Causal Effect Inference for Structured Treatments





Jean Kaddour Yuchen Zhu Qi Liu Matt Kusner Ricardo Silva

Roadmap

- Motivation
- Generalized Robinson Decomposition
- Quasi-Oracle Convergence Rate
- Structured Intervention Networks
- Experiments
- Summary

Roadmap

- **Motivation**
- Generalized Robinson Decomposition
- Quasi-Oracle Convergence Rate
- Structured Intervention Networks
- Experiments
- Summary

Causal Effects of Structured Treatments

🥳

Causal Effects of Structured Treatments

🥳

↓

do(🍸)

Causal Effects of Structured Treatments

🥳

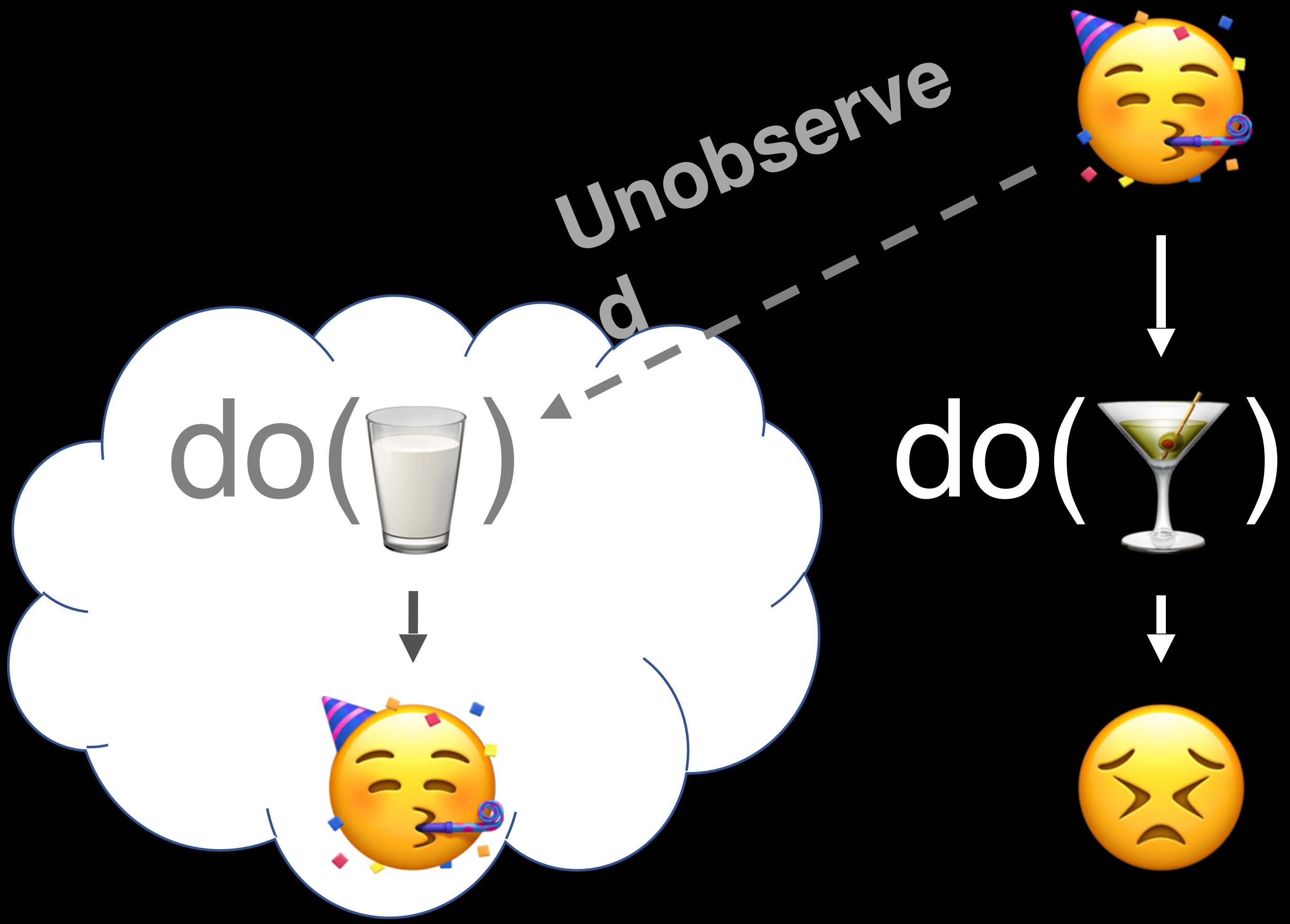
↓

do(🍸)

