

# Causal Effect Estimation with Context and Confounders (Pt. 2)

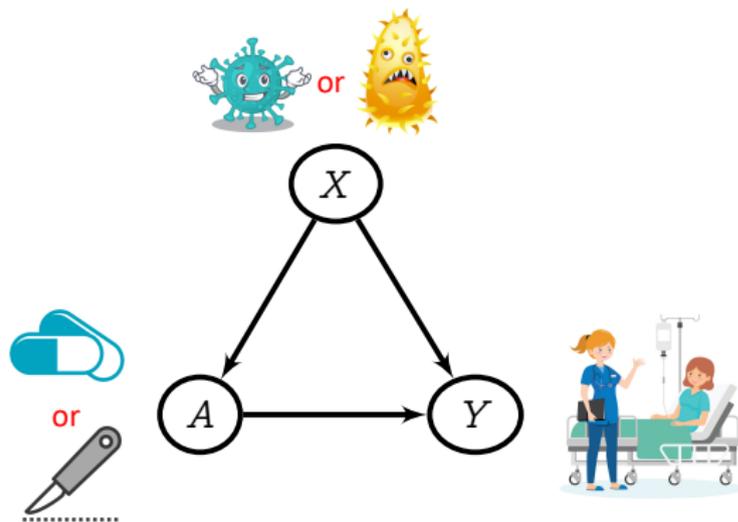
Arthur Gretton

Gatsby Computational Neuroscience Unit, UCL  
Google Deepmind

ML Winter School, MBZUAI 2026

## Observation vs intervention

Conditioning from observation:  $\mathbb{E}[Y|A = a] = \sum_x \mathbb{E}[Y|a, x]p(x|a)$

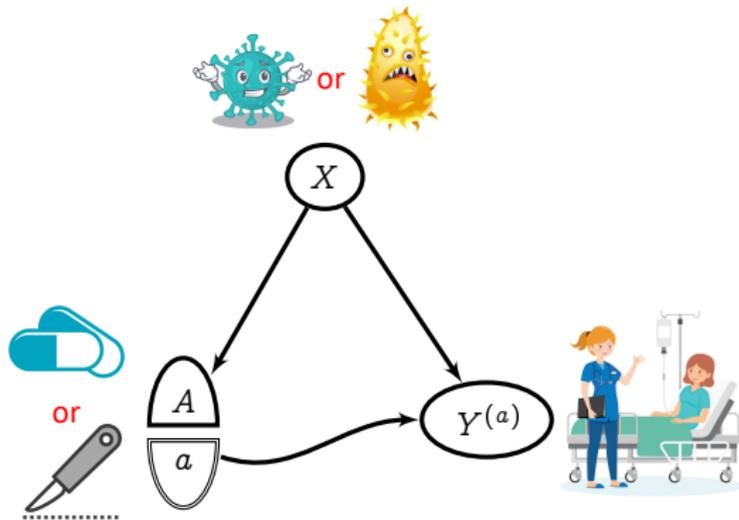


From our *observations* of historical hospital data:

- $P(Y = \text{cured}|A = \text{pills}) = 0.85$
- $P(Y = \text{cured}|A = \text{surgery}) = 0.72$

# Observation vs intervention

Dose-response curve (**intervention**):  $\mathbb{E}[Y^{(a)}] = \sum_x \mathbb{E}[Y|a, x]p(x)$

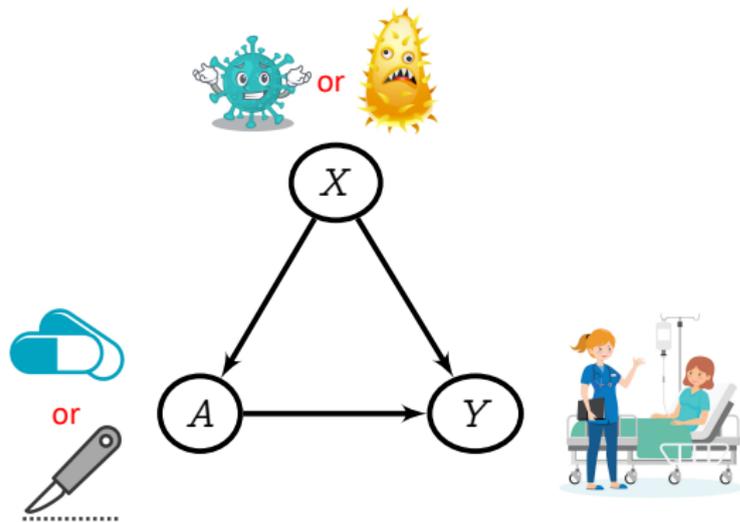


From our *intervention* (making all patients take a treatment):

- $P(Y^{(\text{pills})} = \text{cured}) = 0.64$
- $P(Y^{(\text{surgery})} = \text{cured}) = 0.75$

Richardson, Robins (2013), Single World Intervention Graphs (SWIGs): A Unification of the Counterfactual and Graphical Approaches to Causality

## Some core assumptions



### Assume:

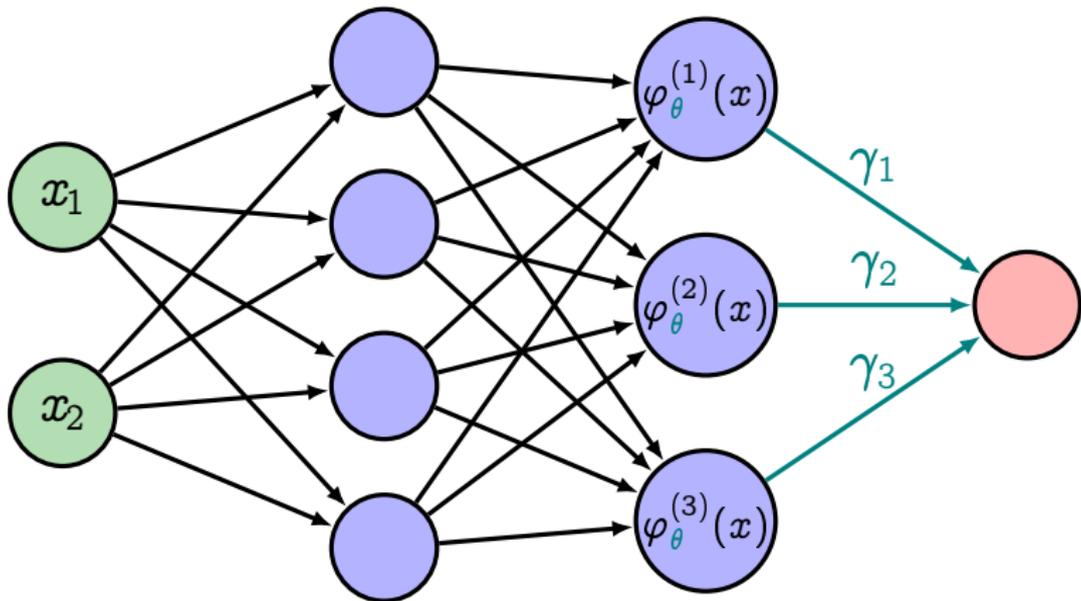
- Stable Unit Treatment Value Assumption (aka “no interference”),
- Conditional exchangeability  $Y^{(a)} \perp\!\!\!\perp A|X$ .
- Overlap.

