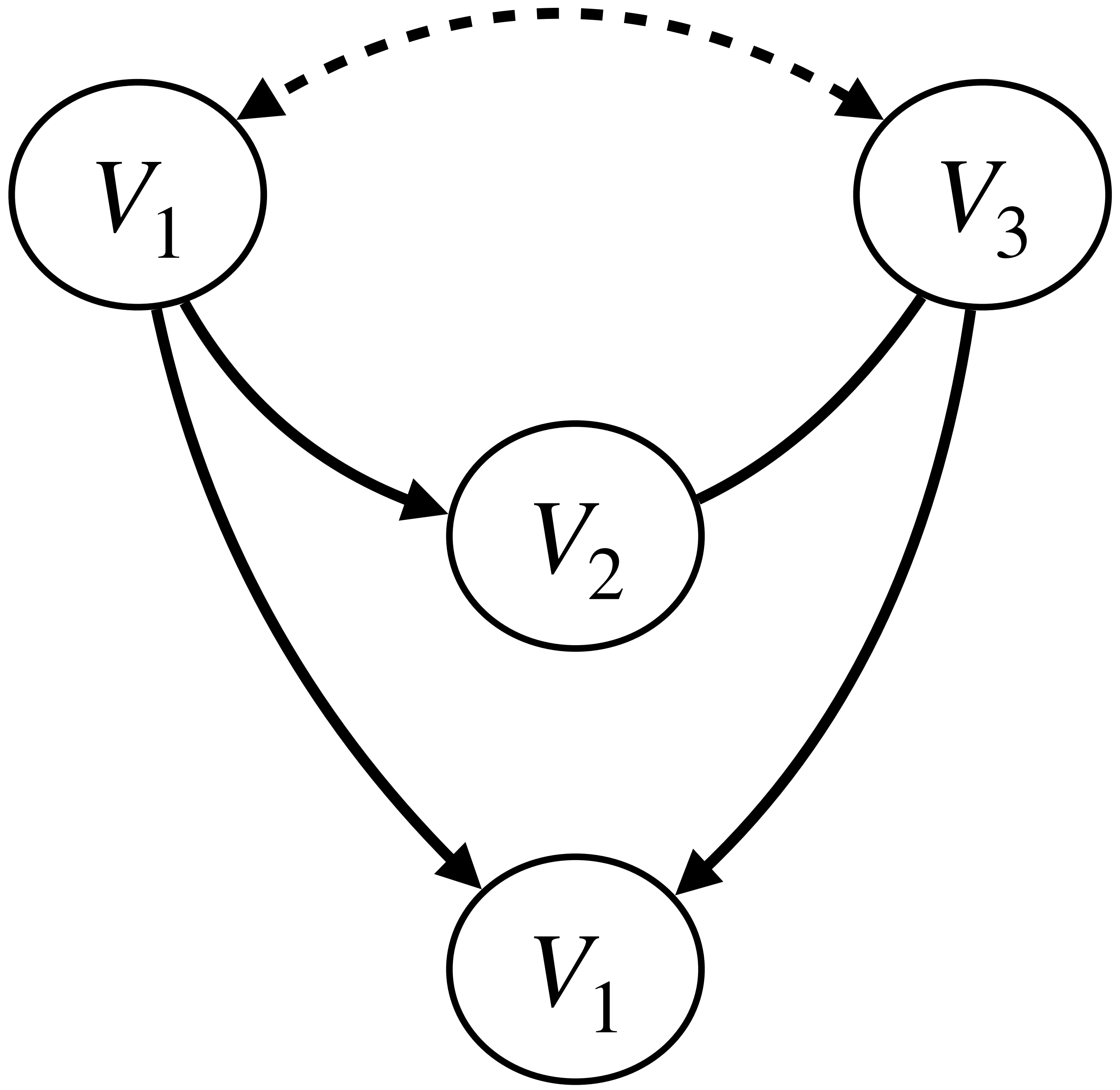


do-Shapley Identifiability: Examples

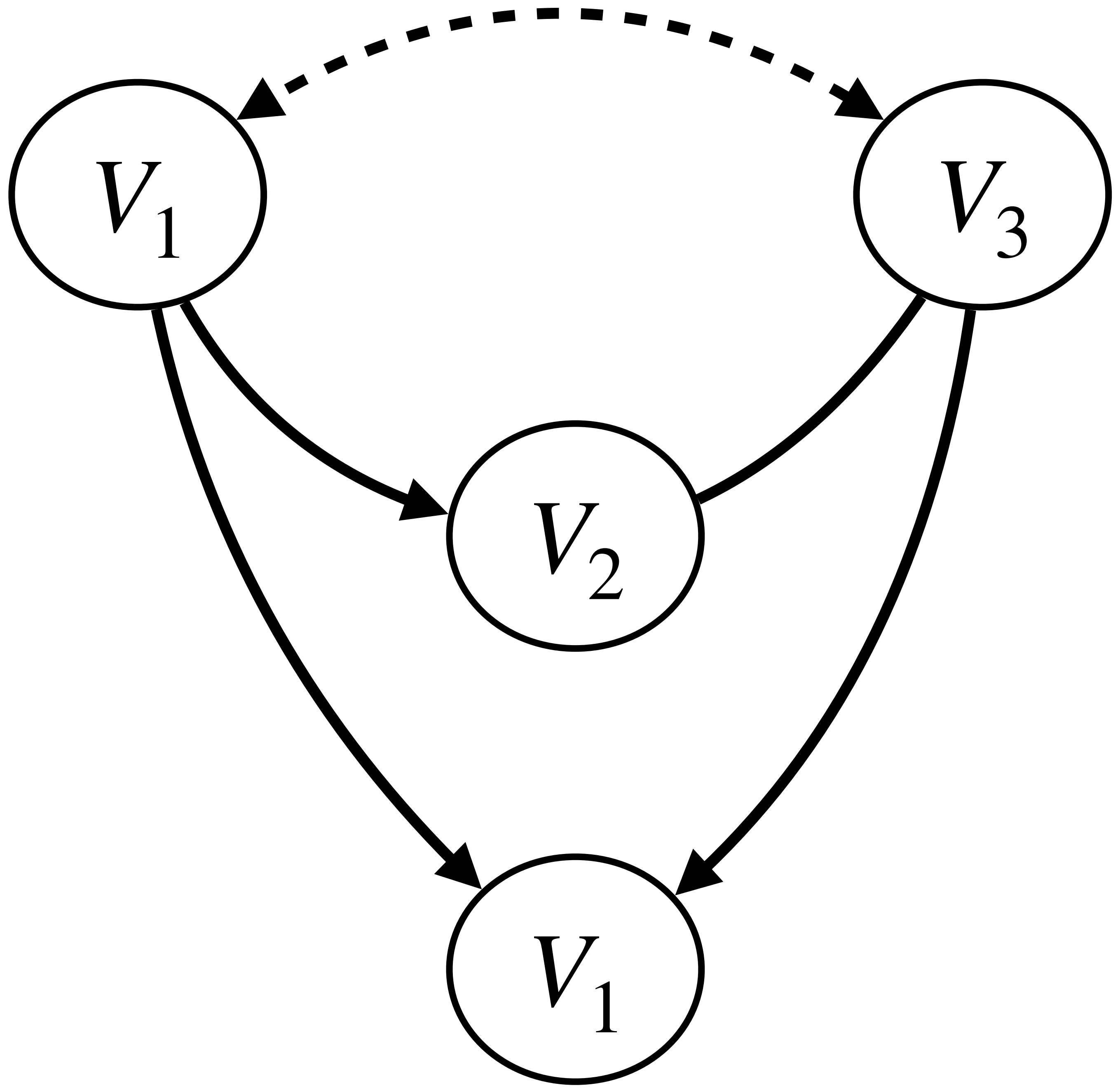




do-Shapley Identifiability: Examples



do-Shapley Identifiability: Examples



$$
\begin{aligned}
&\mathbb{E}\left[Y|do(\mathbf{v}_S)\right] \\
&= \begin{cases}
\sum_{\mathbf{v}_{\overline{S}}} \mathbb{E}\left[Y|\mathbf{v}\right] P(v_2|v_1, v_3)P(\mathbf{v}_S), \text{ if } S \in \{1, 3\}, \\
\sum_{\mathbf{v}_{\overline{S}}} \mathbb{E}\left[Y|\mathbf{v}\right] P(\mathbf{v}_{\overline{S}}), \text{ if } S \in \{\emptyset, 2, \{1, 2\}, \{2, 3\}\}, \\
\sum_{\mathbf{v}_{\overline{S}}} \mathbb{E}\left[Y|\mathbf{v}\right] P(\mathbf{v}_{\overline{S}}|\mathbf{v}_S), \text{if } S \in \{\{1, 3\}\}, \\
\mathbb{E}\left[Y|\mathbf{v}\right] \text{ if } S = \{1, 2, 3\}.
\end{cases}
\end{aligned}
$$

Outline

We develop ***causally*** interpretable ***feature attribution method***.

1. We *axiomatize and characterize* a causally interpretable feature attribution method, and propose do-Shapley values.

2. We provide *identifiability* condition where the do-Shapley values can be inferred from the observational data.

3. We construct a *double/debiased machine learning (DML)* [Chernozhukov et al., 2018] based do-Shapley estimator for practical settings.

Outline

We develop ***causally*** interpretable ***feature attribution method***.

1. We *axiomatize and characterize* a causally interpretable feature attribution method, and propose do-Shapley values.

2. We provide *identifiability* condition where the do-Shapley values can be inferred from the observational data.

3. We construct a *double/debiased machine learning (DML)* [Chernozhukov et al., 2018] based do-Shapley estimator for practical settings.

Outline

We develop ***causally*** interpretable ***feature attribution method***.

1. We *axiomatize and characterize* a causally interpretable feature attribution method, and propose do-Shapley values.

2. We provide *identifiability* condition where the do-Shapley values can be inferred from the observational data.

3. We construct a *double/debiased machine learning (DML)* [Chernozhukov et al., 2018] based do-Shapley estimator for practical settings.

DAG (No latent confounders, in this talk!)

Two components in do-Shapley estimation

$$\phi_{V_i}(\nu_{do}) \equiv \frac{1}{n} \sum_{S \subseteq [n] \setminus \{i\}} \binom{n-1}{|S|}^{-1} \{\nu_{do}(S \cup \{i\}) - \nu_{do}(S)\}.$$

Two components in do-Shapley estimation

$$\phi_{V_i}(\nu_{do}) \equiv \frac{1}{n} \sum_{S \subseteq [n] \setminus \{i\}} \binom{n-1}{|S|}^{-1} \{\nu_{do}(S \cup \{i\}) - \nu_{do}(S)\}.$$

Computing the Shapley value requires

Two components in do-Shapley estimation

$$\phi_{V_i}(\nu_{do}) \equiv \frac{1}{n} \sum_{S \subseteq [n] \backslash \{i\}} \binom{n-1}{|S|}^{-1} \{\nu_{do}(S \cup \{i\}) - \nu_{do}(S)\}.$$

Computing the Shapley value requires

1. Exploring all possible subsets in $[n] \backslash \{i\}$;

Two components in do-Shapley estimation

$$\phi_{V_i}(\nu_{do}) \equiv \frac{1}{n} \sum_{S \subseteq [n] \backslash \{i\}} \binom{n-1}{|S|}^{-1} \{\nu_{do}(S \cup \{i\}) - \nu_{do}(S)\}.$$

Computing the Shapley value requires

1. Exploring all possible subsets in $[n] \backslash \{i\}$;

2. Estimating $\nu_{do}(S)$ from finite samples $\mathscr{D}$.

Two components in do-Shapley estimation

$$\phi_{V_i}(\nu_{do}) \equiv \frac{1}{n} \sum_{S \subseteq [n] \backslash \{i\}} \binom{n-1}{|S|}^{-1} \{\nu_{do}(S \cup \{i\}) - \nu_{do}(S)\}.$$

Computing the Shapley value requires

1. Exploring all possible subsets in $[n] \backslash \{i\}$; Takes exponential computational time!

2. Estimating $\nu_{do}(S)$ from finite samples $\mathcal{D}$.

Two components in do-Shapley estimation

$$\phi_{V_i}(\nu_{do}) \equiv \frac{1}{n} \sum_{S \subseteq [n] \setminus \{i\}} \binom{n-1}{|S|}^{-1} \{\nu_{do}(S \cup \{i\}) - \nu_{do}(S)\}.$$

Computing the Shapley value requires

1. Exploring all possible subsets in $[n]\setminus\{i\}$; Takes exponential computational time!

   Random Permutation based approximation

2. Estimating $\nu_{do}(S)$ from finite samples $\mathcal{D}$.

Two components in do-Shapley estimation

$$\phi_{V_i}(\nu_{do}) \equiv \frac{1}{n} \sum_{S \subseteq [n] \setminus \{i\}} \binom{n-1}{|S|}^{-1} \{\nu_{do}(S \cup \{i\}) - \nu_{do}(S)\}.$$

Computing the Shapley value requires

1. Exploring all possible subsets in $[n]\backslash\{i\}$; Takes exponential computational time!

   Random Permutation based approximation

2. Estimating $\nu_{do}(S)$ from finite samples $\mathscr{D}$. An estimator robust to bias is desirable!

Two components in do-Shapley estimation

$$\phi_{V_i}(\nu_{do}) \equiv \frac{1}{n} \sum_{S \subseteq [n] \setminus \{i\}} \binom{n-1}{|S|}^{-1} \{\nu_{do}(S \cup \{i\}) - \nu_{do}(S)\}.$$

Computing the Shapley value requires

1. Exploring all possible subsets in $[n] \setminus \{i\}$; Takes exponential computational time!

   Random Permutation based approximation

2. Estimating $\nu_{do}(S)$ from finite samples $\mathscr{D}$. An estimator robust to bias is desirable!

   Double/Debiased Machine Learning (DML) [Chernozhukov, 2018]

Two components in do-Shapley estimation

$$\phi_{V_i}(\nu_{do}) \equiv \frac{1}{n} \sum_{S \subseteq [n] \backslash \{i\}} \binom{n-1}{|S|}^{-1} \{\nu_{do}(S \cup \{i\}) - \nu_{do}(S)\}.$$

Computing the Shapley value requires

1. Exploring all possible subsets in $[n] \backslash \{i\}$;

   Random Permutation based approximation

2. Estimating $\nu_{do}(S)$ from finite samples $\mathscr{D}$.

   An estimator robust to bias is desirable!