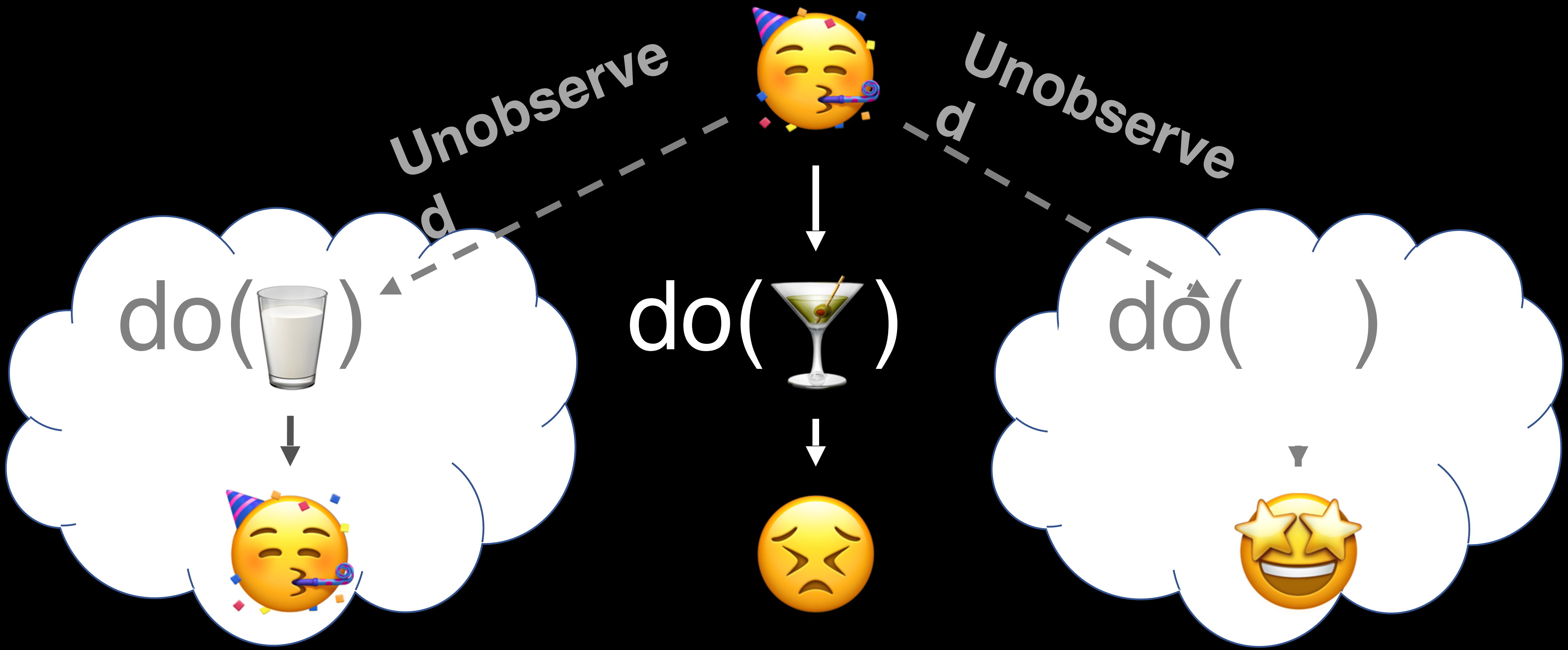
↓

😣

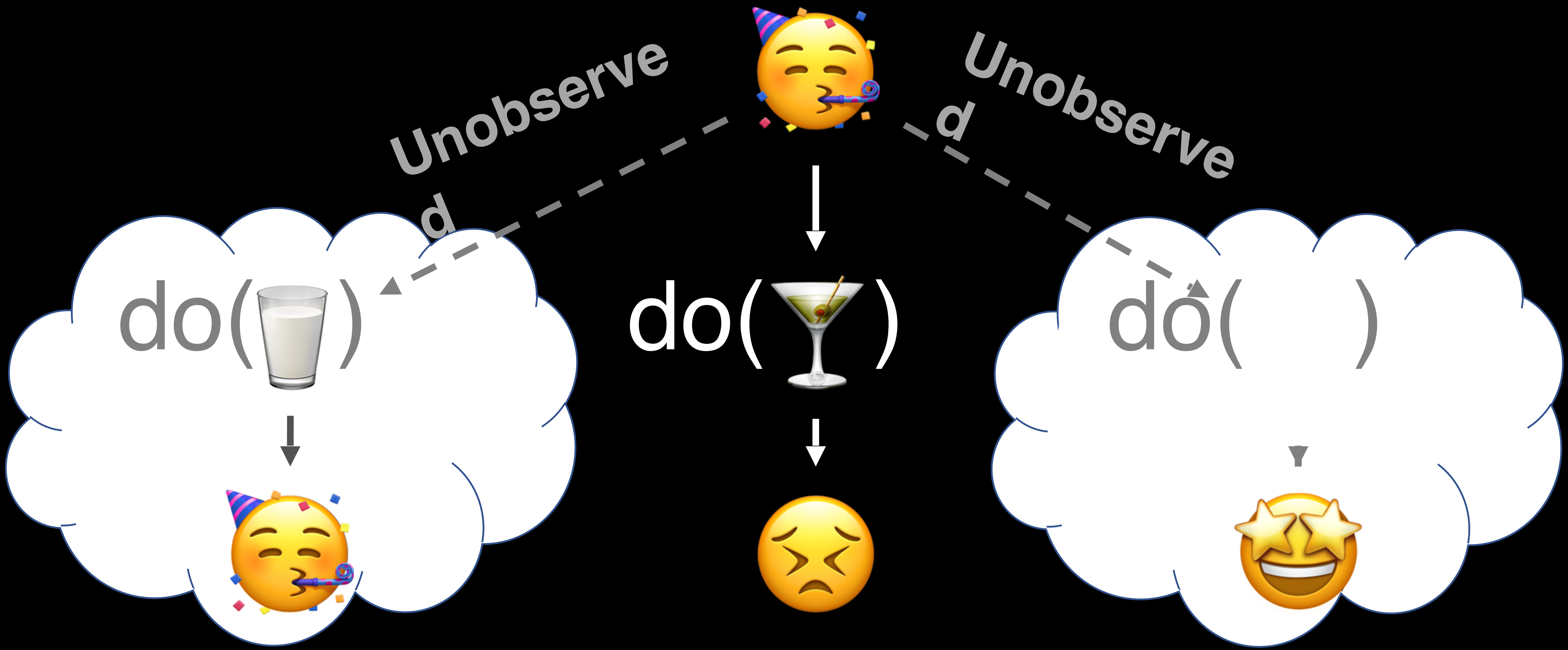
Causal Effects of Structured Treatments



Causal Effects of Structured Treatments

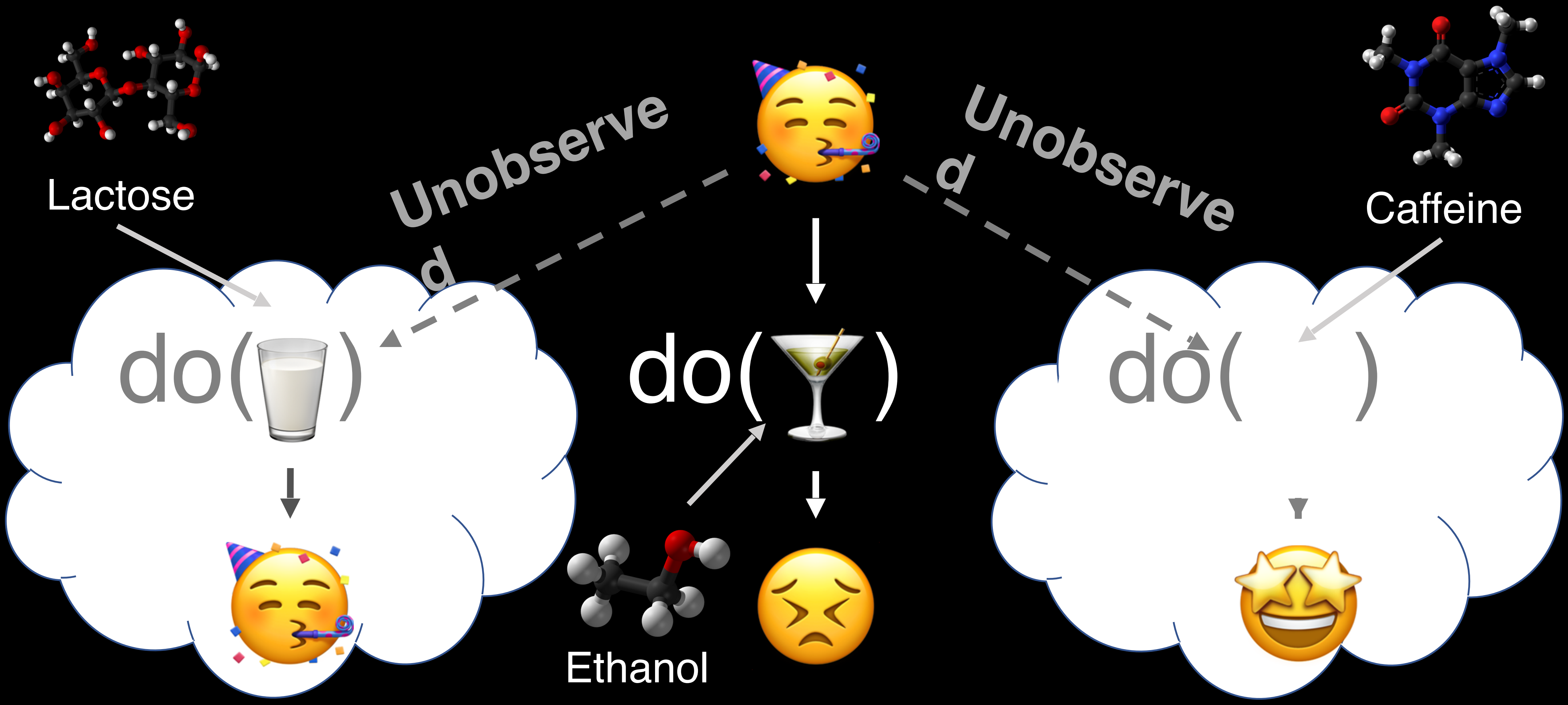


Causal Effects of Structured Treatments



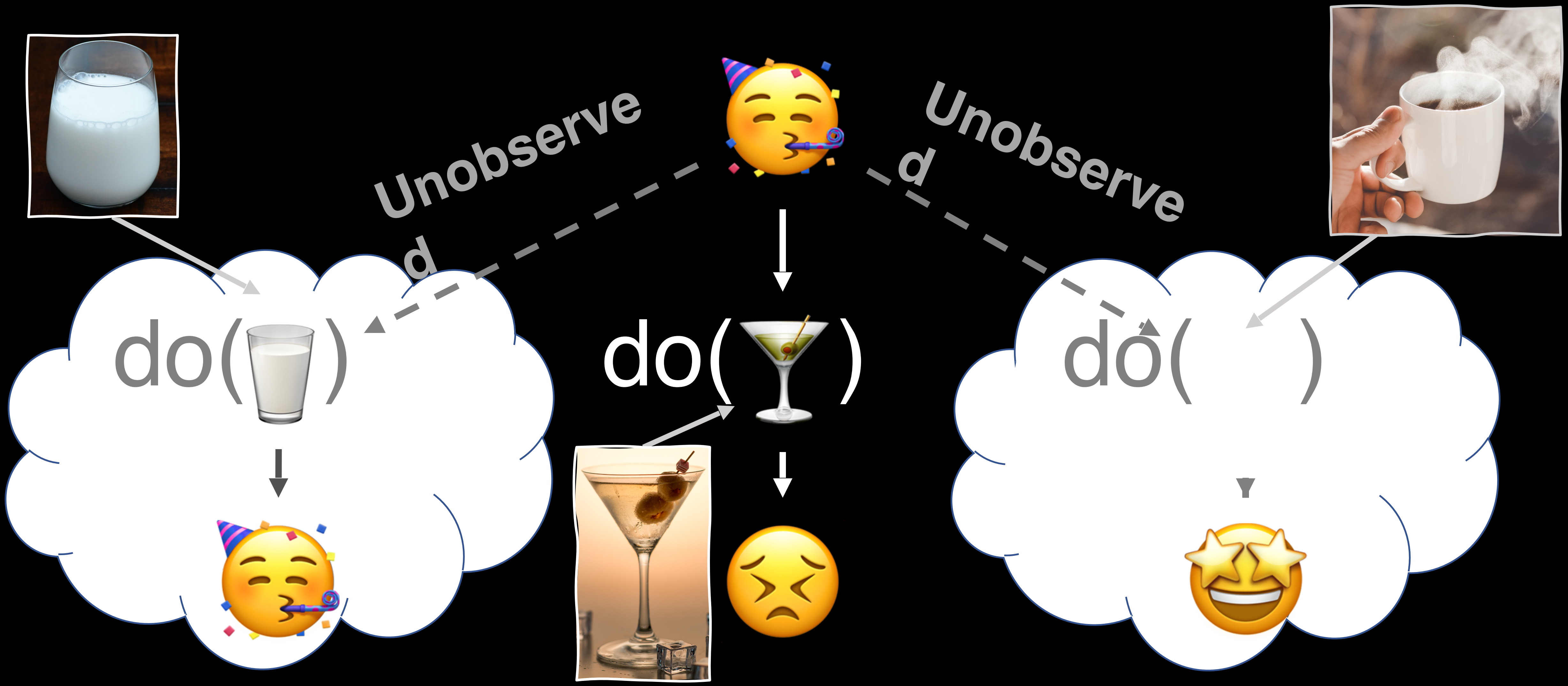
CATE: $\tau(\quad,\quad,\quad) = \mathbb{E}\left[Y \mid \quad, \mathrm{do}(\quad)\right] - \mathbb{E}\left[Y \mid \quad, \mathrm{do}(\quad)\right]$

Causal Effects of Structured Treatments



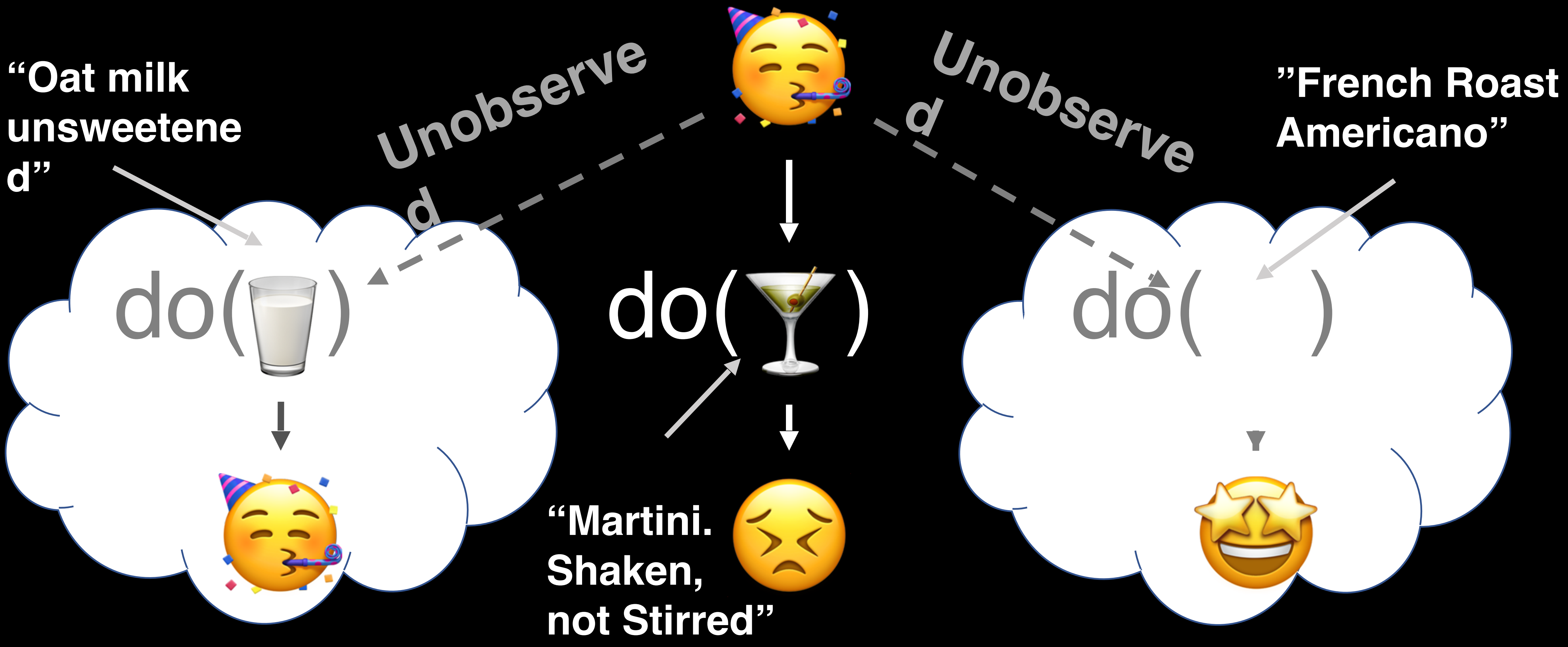
CATE: $\tau(\quad,\quad,\quad) = \mathbb{E}[Y \mid \quad, \mathrm{do}(\quad)] - \mathbb{E}[Y \mid \quad, \mathrm{do}(\quad)]$

Causal Effects of Structured Treatments



CATE: $\tau(\quad,\quad,\quad) = \mathbb{E}\left[Y \mid \quad, \mathrm{do}(\quad)\right] - \mathbb{E}\left[Y \mid \quad, \mathrm{do}(\quad)\right]$

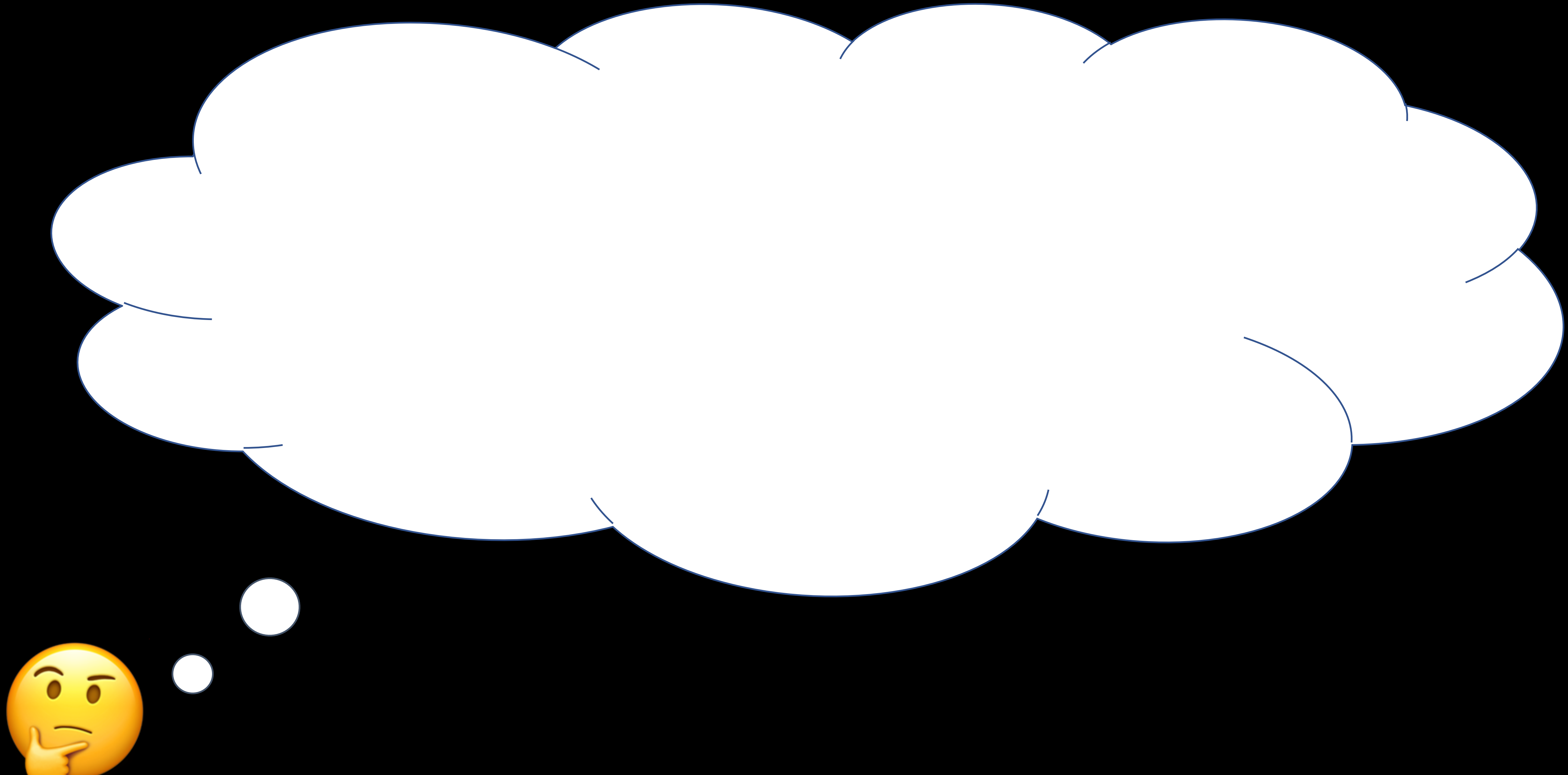
Causal Effects of Structured Treatments



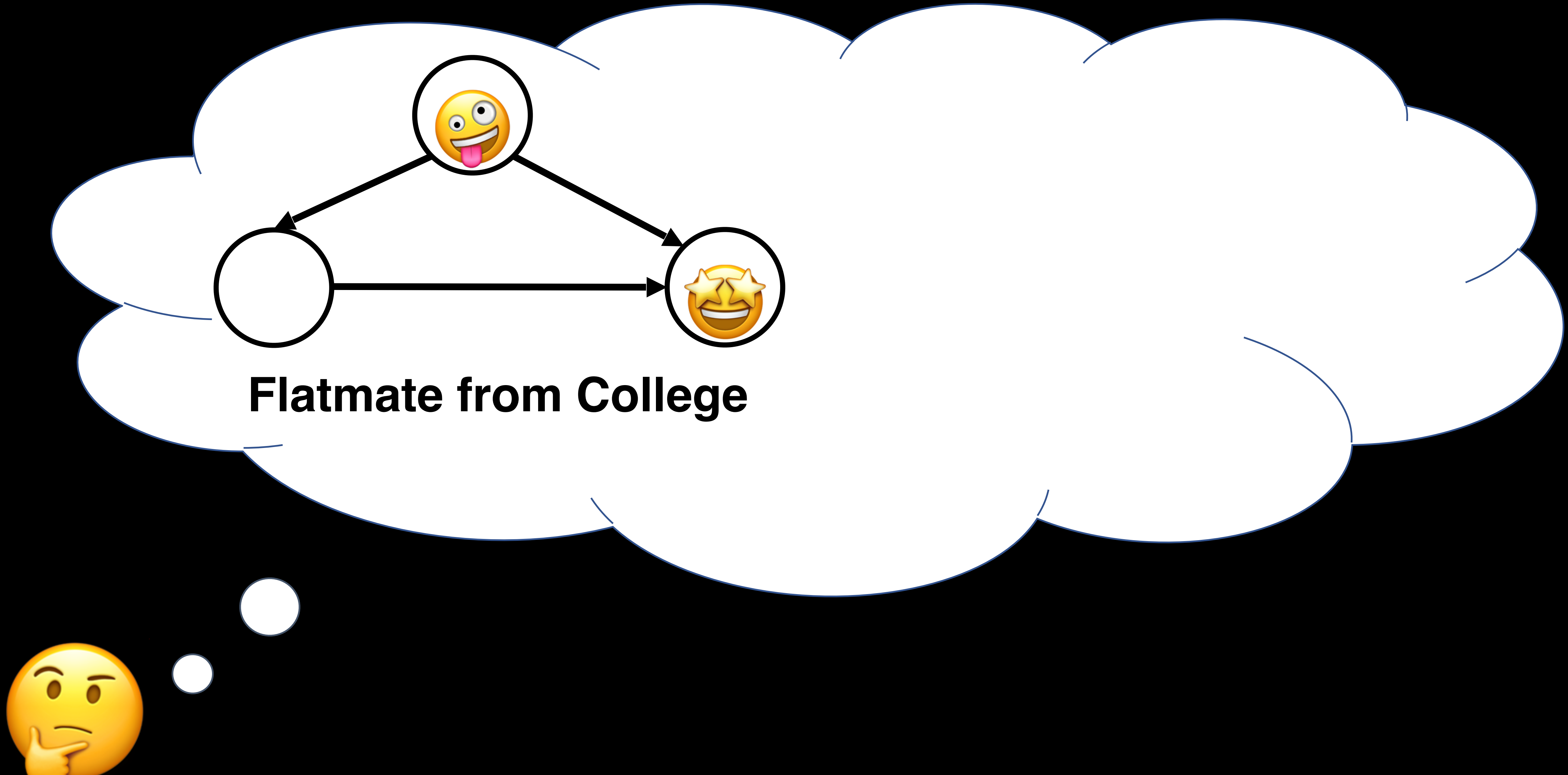
CATE: $\tau(\quad,\quad,\quad) = \mathbb{E}\left[Y \mid \quad, \mathrm{do}(\quad)\right] - \mathbb{E}\left[Y \mid \quad, \mathrm{do}(\quad)\right]$

You only live once!