## One model: linear functions of features

All learned functions will take the form:

$$\gamma(x) = \gamma^\top \varphi_\theta(x)$$

NN approach: **Finite** dictionaries of **learned** neural net features  $\varphi_\theta(x)$   
(linear final layer  $\gamma$ )



## Model fitting: *neural* ridge regression

Learn  $\gamma_0(x) := \mathbb{E}[Y|X = x]$  from **features**  $\varphi_\theta(x_i)$  with outcomes  $y_i$ :

$$\hat{\gamma}_\theta = \arg \min_{\gamma} \left( \sum_{i=1}^n (y_i - \gamma^\top \varphi_\theta(x_i))^2 + \lambda \|\gamma\|^2 \right) \quad (1)$$

## Model fitting: *neural* ridge regression

Learn  $\gamma_0(x) := \mathbb{E}[Y|X = x]$  from **features**  $\varphi_\theta(x_i)$  with outcomes  $y_i$ :

$$\hat{\gamma}_\theta = \arg \min_{\gamma} \left( \sum_{i=1}^n (y_i - \gamma^\top \varphi_\theta(x_i))^2 + \lambda \|\gamma\|^2 \right) \quad (1)$$

Solution for **linear final layer**  $\gamma$ :

$$\hat{\gamma}_\theta = C_{YX}^{(\theta)} (C_{XX}^{(\theta)} + \lambda)^{-1}$$

$$C_{YX}^{(\theta)} = \frac{1}{n} \sum_{i=1}^n [y_i \varphi_\theta(x_i)^\top]$$

$$C_{XX}^{(\theta)} = \frac{1}{n} \sum_{i=1}^n [\varphi_\theta(x_i) \varphi_\theta(x_i)^\top]$$

## Model fitting: *neural* ridge regression

Learn  $\gamma_0(x) := \mathbb{E}[Y|X = x]$  from **features**  $\varphi_\theta(x_i)$  with outcomes  $y_i$ :

$$\hat{\gamma}_\theta = \arg \min_{\gamma} \left( \sum_{i=1}^n (y_i - \gamma^\top \varphi_\theta(x_i))^2 + \lambda \|\gamma\|^2 \right) \quad (1)$$

Solution for **linear final layer**  $\gamma$ :

$$\hat{\gamma}_\theta = C_{YX}^{(\theta)} (C_{XX}^{(\theta)} + \lambda)^{-1}$$

$$C_{YX}^{(\theta)} = \frac{1}{n} \sum_{i=1}^n [y_i \varphi_\theta(x_i)^\top]$$

$$C_{XX}^{(\theta)} = \frac{1}{n} \sum_{i=1}^n [\varphi_\theta(x_i) \varphi_\theta(x_i)^\top]$$

**How to solve for  $\theta$ :**

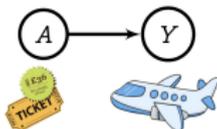
Substitute  $\hat{\gamma}_\theta$  into (1), backprop for remaining  $\theta$ .

More details: Galashov, Da Costa, Xu, Hennig, G, Closed-Form Last Layer Optimization (2025, arxiv:2510.04606)

# Instrumental variable regression

## Illustration: ticket prices for air travel

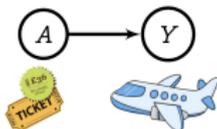
Ticket price  $A$ , seats sold  $Y$ .



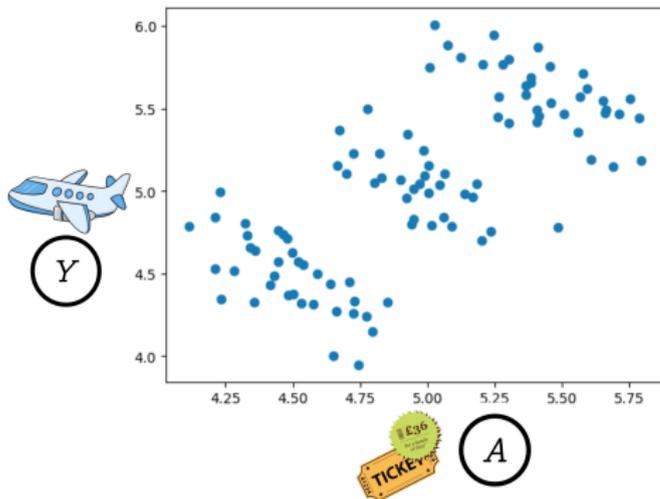
What is the effect on seats sold  $Y^{(a)}$  of intervening on price  $a$ ?

## Illustration: ticket prices for air travel

Ticket price  $A$ , seats sold  $Y$ .



What is the effect on seats sold  $Y^{(a)}$  of intervening on price  $a$ ?

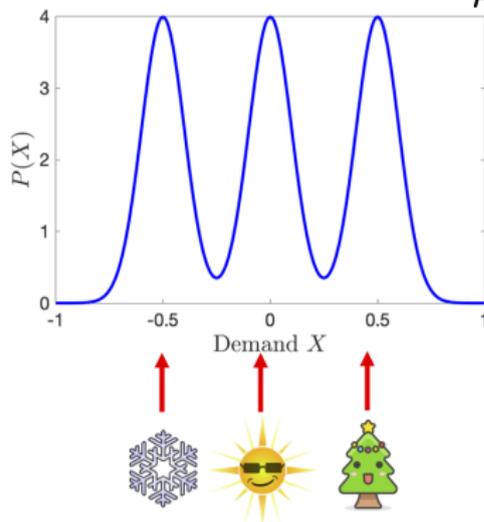
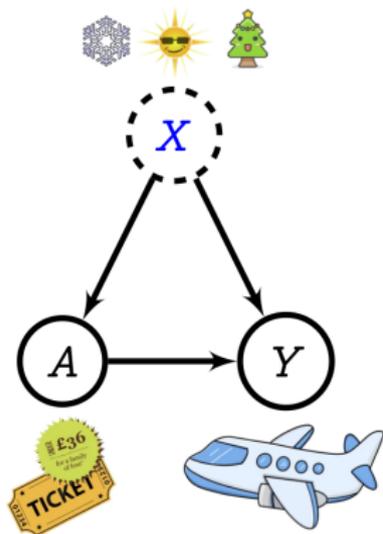


## Illustration: ticket prices for air travel

Unobserved variable  $X$  = **desire for travel**, affects *both* price (via airline algorithms) *and* seats sold.