   Double/Debiased Machine Learning (DML) [Chernozhukov, 2018]

Two components in do-Shapley estimation

$$\phi_{V_i}(\nu_{do}) \equiv \frac{1}{n} \sum_{S \subseteq [n] \setminus \{i\}} \binom{n-1}{|S|}^{-1} \{\nu_{do}(S \cup \{i\}) - \nu_{do}(S)\}.$$

Computing the Shapley value requires

1. Exploring all possible subsets in $[n] \setminus \{i\}$;

   Random Permutation based approximation

2. Estimating $\nu_{do}(S)$ from finite samples $\mathcal{D}$. An estimator robust to bias is desirable!

   Double/Debiased Machine Learning (DML) [Chernozhukov, 2018]

Monte-Carlo approximation for do-Shapley (1)

$$\phi_i \equiv \frac{1}{n} \sum\nolimits_{S \subseteq [n] \setminus \{i\}} \binom{n-1}{|S|}^{-1} \{v(S \cup \{i\}) - v(S)\}.$$

Monte-Carlo approximation for do-Shapley (1)

$$\phi_i \equiv \frac{1}{n} \sum_{S \subseteq [n] \setminus \{i\}} \binom{n-1}{|S|}^{-1} \{v(S \cup \{i\}) - v(S)\}.$$

$$= \frac{1}{n!} \sum_{\pi(\mathbf{V}) \in \text{perm}(\mathbf{V})} \{\nu(v_i, \text{pre}_\pi(v_i)) - \nu(\text{pre}_\pi(v_i))\}$$ [Štrumbelj and Kononenko, 2014]

Monte-Carlo approximation for do-Shapley (1)

$$\phi_i \equiv \frac{1}{n} \sum_{S \subseteq [n] \setminus \{i\}} \binom{n-1}{|S|}^{-1} \{v(S \cup \{i\}) - v(S)\}.$$

$$= \frac{1}{n!} \sum_{\pi(\mathbf{V}) \in \text{perm}(\mathbf{V})} \{\nu(v_i, \text{pre}_\pi(v_i)) - \nu(\text{pre}_\pi(v_i))\}$$ [Štrumbelj and Kononenko, 2014]

all possible permutation of $\mathbf{V} = \{V_i\}_{i=1}^n$

Predecessor of V_i given the fixed permutation $\pi(\mathbf{V})$.

Monte-Carlo approximation for do-Shapley (1)

$$\phi_i \equiv \frac{1}{n} \sum_{S \subseteq [n] \setminus \{i\}} \binom{n-1}{|S|}^{-1} \{v(S \cup \{i\}) - v(S)\}.$$

$$= \frac{1}{n!} \sum_{\pi(\mathbf{V}) \in \mathrm{perm}(\mathbf{V})} \{\nu(v_i, \mathrm{pre}_\pi(v_i)) - \nu(\mathrm{pre}_\pi(v_i))\}$$ [Štrumbelj and Kononenko, 2014]

all possible permutation of $\mathbf{V} = \{V_i\}_{i=1}^n$

Predecessor of V_i given the fixed permutation $\pi(\mathbf{V})$.

$$= \mathbb{E}_{\pi(\mathbf{V})} \left[\nu(v_i, \mathrm{pre}_\pi(v_i)) - \nu(\mathrm{pre}_\pi(v_i))\right]$$

Monte-Carlo approximation for do-Shapley (1)

$$\phi_i \equiv \frac{1}{n} \sum_{S \subseteq [n] \setminus \{i\}} \binom{n-1}{|S|}^{-1} \{v(S \cup \{i\}) - v(S)\}.$$

$$= \frac{1}{n!} \sum_{\pi(\mathbf{V}) \in \mathrm{perm}(\mathbf{V})} \{\nu(v_i, \mathrm{pre}_\pi(v_i)) - \nu(\mathrm{pre}_\pi(v_i))\}$$

[Štrumbelj and Kononenko, 2014]

all possible permutation of $\mathbf{V} = \{V_i\}_{i=1}^n$

Predecessor of V_i given the fixed permutation $\pi(\mathbf{V})$.

$$= \mathbb{E}_{\pi(\mathbf{V})} \left[\nu(v_i, \mathrm{pre}_\pi(v_i)) - \nu(\mathrm{pre}_\pi(v_i))\right]$$

The expectation is over the probability for each permutation order $\pi(\mathbf{V})$, where $P(\pi) = \frac{1}{n!}$.

Monte-Carlo approximation for do-Shapley (2)

$$\phi_i = \mathbb{E}_{\pi(\mathbf{V})}\left[\nu(v_i, \mathrm{pre}_\pi(v_i)) - \nu(\mathrm{pre}_\pi(v_i))\right].$$

Monte-Carlo approximation for do-Shapley (2)

$$\phi_i = \mathbb{E}_{\pi(\mathbf{V})}\left[\nu(v_i, \mathsf{pre}_\pi(v_i)) - \nu(\mathsf{pre}_\pi(v_i))\right].$$

$$\tilde{\phi}_i = \frac{1}{M}\sum_{m=1}^{M}\left\{\nu(v_i, \mathsf{pre}_{\pi_{(m)}}(v_i)) - \nu(\mathsf{pre}_{\pi_{(m)}}(v_i))\right\}$$

Monte-Carlo approximation for do-Shapley (2)

$$\phi_i = \mathbb{E}_{\pi(\mathbf{V})}\left[\nu(v_i, \text{pre}_\pi(v_i)) - \nu(\text{pre}_\pi(v_i))\right].$$

$$\tilde{\phi}_i = \frac{1}{M}\sum_{m=1}^{M}\left\{\nu(v_i, \text{pre}_{\pi_{(m)}}(v_i)) - \nu(\text{pre}_{\pi_{(m)}}(v_i))\right\}$$

- For M number of randomly generated permutations of $\mathbf{V}$ (where each permutations are denoted $\pi_{(m)}$),

Monte-Carlo approximation for do-Shapley (2)

$$\phi_i = \mathbb{E}_{\pi(\mathbf{V})}\left[\nu(v_i, \text{pre}_\pi(v_i)) - \nu(\text{pre}_\pi(v_i))\right].$$

$$\tilde{\phi}_i = \frac{1}{M}\sum_{m=1}^{M}\left\{\nu(v_i, \text{pre}_{\pi_{(m)}}(v_i)) - \nu(\text{pre}_{\pi_{(m)}}(v_i))\right\}$$

- For M number of randomly generated permutations of $\mathbf{V}$ (where each permutations are denoted $\pi_{(m)}$),
- Compute $\nu(v_i, \text{pre}_{\pi_{(m)}}(v_i)) - \nu(\text{pre}_{\pi_{(m)}}(v_i))$ and take an average.

Random permutation-based algorithm

Random permutation-based algorithm

1. Initiate $\phi_{V_i} = 0$ for all $V_i \in \mathbf{V}$.

Random permutation-based algorithm

1. Initiate $\phi_{V_i} = 0$ for all $V_i \in \mathbf{V}$.

2. Generate M randomly generated permutations of $\mathbf{V}$. The permuted variables are $\mathbf{V}_\pi = \{V_{\pi,1}, \cdots, V_{\pi,n}\}$, where $V_{\pi,i}$ is the ith variable in the permutation π.

Random permutation-based algorithm

1. Initiate $\phi_{V_i} = 0$ for all $V_i \in \mathbf{V}$.

2. Generate M randomly generated permutations of $\mathbf{V}$. The permuted variables are $\mathbf{V}_\pi = \{V_{\pi,1}, \cdots, V_{\pi,n}\}$, where $V_{\pi,i}$ is the ith variable in the permutation π.

3. For each $i = 1,2,\cdots, n$, compute

$$\phi_{V_i} \leftarrow \phi_{V_i} + \left\{\nu_{do}(V_{\pi,i}, pre_\pi(V_{\pi,i})) - \nu_{do}(pre_\pi(V_{\pi,i}))\right\}$$

Random permutation-based algorithm

1. Initiate $\phi_{V_i} = 0$ for all $V_i \in \mathbf{V}$.

2. Generate M randomly generated permutations of $\mathbf{V}$. The permuted variables are $\mathbf{V}_\pi = \{V_{\pi,1}, \cdots, V_{\pi,n}\}$, where $V_{\pi,i}$ is the ith variable in the permutation π.

3. For each $i = 1,2,\cdots,n$, compute

$$\phi_{V_i} \leftarrow \phi_{V_i} + \left\{\nu_{do}(V_{\pi,i}, pre_\pi(V_{\pi,i})) - \nu_{do}(pre_\pi(V_{\pi,i}))\right\}$$

4. For each $i = 1,2,\cdots,n$, $\phi_{V_i} \leftarrow (1/M) \cdot \phi_{V_i}$.

Monte-Carlo approximation for do-Shapley (1)

Let $\nu(S) := \mathbb{E}[Y \mid do(\mathbf{v}_S)]$, where $\mathbf{V}_S \subseteq \mathbf{V}$

$$\phi_i \equiv \frac{1}{n} \sum_{S \subseteq [n] \setminus \{i\}} \binom{n-1}{|S|}^{-1} \{v(S \cup \{i\}) - v(S)\}.$$

Monte-Carlo approximation for do-Shapley (1)

Let $\nu(S) := \mathbb{E}[Y \mid do(\mathbf{v}_S)]$, where $\mathbf{V}_S \subseteq \mathbf{V}$

$$\phi_i \equiv \frac{1}{n} \sum_{S \subseteq [n] \setminus \{i\}} \binom{n-1}{|S|}^{-1} \{v(S \cup \{i\}) - v(S)\}.$$

$$= \frac{1}{n!} \sum_{\pi(\mathbf{V}) \in \text{perm}(\mathbf{V})} \{\nu(v_i, \text{pre}_\pi(v_i)) - \nu(\text{pre}_\pi(v_i))\}$$

Monte-Carlo approximation for do-Shapley (1)

Let $\nu(S) := \mathbb{E}[Y \mid do(\mathbf{v}_S)]$, where $\mathbf{V}_S \subseteq \mathbf{V}$

$$\phi_i \equiv \frac{1}{n} \sum_{S \subseteq [n] \setminus \{i\}} \binom{n-1}{|S|}^{-1} \{v(S \cup \{i\}) - v(S)\}.$$

$$= \frac{1}{n!} \sum_{\pi(\mathbf{V}) \in \text{perm}(\mathbf{V})} \{\nu(v_i, \text{pre}_\pi(v_i)) - \nu(\text{pre}_\pi(v_i))\}$$

all possible permutation of $\mathbf{V} = \{V_i\}_{i=1}^{n}$

Predecessor of V_i given the fixed permutation $\pi(\mathbf{V})$.

Monte-Carlo approximation for do-Shapley (1)

Let $\nu(S) := \mathbb{E}[Y \mid do(\mathbf{v}_S)]$, where $\mathbf{V}_S \subseteq \mathbf{V}$

$$\phi_i \equiv \frac{1}{n} \sum_{S \subseteq [n] \setminus \{i\}} \binom{n-1}{|S|}^{-1} \{v(S \cup \{i\}) - v(S)\}.$$

$$= \frac{1}{n!} \sum_{\pi(\mathbf{V}) \in \text{perm}(\mathbf{V})} \{\nu(v_i, \text{pre}_\pi(v_i)) - \nu(\text{pre}_\pi(v_i))\}$$

all possible permutation of $\mathbf{V} = \{V_i\}_{i=1}^n$

Predecessor of V_i given the fixed permutation $\pi(\mathbf{V})$.

$$= \mathbb{E}_{\pi(\mathbf{V})} \left[\nu(v_i, \text{pre}_\pi(v_i)) - \nu(\text{pre}_\pi(v_i))\right]$$

Monte-Carlo approximation for do-Shapley (1)

Let $\nu(S) := \mathbb{E}[Y \mid do(\mathbf{v}_S)]$, where $\mathbf{V}_S \subseteq \mathbf{V}$

$$\phi_i \equiv \frac{1}{n} \sum_{S \subseteq [n] \setminus \{i\}} \binom{n-1}{|S|}^{-1} \{v(S \cup \{i\}) - v(S)\}.$$

$$= \frac{1}{n!} \sum_{\pi(\mathbf{V}) \in \text{perm}(\mathbf{V})} \{\nu(v_i, \text{pre}_\pi(v_i)) - \nu(\text{pre}_\pi(v_i))\}$$

all possible permutation of $\mathbf{V} = \{V_i\}_{i=1}^n$

Predecessor of V_i given the fixed permutation $\pi(\mathbf{V})$.

$$= \mathbb{E}_{\pi(\mathbf{V})} \left[\nu(v_i, \text{pre}_\pi(v_i)) - \nu(\text{pre}_\pi(v_i))\right]$$

The expectation is over the probability for each permutation order $\pi(\mathbf{V})$, where $P(\pi) = \frac{1}{n!}$.

Monte-Carlo approximation for do-Shapley (2)

$$\phi_i = \mathbb{E}_{\pi(\mathbf{V})}\left[\nu(v_i, \text{pre}_\pi(v_i)) - \nu(\text{pre}_\pi(v_i))\right].$$

Monte-Carlo approximation for do-Shapley (2)

$$\phi_i = \mathbb{E}_{\pi(\mathbf{V})}\left[\nu(v_i, \mathrm{pre}_\pi(v_i)) - \nu(\mathrm{pre}_\pi(v_i))\right].$$

$$\tilde{\phi}_i = \frac{1}{M}\sum_{m=1}^{M}\left\{\nu(v_i, \mathrm{pre}_{\pi_{(m)}}(v_i)) - \nu(\mathrm{pre}_{\pi_{(m)}}(v_i))\right\}$$

Monte-Carlo approximation for do-Shapley (2)

$$\phi_i = \mathbb{E}_{\pi(\mathbf{V})}\left[\nu(v_i, \mathrm{pre}_\pi(v_i)) - \nu(\mathrm{pre}_\pi(v_i))\right].$$

$$\tilde{\phi}_i = \frac{1}{M}\sum_{m=1}^{M}\left\{\nu(v_i, \mathrm{pre}_{\pi_{(m)}}(v_i)) - \nu(\mathrm{pre}_{\pi_{(m)}}(v_i))\right\}$$