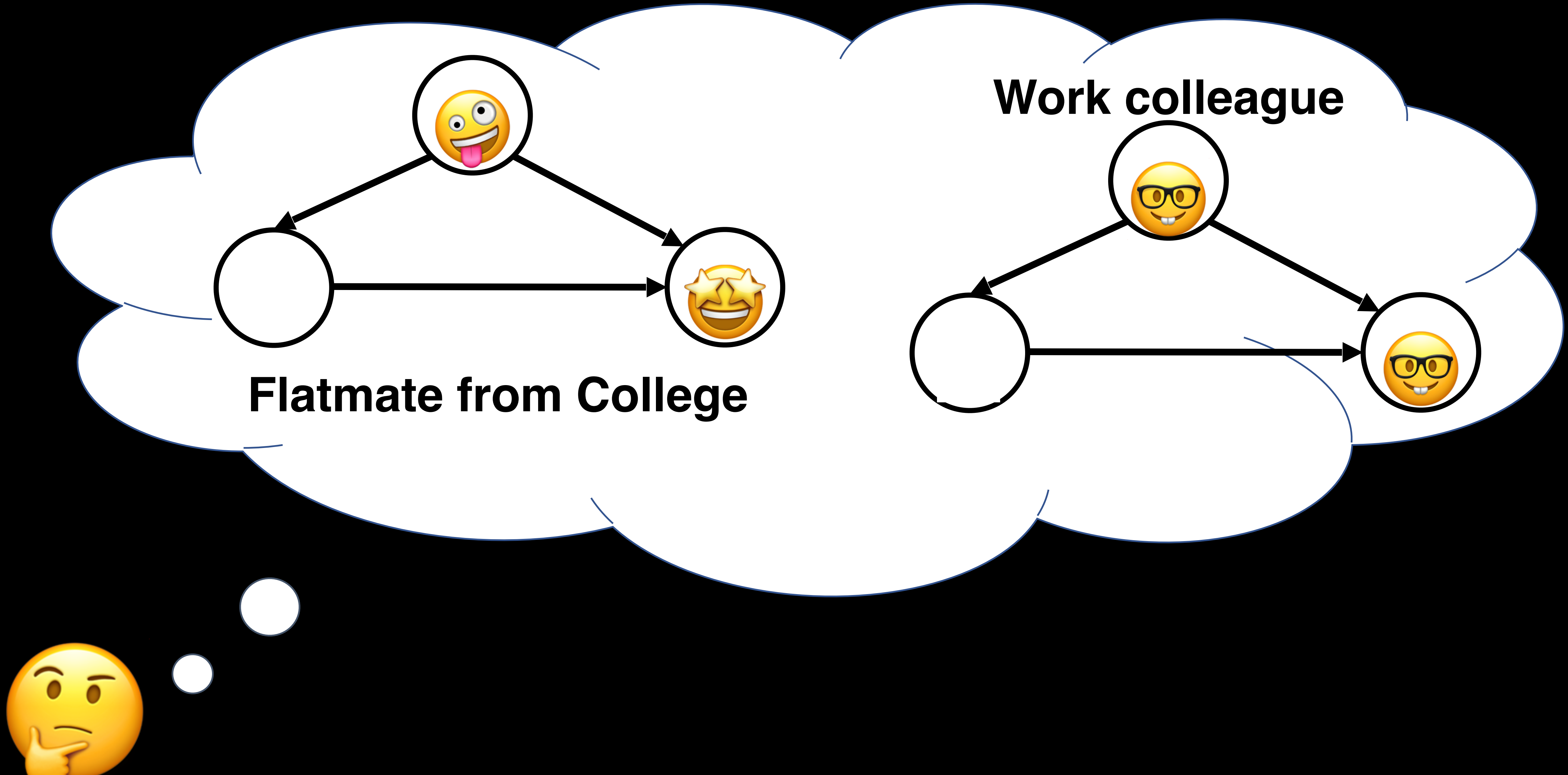
You only live once!



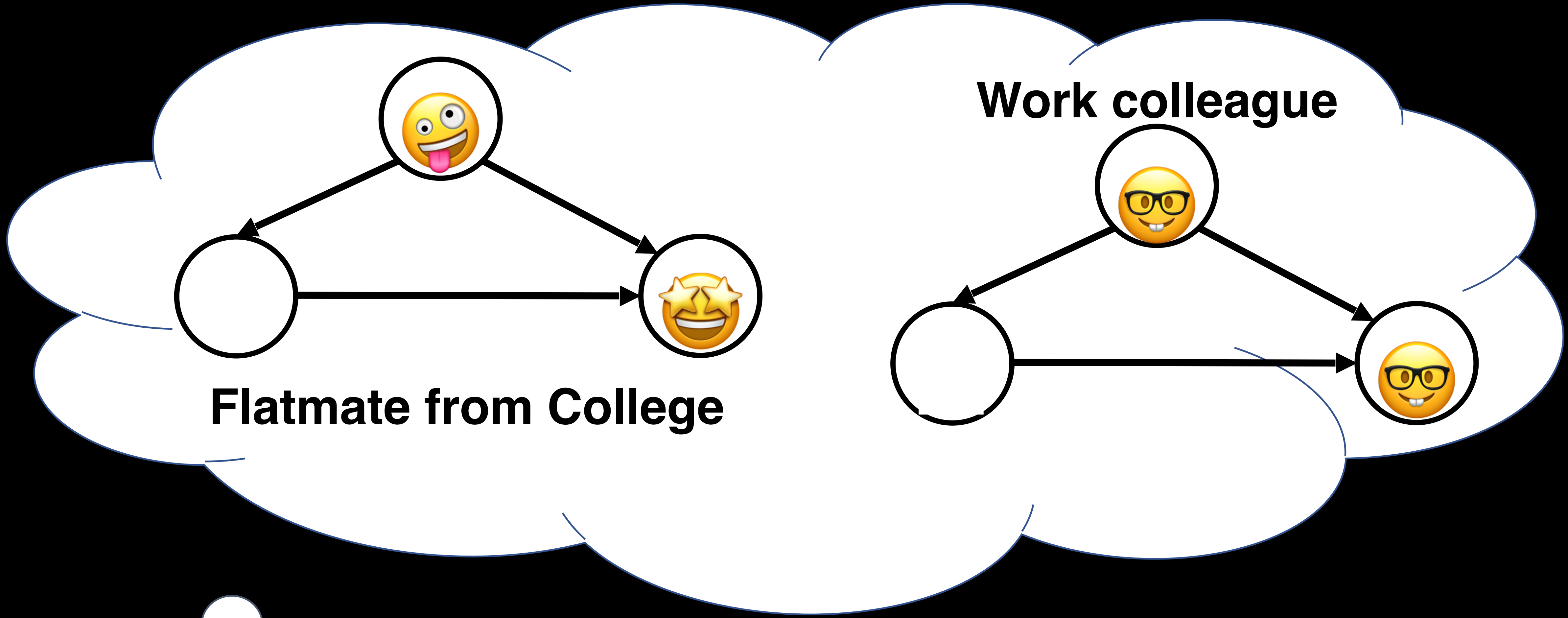
You only live once!



You only live once!

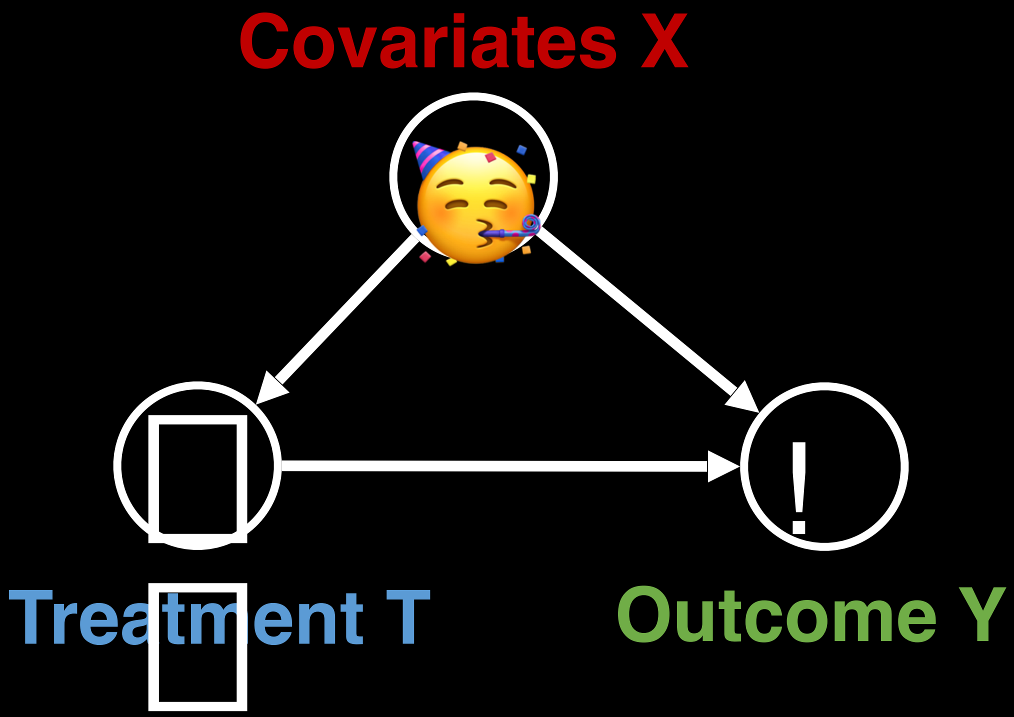


You only live once!

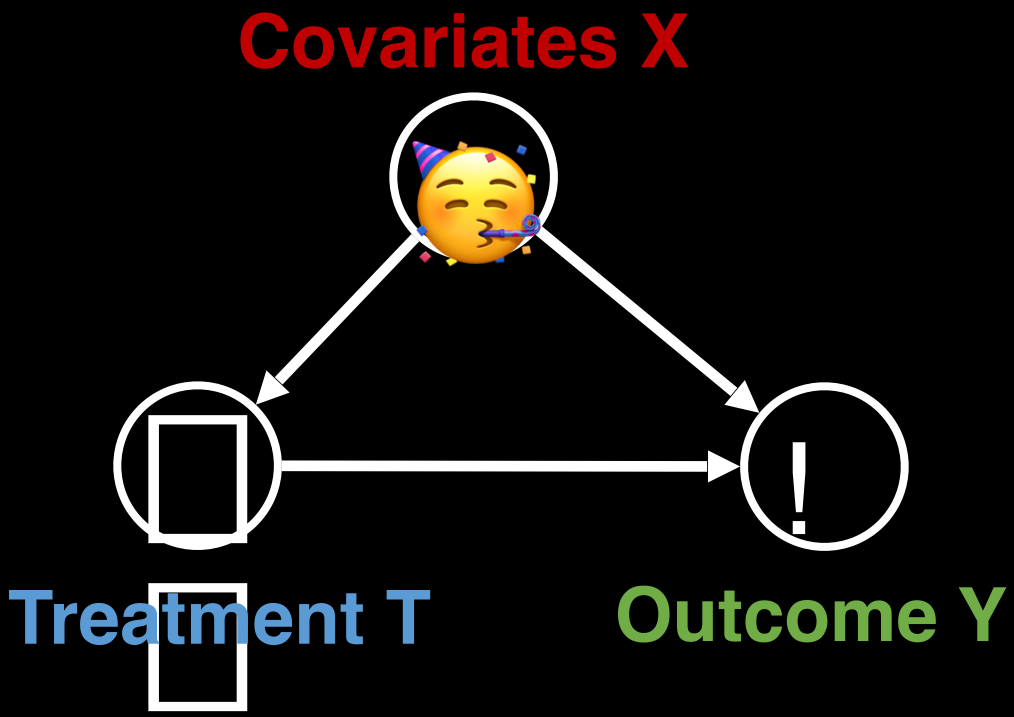


Can we learn from others?

Goal: Estimate CATE from observational data



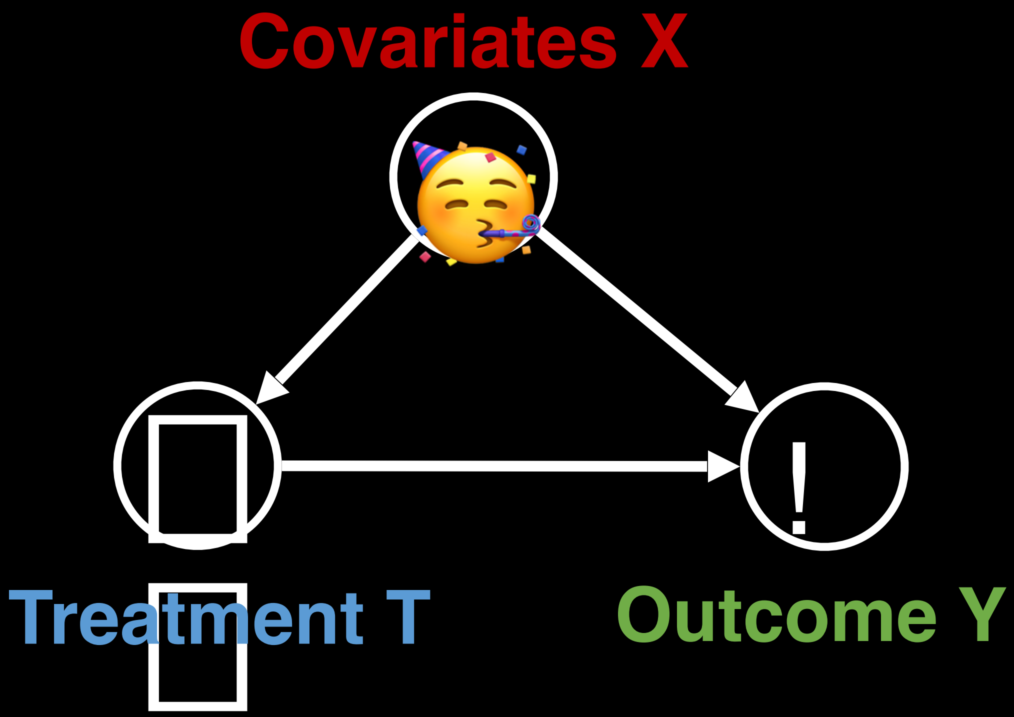
Goal: Estimate CATE from observational data



+

X	T	Y
🥳	🍸	😣
🤪	#	🤩
🤓	🥛	🤓

Goal: Estimate CATE from observational data

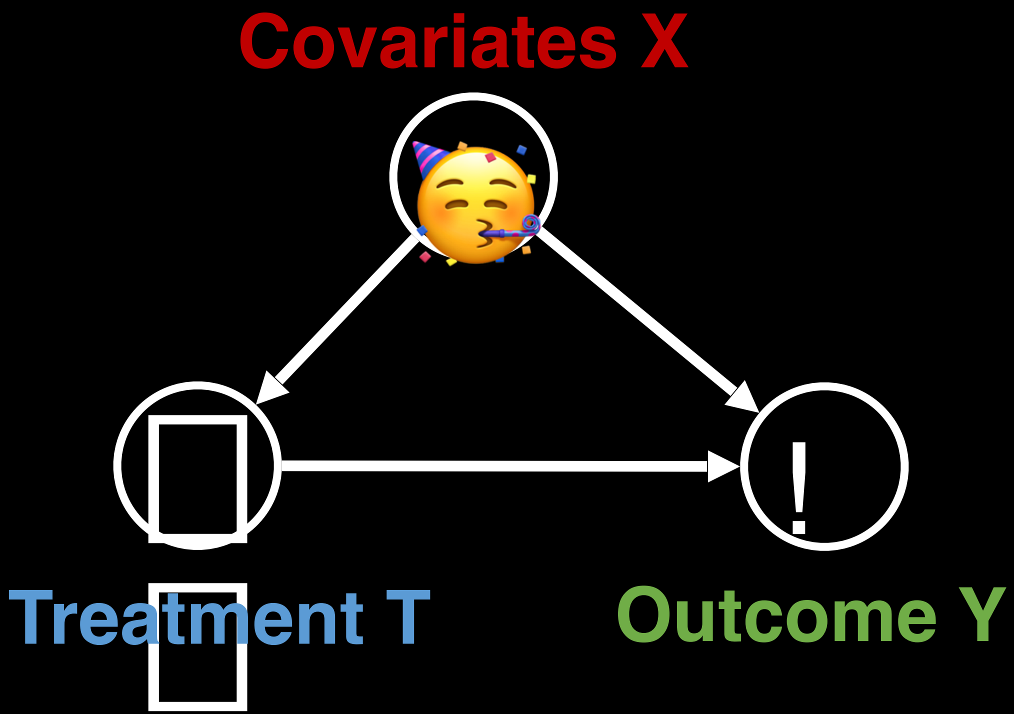


+

X	T	Y
🥳	🍸	😣
🤪	#	🤩
🤓	🥛	🤓

+

Goal: Estimate CATE from observational data



+

X	T	Y
🥳	🍸	😣
🤪	#	🤩
🤓	🥛	🤓

+

=

# #

Wait, can't I simply learn a model

$$\widehat{f} \approx \mathbb{E}[Y \mid \mathbf{X}, \mathbf{T}]$$

Wait, can't I simply learn a model

$$\widehat{f} \approx \mathbb{E}[Y \mid \mathbf{X}, \mathbf{T}]$$

and then subtract predictions

$$\widehat{\tau}\,(\mathbf{t}', \mathbf{t}, \mathbf{x}) = \widehat{f}\,(\mathbf{x}, \mathbf{t}') - \widehat{f}(\mathbf{x}, \mathbf{t})?$$

Technically, yes! But...

Technically, yes! But...