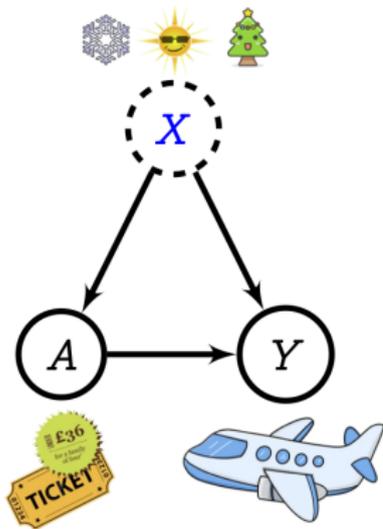
■ **Desire for travel:**

$$X \sim \mathcal{N}(\mu, 0.01)$$
$$\mu \sim \mathcal{U}\left\{-\frac{1}{2}, 0, \frac{1}{2}\right\}$$



## Illustration: ticket prices for air travel

Unobserved variable  $X$  = desire for travel, affects *both* price (via airline algorithms) *and* seats sold.



■ **Desire for travel:**

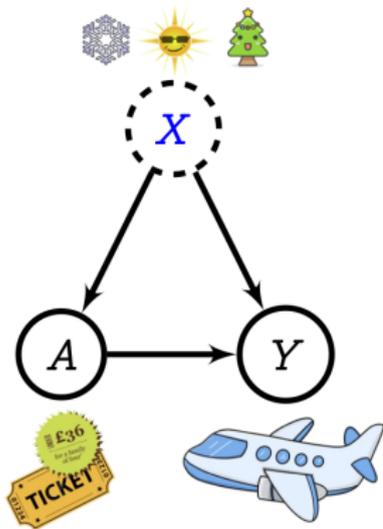
$$X \sim \mathcal{N}(\mu, 0.01)$$
$$\mu \sim \mathcal{U}\left\{-\frac{1}{2}, 0, \frac{1}{2}\right\}$$

■ **Price:**

$$A = X + Z,$$
$$Z \sim \mathcal{N}(5, 0.04)$$

## Illustration: ticket prices for air travel

Unobserved variable  $X$  = **desire for travel**, affects *both* price (via airline algorithms) *and* seats sold.



■ **Desire for travel:**

$$X \sim \mathcal{N}(\mu, 0.01)$$
$$\mu \sim \mathcal{U}\left\{-\frac{1}{2}, 0, \frac{1}{2}\right\}$$

■ **Price:**

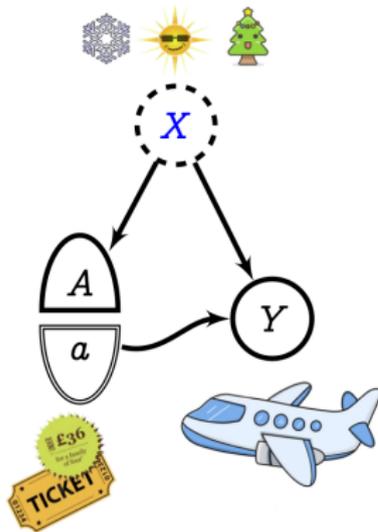
$$A = X + Z,$$
$$Z \sim \mathcal{N}(5, 0.04)$$

■ **Seats sold:**

$$Y = 10 - A + 2X$$

## Illustration: ticket prices for air travel

Unobserved variable  $X$  = desire for travel, affects *both* price (via airline algorithms) *and* seats sold.



- Desire for travel:

$$X \sim \mathcal{N}(\mu, 0.01)$$
$$\mu \sim \mathcal{U}\left\{-\frac{1}{2}, 0, \frac{1}{2}\right\}$$

- Price:

$$A = X + Z,$$
$$Z \sim \mathcal{N}(5, 0.04)$$

- Seats sold:

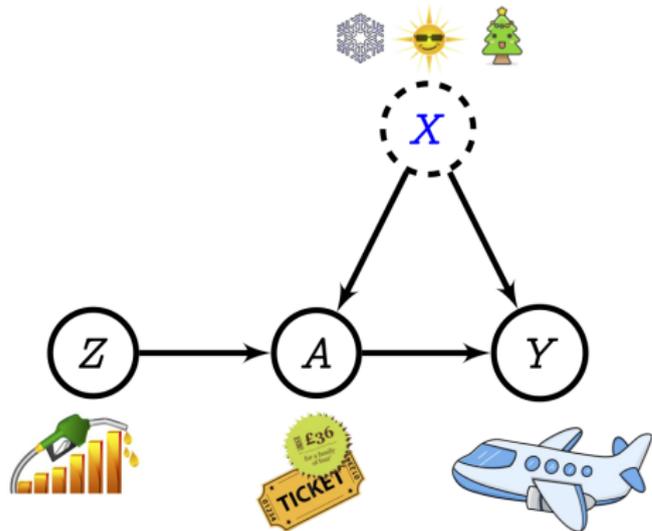
$$Y = 10 - A + 2X$$

Dose-response curve:

$$\text{DR}(a) = \mathbb{E}[Y^{(a)}] = \int (10 - a + 2X) dp(X) = 10 - a$$

## Illustration: ticket prices for air travel

Unobserved variable  $X$  = **desire for travel**, affects *both* price (via airline algorithms) *and* seats sold.



- **Desire for travel:**

$$X \sim \mathcal{N}(\mu, 0.01)$$
$$\mu \sim \mathcal{U}\left\{-\frac{1}{2}, 0, \frac{1}{2}\right\}$$

- **Price:**

$$A = X + Z,$$
$$Z \sim \mathcal{N}(5, 0.04)$$

- **Seats sold:**

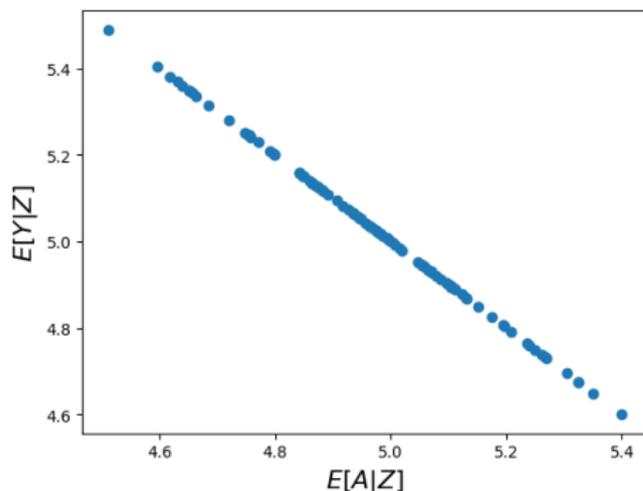
$$Y = 10 - A + 2X$$

$Z$  is an **instrument** (cost of fuel). Condition on  $Z$ ,

$$\mathbb{E}[Y|Z] = 10 - \mathbb{E}[A|Z] + \underbrace{2\mathbb{E}[X|Z]}_{=0}$$

## Illustration: ticket prices for air travel

Unobserved variable  $X$  = desire for travel, affects *both* price (via airline algorithms) *and* seats sold.