- For M number of randomly generated permutations of $\mathbf{V}$ (where each permutations are denoted $\pi_{(m)}$),

do-DML-Shapley

do-DML-Shapley

Recall the random permutation based Shapley approximation is

$$\tilde{\phi}_i = \frac{1}{M}\sum_{m=1}^{M}\left\{\nu(v_i, \text{pre}_{\pi_{(m)}}(v_i)) - \nu(\text{pre}_{\pi_{(m)}}(v_i))\right\}$$

do-DML-Shapley

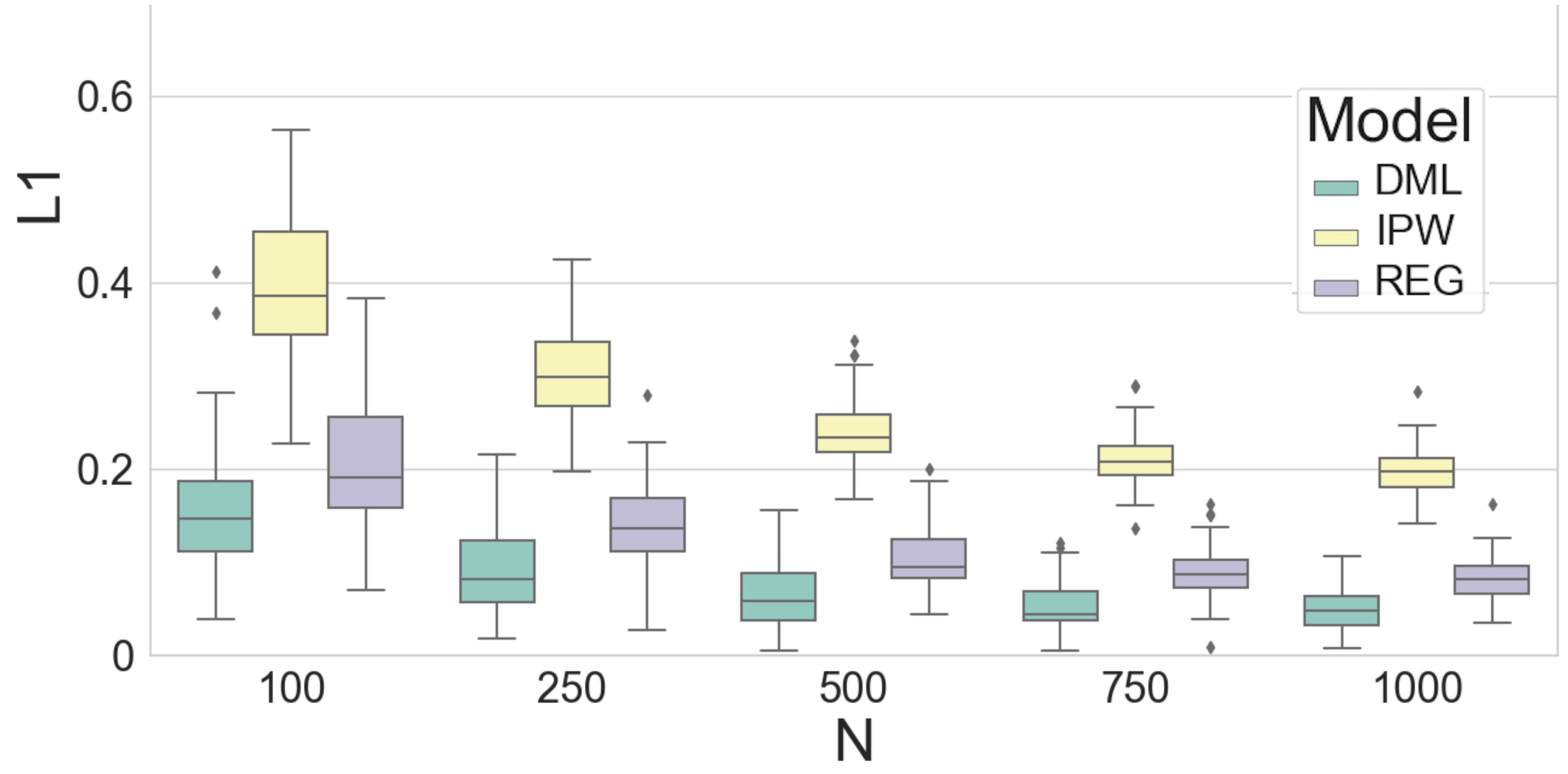
do-DML-Shapley

Recall the random permutation based Shapley approximation is

$$\widehat{\phi}_{V_i}(T) = \frac{1}{M}\sum_{m=1}^{M}\left\{T(v_i, \text{pre}_{\pi_{(m)}}(v_i)) - T(\text{pre}_{\pi_{(m)}}(v_i))\right\}$$

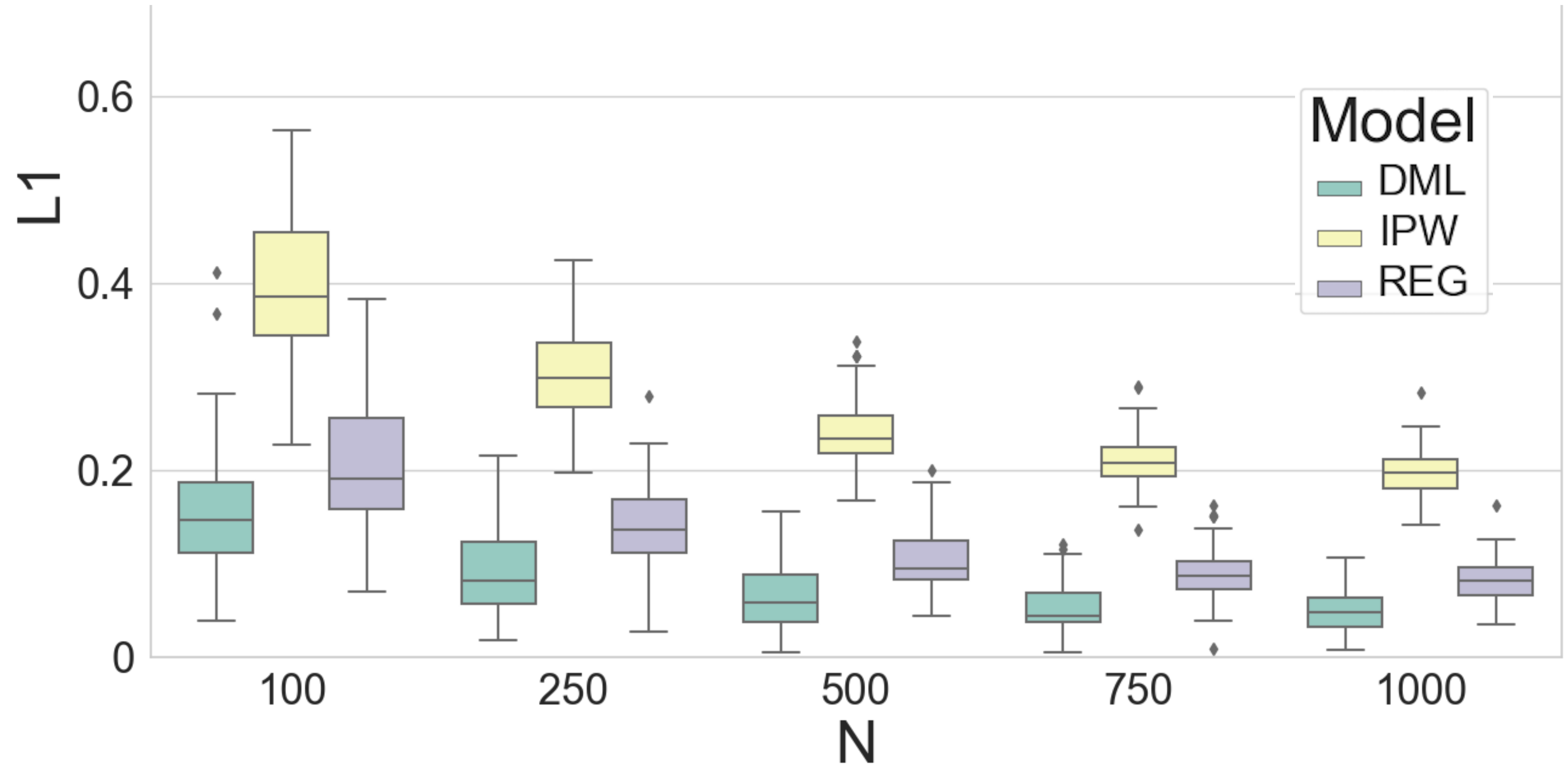
Simulation

Empirical Study: DML Property



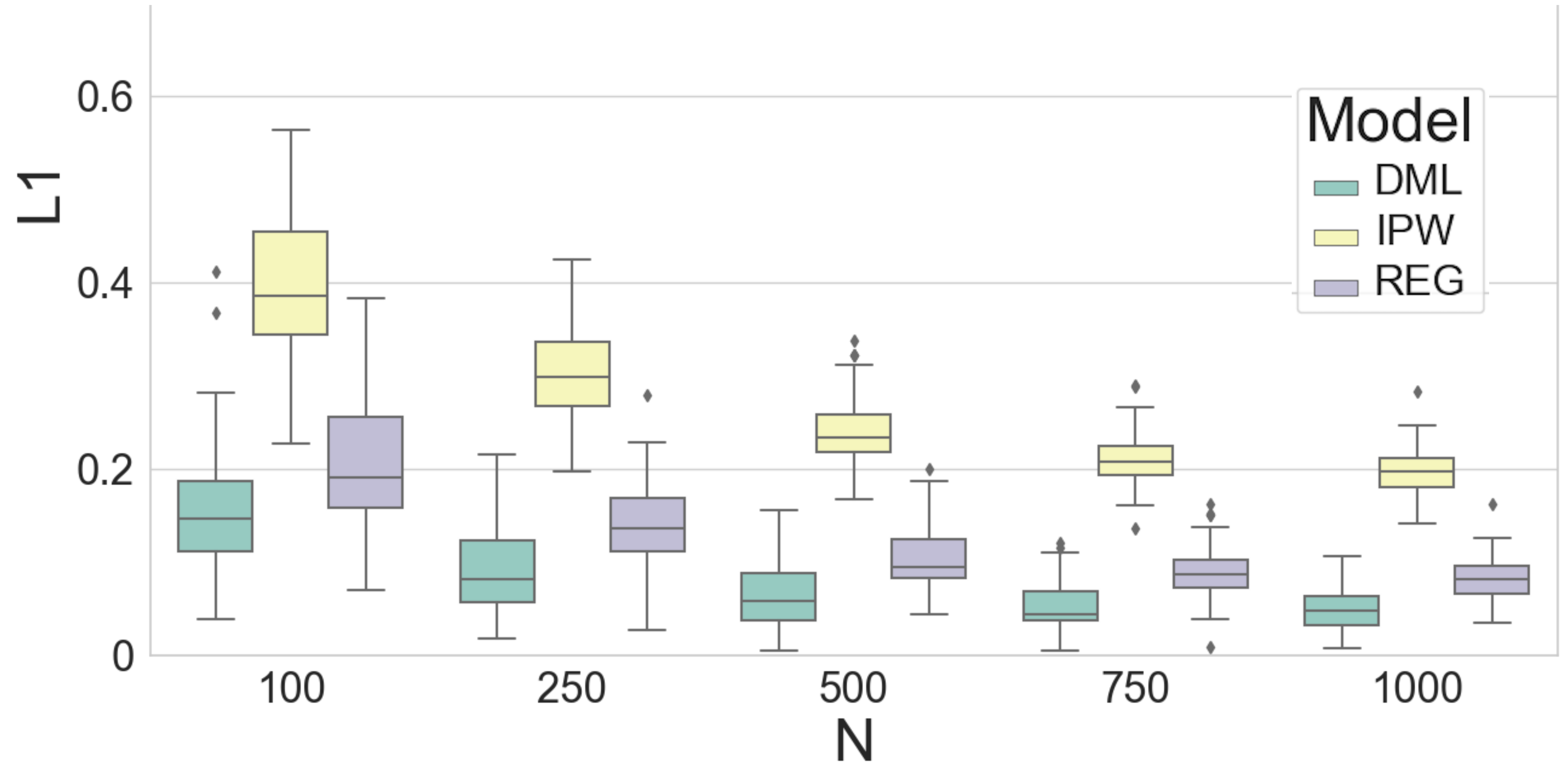
Empirical Study: DML Property

We compared the DML-based do-Shapley estimator with other existing estimators when the $\mathbb{E}[Y \mid do(\mathbf{v}_S)]$ is given as mSBD adjustment:



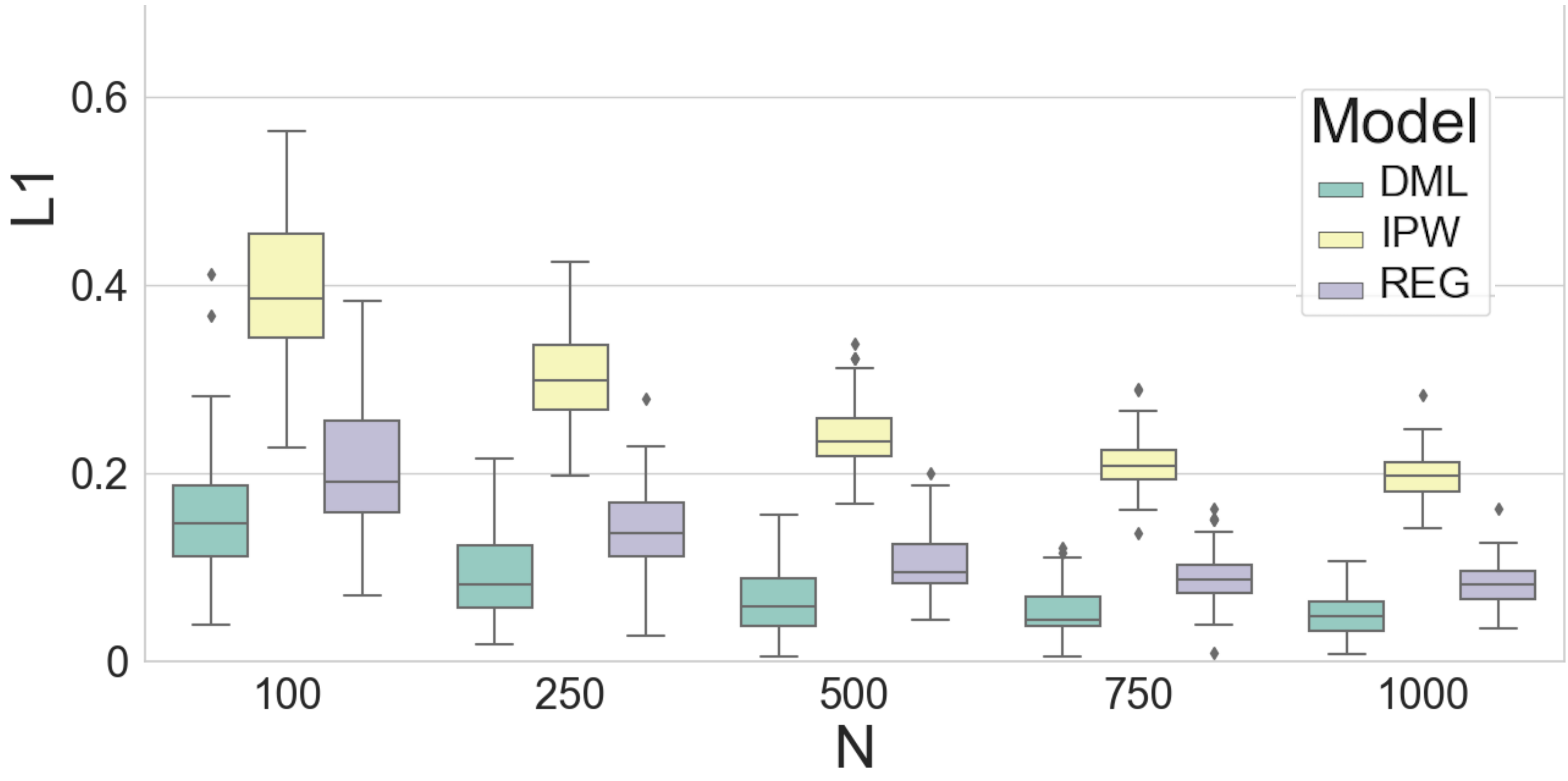
Empirical Study: DML Property

We compared the DML-based do-Shapley estimator with other existing estimators when the $\mathbb{E}[Y \mid do(\mathbf{v}_S)]$ is given as mSBD adjustment:



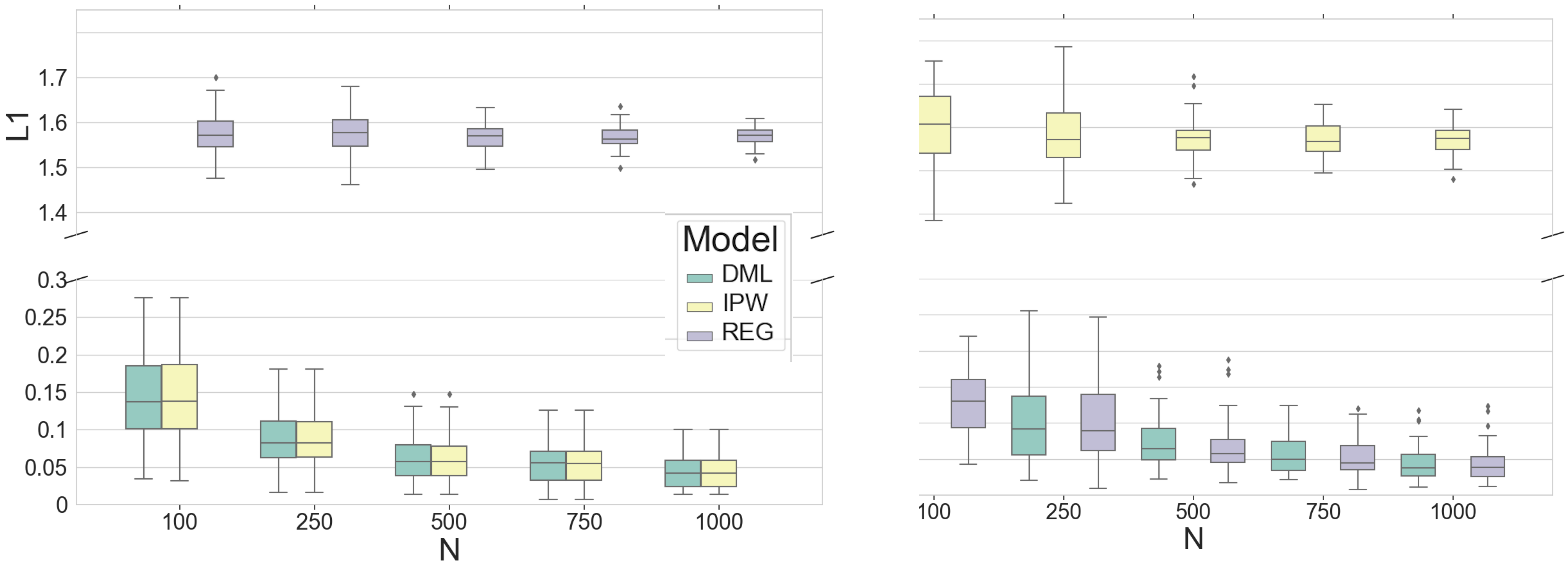
Empirical Study: DML Property

We compared the DML-based do-Shapley estimator with other existing estimators when the $\mathbb{E}[Y \mid do(\mathbf{v}_S)]$ is given as mSBD adjustment:



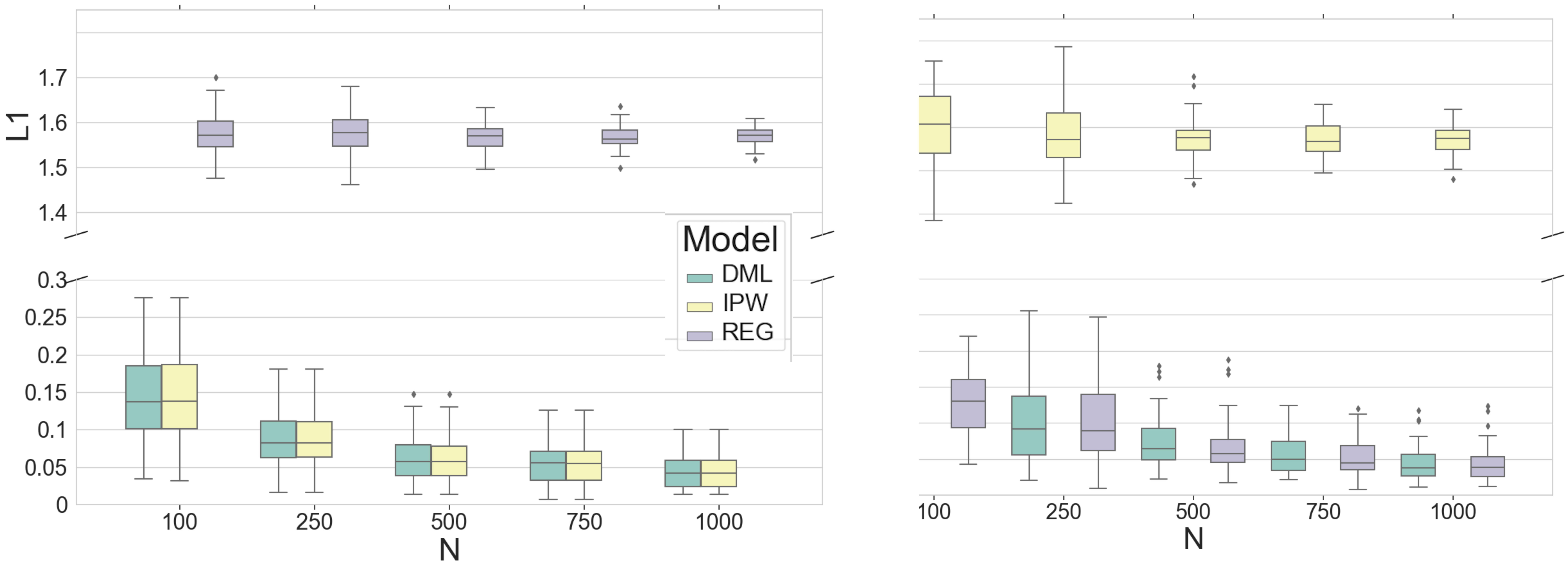
The DML estimator converges faster than competing estimators.

Empirical Study: DML Property

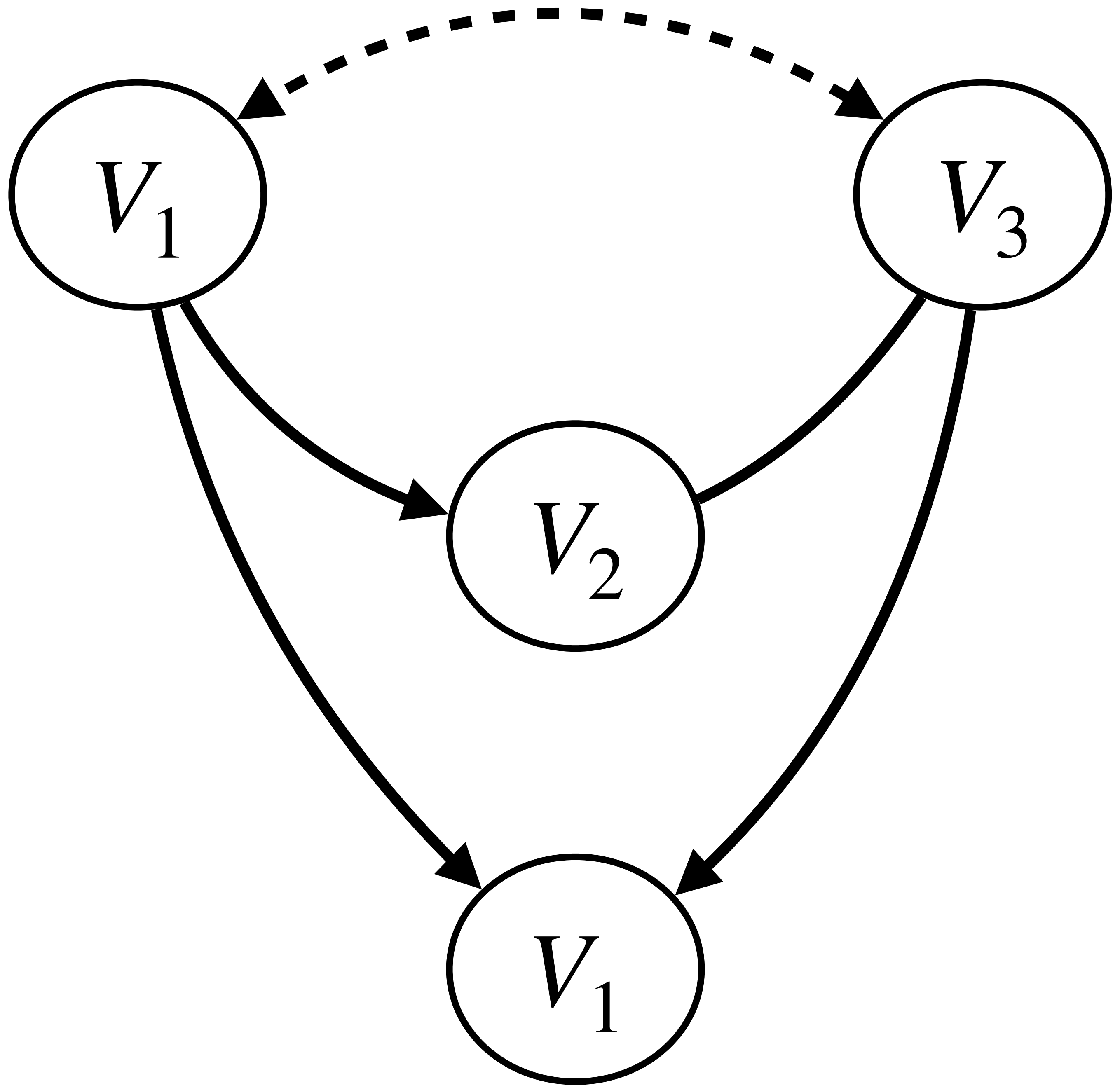


Empirical Study: DML Property

When nuisances corresponding to the IPW, REG estimators are misspecified, the DML estimator converges fast.