Technically, yes! But...

“When solving a problem of interest,

do not solve a more general problem

as an intermediate step.

Technically, yes! But...

“When solving a problem of interest,

do not solve a more general problem

as an intermediate step.

Try to get the answer that you really

need but not a more general one.”

Vladimir Vapnik, 2006.

Why targeting the effect directly helps

Why targeting the effect directly helps

- **Reason 1**: the effect often exhibits a simpler structure

Why targeting the effect directly helps

- **Reason 1**: the effect often exhibits a simpler structure
  - For example, $\widehat{f}(\mathbf{X}, \mathbf{T})$ can be very non-smooth for rarely treated **X**

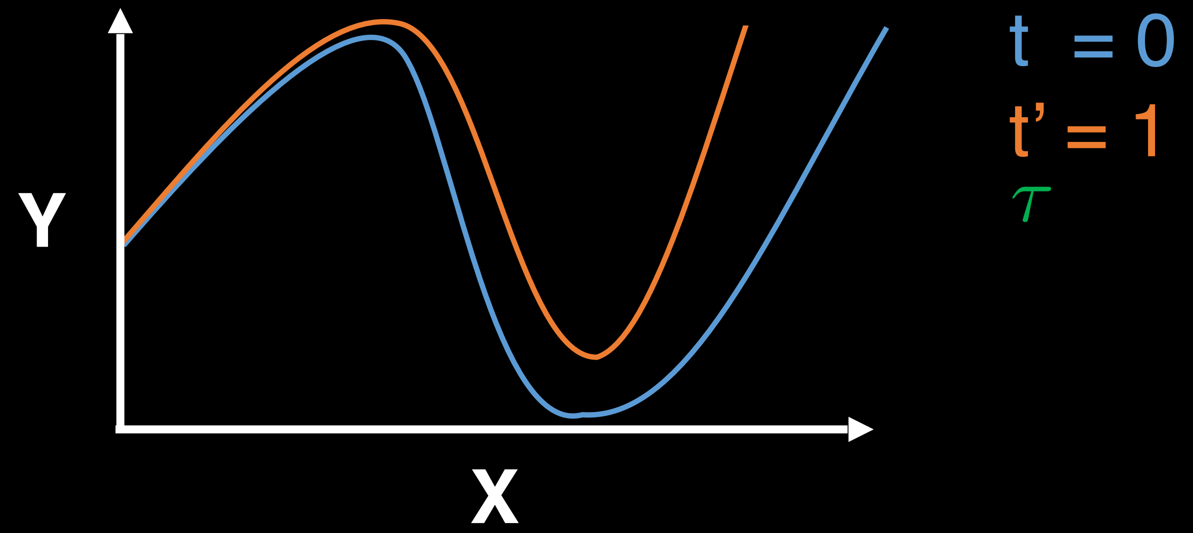


Figure adopted from "On Inductive Biases for Heterogeneous Treatment Effect Estimation", Curth & van der Schaar, NeurIPS 2021.

Why targeting the effect directly helps

- **Reason 1**: the effect often exhibits a simpler structure
  - For example, $\widehat{f}(\mathbf{X}, \mathbf{T})$ can be very non-smooth for rarely treated **X**
  - However, $f(\mathbf{X}, \mathbf{t}') - f(\mathbf{X}, \mathbf{t})$ might be (almost) linear across **X**

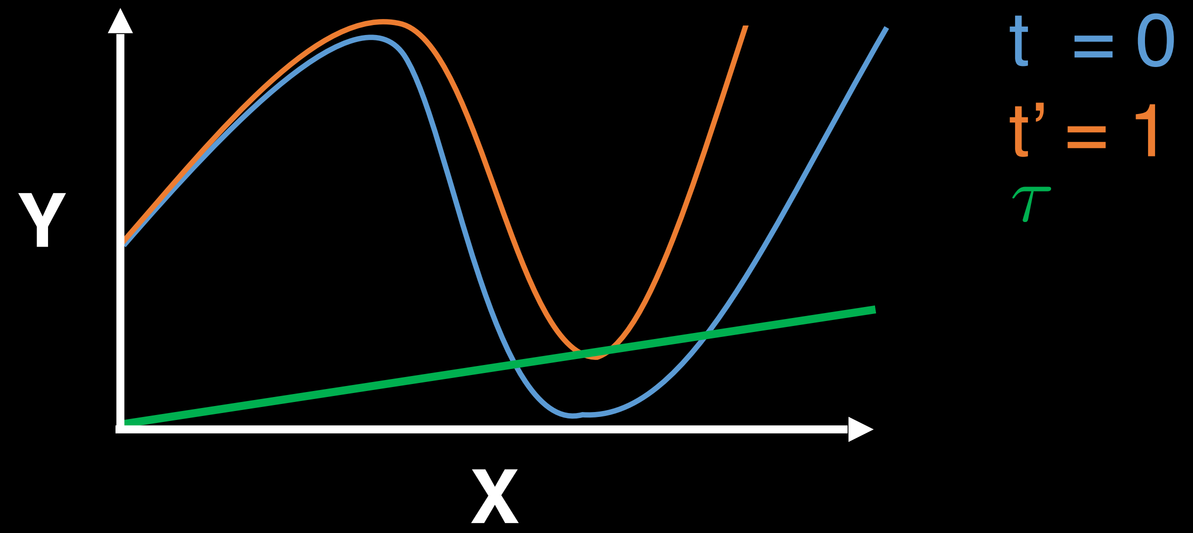


Figure adopted from "On Inductive Biases for Heterogeneous Treatment Effect Estimation", Curth & van der Schaar, NeurIPS 2021.

Why targeting the effect directly helps

- **Reason 1**: the effect often exhibits a simpler structure
  - For example, $\widehat{f}(\mathbf{X}, \mathbf{T})$ can be very non-smooth for rarely treated **X**
  - However, $f(\mathbf{X}, \mathbf{t}') - f(\mathbf{X}, \mathbf{t})$ might be (almost) linear across **X**

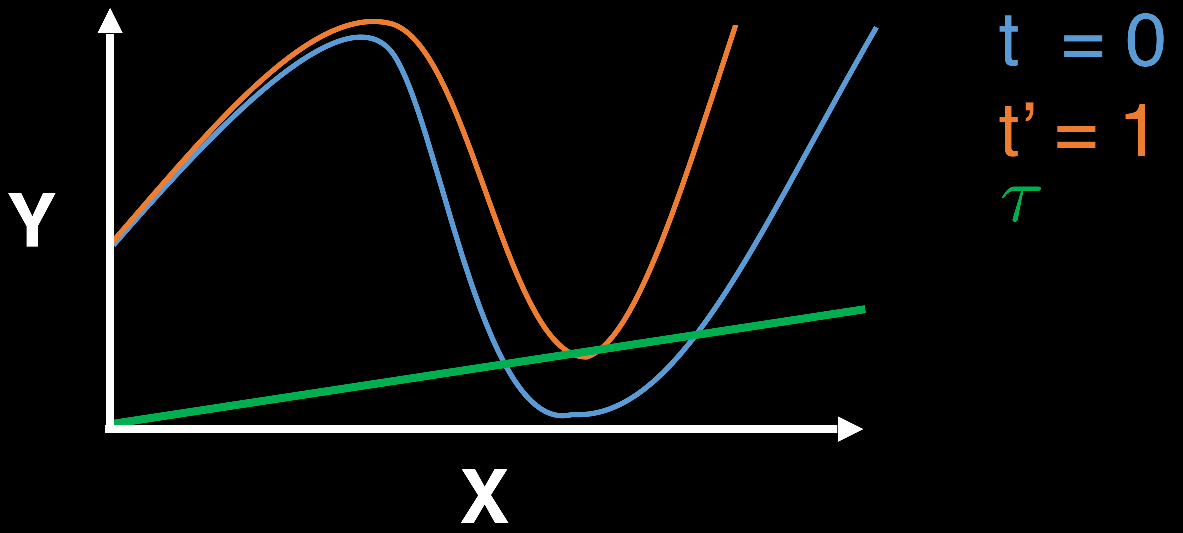


Figure adopted from "On Inductive Biases for Heterogeneous Treatment Effect Estimation", Curth & van der Schaar, NeurIPS 2021.

Why targeting the effect directly helps

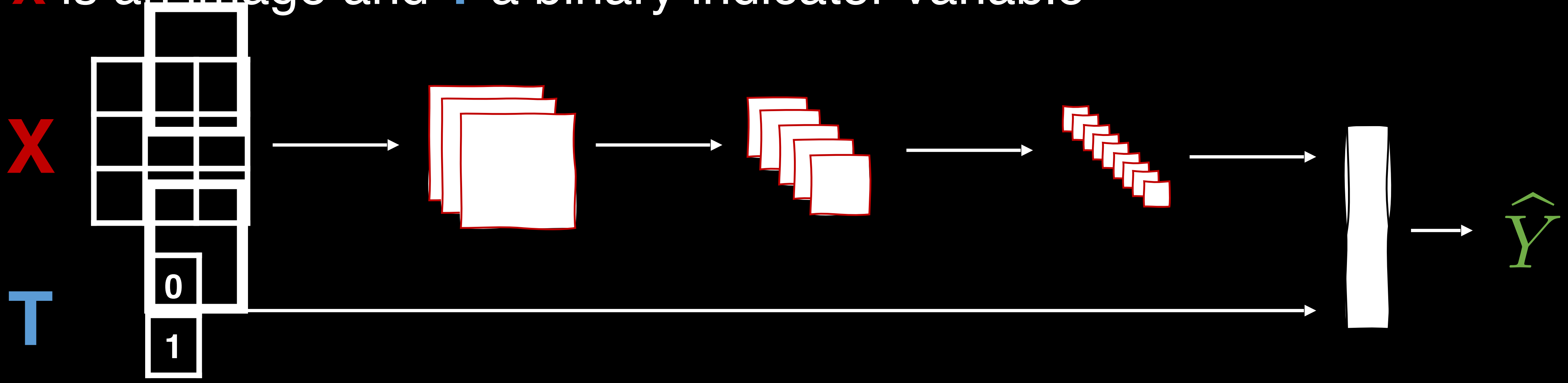
- **Reason 2**: it reduces regularization bias

Why targeting the effect directly helps

- **Reason 2**: it reduces regularization bias
  - e.g., consider **X** and **T** with vastly different dimensionalities

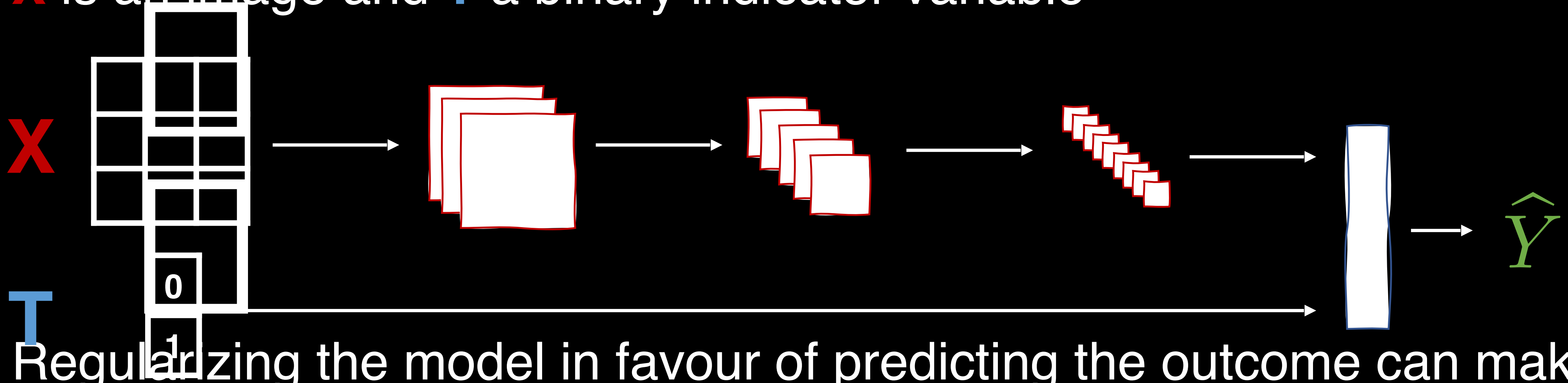
Why targeting the effect directly helps

- **Reason 2**: it reduces regularization bias
  - e.g., consider X and T with vastly different dimensionalities
  - X is an image and T a binary indicator variable



Why targeting the effect directly helps

- **Reason 2**: it reduces regularization bias
  - e.g., consider X and T with vastly different dimensionalities
  - X is an image and T a binary indicator variable



  - Regularizing the model in favour of predicting the outcome can make it ignore the lower-dimensional variable (towards zero-effect)

Previous work in CATE estimation

- **S-Learner (Hill, 2011)**
- **T-Learner (Athey & Imbens, 2016)**
- **R-Learner (Nie & Wager, 2017)**
- **CFRnet/TARnet (Shalit et al., 2017)**
- **X-Learner (Künzel et al., 2018)**
- Perfect Match (Schwab et al. 2018)
- **Multitask-Learner (Alaa & van der Schaar, 2018)**
- **Bayesian Causal Forest (Hahn et al., 2020)**
- **VCnet (Nie et al., 2021)**
- **FlexTENet (Curth & van der Schaar, 2021)**

…

Previous work in CATE estimation

- **S-Learner (Hill, 2011)**
- **T-Learner (Athey & Imbens, 2016)**
- **R-Learner (Nie & Wager, 2017)**
- **CFRnet/TARnet (Shalit et al., 2017)**
- **X-Learner (Künzel et al., 2018)**
- Perfect Match (Schwab et al. 2018)
- **Multitask-Learner (Alaa & van der Schaar, 2018)**
- **Bayesian Causal Forest (Hahn et al., 2020)**
- **VCnet (Nie et al., 2021)**
- **FlexTENet (Curth & van der Schaar, 2021)**

…

Most of these deal with binary or scalar-continuous treatments.

Why structured treatments?