- Desire for travel:

$$X \sim \mathcal{N}(\mu, 0.01)$$
$$\mu \sim \mathcal{U}\left\{-\frac{1}{2}, 0, \frac{1}{2}\right\}$$

- Price:

$$A = X + Z,$$
$$Z \sim \mathcal{N}(5, 0.04)$$

- Seats sold:

$$Y = 10 - A + 2X$$

$Z$  is an instrument (cost of fuel). Condition on  $Z$ ,

$$\mathbb{E}[Y|Z] = 10 - \mathbb{E}[A|Z] + \underbrace{2\mathbb{E}[X|Z]}_{=0}$$

Regressing from  $\mathbb{E}[A|Z]$  to  $\mathbb{E}[Y|Z]$  recovers causal relation!

# Instrumental variable regression

## The Sveriges Riksbank Prize in Economic Sciences in Memory of Alfred Nobel 2021



© Nobel Prize Outreach. Photo: Paul Kennedy

**David Card**

Prize share: 1/2



© Nobel Prize Outreach. Photo: Risdon Photography

**Joshua D. Angrist**

Prize share: 1/4



© Nobel Prize Outreach. Photo: Paul Kennedy

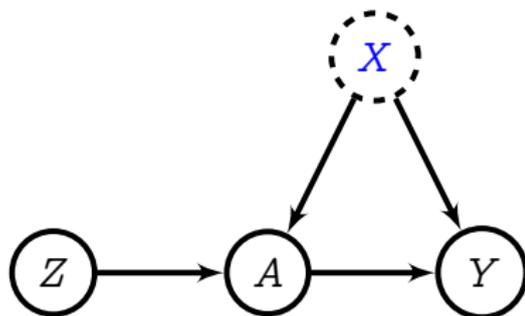
**Guido W. Imbens**

Prize share: 1/4

The Sveriges Riksbank Prize in Economic Sciences in Memory of Alfred Nobel 2021 was divided, one half awarded to David Card "for his empirical contributions to labour economics", the other half jointly to Joshua D. Angrist and Guido W. Imbens "for their methodological contributions to the analysis of causal relationships"

# Instrumental variable regression with NN features

- $X$ : unobserved confounder.
- $A$ : treatment
- $Y$ : outcome
- $Z$ : instrument



## Assumptions

$$\mathbb{E}[X|Z] = 0$$

$$Z \not\perp A$$

$$(Y \perp Z|A)_{G_{\bar{A}}}$$

$$Y = \gamma^\top \phi_\theta(A) + X$$

# Instrumental variable regression with NN features

- $X$ : unobserved confounder.
- $A$ : treatment
- $Y$ : outcome
- $Z$ : instrument

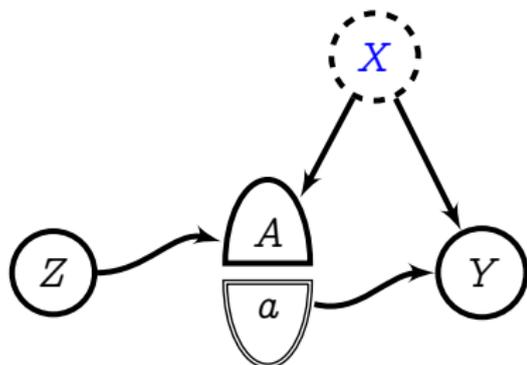
## Assumptions

$$\mathbb{E}[X|Z] = 0$$

$$Z \not\perp A$$

$$(Y \perp Z|A)_{G_{\bar{A}}}$$

$$Y = \gamma^\top \phi_\theta(A) + X$$

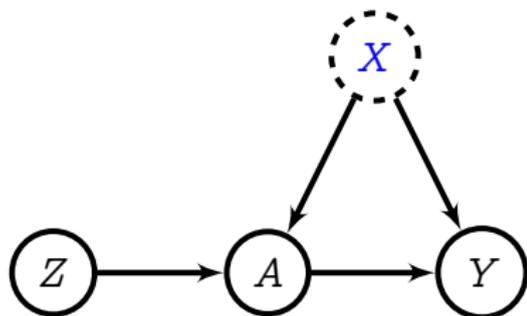


Dose-response curve:

$$\text{DR}(a) = \int \mathbb{E}(Y|X, a) dp(X) = \gamma^\top \phi_\theta(a)$$

# Instrumental variable regression with NN features

- $X$ : unobserved confounder.
- $A$ : treatment
- $Y$ : outcome
- $Z$ : instrument



## Assumptions

$$\mathbb{E}[X|Z] = 0$$

$$Z \not\perp A$$

$$(Y \perp Z|A)_{G_{\bar{A}}}$$

$$Y = \gamma^\top \phi_\theta(A) + X$$

IV regression: Condition both sides on  $Z$ ,

$$\mathbb{E}[Y|Z] = \gamma^\top \mathbb{E}[\phi_\theta(A)|Z] + \underbrace{\mathbb{E}[X|Z]}_{=0}$$

## Plain linear regression: what goes wrong?

Output  $y \in \mathbb{R}$ , noise  $X \in \mathbb{R}$ , input  $A$  with NN features  $\varphi_\theta(a)$ .

Crucially,  $X \not\perp A$  and

$$C_{ax} := \mathbb{E}[\varphi_\theta(A)X] \neq 0$$

## Plain linear regression: what goes wrong?

Output  $y \in \mathbb{R}$ , noise  $X \in \mathbb{R}$ , input  $A$  with NN features  $\varphi_\theta(a)$ .

Crucially,  $X \not\perp A$  and

$$C_{ax} := \mathbb{E}[\varphi_\theta(A)X] \neq 0$$

Dose-response curve:

$$y = \gamma_0^\top \varphi_\theta(a) + X \quad \mathbb{E}(X) = 0$$

$$\text{DR} := \mathbb{E}(Y^{(a)}) = \int (\gamma_0^\top \varphi_\theta(a) + X) dP(X) = \gamma_0^\top \varphi_\theta(a).$$

## Plain linear regression: what goes wrong?

Output  $y \in \mathbb{R}$ , noise  $X \in \mathbb{R}$ , input  $A$  with NN features  $\varphi_\theta(a)$ .

Crucially,  $X \not\perp A$  and

$$C_{ax} := \mathbb{E}[\varphi_\theta(A)X] \neq 0$$

Dose-response curve:

$$y = \gamma_0^\top \varphi_\theta(a) + X \quad \mathbb{E}(X) = 0$$

$$\text{DR} := \mathbb{E}(Y^{(a)}) = \int (\gamma_0^\top \varphi_\theta(a) + X) dP(X) = \gamma_0^\top \varphi_\theta(a).$$

**Problem:** Least-squares loss for  $\gamma$ :

$$\mathcal{L}(\gamma, \theta) = \mathbb{E} \left\| Y - \gamma^\top \varphi_\theta(A) - X \right\|^2$$

## Plain linear regression: what goes wrong?

Output  $y \in \mathbb{R}$ , noise  $X \in \mathbb{R}$ , input  $A$  with NN features  $\varphi_\theta(a)$ .