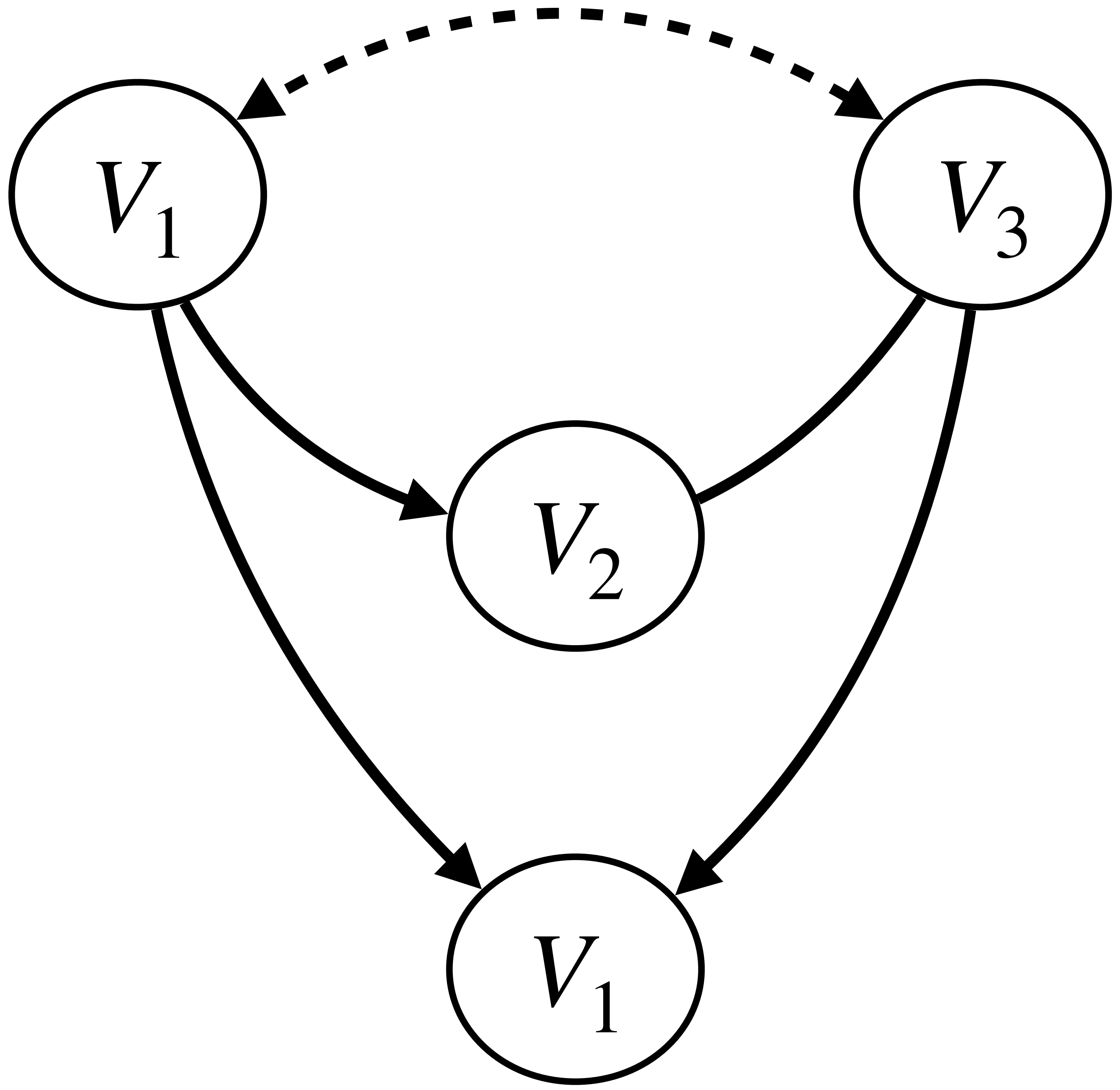


A simulation result



$Y = 3V_1 + 0.4V_2 + V_3 + U_Y$

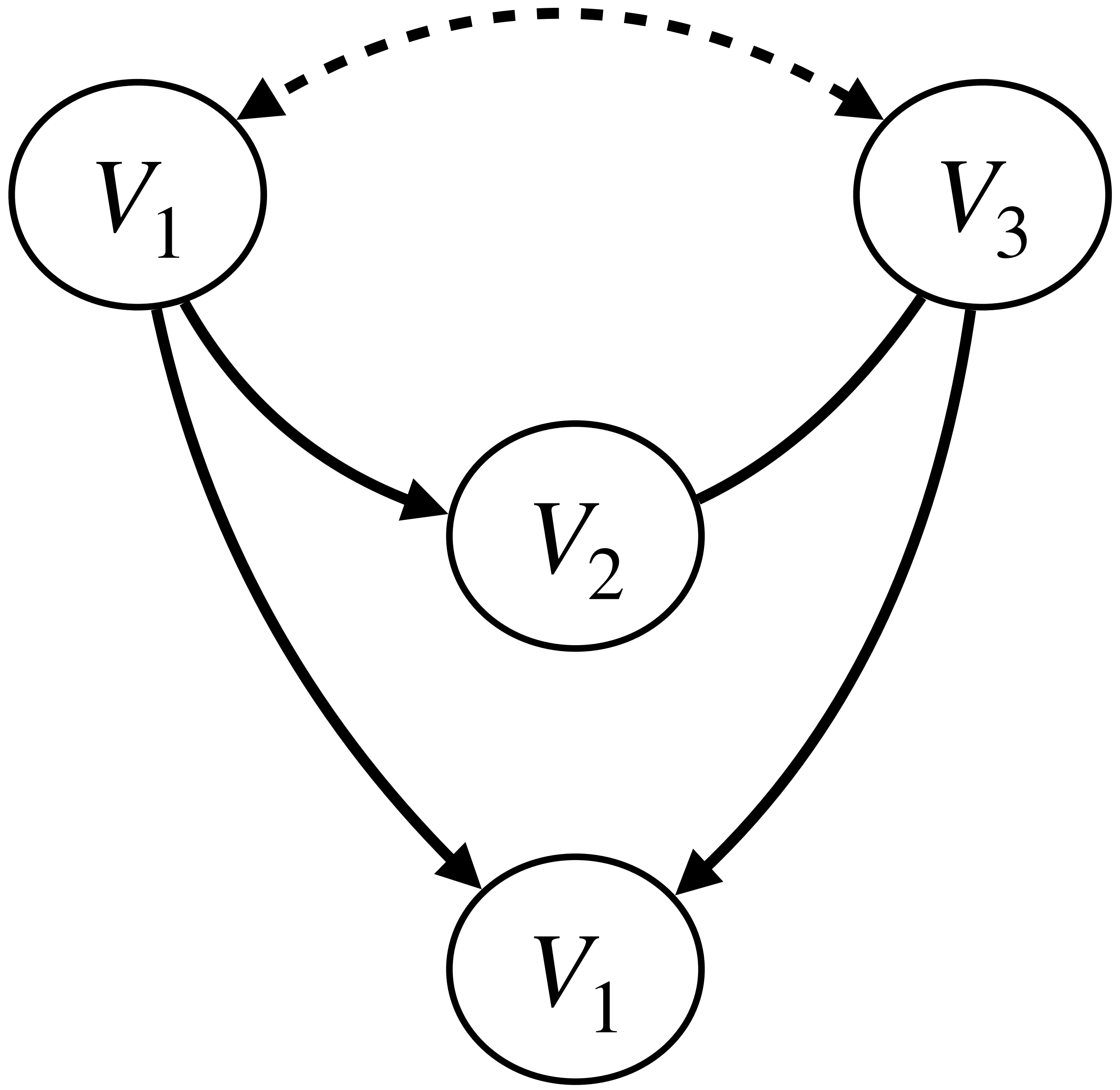
A simulation result



$$Y = 3V_1 + 0.4V_2 + V_3 + U_Y$$

We designed the DGP s.t. the importances are ordered as $V_1 > V_3 > V_2$.

A simulation result

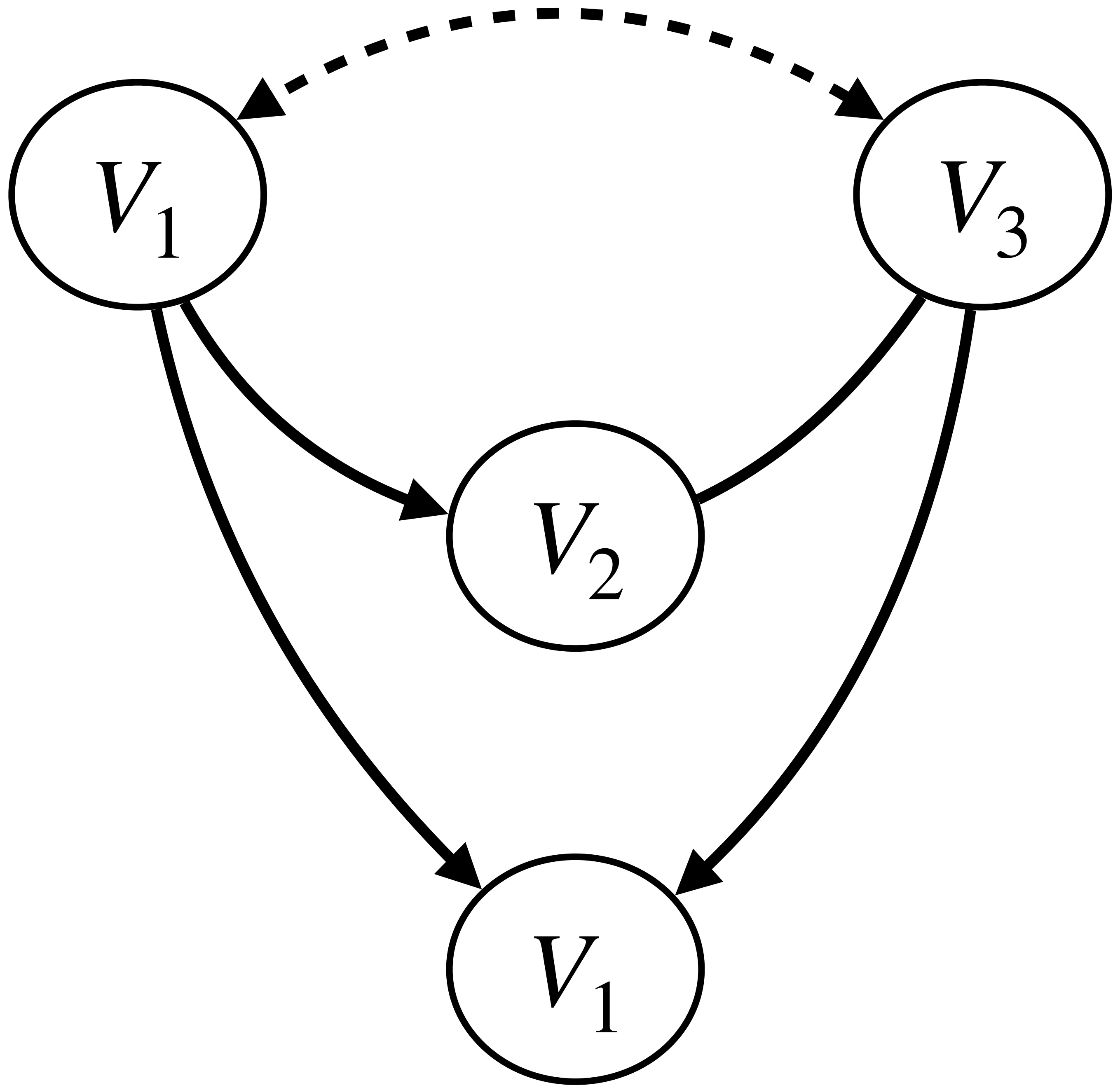


$$Y = 3V_1 + 0.4V_2 + V_3 + U_Y$$

We designed the DGP s.t. the importances are ordered as $V_1 > V_3 > V_2$.

We compared the DML-based do-Shapley based method with the conditional-Shapley.

A simulation result



$$Y = 3V_1 + 0.4V_2 + V_3 + U_Y$$

We designed the DGP s.t. the importances are ordered as $V_1 > V_3 > V_2$.

We compared the DML-based do-Shapley based method with the conditional-Shapley.

The DML-based do-Shapley ranks $V_1 > V_3 > V_2$, while the conditional Shapley ranks V_2 as the most important one, in our scenario.

Conclusion

Conclusion

We develop ***causally*** interpretable ***feature attribution method***.

Conclusion

We develop ***causally*** interpretable ***feature attribution method***.

1. We *axiomatize and characterize* a causally interpretable feature attribution method, and propose do-Shapley values.

Conclusion

We develop ***causally*** interpretable ***feature attribution method***.

1. We *axiomatize and characterize* a causally interpretable feature attribution method, and propose do-Shapley values.

2. We provide *identifiability* condition where the do-Shapley values can be inferred from the observational data.

Conclusion

We develop ***causally*** interpretable ***feature attribution method***.

1. We *axiomatize and characterize* a causally interpretable feature attribution method, and propose do-Shapley values.

2. We provide *identifiability* condition where the do-Shapley values can be inferred from the observational data.

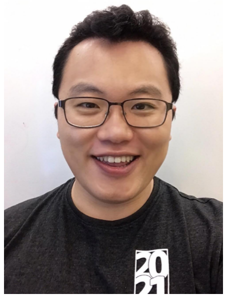
3. We construct a *double/debiased machine learning (DML)* [Chernozhukov et al., 2018] based do-Shapley estimator for practical settings.

Shortcut Learning in Machine Learning:
Challenges, Analysis, Solutions

https://sites.google.com/view/facct22-shortcut-learning/home

Sanghyuk Chun
NAVER AI Lab
https://sanghyukchun.github.io/home/

Kyungwoo Song
University of Seoul
https://mlai.uos.ac.kr/



Yonghan Jung
Purdue University
http://yonghanjung.me/

Introduction to Shortcut Learning

Sanghyuk Chun

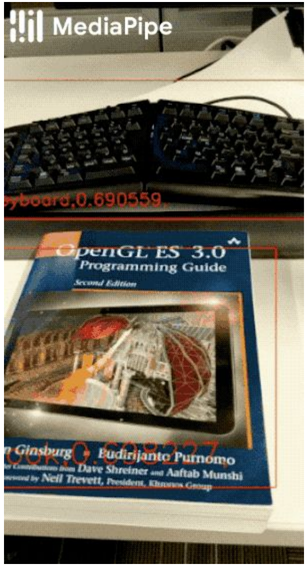
NAVER AI Lab

https://sanghyukchun.github.io/home/

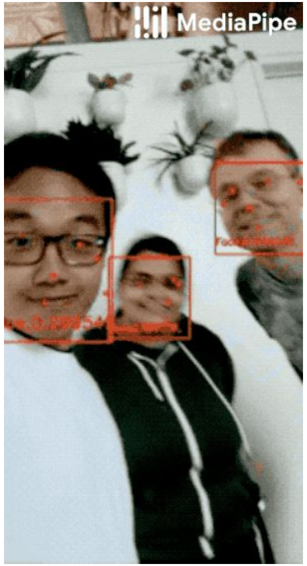
Machine Learning (ML) opens a new stage of automation.

Pose estimation

Object detection



Face recognition



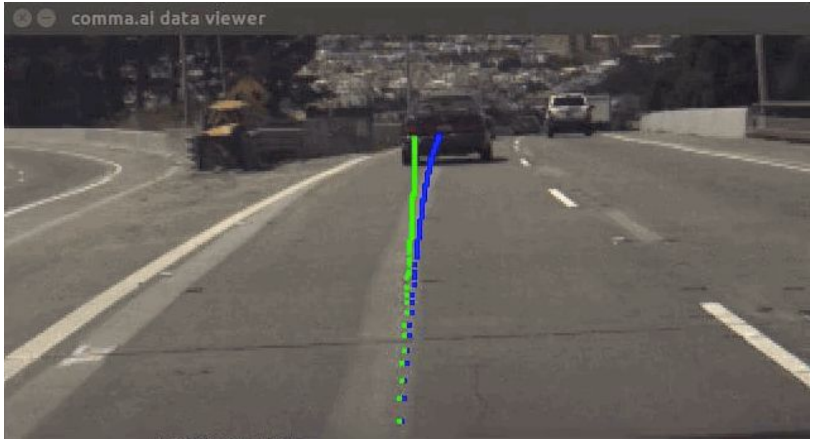
In-the-wild examples from https://google.github.io/mediapipe/

Machine Learning (ML) opens a new stage of automation.

https://towardsdatascience.com/some-experiments-using-github-copilot-with-python-90f8065fb72e

Machine Learning (ML) opens a new stage of automation.

Line tracing for self-driving cars



Semantic segmentation for self-driving cars

https://github.com/commaai/research
https://studentsxstudents.com/using-semantic-segmentation-to-give-a-self-driving-car-the-ability-to-see-6c97425ec562

However, AI often cannot understand the problem itself.

- An object detection model is easily fooled by a semantically meaningless patch image (failed to detect “person” if the patch is near the person)

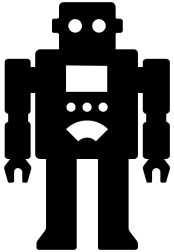
Video from: https://youtu.be/MIbFvK2S9g8?t=54

Thys, et al. “Fooling Automated Surveillance Cameras: Adversarial Patches to Attack Person Detection”, CVPR 2019

However, AI often cannot understand the problem itself.

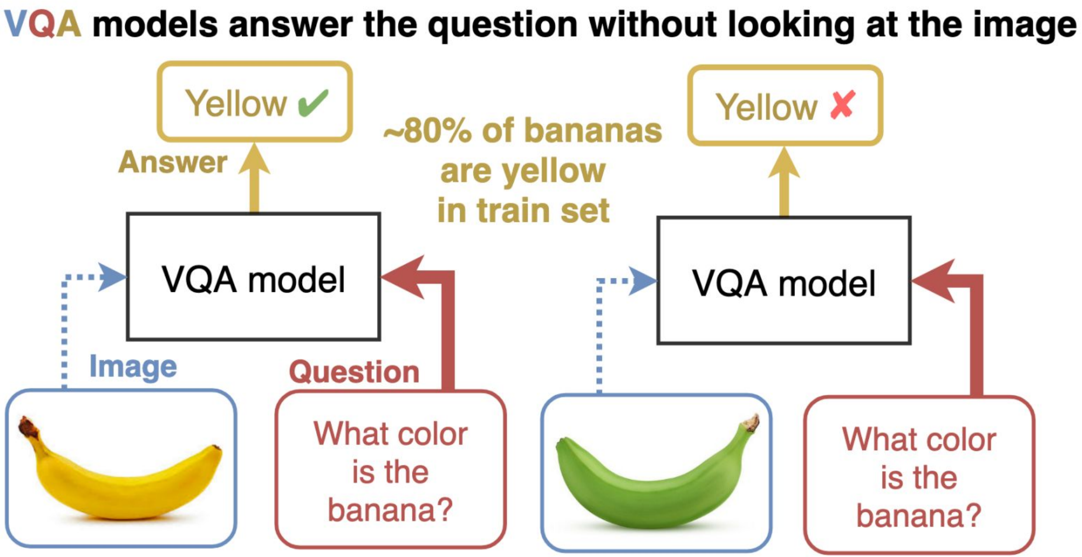
- A self-driving car thinks “Burger King sign 🍔” is a “stop sign 🛑”

https://www.youtube.com/watch?v=jheBCOpE9ws



Both signs are
“Stop sign 🛑”

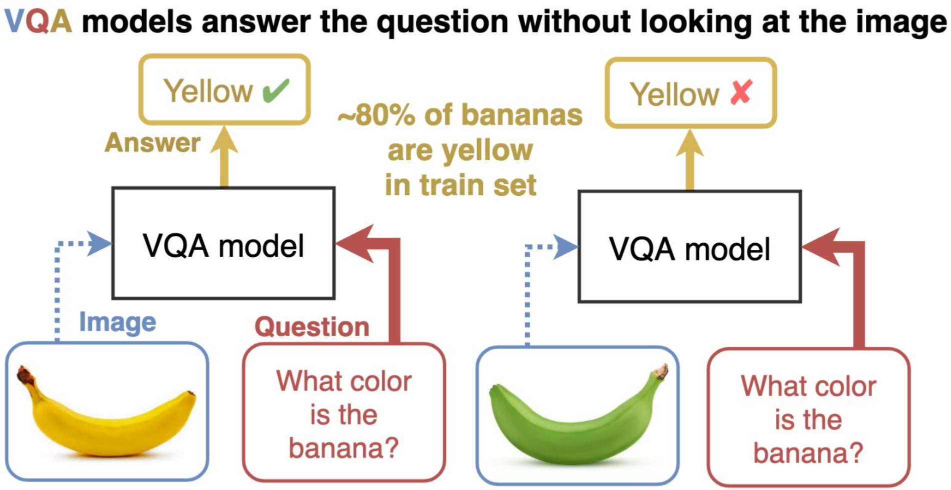
ML models often rely on “easy-to-learn shortcuts” without an understanding of the problem itself.