1. **Data-Efficiency**

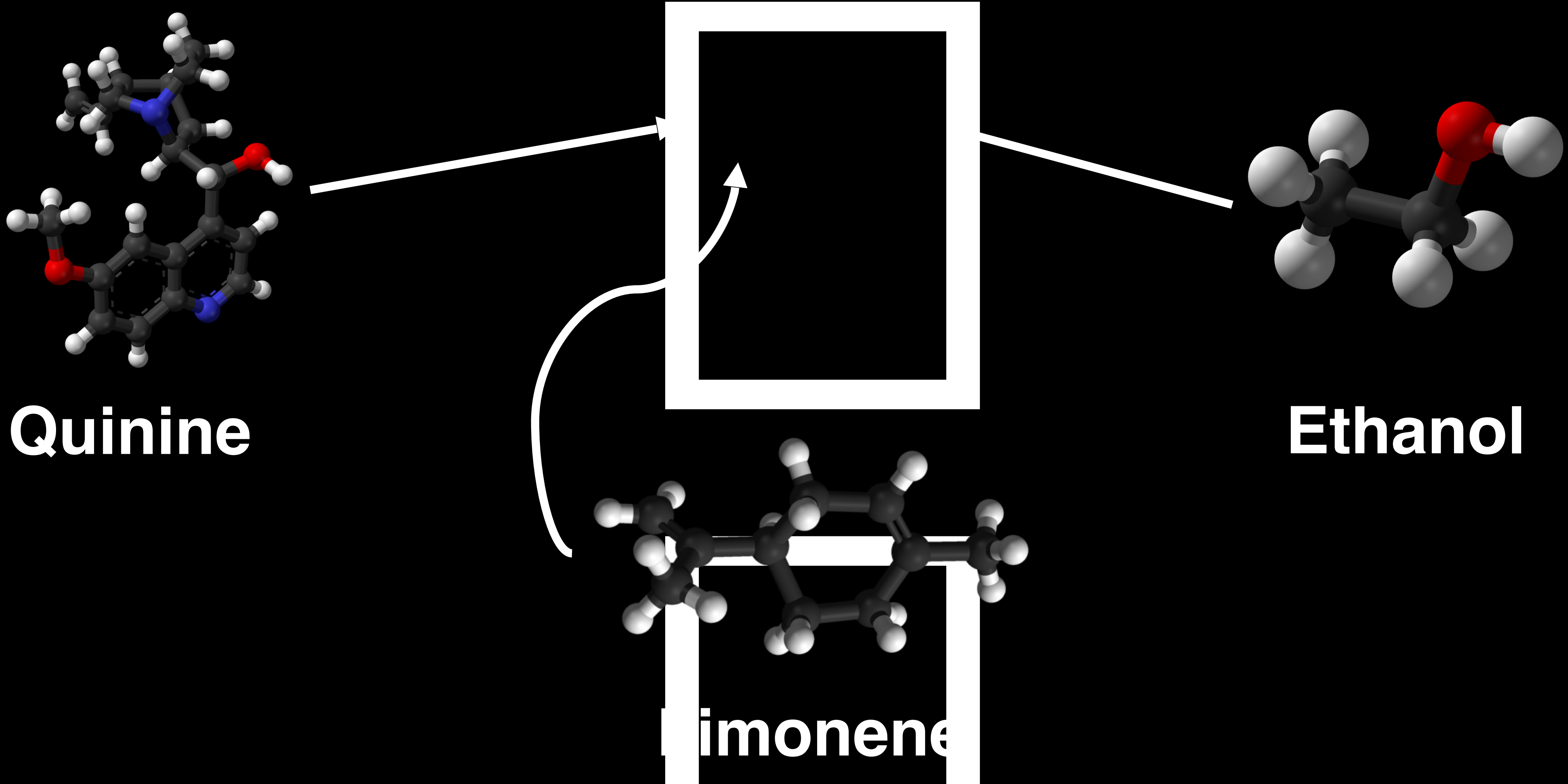
2. **(Infinitely)-Many-Treatments-Setup**

3. **Generalization to unseen treatments**

1. Data-Efficiency

1. Data-Efficiency

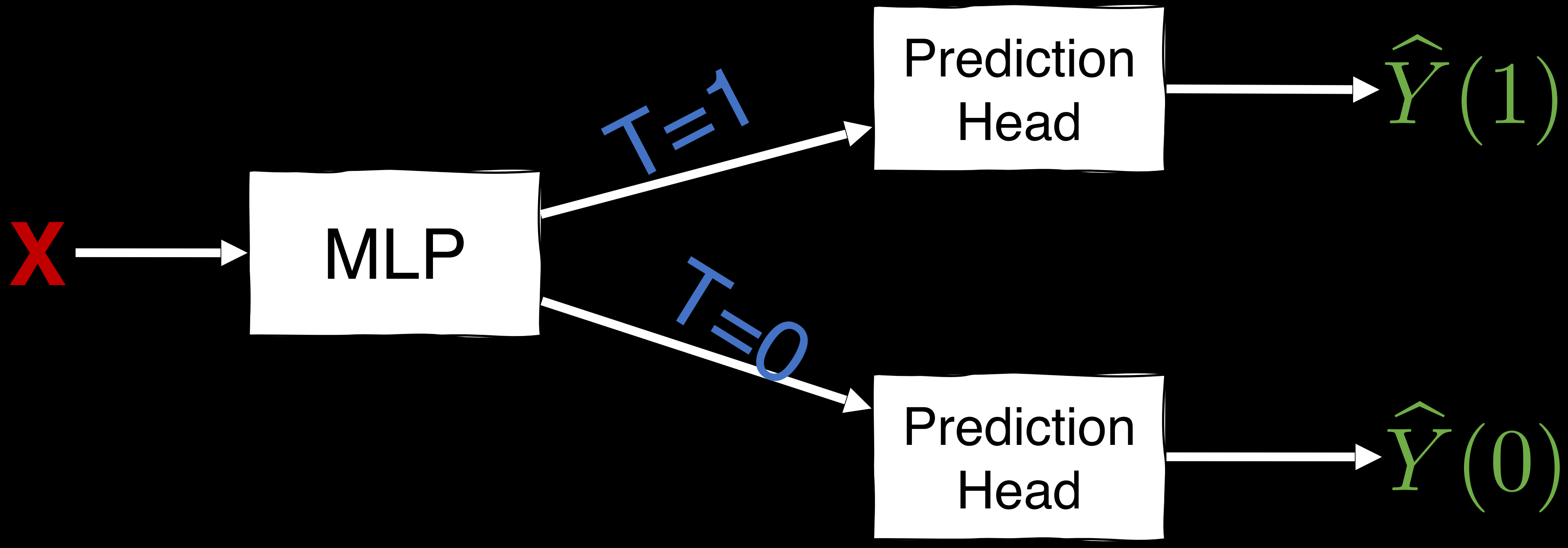
Example: the molecular graph of a drink



2. (Infinitely-)Many-Treatments settings

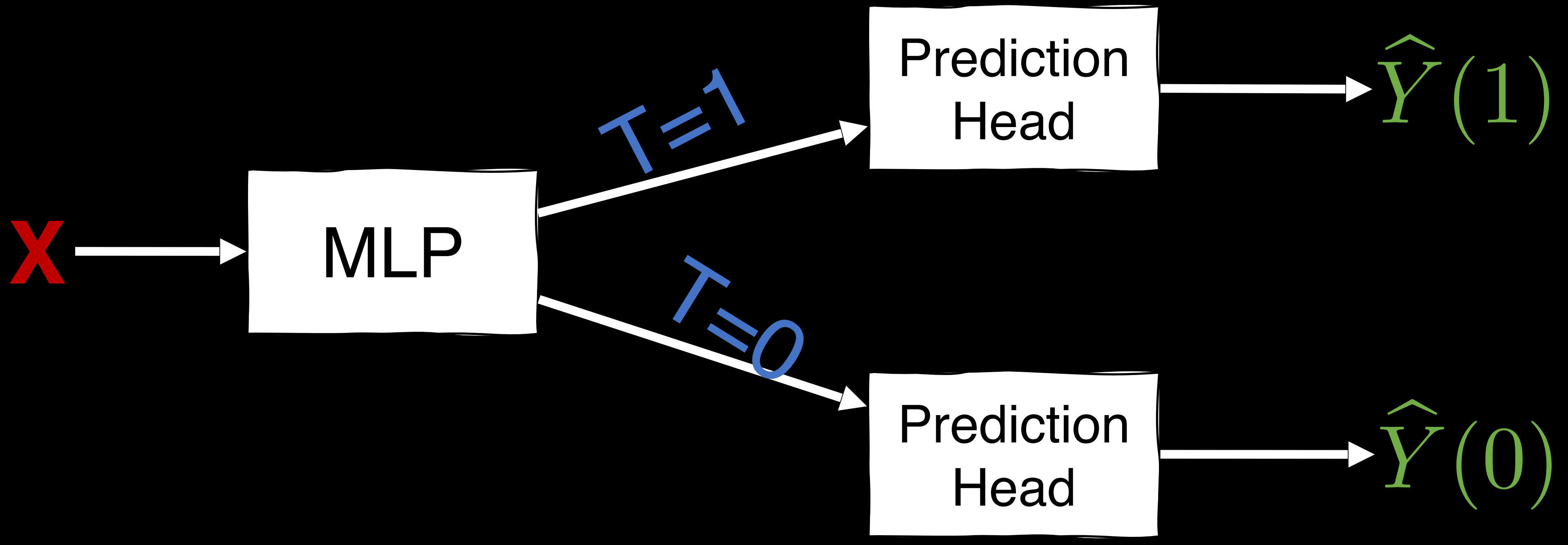
2. (Infinitely-)Many-Treatments settings

Typical NN architectures for CATE estimation



2. (Infinitely-)Many-Treatments settings

Typical NN architectures for CATE estimation

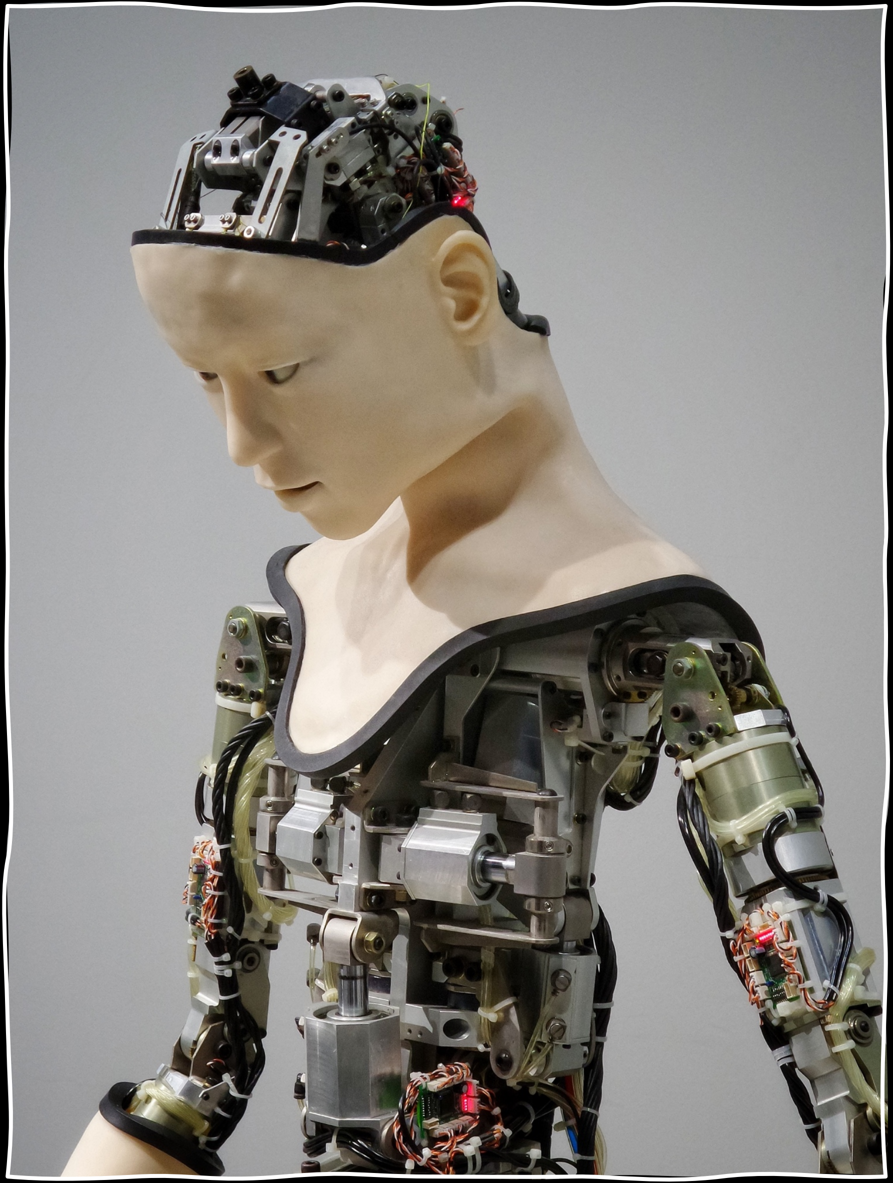


This does not scale well in the treatment options!

2. (Infinitely-)Many-Treatments settings

3. Generalization to unseen treatments

3. Generalization to unseen treatments



How to deal with structured treatments?

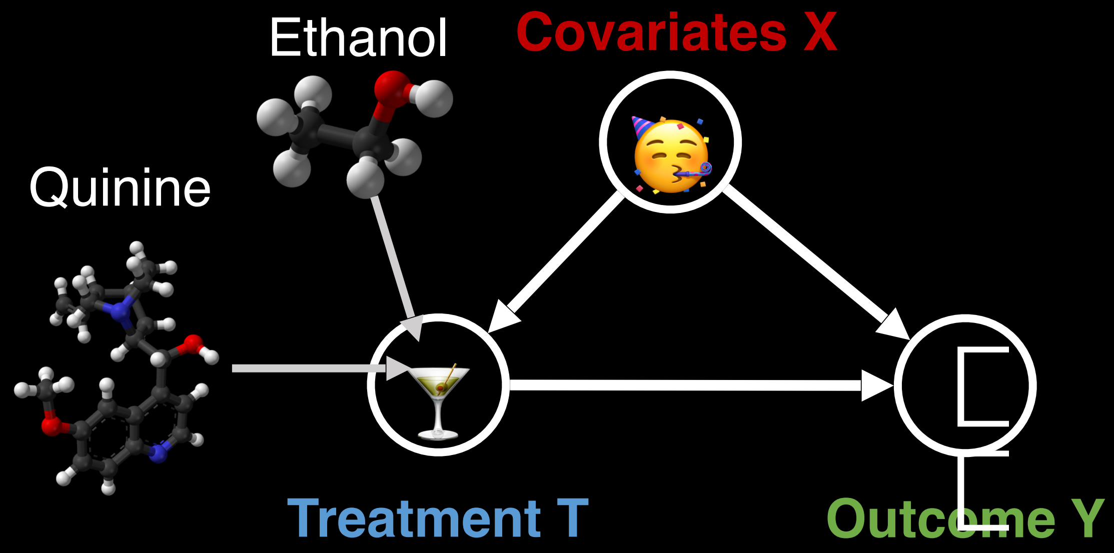
How to deal with structured treatments?

- **Issue 1: H**ow can we represent the CATE with struct. treatments?

How to deal with structured treatments?

- **Issue 1: H**ow can we represent the CATE with struct. treatments?

➢**Goal:** Taking covariates **AND** treatment features into account

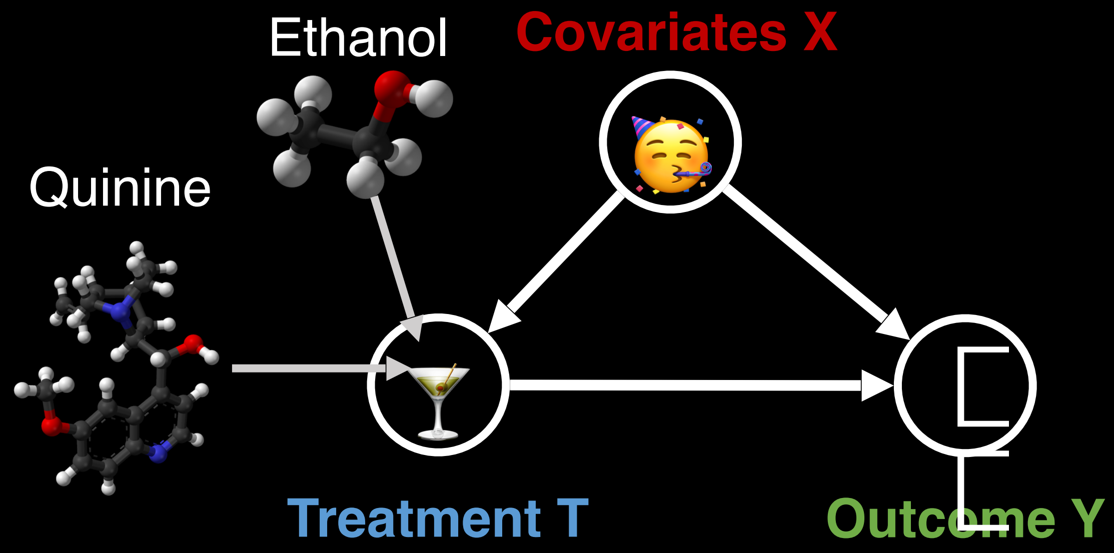


How to deal with structured treatments?

- **Issue 1: H**ow can we represent the CATE with struct. treatments?

➢**Goal:** Taking covariates **AND** treatment features into account

- **Issue 2**: How can we learn the relevant feature maps of the effect?