Crucially,  $X \not\perp A$  and

$$C_{ax} := \mathbb{E}[\varphi_\theta(A)X] \neq 0$$

Dose-response curve:

$$y = \gamma_0^\top \varphi_\theta(a) + X \quad \mathbb{E}(X) = 0$$

$$\text{DR} := \mathbb{E}(Y^{(a)}) = \int (\gamma_0^\top \varphi_\theta(a) + X) dP(X) = \gamma_0^\top \varphi_\theta(a).$$

**Problem:** Least-squares loss for  $\gamma$ :

$$\mathcal{L}(\gamma, \theta) = \mathbb{E} \left\| Y - \gamma^\top \varphi_\theta(A) - X \right\|^2$$

Minimizing for  $\gamma$ ,

$$\begin{aligned} \gamma_0 &= C_{aa}^{-1} (C_{ay} - C_{ax}) & C_{aa} &= \mathbb{E}[\varphi_\theta(A)\varphi_\theta(A)^\top] \\ & & C_{ay} &= \mathbb{E}[\varphi_\theta(A)Y] \end{aligned}$$

...but we don't have  $C_{ax}$ .

# Two-stage least squares for IV regression

Kernel features (NeurIPS 2019):

arXiv.org > cs > arXiv:1906.00232  [Help](#) | [Ad](#)

**Computer Science > Machine Learning**

*[Submitted on 1 Jun 2019 (v1), last revised 15 Jul 2020 (this version, v6)]*

**Kernel Instrumental Variable Regression**

Rahul Singh, Maneesh Sahani, Arthur Gretton



NN features (ICLR 2021):

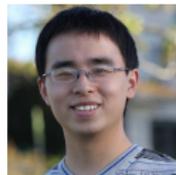
arXiv > cs > arXiv:2010.07154  [Help](#) | [Ad](#)

**Computer Science > Machine Learning**

*[Submitted on 14 Oct 2020 (v1), last revised 1 Nov 2020 (this version, v3)]*

**Learning Deep Features in Instrumental Variable Regression**

Liyuan Xu, Yutian Chen, Siddarth Srinivasan, Nando de Freitas, Arnaud Doucet, Arthur Gretton



Code for NN and kernel IV methods:

<https://github.com/liyuan9988/DeepFeatureIV/>

# Two-stage least squares for IV regression

## Kernel features (NeurIPS 2019):

arXiv.org > cs > arXiv:1906.00232

Search...  
Help | Ad

Computer Science > Machine Learning

[Submitted on 1 Jun 2019 (v1), last revised 15 Jul 2020 (this version, v6)]

### Kernel Instrumental Variable Regression

Rahul Singh, Maneesh Sahani, Arthur Gretton



## NN features (ICLR 2021):

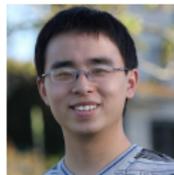
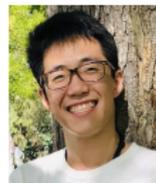
arXiv > cs > arXiv:2010.07154

Computer Science > Machine Learning

[Submitted on 14 Oct 2020 (v1), last revised 1 Nov 2020 (this version, v3)]

### Learning Deep Features in Instrumental Variable Regression

Liyuan Xu, Yutian Chen, Siddarth Srinivasan, Nando de Freitas, Arnaud Doucet, Arthur Gretton



Code for NN and kernel IV methods:

<https://github.com/liyuan9988/DeepFeatureIV/>

## IV using neural net features

Stage 2 regression (IV): learn NN features  $\phi_\theta(A)$  and linear layer  $\gamma$  to obtain  $Y$  with RR loss:

$$\mathbb{E}_{YZ} \left[ (Y - \gamma^\top \mathbb{E}[\phi_\theta(A)|Z])^2 \right] + \lambda_2 \|\gamma\|^2$$

## IV using neural net features

Stage 2 regression (IV): learn NN features  $\phi_\theta(A)$  and linear layer  $\gamma$  to obtain  $Y$  with RR loss:

$$\mathbb{E}_{YZ} \left[ (Y - \gamma^\top \mathbb{E}[\phi_\theta(A)|Z])^2 \right] + \lambda_2 \|\gamma\|^2$$

Stage 1 regression: learn NN features  $\phi_\zeta(Z)$  and linear layer  $F$ :

$$\mathbb{E}[\phi_\theta(A)|Z] \approx F \phi_\zeta(Z)$$

with RR loss

$$\mathbb{E} \|\phi_\theta(A) - F \phi_\zeta(Z)\|^2 + \lambda_1 \|F\|_{HS}^2$$

## IV using neural net features

Stage 2 regression (IV): learn NN features  $\phi_\theta(A)$  and linear layer  $\gamma$  to obtain  $Y$  with RR loss:

$$\mathbb{E}_{YZ} \left[ (Y - \gamma^\top \mathbb{E}[\phi_\theta(A)|Z])^2 \right] + \lambda_2 \|\gamma\|^2$$

Stage 1 regression: learn NN features  $\phi_\zeta(Z)$  and linear layer  $F$ :

$$\mathbb{E}[\phi_\theta(A)|Z] \approx F \phi_\zeta(Z)$$

with RR loss

$$\mathbb{E} \|\phi_\theta(A) - F \phi_\zeta(Z)\|^2 + \lambda_1 \|F\|_{HS}^2$$

Challenge: how to learn  $\theta$ ?

## IV using neural net features

Stage 2 regression (IV): learn NN features  $\phi_\theta(A)$  and linear layer  $\gamma$  to obtain  $Y$  with RR loss:

$$\mathbb{E}_{YZ} \left[ (Y - \gamma^\top \mathbb{E}[\phi_\theta(A)|Z])^2 \right] + \lambda_2 \|\gamma\|^2$$

Stage 1 regression: learn NN features  $\phi_\zeta(Z)$  and linear layer  $F$ :

$$\mathbb{E}[\phi_\theta(A)|Z] \approx F \phi_\zeta(Z)$$

with RR loss

$$\mathbb{E} \|\phi_\theta(A) - F \phi_\zeta(Z)\|^2 + \lambda_1 \|F\|_{HS}^2$$

Challenge: how to learn  $\theta$ ?

From Stage 2 regression?