Cadene, et al. “RUBi: Reducing Unimodal Biases for Visual Question Answering”, NeurIPS 2019

“Shortcut learning” problem?

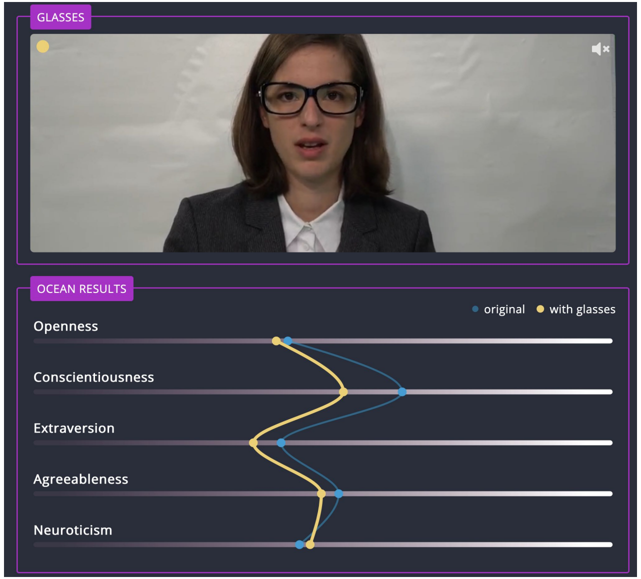
- When a model does not make a decision based on **“desired”** features (considering both question and image – color in this case), but **“undesired”** features (ignoring image), there exists **a shortcut learning problem.**

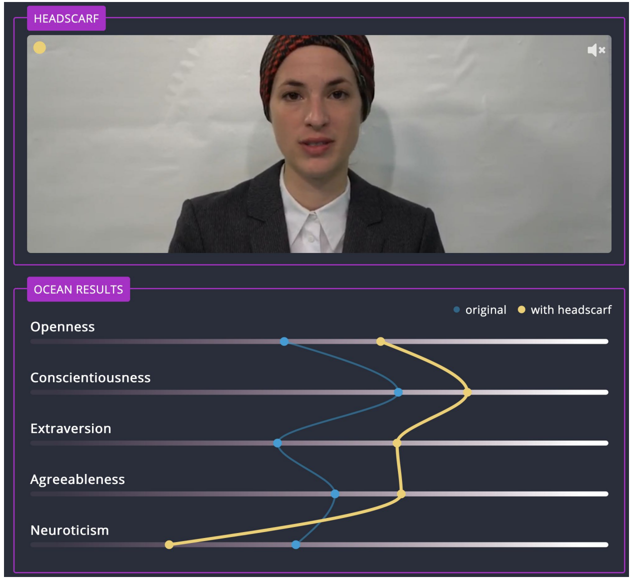


Cadene, et al. “RUBi: Reducing Unimodal Biases for Visual Question Answering”, NeurIPS 2019

Shortcut learning in human-related applications.

- When the actress acts **the same script and the same action** but **with different appearance (with glasses or with headscarf)**, **the predictions vary significantly!**

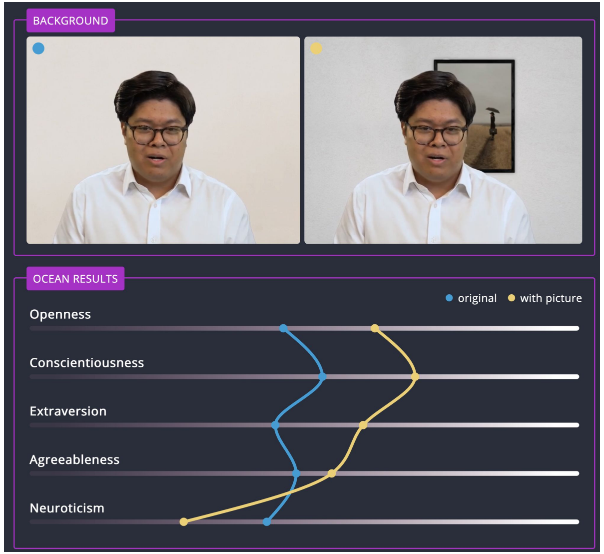


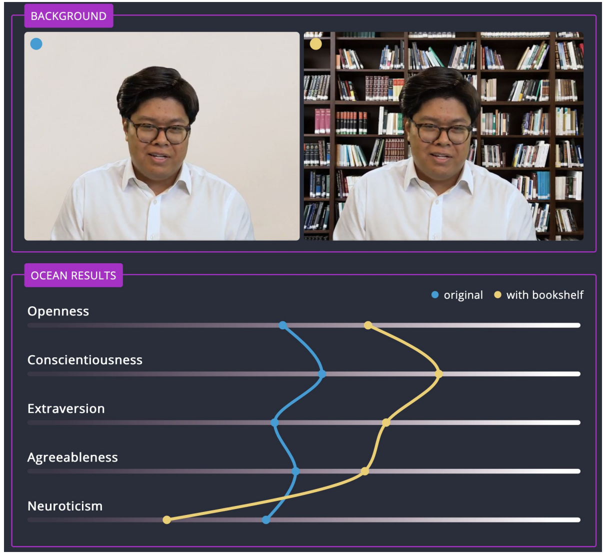


Objective or Biased https://interaktiv.br.de/ki-bewerbung/en/

Shortcut learning in human-related applications.

- Similarly, **the predictions vary significantly** with **the same script and the same action** but **with different backgrounds (with picture, with bookshelf)**.

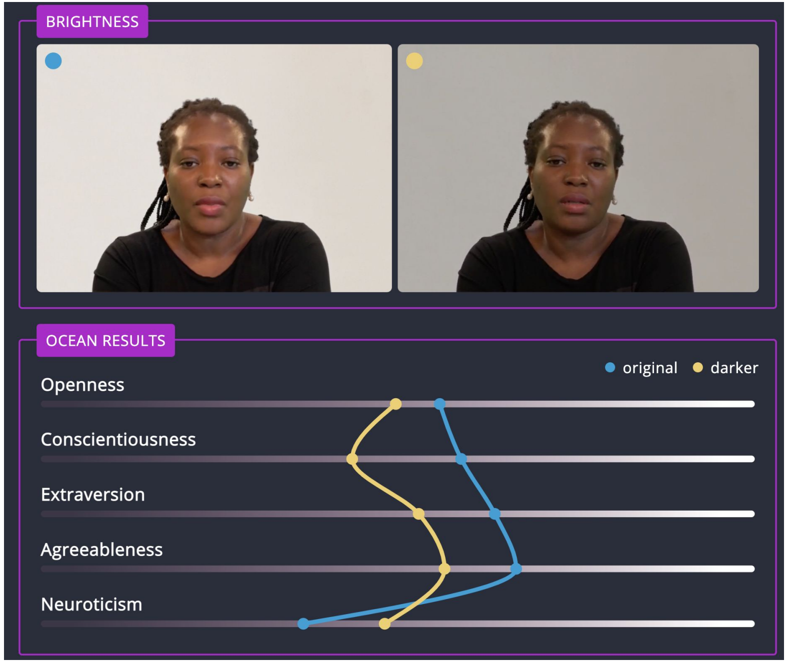




Objective or Biased https://interaktiv.br.de/ki-bewerbung/en/

Shortcut learning in human-related applications.

- Even for different brightness settings!



Objective or Biased https://interaktiv.br.de/ki-bewerbung/en/

Summary of Part 1

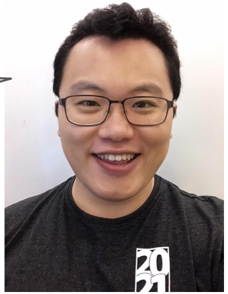
- The **shortcut learning problem** happens when the model makes a decision based on **“undesired”** feature, not **“desired”** feature.
- There exist a lot of examples of shortcut learning in machine learning algorithms.
- Naive learning strategy will lead to shortcut learning if we do not consider what is the desired feature for the given task.

References

- Geirhos, et al. “Shortcut Learning in Deep Neural Networks”, Nature Machine Intelligence 2020
- Scimeca, et al. “Which Shortcut Cues Will DNNs Choose? A Study from the Parameter-Space Perspective”, ICLR 2022.
- Objective or Biased https://interaktiv.br.de/ki-bewerbung/en/
- Cadene, et al. “RUBi: Reducing Unimodal Biases for Visual Question Answering”, NeurIPS 2019

Understanding Shortcut Learning through the Lens of Causality & Invariance

Tech report for this part is available in our tutorial website!



Light Talk!

Yonghan Jung

Purdue University

http://yonghanjung.me/

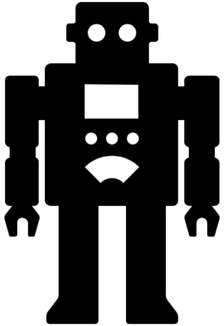
Motivational Example for Shortcut Learning - 1

Consider the task of classifying images of “boat” and “car”.

Data

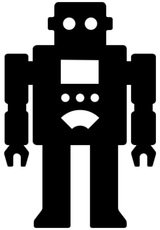
Train

I am learning a *decision rule* for classifying “boat” and “car”!



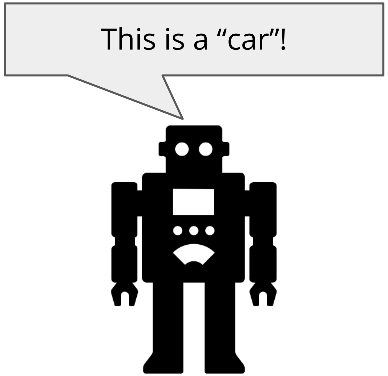
Classification

This is a “boat”!



Motivational Example for Shortcut Learning - 2

Modern ML models oftentimes make a mistake…



What’s going on…?

Motivational Example for Shortcut Learning - 3

The mistakes happened when ML models used *unintended / undesired* features (e.g., background) as a decision rule.

Data

Train

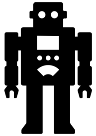
[water] background means “boat”!

[road] background means “car”!

... and it works pretty well for the training data!

For a new data

There is a road background!



This is a “car”!

Definition of Shortcut Learning

This phenomenon has been named “shortcut learning.”

Shortcut Learning [Geirhos et al., 2020]

A shortcut learning is a phenomenon in which ML models fail to generalize for a new sample due to taking *unintended / undesired features* called ***shortcuts*** (e.g., background objects) in establishing decision rules.

Overview of This Part

1. We will provide a formal understanding of shortcut learning through causality.

2. We will propose two approaches for preventing shortcut learning, which both suggest using causal features (a set of features that directly causes true labels).

    i. Approach 1. Avoid causally irrelevant features to the true label as much as possible.

    ii. Approach 2. Find an ML model that works best for all heterogeneous data generating processes.

3. We will provide a principle for identifying causal features by leveraging the causal invariance property.

Overview

1. We will provide a formal understanding of shortcut learning through causality.

2. We will propose two approaches for preventing shortcut learning, which both suggest using causal features (a set of features that directly causes true labels).

   i. Approach 1. Avoid causally irrelevant features to the true label as much as possible.

   ii. Approach 2. Find an ML model that works best for all heterogeneous data generating processes.

3. We will provide a principle for identifying causal features by leveraging the causal invariance property.

Expressing Data Generating Process with Multiple Functions

Data and Label

“Boat”

Data Generating Process

[water] $\leftarrow f_B(U_B)$, where U_W is some unknown variable. (generating function for the background)

[boat] $\leftarrow f_T(U_T)$, where U_B is some unknown variable. (generating function for the target)

“Boat”

[Label] $\leftarrow f_Y(U_{Y,}\ T)$, where T is a [boat] object. (generating function for the label)

Structural Causal Model as a Data Generating Process

This view of the data generating process is formalized as a *Structural Causal Model (SCM)*.

Structural Causal Models (SCM) [Pearl, 2000]

A structural causal model is a tuple $\mathcal{M} := \langle \mathbf{V}, \mathbf{U}, \mathbf{F}, P(\mathbf{U}) \rangle$

- **V** is a set of observed variables: $\mathbf{V} = \{V_1, ..., V_n\}$
- **U** is a set of latent variables
- **F** is a set of functions $\{F_{v_i}\}$ determining the value of V_i; i.e., $V_i = F_{v_i}(PA_i, U_i)$ where $PA_i \subseteq \mathbf{V}$ and $U_i \subseteq \mathbf{U}$.
- **P(U)** is a distribution over **U.**

Example: SCM as a Data Generating Process

SCM

$B \longleftarrow f_B(U_B)$ (Background like [water])

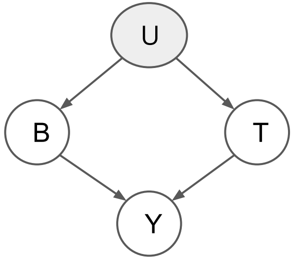
$T \longleftarrow f_T(U_T)$ (Target like [boat])

$Y \longleftarrow f_Y(B, U_Y)$ (Label)

$P(B,T,Y)$ from the functions and $P(U)$

Data

Graph



- SCM generates the data.
- Instead of access to the SCM, we have a graph, as a qualitative description.

Encoding Intervention on DGP in SCM

Intervention in SCM

B 👉 [Road] Formally, *do*(B = [Road])
(Pearl, 2000)

$T \leftarrow f_T(U_T)$ (Target like [boat])

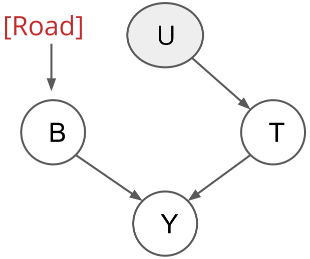
$Y \leftarrow f_Y(B, U_Y)$ (Label)

P(B,T,Y) from the functions and P(U)

Intervene to force the background to be “Road”

Data

Graph



- Intervened SCM (with do-operator) generates the data.
- Graphically, the intervention is expressed as cutting all incoming edges.

Submodel: Encoding Intervention through SCM

The SCM (Data generating process) *induced by intervention* is called a ***submodel*** of the SCM.

Submodels of the SCM [Pearl, 2000]

Given SCM $M := <\mathbf{V}, \mathbf{U}, \mathbf{F}, P(\mathbf{U})>$, the submodel M_{V_i} is the SCM induced by replacing a function $V_i \leftarrow F_{v_i}$ as a fixed constant $V_i \leftarrow v_i$ (i.e., operating $do(V_i = v_i)$).