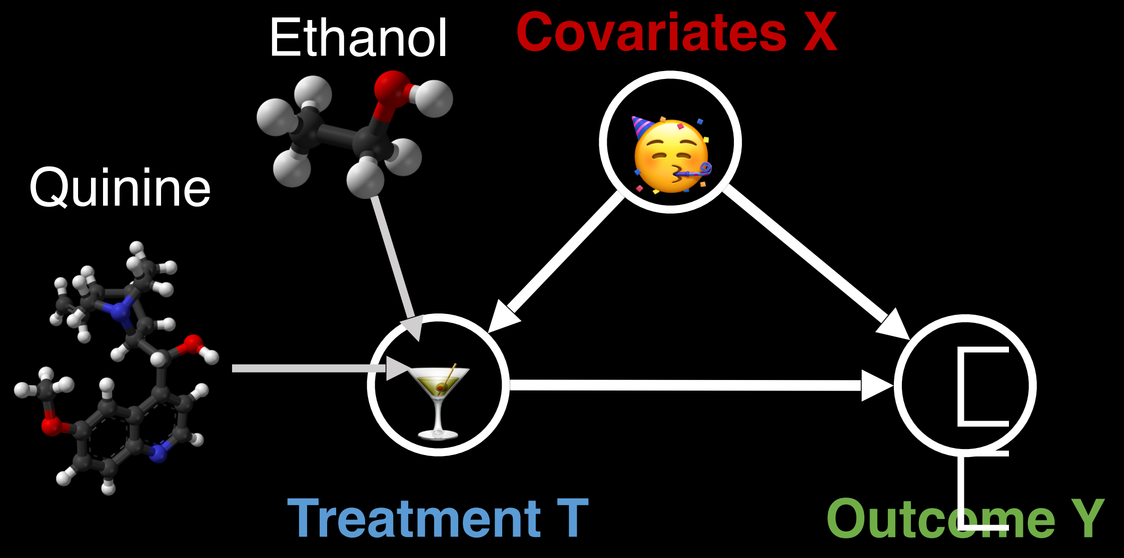
How to deal with structured treatments?

- **Issue 1: H**ow can we represent the CATE with struct. treatments?

➢**Goal:** Taking covariates **AND** treatment features into account

- **Issue 2**: How can we learn the relevant feature maps of the effect?

➢**Goal:** Deriving a trainable objective that targets the effect



Roadmap

- Motivation
- **Generalized Robinson Decomposition**
- Quasi-Oracle Convergence Rate
- Structured Intervention Networks
- Experiments
- Summary

Robinson Decomposition/R-Learner[1]

1 | Xinkun Nie and Stefan Wager. Quasi-Oracle Estimation of Heterogeneous Treatment Effects. Biometrika, 2021.

Robinson Decomposition/R-Learner[1]

- Allows us to construct a learnable objective of the binary CATE

1 | Xinkun Nie and Stefan Wager. Quasi-Oracle Estimation of Heterogeneous Treatment Effects. Biometrika, 2021.

Robinson Decomposition/R-Learner[1]

- Allows us to construct a learnable objective of the binary CATE

- Define the *propensity score* $e(\mathbf{x}) \triangleq p(T = 1 \mid \mathbf{x})$

1 | Xinkun Nie and Stefan Wager. Quasi-Oracle Estimation of Heterogeneous Treatment Effects. Biometrika, 2021.

Robinson Decomposition/R-Learner[1]

- Allows us to construct a learnable objective of the binary CATE
- Define the *propensity score* $e(\mathbf{x}) \triangleq p(T = 1 \mid \mathbf{x})$
- Define the *conditional mean outcome* $m(\mathbf{x}) \triangleq \mathbb{E}[Y \mid \mathbf{x}]$

1 | Xinkun Nie and Stefan Wager. Quasi-Oracle Estimation of Heterogeneous Treatment Effects. Biometrika, 2021.

Robinson Decomposition/R-Learner[1]

- Allows us to construct a learnable objective of the binary CATE

- Define the *propensity score* $e(\mathbf{x}) \triangleq p(T = 1 \mid \mathbf{x})$

- Define the *conditional mean outcome* $m(\mathbf{x}) \triangleq \mathbb{E}[Y \mid \mathbf{x}]$

- Define $\tilde{y}_i \triangleq y_i - \widehat{m}\left(\mathbf{x}_i\right)$ and $\tilde{t}_i \triangleq t_i - \widehat{e}\left(\mathbf{x}_i\right)$, we yield the objective

$$\widehat{\tau}_{\mathrm{b}}(\cdot) = \arg\min_{\tau_{\mathrm{b}}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(\tilde{y}_i - \tilde{t}_i \times \tau_{\mathrm{b}}\left(\mathbf{x}_i\right)\right)^2 + \Lambda\left(\tau_{\mathrm{b}}(\cdot)\right) \right\}$$

- We call $\widehat{m}(\mathbf{x})$ and $\widehat{e}(\mathbf{x})$ the (estimated) **nuisance components**

1 | Xinkun Nie and Stefan Wager. Quasi-Oracle Estimation of Heterogeneous Treatment Effects. Biometrika, 2021.

Generalizing RD to structured treatments I

Generalizing RD to structured treatments I

- **Product Effect Assumption**: Re-parameterize the outcome surface as

$$Y = g(\mathbf{X})^\top h(\mathbf{T}) + \varepsilon$$

where $g : \mathcal{X} \to \mathbb{R}^d, h : \mathcal{T} \to \mathbb{R}^d$ are feature maps

Generalizing RD to structured treatments I

- **Product Effect Assumption**: Re-parameterize the outcome surface as

$$Y = g(\mathbf{X})^\top h(\mathbf{T}) + \varepsilon$$

where $g : \mathcal{X} \to \mathbb{R}^d, h : \mathcal{T} \to \mathbb{R}^d$ are feature maps

- **Universality property**: As we let the dimensionality of $g(\cdot)$ and $h(\cdot)$ grow, we may approximate any bounded function in $\mathcal{C}\left(\mathcal{X} \times \mathcal{T}\right)$

Generalizing RD to structured treatments I

- **Product Effect Assumption**: Re-parameterize the outcome surface as

$$Y = g(\mathbf{X})^\top h(\mathbf{T}) + \varepsilon$$

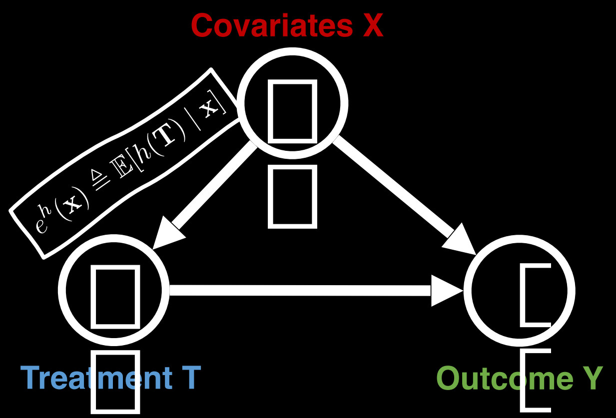
where $g : \mathcal{X} \to \mathbb{R}^d, h : \mathcal{T} \to \mathbb{R}^d$ are feature maps

- **Universality property**: As we let the dimensionality of $g(\cdot)$ and $h(\cdot)$ grow, we may approximate any bounded function in $\mathcal{C}\left(\mathcal{X} \times \mathcal{T}\right)$

~~Issue 1: How can we represent the CATE effect?~~ $\tau\left(\mathbf{t}', \mathbf{t}, \mathbf{x}\right) = g(\mathbf{x})^\top \left(h\left(\mathbf{t}'\right) - h(\mathbf{t})\right)$

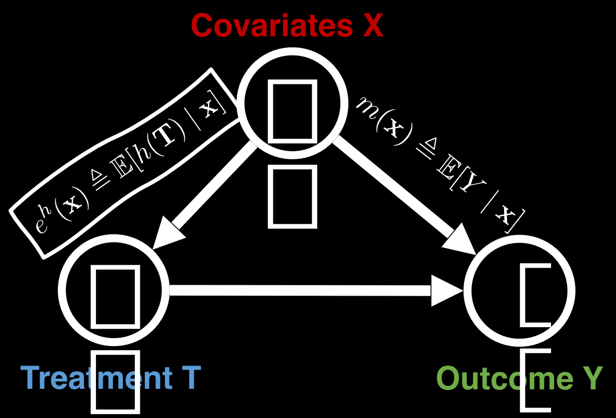
Generalizing RD to structured treatments II

- Define *propensity features* $\boxed{e^h(\mathbf{x}) \triangleq \mathbb{E}[h(\mathbf{T}) \mid \mathbf{x}]}$

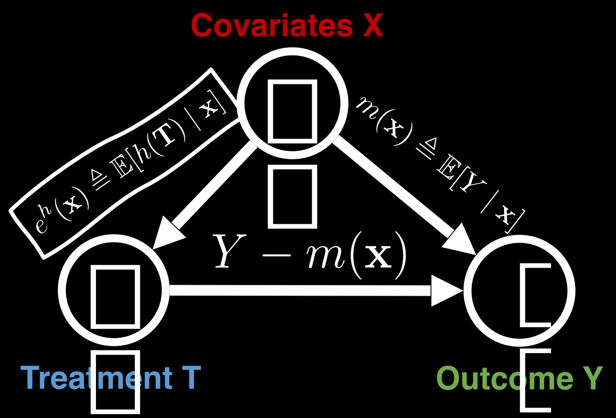


Generalizing RD to structured treatments II

- Define *propensity features* $\boxed{e^h(\mathbf{x}) \triangleq \mathbb{E}[h(\mathbf{T}) \mid \mathbf{x}]}$
- Recall $m(\mathbf{x}) \triangleq \mathbb{E}[Y \mid \mathbf{x}] = g(\mathbf{x})^\top e^h(\mathbf{x})$



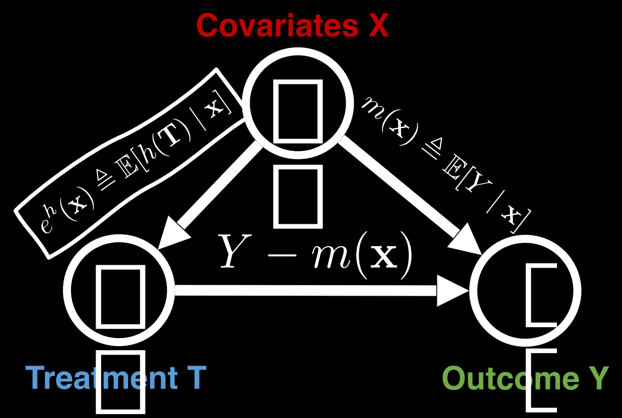
Generalizing RD to structured treatments II



- Define *propensity features* $\boxed{e^h(\mathbf{x}) \triangleq \mathbb{E}[h(\mathbf{T}) \mid \mathbf{x}]}$
- Recall $m(\mathbf{x}) \triangleq \mathbb{E}[Y \mid \mathbf{x}] = g(\mathbf{x})^\top e^h(\mathbf{x})$
- Following the same steps as for the binary treatment case, we yield

$$\boxed{Y - m(\mathbf{X}) = g(\mathbf{X})^\top \left(h(\mathbf{T}) - e^h(\mathbf{X})\right) + \varepsilon}$$

Generalizing RD to structured treatments II



- Define *propensity features* $\boxed{e^h(\mathbf{x}) \triangleq \mathbb{E}[h(\mathbf{T}) \mid \mathbf{x}]}$
- Recall $m(\mathbf{x}) \triangleq \mathbb{E}[Y \mid \mathbf{x}] = g(\mathbf{x})^\top e^h(\mathbf{x})$
- Following the same steps as for the binary treatment case, we yield

$$\boxed{Y - m(\mathbf{X}) = g(\mathbf{X})^\top \left(h(\mathbf{T}) - e^h(\mathbf{X})\right) + \varepsilon}$$

~~Issue 2: How can we learn the relevant feature maps of the effect?~~

Solution: For a fixed $h(\cdot)$ a generalization to structured treatments is

$$\widehat{g}(\cdot) = \arg\min_{g} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(Y_i - \widehat{m}\left(\mathbf{X}_i\right) - g\left(\mathbf{X}_i\right)^\top \left(h\left(\mathbf{T}_i\right) - \widehat{e}^h\left(\mathbf{X}_i\right)\right)\right)^2 \right\}$$

Roadmap

- Motivation
- **Generalized Robinson Decomposition**
- Quasi-Oracle Convergence Rate
- Structured Intervention Networks
- Experiments
- Summary

Why is our decomposition useful*?

** Main statement in Theorem 2 of paper.*

Why is our decomposition useful*?

$$\hat{f}(\mathbf{x}, \mathbf{t}) := \Psi(\mathbf{x})^T \Theta \Phi(\mathbf{t})$$

** Main statement in Theorem 2 of paper.*

Why is our decomposition useful*?

$$\hat{f}(\mathbf{x}, \mathbf{t}) := \Psi(\mathbf{x})^T \Theta \Phi(\mathbf{t})$$

$$f^*(\mathbf{x}, \mathbf{t}) := \mathbb{E}[Y | \mathbf{x}, \mathbf{t}]$$

** Main statement in Theorem 2 of paper.*

Why is our decomposition useful*?

$$\hat{f}(\cdot_{\mathbf{x}}, \cdot_{\mathbf{t}}) := \Psi(\cdot_{\mathbf{x}})^T \Theta \Phi(\cdot_{\mathbf{t}})$$

$$\downarrow$$

$$f^*(\cdot_{\mathbf{x}}, \cdot_{\mathbf{t}}) := \mathbb{E}[Y \mid \cdot_{\mathbf{x}}, \cdot_{\mathbf{t}}]$$

** Main statement in Theorem 2 of paper.*

Why is our decomposition useful*?

$$\boxed{\begin{aligned} \hat{f}(\cdot_{\mathbf{x}}, \cdot_{\mathbf{t}}) &:= \Psi(\cdot_{\mathbf{x}})^T \Theta \Phi(\cdot_{\mathbf{t}}) \\ &\Big\downarrow \tilde{O}(n^{-\frac{1}{2(1+p)}}) \\ f^*(\cdot_{\mathbf{x}}, \cdot_{\mathbf{t}}) &:= \mathbb{E}[Y \mid \cdot_{\mathbf{x}}, \cdot_{\mathbf{t}}] \end{aligned}}$$

** Main statement in Theorem 2 of paper.*

Why is our decomposition useful*?

$$\hat{f}(\cdot_{\mathbf{x}}, \cdot_{\mathbf{t}}) := \Psi(\cdot_{\mathbf{x}})^T \Theta \Phi(\cdot_{\mathbf{t}})$$

$$\Big\downarrow \ \tilde{O}(n^{-\frac{1}{2(1+p)}})$$

$$f^*(\cdot_{\mathbf{x}}, \cdot_{\mathbf{t}}) := \mathbb{E}[Y \mid \cdot_{\mathbf{x}}, \cdot_{\mathbf{t}}]$$

$$\hat{m}(\cdot_{\mathbf{x}}) \longrightarrow m(\cdot_{\mathbf{x}})$$

$$\hat{e}^h(\cdot_{\mathbf{x}}) \longrightarrow e^h(\cdot_{\mathbf{x}})$$

** Main statement in Theorem 2 of paper.*

Why is our decomposition useful*?

$$\hat{f}(\cdot_{\mathbf{x}}, \cdot_{\mathbf{t}}) := \Psi(\cdot_{\mathbf{x}})^T \Theta \Phi(\cdot_{\mathbf{t}})$$

$$\Big\downarrow \; \tilde{O}(n^{-\frac{1}{2(1+p)}})$$

$$f^*(\cdot_{\mathbf{x}}, \cdot_{\mathbf{t}}) := \mathbb{E}[Y \mid \cdot_{\mathbf{x}}, \cdot_{\mathbf{t}}]$$

$$\hat{m}(\cdot_{\mathbf{x}}) \xrightarrow{O(n^{-\frac{1}{4}})} m(\cdot_{\mathbf{x}})$$

$$\hat{e}^h(\cdot_{\mathbf{x}}) \xrightarrow{O(n^{-\frac{1}{4}})} e^h(\cdot_{\mathbf{x}})$$

** Main statement in Theorem 2 of paper.*

Why is our decomposition useful*?

$$\hat{f}(\cdot_{\mathbf{x}}, \cdot_{\mathbf{t}}) := \Psi(\cdot_{\mathbf{x}})^T \Theta \Phi(\cdot_{\mathbf{t}})$$

$$\downarrow \tilde{O}(n^{-\frac{1}{2(1+p)}})$$

$$f^*(\cdot_{\mathbf{x}}, \cdot_{\mathbf{t}}) := \mathbb{E}[Y \mid \cdot_{\mathbf{x}}, \cdot_{\mathbf{t}}]$$

$$\hat{m}(\cdot_{\mathbf{x}}) \xrightarrow{O(n^{-\frac{1}{4}})} m(\cdot_{\mathbf{x}})$$

$$\hat{e}^h(\cdot_{\mathbf{x}}) \xrightarrow{O(n^{-\frac{1}{4}})} e^h(\cdot_{\mathbf{x}})$$

$$\text{Overlap: } \mathcal{P}_{\Psi(X)\times\Phi(T)} > 0$$

** Main statement in Theorem 2 of paper.*

What does this mean?