## IV using neural net features

Stage 2 regression (IV): learn NN features  $\phi_\theta(A)$  and linear layer  $\gamma$  to obtain  $Y$  with RR loss:

$$\mathbb{E}_{YZ} \left[ (Y - \gamma^\top \mathbb{E}[\phi_\theta(A)|Z])^2 \right] + \lambda_2 \|\gamma\|^2$$

Stage 1 regression: learn NN features  $\phi_\zeta(Z)$  and linear layer  $F$ :

$$\mathbb{E}[\phi_\theta(A)|Z] \approx F \phi_\zeta(Z)$$

with RR loss

$$\mathbb{E} \|\phi_\theta(A) - F \phi_\zeta(Z)\|^2 + \lambda_1 \|F\|_{HS}^2$$

Challenge: how to learn  $\theta$ ?

From Stage 2 regression?

...which requires  $\mathbb{E}[\phi_\theta(A)|Z]$  from Stage 1 regression

## IV using neural net features

Stage 2 regression (IV): learn NN features  $\phi_\theta(A)$  and linear layer  $\gamma$  to obtain  $Y$  with RR loss:

$$\mathbb{E}_{YZ} \left[ (Y - \gamma^\top \mathbb{E}[\phi_\theta(A)|Z])^2 \right] + \lambda_2 \|\gamma\|^2$$

Stage 1 regression: learn NN features  $\phi_\zeta(Z)$  and linear layer  $F$ :

$$\mathbb{E}[\phi_\theta(A)|Z] \approx F \phi_\zeta(Z)$$

with RR loss

$$\mathbb{E} \|\phi_\theta(A) - F \phi_\zeta(Z)\|^2 + \lambda_1 \|F\|_{HS}^2$$

Challenge: how to learn  $\theta$ ?

From Stage 2 regression?

...which requires  $\mathbb{E}[\phi_\theta(A)|Z]$  from Stage 1 regression

...which requires  $\phi_\theta(A)$ ... which requires  $\theta$ ...

## IV using neural net features

Stage 2 regression (IV): learn NN features  $\phi_\theta(A)$  and linear layer  $\gamma$  to obtain  $Y$  with RR loss:

$$\mathbb{E}_{YZ} \left[ (Y - \gamma^\top \mathbb{E}[\phi_\theta(A)|Z])^2 \right] + \lambda_2 \|\gamma\|^2$$

Stage 1 regression: learn NN features  $\phi_\zeta(Z)$  and linear layer  $F$ :

$$\mathbb{E}[\phi_\theta(A)|Z] \approx F \phi_\zeta(Z)$$

with RR loss

$$\mathbb{E} \|\phi_\theta(A) - F \phi_\zeta(Z)\|^2 + \lambda_1 \|F\|_{HS}^2$$

Challenge: how to learn  $\theta$ ?

From Stage 2 regression?

...which requires  $\mathbb{E}[\phi_\theta(A)|Z]$  from Stage 1 regression

...which requires  $\phi_\theta(A)$ ... which requires  $\theta$ ...

**Use the linear final layers!** (i.e.  $\gamma$  and  $F$ )

## IV using neural net features

Stage 1 regression: learn NN features  $\phi_\zeta(Z)$  and linear layer  $F$ :

$$\mathbb{E}[\phi_\theta(A)|Z] \approx F\phi_\zeta(Z)$$

with RR loss

$$\mathbb{E} \left[ \|\phi_\theta(A) - F\phi_\zeta(Z)\|^2 \right] + \lambda_1 \|F\|_{HS}^2$$

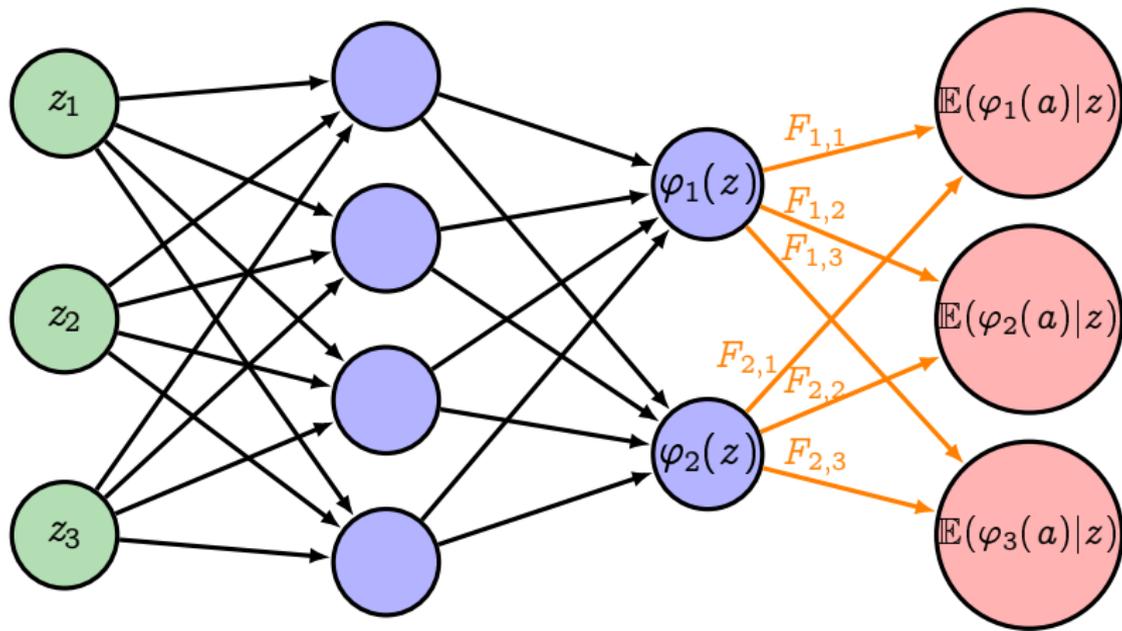
## IV using neural net features

Stage 1 regression: learn NN features  $\phi_\zeta(Z)$  and linear layer  $F$ :

$$\mathbb{E}[\phi_\theta(A)|Z] \approx F\phi_\zeta(Z)$$

with RR loss

$$\mathbb{E} \left[ \|\phi_\theta(A) - F\phi_\zeta(Z)\|^2 \right] + \lambda_1 \|F\|_{HS}^2$$



## IV using neural net features

Stage 1 regression: learn NN features  $\phi_\zeta(Z)$  and linear layer  $F$ :

$$\mathbb{E}[\phi_\theta(A)|Z] \approx F\phi_\zeta(Z)$$

with RR loss

$$\mathbb{E}[\|\phi_\theta(A) - F\phi_\zeta(Z)\|^2] + \lambda_1 \|F\|_{HS}^2$$

$\hat{F}_{\theta,\zeta}$  in closed form wrt  $\phi_\theta, \phi_\zeta$ :

$$\hat{F}_{\theta,\zeta} = C_{AZ}(C_{ZZ} + \lambda_1 I)^{-1} \quad C_{AZ} = \mathbb{E}[\phi_\theta(A)\phi_\zeta^\top(Z)]$$
$$C_{ZZ} = \mathbb{E}[\phi_\zeta(Z)\phi_\zeta^\top(Z)]$$