Environments: A set of submodels of SCMs.

An original image ([boat in the water]) and its perturbed example ([boat in the road]) can be viewed as objects generated by an SCM M and its submodels M_{Vi}.

Environments [Peters et al., 2016]

Environments $\mathcal{E}$ is a set of an SCM M and its submodels M_{W}. We will call individual SCM in $\mathcal{E}$ as an ***environment***.

Assumption: Environments $\mathcal{E}$ are data generating processes of given samples.

Problem Setup

Our task

Construct the ML model working well for all environment in $\mathcal{E}$.

Environments

M

M_{V1}

...

M_{Vk}

Samples

D

D_{V1}

...

D_{Vk}

ML models

$f(\mathbf{V})$ that works well for all environments.

Overview

1. We will provide a formal understanding of shortcut learning through causality.

2. We will propose two approaches for preventing shortcut learning, which both suggest using causal features (a set of features that directly causes true labels).

    i. Approach 1. Avoid causally irrelevant features to the true label as much as possible.

    ii. Approach 2. Find an ML model that works best for all heterogeneous data generating processes.

3. We will provide a principle for identifying causal features by leveraging the causal invariance property.

Overview

1. We will provide a formal understanding of shortcut learning through causality.

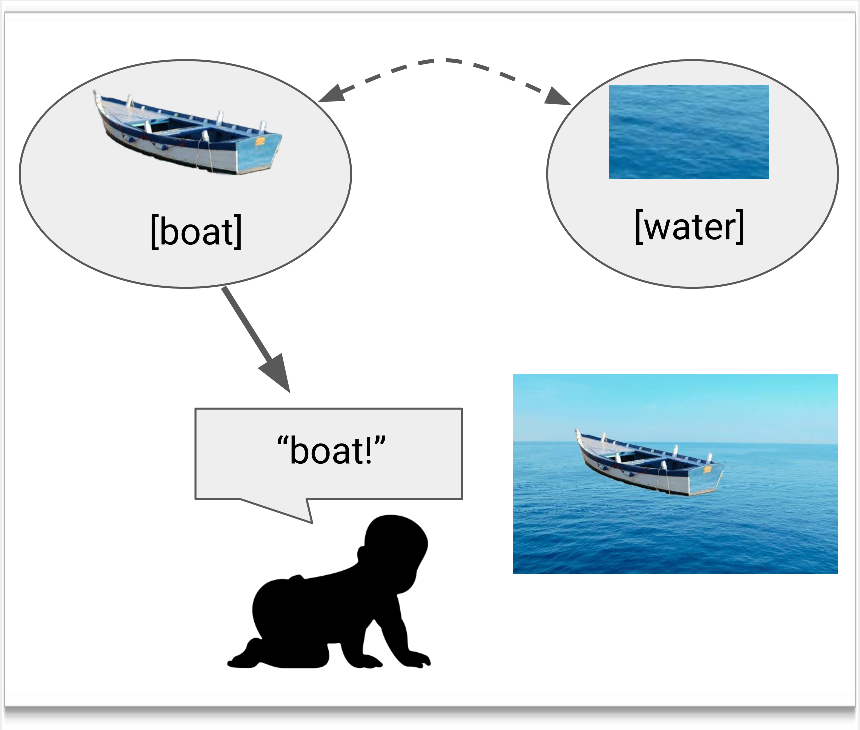
2. We will propose two approaches for preventing shortcut learning, which both suggest using causal features (a set of features that directly causes true labels).

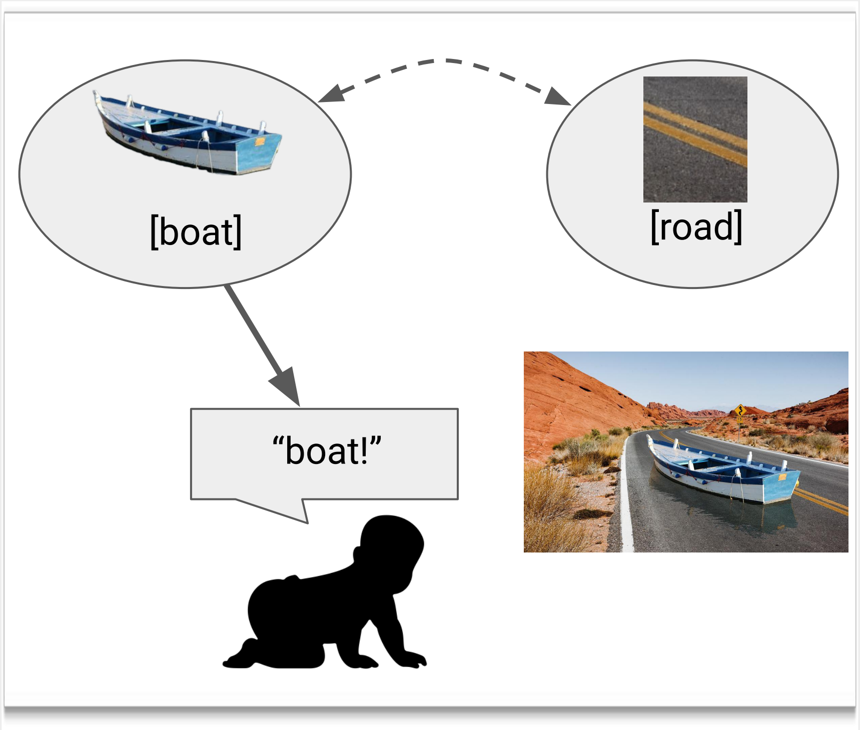
   i. Approach 1. Avoid causally irrelevant features to the true label as much as possible.

   ii. Approach 2. Find an ML model that works best for all heterogeneous data generating processes.

3. We will provide a principle for identifying causal features by leveraging the causal invariance property.

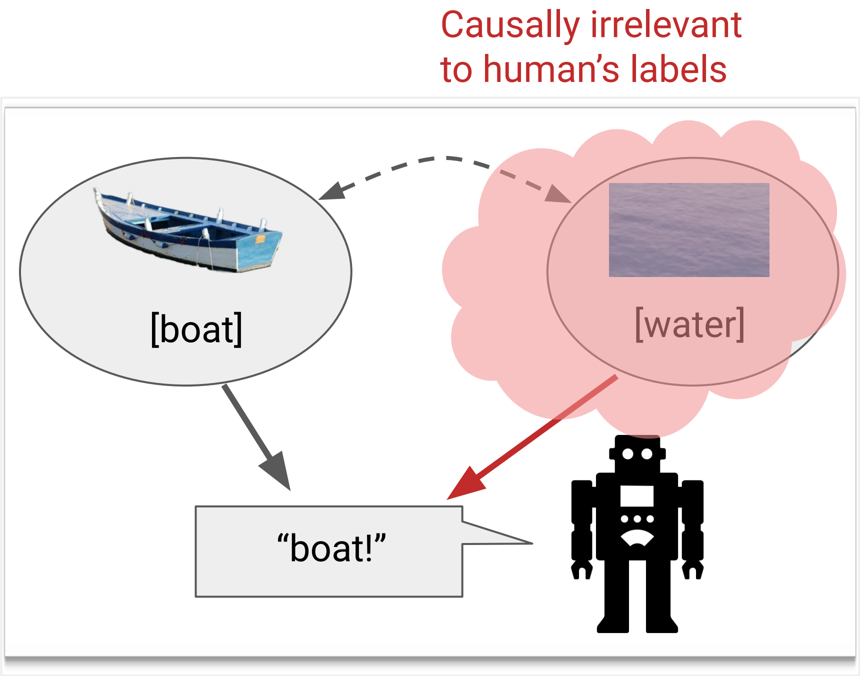
How do Humans Classify (How are True Labels Generated)?

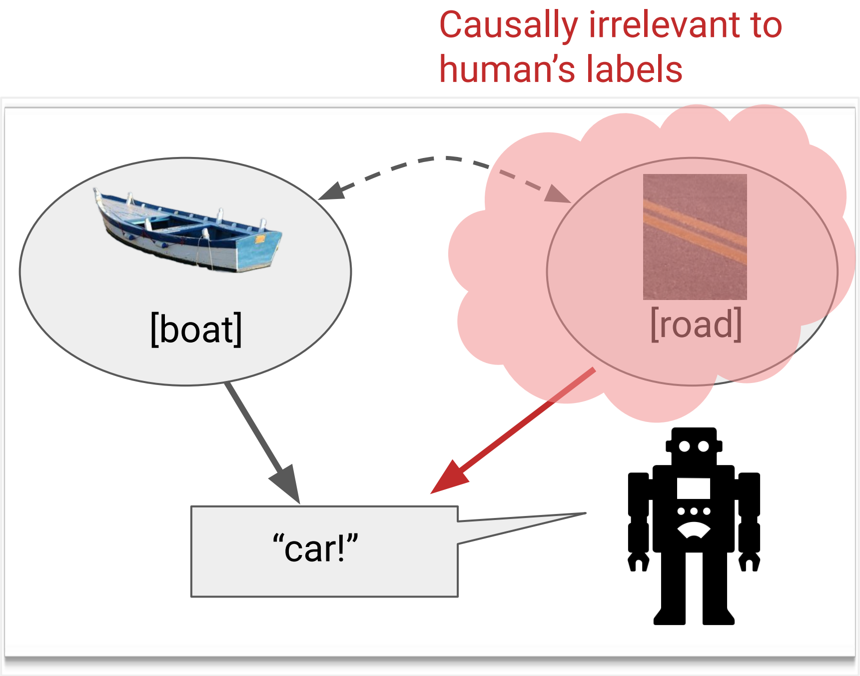




Humans don't use background (*unintended/undesired*) objects ([water], [road]).

ML Models Contaminated by Shortcut Learning





ML models use background features causally irrelevant to the humans' label for their decision rules.

Shortcut Learning and Causally Irrelevant Features

Shortcut Learning: ML models fail due to the usage of *unintended/undesired* features (backgrounds) for their decision rule.

(Rewritten) Shortcut Learning: ML models fail due to the usage of *causally irrelevant features (to the label)* (features that aren’t causing humans’ true label) for their decision rule.

Approach 1. Avoid Causally Irrelevant Features

(Rewritten) Shortcut Learning: ML models fail due usage of *causally irrelevant features* (features that aren't causing humans' true label) for their decision rule.

Approach 1. Avoid Causally Irrelevant Features

Construct the ML models avoiding causally irrelevant features as much as possible.

Definition of Causal Irrelevance - 1

A set of variables $\mathbf{X}$ is said to be *causally irrelevant* to $\mathbf{Y}$ (given other intervention $\mathbf{W}$) if intervening on $\mathbf{X}$ ($\mathrm{do}(\mathbf{X})$) doesn't affect $\mathbf{Y}$, given $\mathrm{do}(\mathbf{W})$.

Causal Irrelevance (Pearl, 2000)

A set of variables $\mathbf{X}$ is said to be *causally irrelevant* to $\mathbf{Y}$ given $\mathbf{W}$ if,

$$P(\mathbf{Y}=\mathbf{y} \mid \mathrm{do}(\mathbf{X}=\mathbf{x}), \mathrm{do}(\mathbf{W}=\mathbf{w})) = P(\mathbf{Y}=\mathbf{y} \mid \mathrm{do}(\mathbf{X}=\mathbf{x}`), \mathrm{do}(\mathbf{W}=\mathbf{w}))$$

for any realizations $(\mathbf{y}, \mathbf{x}, \mathbf{w}, \mathbf{x}')$ s.t. $\mathbf{x} \neq \mathbf{x}'$.

Definition of Causal Irrelevance - 2

A set of variables $\mathbf{X}$ is said to be *causally irrelevant* to $\mathbf{Y}$ if intervening on $\mathbf{X}$ ($\mathrm{do}(\mathbf{X})$) doesn't affect $\mathbf{Y}$ given the intervention on the rest of variables.

Causal Irrelevance Set

A set of variables X is said to be ***causally irrelevant set*** to Y if,

$$P(\mathbf{Y}=\mathbf{y} \mid \mathrm{do}(\mathbf{X}=\mathbf{x}), \mathrm{do}(\mathbf{V}\backslash\mathbf{X}=\mathbf{v}\backslash\mathbf{x})) = P(\mathbf{Y}=\mathbf{y} \mid \mathrm{do}(\mathbf{X}=\mathbf{x}`), \mathrm{do}(\mathbf{V}\backslash\mathbf{X}=\mathbf{v}\backslash\mathbf{x}))$$

for any realizations $(\mathbf{y}, \mathbf{x}, \mathbf{v}\backslash\mathbf{x}, \mathbf{x}')$ s.t. $\mathbf{x} \neq \mathbf{x}'$.

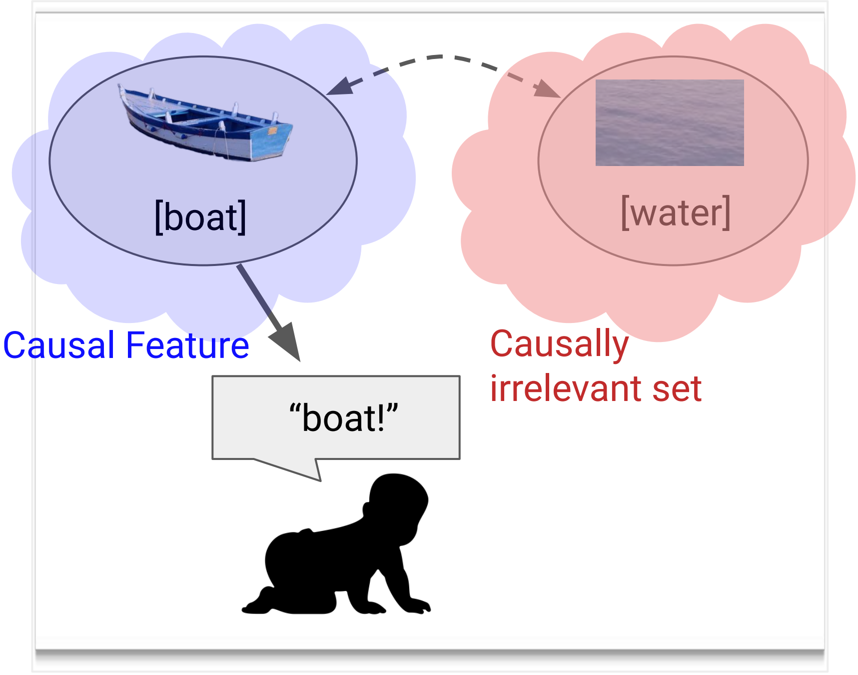
Graphical Interpretation of Causal Irrelevant Set

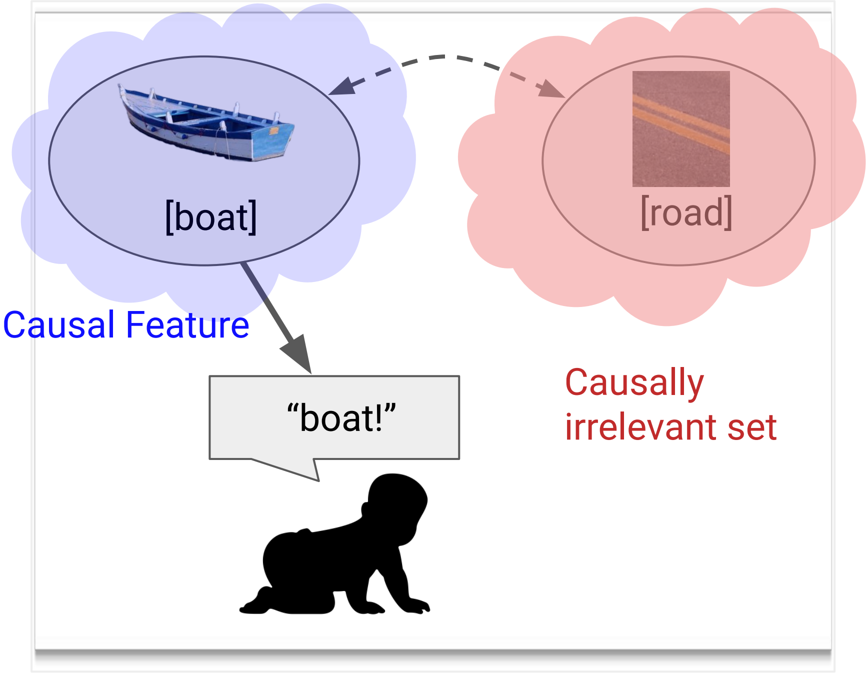
Causal Features (of the true label Y): $\mathrm{PA}_{\mathrm{Y}} \subseteq \mathbf{V}$ is called ***causal features*** of Y if it is a parental set of Y (direct cause) in the graph induced by the SCM.