- The target or nuisance functions cannot converge faster than $O(n^{-1/2})$

What does this mean?

- The target or nuisance functions cannot converge faster than $O(n^{-1/2})$
- Usually this rate caps the rate of the target function - see the discussion in e.g. *Chernozhukov et al.,* 2018 (Double Machine Learning)

What does this mean?

- The target or nuisance functions cannot converge faster than $O(n^{-1/2})$
- Usually this rate caps the rate of the target function - see the discussion in e.g. *Chernozhukov et al.,* 2018 (Double Machine Learning)

We show that in the fixed features setting, the target function converges at almost $n^{-\frac{1}{2(1+p)}}$ rate as long as the nuisance functions converge at $n^{-1/4}$ rate.

Proof outline - Notation

The regret quantities:

$$R(\boldsymbol{\Theta}) = L(\boldsymbol{\Theta}) - L(\boldsymbol{\Theta}^*)$$
$$\tilde{R}_n(\boldsymbol{\Theta}) = \tilde{L}_n(\boldsymbol{\Theta}) - \tilde{L}_n(\boldsymbol{\Theta}^*)$$
$$\hat{R}_n(\boldsymbol{\Theta}) = \hat{L}_n(\boldsymbol{\Theta}) - \hat{L}_n(\boldsymbol{\Theta}^*)$$

$$L(f_{\boldsymbol{\Theta}}) = L(\boldsymbol{\Theta}) = \mathbb{E}\left[\left\{(Y - m^*(\mathbf{X})) - \boldsymbol{\alpha}(\mathbf{X})^T \boldsymbol{\Theta}(\boldsymbol{\beta}(\mathbf{T}) - e^P(\mathbf{X}))\right\}^2\right] \tag{39}$$

$$\tilde{L}_n(f_{\boldsymbol{\Theta}}) = \tilde{L}_n(\boldsymbol{\Theta}) = \sum_{l=1}^{n}\left[\left\{(Y - m^*(\mathbf{X}_l)) - \boldsymbol{\alpha}(\mathbf{X}_l)^T \boldsymbol{\Theta}(\boldsymbol{\beta}(\mathbf{T}_l) - e^P(\mathbf{X}_l))\right\}^2\right] \tag{40}$$

$$\hat{L}_n(f_{\boldsymbol{\Theta}}) = \hat{L}_n(\boldsymbol{\Theta}) = \sum_{l=1}^{n}\left[\left\{(Y - \hat{m}(\mathbf{X}_l)) - \boldsymbol{\alpha}(\mathbf{X}_l)^T \boldsymbol{\Theta}(\boldsymbol{\beta}(\mathbf{T}_l) - \hat{e}^P(\mathbf{X}_l))\right\}^2\right] \tag{41}$$

Proof outline - oracle rate

The regret quantities:

$$R(\boldsymbol{\Theta}) = L(\boldsymbol{\Theta}) - L(\boldsymbol{\Theta}^*)$$
$$\tilde{R}_n(\boldsymbol{\Theta}) = \tilde{L}_n(\boldsymbol{\Theta}) - \tilde{L}_n(\boldsymbol{\Theta}^*)$$
$$\hat{R}_n(\boldsymbol{\Theta}) = \hat{L}_n(\boldsymbol{\Theta}) - \hat{L}_n(\boldsymbol{\Theta}^*)$$

Lemma 4. *Let $\check{L}(f_\Theta \in \mathcal{H}_c)$ be a loss function, and $\check{R}(f_\Theta; c) = \check{L}(f_\Theta) - \check{L}(f_{\Theta_c})$ be the associated c-regret. Suppose $\rho(r)$ is a positive, continuous, increasing function. If, $\forall\ 1 \leq c \leq C$ and some $k > 1$, the following inequality holds for all $f_\Theta \in \mathcal{H}_c$:*

$$\frac{1}{k}\check{R}(f_\Theta; c) - \rho(c) \leq R(f_\Theta; c) \leq k\check{R}(f_\Theta; c) + \rho(c) \tag{45}$$

Then, writing $\kappa_1 = 2k + \frac{1}{k}$ and $\kappa_2 = 2k^2 + 3$, any solution to the regularized minimization problem with $\Lambda(c) \geq \rho(c)$,

$$f_{\check{\boldsymbol{\Theta}}} \in \arg\min_{f_\Theta \in \mathcal{H}_C}\{\check{L}(f_\Theta) + \kappa_1 \Lambda(f_\Theta)_{\mathcal{H}}\} \tag{46}$$

also satisfied the following risk bound:

$$L(f_{\check{\boldsymbol{\Theta}}}) \leq \inf_{f_\Theta \in \mathcal{H}_C}\{L(f_\Theta) + \kappa_2 \Lambda(f_\Theta)_{\mathcal{H}} \tag{47}$$

Proof outline - oracle rate

Mendelson and Neeman (2010) for $\tilde{R}$:

$$\rho_n(c) = U(\epsilon)\left\{1 + \log(n) + \log\left(\log(c+e)\right)\right\}\left(\frac{(c+1)^p \log(n)}{\sqrt{n}}\right)^{2/(1+p)} \quad (53)$$

With 53, Lemma 4 immediately implies that penalized regression over $\mathcal{H}_C$ with the oracle loss function $\tilde{L}_n(\cdot)$ and regularizer $\kappa_1 \rho_n(c)$ satisfies the bound below with high probability:

$$R(\tilde{\mathbf{\Theta}}_n) = L(\tilde{\mathbf{\Theta}}_n) - L(\mathbf{\Theta}^*) \leq \inf_{\mathbf{\Theta} \in \mathcal{H}_C} \{L((\mathbf{\Theta}) + \kappa_2 \rho_n(\|\mathbf{\Theta}\|_{\mathcal{H}})\} - L(\mathbf{\Theta}^*) \quad (54)$$

Proof outline - oracle rate

Furthermore, Corollary 2.7 in [36] gives that for any $1 < c < C$,

$$\inf_{\mathbf{\Theta} \in \mathcal{H}_C} \{L(\mathbf{\Theta}) + \kappa_2 \rho_n(\|\mathbf{\Theta}\|_{\mathcal{H}})\} \leq L(\mathbf{\Theta}^*) + \{L(\mathbf{\Theta}_c^*) - L(\mathbf{\Theta}^*)\} + \kappa_2 \rho_n(c) \tag{55}$$

Finally, note that for large enough c,

$$\left\{L\left(\mathbf{\Theta}_c^*\right) - L\left(\mathbf{\Theta}^*\right)\right\} = 0, \tag{56}$$

so the error is dominated by $\rho_n(c)$, at

$$R\left(\tilde{\mathbf{\Theta}}_n\right) = \mathcal{O}\left(\left(\log(n)\right)^{\frac{3+p}{1+p}} n^{-\frac{1}{1+p}}\right) = \tilde{\mathcal{O}}(n^{-\frac{1}{1+p}}), \tag{57}$$

where $\tilde{\mathcal{O}}$ notation ignores the logarithmic factors.

Proof outline - Bridging $\hat{R}$ and $\tilde{R}$

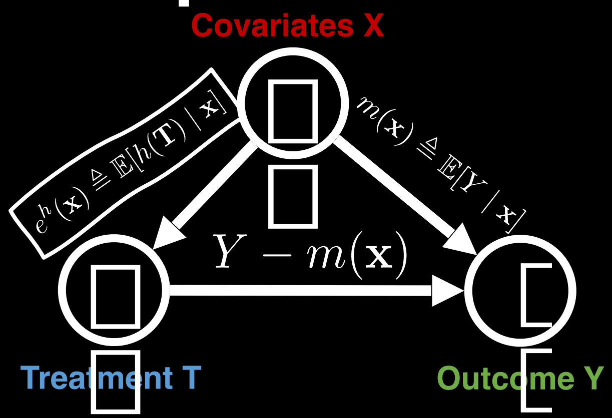
Overlap + Boundedness of Y:

$$\left|\hat{R}_n(\mathbf{\Theta}; c) - \tilde{R}_n(\mathbf{\Theta}; c)\right| \leq 0.125 R(\mathbf{\Theta}; c) + o(\rho_n(c)) \tag{119}$$

Roadmap

- Motivation
- **Generalized Robinson Decomposition**
- Quasi-Oracle Convergence Rate
- Structured Intervention Networks
- Experiments
- Summary

Recap: Generalized Robinson Decomposition



- Define *propensity features* $\boxed{e^h(\mathbf{x}) \triangleq \mathbb{E}[h(\mathbf{T}) \mid \mathbf{x}]}$
- Recall $m(\mathbf{x}) \triangleq \mathbb{E}[Y \mid \mathbf{x}] = g(\mathbf{x})^\top e^h(\mathbf{x})$
- Following the same steps as for the binary treatment case, we yield

$$\boxed{Y - m(\mathbf{X}) = g(\mathbf{X})^\top \left(h(\mathbf{T}) - e^h(\mathbf{X})\right) + \varepsilon}$$

~~Issue 2: How can we learn the relevant feature maps of the effect?~~

Solution: For a fixed $h(\cdot)$, a generalization to structured treatments is

$$\widehat{g}(\cdot) = \arg\min_{g} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(Y_i - \widehat{m}\left(\mathbf{X}_i\right) - g\left(\mathbf{X}_i\right)^\top \left(h\left(\mathbf{T}_i\right) - \widehat{e}^h\left(\mathbf{X}_i\right)\right)\right)^2 \right\}$$

Recap: Generalized Robinson Decomposition

How can we turn this into a practical learning algorithm?

~~Issue 2: How can we learn the relevant feature maps of the effect?~~

Solution: For a fixed $h(\cdot)$, a generalization to structured treatments is

$$\widehat{g}(\cdot) = \arg\min_{g} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(Y_i - \widehat{m}\left(\mathbf{X}_i\right) - g\left(\mathbf{X}_i\right)^\top \left(h\left(\mathbf{T}_i\right) - \widehat{e}^h\left(\mathbf{X}_i\right) \right) \right)^2 \right\}$$

Two-Stage Training Procedure

Two-Stage Training Procedure

- **Stage 1**: Learn parameters of $\widehat{m}_{\boldsymbol{\theta}}(\mathbf{X})$ based on objective

$$J_m(\boldsymbol{\theta}) = \sum_{i=1}^{m} \Big(y_i - \widehat{m}_{\boldsymbol{\theta}}\left(\mathbf{x}_i\right) \Big)^2$$

Two-Stage Training Procedure

- **Stage 1**: Learn parameters of $\widehat{m}_{\boldsymbol{\theta}}(\mathbf{X})$ based on objective

$$J_m(\boldsymbol{\theta}) = \sum_{i=1}^{m} \Big(y_i - \widehat{m}_{\boldsymbol{\theta}}\left(\mathbf{x}_i\right)\Big)^2$$

- Stage **2: Alternate between optimizing** $\widehat{g}_{\psi}(\mathbf{X}), \widehat{h}_{\boldsymbol{\phi}}(\mathbf{T})$ **and** $\hat{e}_{\eta}^{h}(\mathbf{X})$

Two-Stage Training Procedure

- **Stage 1**: Learn parameters of $\widehat{m}_{\boldsymbol{\theta}}(\mathbf{X})$ based on objective

$$J_m(\boldsymbol{\theta}) = \sum_{i=1}^{m} \left(y_i - \widehat{m}_{\boldsymbol{\theta}}\left(\mathbf{x}_i\right)\right)^2$$

- Stage **2: Alternate between optimizing** $\widehat{g}_{\psi}(\mathbf{X}), \widehat{h}_{\boldsymbol{\phi}}(\mathbf{T})$ **and** $\hat{e}^h_{\eta}(\mathbf{X})$
  - **a**: Freeze $\widehat{m}_{\boldsymbol{\theta}}(\mathbf{X})$ and $\hat{e}^h_{\eta}(\mathbf{X})$ to optimize $\widehat{g}_{\psi}(\mathbf{X}), \widehat{h}_{\boldsymbol{\phi}}(\mathbf{T})$ based on

$$J_{g,h}(\boldsymbol{\phi}, \boldsymbol{\psi}) = \sum_{i=1}^{n} \left(y_i - \left\{\widehat{m}_{\boldsymbol{\theta}}\left(\mathbf{x}_i\right) + \widehat{g}_{\psi}\left(\mathbf{x}_i\right)^{\top}\left(\widehat{h}_{\phi}\left(\mathbf{t}_i\right) - \widehat{e}^h_{\boldsymbol{\eta}}\left(\mathbf{x}_i\right)\right)\right\}\right)^2$$

Two-Stage Training Procedure

- **Stage 1**: Learn parameters of $\widehat{m}_{\boldsymbol{\theta}}(\mathbf{X})$ based on objective

$$J_m(\boldsymbol{\theta}) = \sum_{i=1}^{m} \Big(y_i - \widehat{m}_{\boldsymbol{\theta}}\left(\mathbf{x}_i\right)\Big)^2$$

- Stage **2: Alternate between optimizing** $\widehat{g}_{\psi}(\mathbf{X}), \widehat{h}_{\boldsymbol{\phi}}(\mathbf{T})$ **and** $\hat{e}^h_{\eta}(\mathbf{X})$
  - **a**: Freeze $\widehat{m}_{\boldsymbol{\theta}}(\mathbf{X})$ and $\hat{e}^h_{\eta}(\mathbf{X})$ to optimize $\widehat{g}_{\psi}(\mathbf{X}), \widehat{h}_{\boldsymbol{\phi}}(\mathbf{T})$ based on

$$J_{g,h}(\boldsymbol{\phi}, \boldsymbol{\psi}) = \sum_{i=1}^{n} \left(y_i - \left\{\widehat{m}_{\boldsymbol{\theta}}\left(\mathbf{x}_i\right) + \widehat{g}_{\psi}\left(\mathbf{x}_i\right)^{\top} \left(\widehat{h}_{\phi}\left(\mathbf{t}_i\right) - \widehat{e}^h_{\boldsymbol{\eta}}\left(\mathbf{x}_i\right)\right)\right\}\right)^2$$

  - b: Freeze $\widehat{m}_{\boldsymbol{\theta}}(\mathbf{X})$ and $\widehat{g}_{\psi}(\mathbf{X}), \widehat{h}_{\boldsymbol{\phi}}(\mathbf{T})$ to optimize $\hat{e}^h_{\eta}(\mathbf{X})$ based on

$$J_{e^h}(\boldsymbol{\eta}) = \sum_{i=1}^{n} \sum_{j=1}^{d} \left(\widehat{h}_{\boldsymbol{\phi}}\left(\mathbf{t}_i\right)^{(j)} - \widehat{e}^h_{\boldsymbol{\eta}}\left(\mathbf{x}_i\right)^{(j)}\right)^2$$

Algorithm

a SIN Training.