## IV using neural net features

Stage 1 regression: learn NN features  $\phi_\zeta(Z)$  and linear layer  $F$ :

$$\mathbb{E}[\phi_\theta(A)|Z] \approx F\phi_\zeta(Z)$$

with RR loss

$$\mathbb{E} \left[ \|\phi_\theta(A) - F\phi_\zeta(Z)\|^2 \right] + \lambda_1 \|F\|_{HS}^2$$

$\hat{F}_{\theta,\zeta}$  in closed form wrt  $\phi_\theta, \phi_\zeta$ :

$$\hat{F}_{\theta,\zeta} = C_{AZ}(C_{ZZ} + \lambda_1 I)^{-1} \quad C_{AZ} = \mathbb{E}[\phi_\theta(A)\phi_\zeta^\top(Z)]$$
$$C_{ZZ} = \mathbb{E}[\phi_\zeta(Z)\phi_\zeta^\top(Z)]$$

Plug  $\hat{F}_{\theta,\zeta}$  into S1 loss, take gradient steps for  $\zeta$  (...but not  $\theta$ ...)

## Stage 2: IV regression

Stage 2 regression (IV): learn NN features  $\phi_\theta(A)$  and linear layer  $\gamma$  to obtain  $Y$  with RR loss:

$$\mathcal{L}_2(\gamma, \theta) = \mathbb{E}_{YZ} \left[ (Y - \gamma^\top \mathbb{E}[\phi_\theta(A)|Z])^2 \right] + \lambda_2 \|\gamma\|^2$$

## Stage 2: IV regression

Stage 2 regression (IV): learn NN features  $\phi_\theta(A)$  and linear layer  $\gamma$  to obtain  $Y$  with RR loss:

$$\begin{aligned}\mathcal{L}_2(\gamma, \theta) &= \mathbb{E}_{YZ} \left[ (Y - \gamma^\top \mathbb{E}[\phi_\theta(A)|Z])^2 \right] + \lambda_2 \|\gamma\|^2 \\ &= \mathbb{E}_{YZ} \left[ (Y - \gamma^\top \underbrace{\hat{F}_{\theta, \zeta} \phi_\zeta(Z)}_{\text{Stage 1}})^2 \right] + \lambda_2 \|\gamma\|^2\end{aligned}$$

## Stage 2: IV regression

Stage 2 regression (IV): learn NN features  $\phi_\theta(A)$  and linear layer  $\gamma$  to obtain  $Y$  with RR loss:

$$\begin{aligned}\mathcal{L}_2(\gamma, \theta) &= \mathbb{E}_{YZ} \left[ (Y - \gamma^\top \mathbb{E}[\phi_\theta(A)|Z])^2 \right] + \lambda_2 \|\gamma\|^2 \\ &= \mathbb{E}_{YZ} \left[ (Y - \gamma^\top \hat{F}_{\theta, \zeta} \phi_\zeta(Z))^2 \right] + \lambda_2 \|\gamma\|^2\end{aligned}$$

$\hat{\gamma}_\theta$  in closed form wrt  $\phi_\theta$ :

$$\begin{aligned}\hat{\gamma}_\theta &:= \tilde{\mathcal{C}}_{YA|Z} (\tilde{\mathcal{C}}_{AA|Z} + \lambda_2 I)^{-1} & \tilde{\mathcal{C}}_{YA|Z} &= \mathbb{E} \left[ Y [\hat{F}_{\theta, \zeta} \phi_\zeta(Z)]^\top \right] \\ & & \tilde{\mathcal{C}}_{AA|Z} &= \mathbb{E} \left[ [\hat{F}_{\theta, \zeta} \phi_\zeta(Z)] [\hat{F}_{\theta, \zeta} \phi_\zeta(Z)]^\top \right]\end{aligned}$$

## Stage 2: IV regression

Stage 2 regression (IV): learn NN features  $\phi_\theta(A)$  and linear layer  $\gamma$  to obtain  $Y$  with RR loss:

$$\begin{aligned}\mathcal{L}_2(\gamma, \theta) &= \mathbb{E}_{YZ} \left[ (Y - \gamma^\top \mathbb{E}[\phi_\theta(A)|Z])^2 \right] + \lambda_2 \|\gamma\|^2 \\ &= \mathbb{E}_{YZ} \left[ (Y - \gamma^\top \hat{F}_{\theta, \zeta} \phi_\zeta(Z))^2 \right] + \lambda_2 \|\gamma\|^2\end{aligned}$$

$\hat{\gamma}_\theta$  in closed form wrt  $\phi_\theta$ :

$$\begin{aligned}\hat{\gamma}_\theta &:= \tilde{\mathcal{C}}_{YA|Z} (\tilde{\mathcal{C}}_{AA|Z} + \lambda_2 I)^{-1} & \tilde{\mathcal{C}}_{YA|Z} &= \mathbb{E} \left[ Y [\hat{F}_{\theta, \zeta} \phi_\zeta(Z)]^\top \right] \\ & & \tilde{\mathcal{C}}_{AA|Z} &= \mathbb{E} \left[ [\hat{F}_{\theta, \zeta} \phi_\zeta(Z)] [\hat{F}_{\theta, \zeta} \phi_\zeta(Z)]^\top \right]\end{aligned}$$

**From linear final layers in Stages 1,2:**

Learn  $\phi_\theta(A)$  by plugging  $\hat{\gamma}_\theta$  into S2 loss, taking gradient steps for  $\theta$

## Stage 2: IV regression

Stage 2 regression (IV): learn NN features  $\phi_\theta(A)$  and linear layer  $\gamma$  to obtain  $Y$  with RR loss:

$$\begin{aligned}\mathcal{L}_2(\gamma, \theta) &= \mathbb{E}_{YZ} \left[ (Y - \gamma^\top \mathbb{E}[\phi_\theta(A)|Z])^2 \right] + \lambda_2 \|\gamma\|^2 \\ &= \mathbb{E}_{YZ} \left[ (Y - \gamma^\top \hat{F}_{\theta, \zeta} \phi_\zeta(Z))^2 \right] + \lambda_2 \|\gamma\|^2\end{aligned}$$

$\hat{\gamma}_\theta$  in closed form wrt  $\phi_\theta$ :

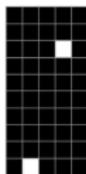
$$\begin{aligned}\hat{\gamma}_\theta &:= \tilde{\mathcal{C}}_{YA|Z} (\tilde{\mathcal{C}}_{AA|Z} + \lambda_2 I)^{-1} & \tilde{\mathcal{C}}_{YA|Z} &= \mathbb{E} \left[ Y [\hat{F}_{\theta, \zeta} \phi_\zeta(Z)]^\top \right] \\ & & \tilde{\mathcal{C}}_{AA|Z} &= \mathbb{E} \left[ [\hat{F}_{\theta, \zeta} \phi_\zeta(Z)] [\hat{F}_{\theta, \zeta} \phi_\zeta(Z)]^\top \right]\end{aligned}$$

**From linear final layers in Stages 1,2:**

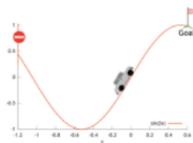
Learn  $\phi_\theta(A)$  by plugging  $\hat{\gamma}_\theta$  into S2 loss, taking gradient steps for  $\theta$   
...but  $\zeta$  changes with  $\theta$

...so **alternate first and second stages** until convergence.