Graphical Interpretation of Causal Irrelevance Set

$\mathbf{X}$ is said to be ***causally irrelevant set*** to $\mathbf{Y}$ if $\mathbf{X}$ doesn't contain any causal features.

Humans don't Use Causally Irrelevant Features





Formal Interpretation of Approach 1

(Informal) Approach 1

Avoid causally irrelevant features *as much as possible.*

(Formal) Approach 1 [Theorem 1]

- The *largest causal irrelevant set* is $\mathbf{V} \setminus \mathrm{PA}_Y$, all variables except causal features PA_Y.
- Therefore, to prevent the shortcut learning, construct the models using causal features PA_Y.

Overview

1. We will provide a formal understanding of shortcut learning through causality.

2. We will propose two approaches for preventing shortcut learning, which both suggest using causal features that directly cause true labels.

   i. Approach 1. Avoid causally irrelevant features to the true label as much as possible.

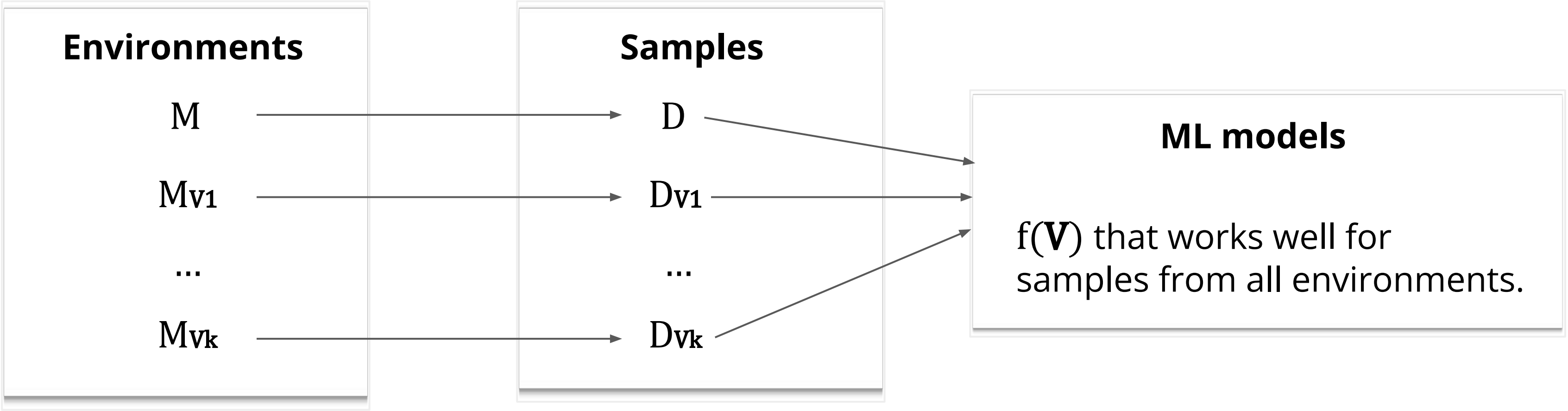
   ii. Approach 2. Find an ML model that works best for all heterogeneous data generating processes.

3. We will provide a principle for identifying causal features by leveraging the causal invariance property.

(Recap) Problem Setup - Task

Our task

Construct the ML model working well for all environment in $\mathcal{E}$.



Approach 2. Model working well for all environments

Approach 2

Find the performant ML model that ***works well*** for all environments, even ***in the worst environment*** (e.g., [boat in the road]).

Formalization of Approach 2

Model that working best in the worst case.

$$\operatorname{argmin}_{f \in \mathcal{F}} \max_{P \in \mathcal{P}(\mathcal{E})} E_P[L(f(\mathbf{V}), Y)]$$

- $L(f(\mathbf{V}), Y)$ is a prediction error of the model $f(V)$ to Y
- $E_P[L(f(\mathbf{V}), Y)]$ is an expected error w.r.t. a distribution P.
- $\mathcal{P}(\mathcal{E})$ is a set of distributions induced by submodels in an environment $\mathcal{E}$.
- $\mathcal{F}$ is a class of the ML model f.

The task is to **minimize** the expected error of the ML model even in the **worst** environment that maximizes the expected error.

Model with Causal Features is Performant.

Model with causal features works well in the worst case.

$$\text{argmin}_f \ \max_{P \in \mathcal{P}(\mathcal{E})} E_P[L(f(\mathbf{V}), Y)]$$

- If $L(f(\mathbf{V}), Y) = E[(f(\mathbf{V})-Y)^2]$ (Regression), the solution is $\mathbf{E[Y \mid PA_Y]}$ (Rojas-Callura et al. 2018)
- If $L(f(\mathbf{V}), Y) = E[\mathbb{1}(f(\mathbf{V}) \neq Y)]$ (Classification), the solution is $\mathbf{argmax}_y \, \mathbf{P(Y=y \mid PA_Y)}$

Takeaway: A ML model working well even in the worst environment can be found by constructing the model for the ***relation b/w the true label and its causal features.***

Interpretation of Approach 2

Approach 2

Find the performant ML model that ***works well*** for all environments, even ***in the worst environment*** (e.g., [boat in the road])

Implication of Approach 2 [Theorems (2,3)]

To prevent the shortcut learning, construct the models using causal features!

Approaches (1,2) imply ML models with Causal Features

Approach 1

Avoid causally irrelevant features as much as possible.

Approach 2

Find the model working well in the worst environment.

Causal Features

Construct the model using causal features PA_Y

Overview

1. We will provide a formal understanding of shortcut learning through causality.

2. We will propose two approaches for preventing shortcut learning, which both suggest using causal features (a set of features that directly causes true labels).

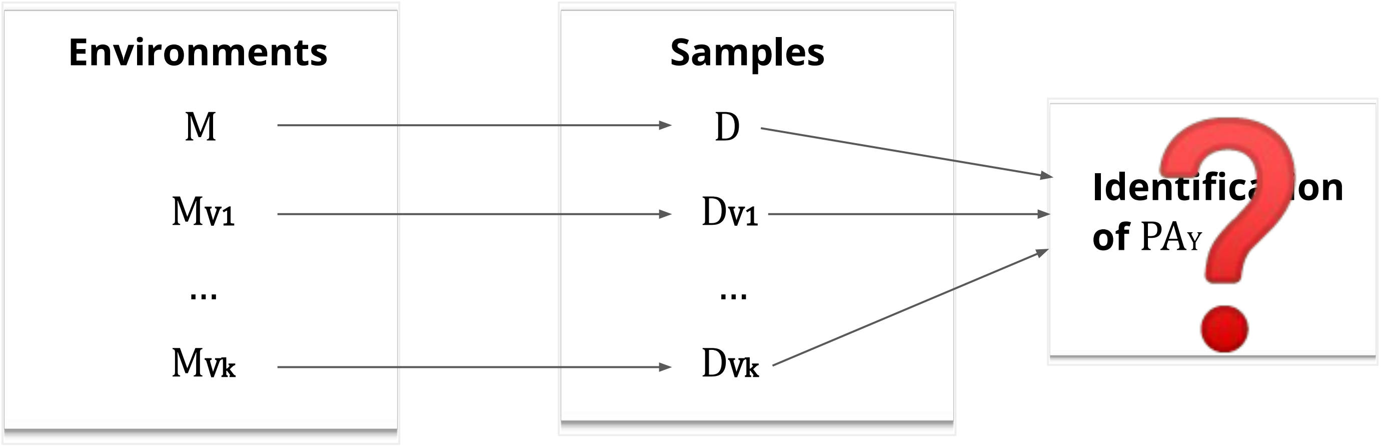
    i. Approach 1. Avoid causally irrelevant features to the true label as much as possible.

    ii. Approach 2. Find an ML model that works best for all heterogeneous data generating processes.

3. We will provide a principle for identifying causal features by leveraging the causal invariance property.

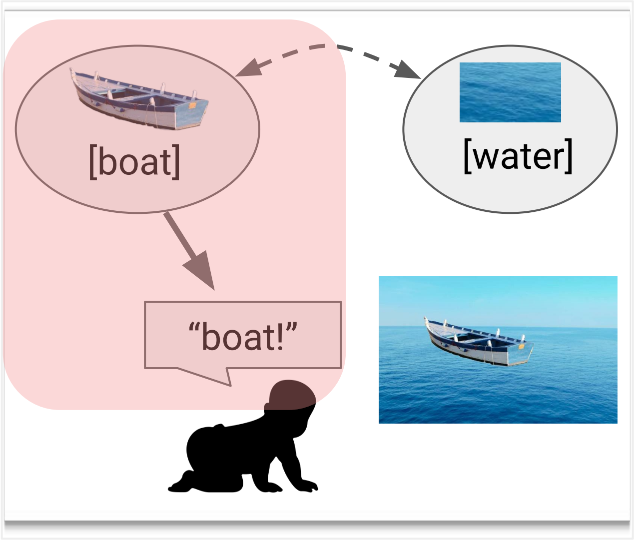
Identification of Causal Features is Difficult in Practice

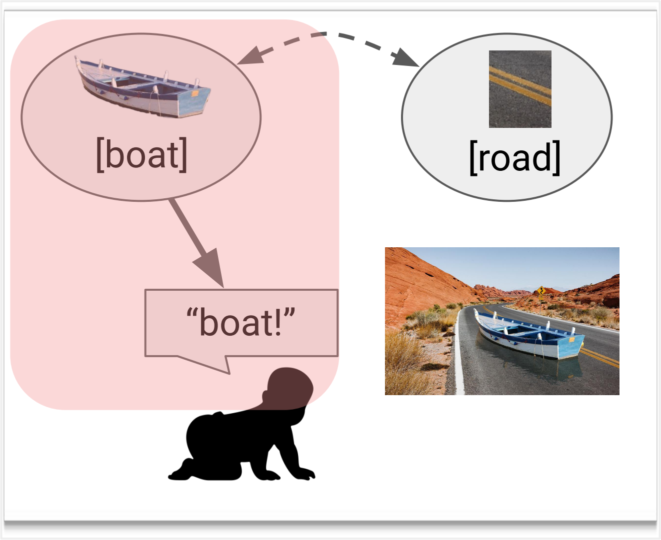
- We discussed that ML models should be constructed using causal features PA_Y.
- When causal graphs are unknown, PA_Y (parental nodes of Y), is hardly identifiable.



How can we identify causal features?

Property of Causal Feature: Causal Invariance





The relation (Y, PA_Y) (the true label and its causal feature) is preserved on different environments (e.g., [boat in the water], [boat in road]) generating perturbed examples.

Causal Invariance: Property of Causal Feature

(Informal) Causal Invariance: The probabilistic relation b/w the label and causal features (Y, PA_Y) is invariant over all environments.

Causal Invariance

Suppose Y is not connected by bidirected paths (Equivalently, U_Y, the hidden/noise variable affecting Y, is independent of all other variables). Then, for any environments M_1, M_2,

$$P_{\mathcal{M}_1}(Y \mid PA_Y) = P_{\mathcal{M}_2}(Y \mid PA_Y)$$

Test Function: Deriving Causal Features from Invariance

Observation: Finding the set $\mathbf{X}$ invariant to Y is easier than finding PA_Y.

- **Example 1**: $\mathbf{X}$ invariant to Y if $P_{\mathcal{M}1}(Y \mid \mathbf{X}) = P_{\mathcal{M}2}(Y \mid \mathbf{X})$ for all environments $\mathcal{M}1, \mathcal{M}2$ in $\mathcal{E}$. [Peters et al., 2016]

- **Example 2**: $\mathbf{X}$ invariant to Y if $(Y, \mathbf{V} \backslash \mathbf{X})$ are independent conditioned on $\mathbf{X}$ (i.e., $P_{\mathcal{M}}(Y \mid \mathbf{V}) = P_{\mathcal{M}}(Y \mid \mathbf{X})$) for all environments $\mathcal{M}$ in $\mathcal{E}$. [Heinze-Deml et al., 2018]

Test Function

$T_{\mathcal{E}}(\mathbf{X},Y)=1$ if the relation b/w $(\mathbf{X},Y)$ is invariant for all given environments in $\mathcal{E}$.

Identification of Causal Features

Remark: The causal feature $\mathbf{X} = \mathrm{PA}_Y$ is the ***smallest set*** satisfying $T_{\mathcal{E}}(\mathbf{X}, Y) = 1$ (i.e., the causal feature is the smallest invariance set).

Identifying Causal Features in high probability [Theorem 4]

Suppose $T_{\mathcal{E}}(\mathbf{X}, Y)$ can capture the invariant set in high probability. Then, ***the smallest set passing the test***; i.e., $\bigcap_{\mathbf{X} \subseteq \mathbf{V}} \{\mathbf{X} \text{ s.t. } T_{\mathcal{E}}(\mathbf{X}, Y) = 1\}$, ***is the causal feature*** in high probability!

Take-Home Message

Approach 1

Avoid causally irrelevant features as much as possible.

Approach 2

Find the model working well in the worst environment.

Causal Features

Construct the model using causal features PA_Y

Invariant Features

The smallest invariant set is the causal feature.

Take-home message: (1) Take the smallest invariant set for all environments, and (2) Build the model based on this set b/c they are causal features (in high prob.)

Summary of Part 2

- We formalize the problem of learning the ML model robust to the shortcut learning w.r.t. Structural Causal Models.

- We proposed two approaches – (1) Avoid causally irrelevant features, and (2) Find the most performant models in the worst environment. These two approaches lead to the same conclusion – Construct the model using causal features.

- Identifying causal features is hard when the graph is absent. To circumvent this challenge, we propose to use the smallest invariant feature since it captures the causal feature.

Q&A for Part 2