Input: Stage 1 data $\mathcal{D}_1 := \{(\mathbf{x}_i, y_i)\}_{i=1}^{m}$, Stage 2 data $\mathcal{D}_2 := \{(\mathbf{x}_i, \mathbf{t}_i, y_i)\}_{i=1}^{n}$. Step sizes $\lambda_{\boldsymbol{\theta}}, \lambda_{\boldsymbol{\eta}}, \lambda_{\boldsymbol{\psi}}, \lambda_{\boldsymbol{\phi}}$. Number of update steps K. Mini-batch sizes B_1, B_2.

Initialize parameters: $\boldsymbol{\theta}, \boldsymbol{\eta}, \boldsymbol{\psi}, \boldsymbol{\phi}$
while not converged **do** ▷ *Stage 1*
  Sample mini-batch $\{(\mathbf{x}_b, y_b)\}_{b=1}^{m_{B_1}}$
  Evaluate $J_m(\boldsymbol{\theta})$
  Update $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \lambda_{\boldsymbol{\theta}} \widehat{\nabla}_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$
end while
while not converged **do** ▷ *Stage 2*
  Sample mini-batch $\{(\mathbf{x}_b, \mathbf{t}_b, y_b)\}_{b=1}^{m_{B_2}}$
  Evaluate $J_{g,h}(\boldsymbol{\psi}, \boldsymbol{\phi}), J_{e^h}(\boldsymbol{\eta})$
  for $k = 1$ to K **do**
    Update $\boldsymbol{\phi} \leftarrow \boldsymbol{\phi} - \lambda_{\boldsymbol{\phi}} \widehat{\nabla}_{\boldsymbol{\phi}} J_{g,h}(\boldsymbol{\psi}, \boldsymbol{\phi})$
    Update $\boldsymbol{\psi} \leftarrow \boldsymbol{\psi} - \lambda_{\boldsymbol{\psi}} \widehat{\nabla}_{\boldsymbol{\psi}} J_{g,h}(\boldsymbol{\psi}, \boldsymbol{\phi})$
  end for
  Update $\boldsymbol{\eta} \leftarrow \boldsymbol{\eta} - \lambda_{\boldsymbol{\eta}} \widehat{\nabla}_{\boldsymbol{\eta}} J_{e^h}(\boldsymbol{\eta})$
end while

280 character PyTorch-like code

```
# Initialize submodels and optimizers
m, e, g, h = MLP(), MLP(), MLP(), GNN()
m_opt, e_opt, g_opt, h_opt = Adam(m.params(), m_lr), Adam(e.params(), e_lr), ...

# Stage 1: Train m(x)
for batch in train_loader:
    X, Y = batch.X, batch.Y
    m_opt.zero_grad()
    F.mse_loss(m(X), Y).backward()
    m_opt.step()

# Stage 2: Train g(x), h(t), e(x)
for batch in train_loader:
    X, T, Y = batch.X, batch.T, batch.Y
    for _ in range(num_update_steps):
        g_opt.zero_grad()
        h_opt.zero_grad()
        F.mse_loss((g(X)*(h(T) - e(X))).sum(-1), (Y-m(X))).backward()
        g_opt.step()
        h_opt.step()
    e_opt.zero_grad()
    F.mse_loss(e(X), h(T)).backward()
    e_opt.step()
```

Roadmap

- Motivation
- **Generalized Robinson Decomposition**
- Quasi-Oracle Convergence Rate
- Structured Intervention Networks
- Experiments
- Summary

Experimental Setup

- **Data**: Two semi-synthetic datasets involving graph-treatments

Small-World (SW)
X: Samples from multivar. uniform dist.
T: Watts–Strogatz small-world graphs

The Cancer Genomic Atlas (TCGA)[1]
X: Gene expression data of cancer patients

1 | Data generated by the TCGA Research Network: https://www.cancer.gov/tcga.
2 | L. Ruddigkeit, et al., Enumeration of 166 billion organic small molecules in the chemical universe database GDB-17, 2012.
3 | Harada & Kashima, GraphITE: Estimating Individual Effects of Graph-structured Treatments, 2020.

Experimental Setup

- **Data**: Two semi-synthetic datasets involving graph-treatments

Small-World (SW)
X: Samples from multivar. uniform dist.
T: Watts–Strogatz small-world graphs

The Cancer Genomic Atlas (TCGA)[1]
X: Gene expression data of cancer patients

- **Tasks**: Predicting in-sample/out-sample CATEs

1 | Data generated by the TCGA Research Network: https://www.cancer.gov/tcga.
2 | L. Ruddigkeit, et al., Enumeration of 166 billion organic small molecules in the chemical universe database GDB-17, 2012.
3 | Harada & Kashima, GraphITE: Estimating Individual Effects of Graph-structured Treatments, 2020.

Experimental Setup

- **Data**: Two semi-synthetic datasets involving graph-treatments

Small-World (SW)
X: Samples from multivar. uniform dist.
T: Watts–Strogatz small-world graphs

The Cancer Genomic Atlas (TCGA)[1]
X: Gene expression data of cancer patients

- **Tasks**: Predicting in-sample/out-sample CATEs

- **Baselines:** GraphITE[3], **Vanilla Regression (**GNN/CAT), Zero

1 | Data generated by the TCGA Research Network: https://www.cancer.gov/tcga.
2 | L. Ruddigkeit, et al., Enumeration of 166 billion organic small molecules in the chemical universe database GDB-17, 2012.
3 | Harada & Kashima, GraphITE: Estimating Individual Effects of Graph-structured Treatments, 2020.

Experimental Setup

- **Data**: Two semi-synthetic datasets involving graph-treatments

Small-World (SW)
X: Samples from multivar. uniform dist.
T: Watts–Strogatz small-world graphs

The Cancer Genomic Atlas (TCGA)[1]
X: Gene expression data of cancer patients

- **Tasks**: Predicting in-sample/out-sample CATEs
- **Baselines:** GraphITE[3], **Vanilla Regression (**GNN/CAT), Zero
- **Metric**: (Un-)Weighted expected Precision in Estimation of Het. Effects

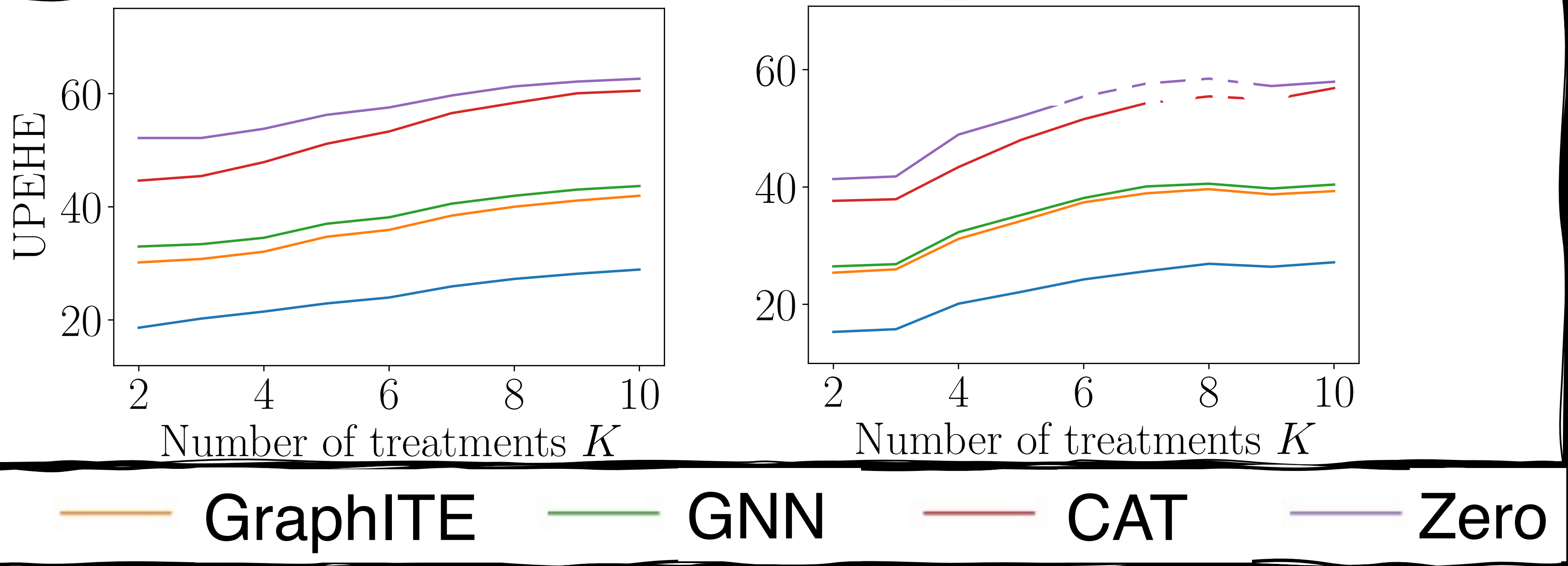
$$\epsilon_{\mathrm{UPEHE(WPEHE)}} \triangleq \int_{\mathcal{X}} \left(\widehat{\tau}\left(\mathbf{t}', \mathbf{t}, \mathbf{x}\right) - \tau\left(\mathbf{t}', \mathbf{t}, \mathbf{x}\right)\right)^2 p(\mathbf{t} \mid \mathbf{x}) p\left(\mathbf{t}' \mid \mathbf{x}\right) p(\mathbf{x}) \mathrm{d}\mathbf{x}$$

1 | Data generated by the TCGA Research Network: https://www.cancer.gov/tcga.
2 | L. Ruddigkeit, et al., Enumeration of 166 billion organic small molecules in the chemical universe database GDB-17, 2012.
3 | Harada & Kashima, GraphITE: Estimating Individual Effects of Graph-structured Treatments, 2020.

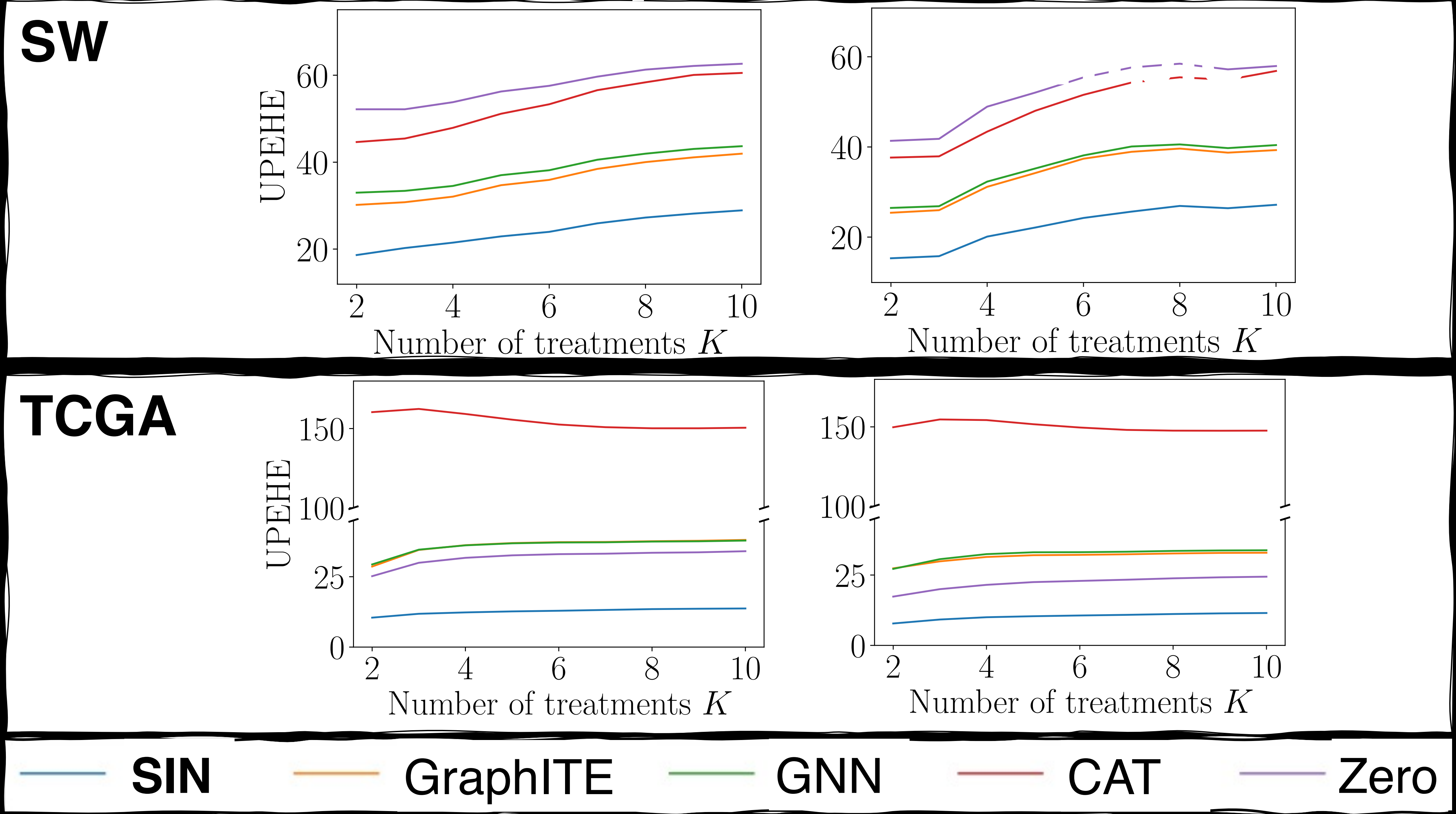
Results: **In-Sample** **Out-Sample**

Results: **In-Sample** **Out-**

SW



Results: **In-Sample** **Out-**



WPEHE for most likely K=6 treatments

Method	SW		TCGA	
	In-sample	Out-sample	In-sample	Out-sample
Zero	56.26 ± 8.12	53.77 ± 8.93	26.63 ± 7.55	17.94 ± 4.86
CAT	51.75 ± 8.85	49.76 ± 9.73	155.88 ± 52.82	146.62 ± 42.32
GNN	37.10 ± 6.84	36.74 ± 7.42	30.67 ± 8.29	27.57 ± 7.95
GraphITE	34.81 ± 6.70	35.94 ± 8.07	30.31 ± 8.96	27.48 ± 8.95
SIN	$\mathbf{23.00 \pm 4.56}$	$\mathbf{23.19 \pm 5.56}$	$\mathbf{10.98 \pm 3.45}$	$\mathbf{8.15 \pm 1.46}$

Roadmap

- Motivation
- **Generalized Robinson Decomposition**
- Quasi-Oracle Convergence Rate
- Structured Intervention Networks
- Experiments
- Summary

Summary

- What?

CATE **Structured treatments (e.g. graphs)**

$$\tau(\quad,\quad,\quad) = \mathbb{E}\left[Y \mid \quad, \mathrm{do}(\quad)\right] - \mathbb{E}\left[Y \mid \quad, \mathrm{do}(\quad)\right]$$

Summary

CATE **Structured treatments (e.g. graphs)**

- What?

$$\tau(\quad,\quad,\quad) = \mathbb{E}\left[Y \mid \quad, \mathrm{do}(\quad)\right] - \mathbb{E}\left[Y \mid \quad, \mathrm{do}(\quad)\right]$$

- Why?

1. Data-Efficiency
2. (Infinitely-)Many-Treatments-Settings
3. Generalization to unseen treatments

Summary

CATE **Structured treatments (e.g. graphs)**

- What? $\tau(\ ,\ ,\ ) = \mathbb{E}\left[Y \mid\ , \mathrm{do}(\ )\right] - \mathbb{E}\left[Y \mid\ , \mathrm{do}(\ )\right]$
- Why?

1. Data-Efficiency
2. (Infinitely-)Many-Treatments-Settings
3. Generalization to unseen treatments

- How?

Generalized Robinson Decomposition

$$Y - m(\mathbf{X}) = g(\mathbf{X})^\top \left(h(\mathbf{T}) - e^h(\mathbf{X})\right) + \varepsilon$$