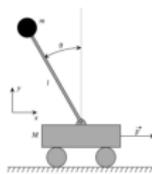
# Neural IV in reinforcement learning



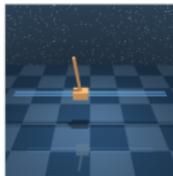
(a) Catch



(b) Mountain Car



(c) Cartpole



(a) Cartpole Swingup



(b) Cheetah Run



(c) Humanoid Run



(d) Walker Walk

Policy evaluation: want Q-value:

$$Q^\pi(s, a) = \mathbb{E} \left[ \sum_{t=0}^{\infty} \gamma^t R_t \mid S_0 = s, A_0 = a \right]$$

for policy  $\pi(A|S = s)$ .

Osband et al (2019). Behaviour suite for reinforcement learning. <https://github.com/deepmind/bsuite>

Tassa et al. (2020). dm\_control: Software and tasks for continuous control.

[https://github.com/deepmind/dm\\_control](https://github.com/deepmind/dm_control)

## Application of IV: reinforcement learning

Q value is a minimizer of Bellman loss

$$\mathcal{L}_{\text{Bellman}} = \mathbb{E}_{SAR} \left[ (R + \gamma \mathbb{E} [Q^\pi(S', A') | S, A] - Q^\pi(S, A))^2 \right].$$

Corresponds to “IV-like” problem

$$\mathcal{L}_{\text{Bellman}} = \mathbb{E}_{YZ} \left[ (Y - \mathbb{E}[f(X) | Z])^2 \right]$$

with

$$Y = R,$$

$$X = (S', A', S, A)$$

$$Z = (S, A),$$

$$f_0(X) = Q^\pi(s, a) - \gamma Q^\pi(s', a')$$

RL experiments and data:

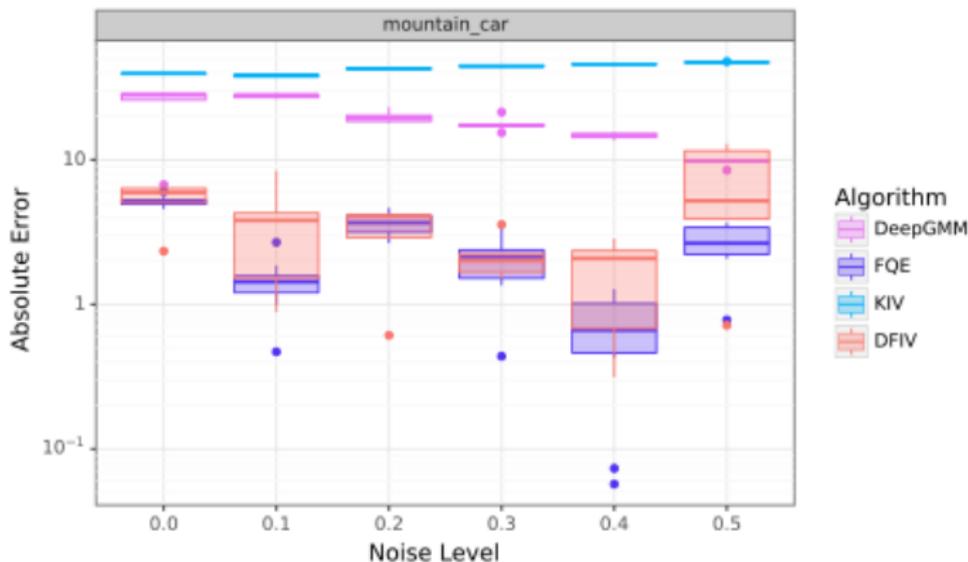
<https://github.com/liyuan9988/IVOPEwithACME>

Bradtke and Barto (1996). Linear least-squares algorithms for temporal difference learning.

Xu, Chen, Srinivasan, De Freitas, Doucet, G. (2021)

Chen, Xu, Gulcehre, Le Paine, G, De Freitas, Doucet (2022). On Instrumental Variable Regression in Deep Offline Policy Evaluation. 18743

# Results on mountain car problem



Good performance compared with FQE.

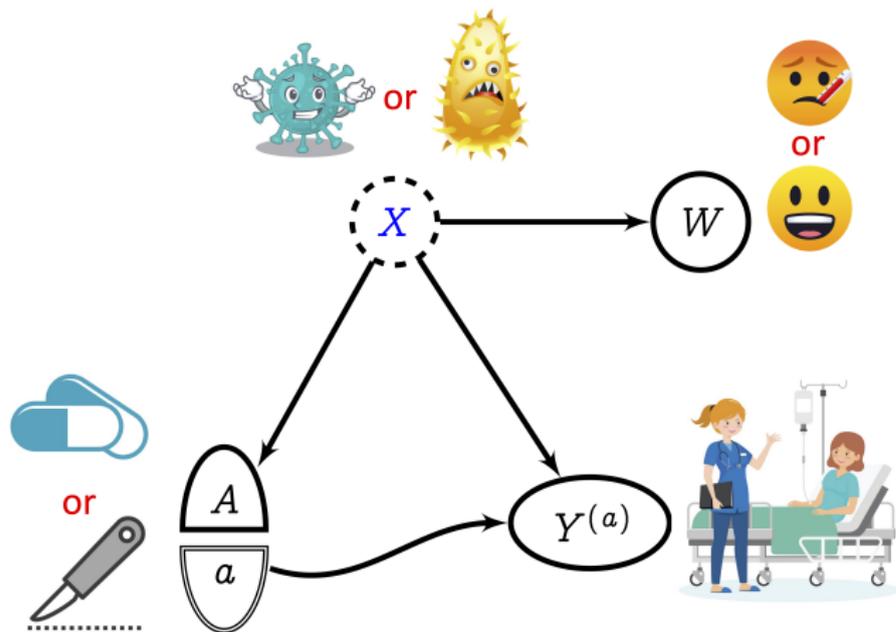
**Warning:** IV assumption can fail when regression underfits. See papers for details.

Xu, Chen, Srinivasan, De Freitas, Doucet, G. (2021)

Chen, Xu, Gulcehre, Le Paine, G, De Freitas, Doucet (2022). On Instrumental Variable Regression for Deep Offline Policy Evaluation.

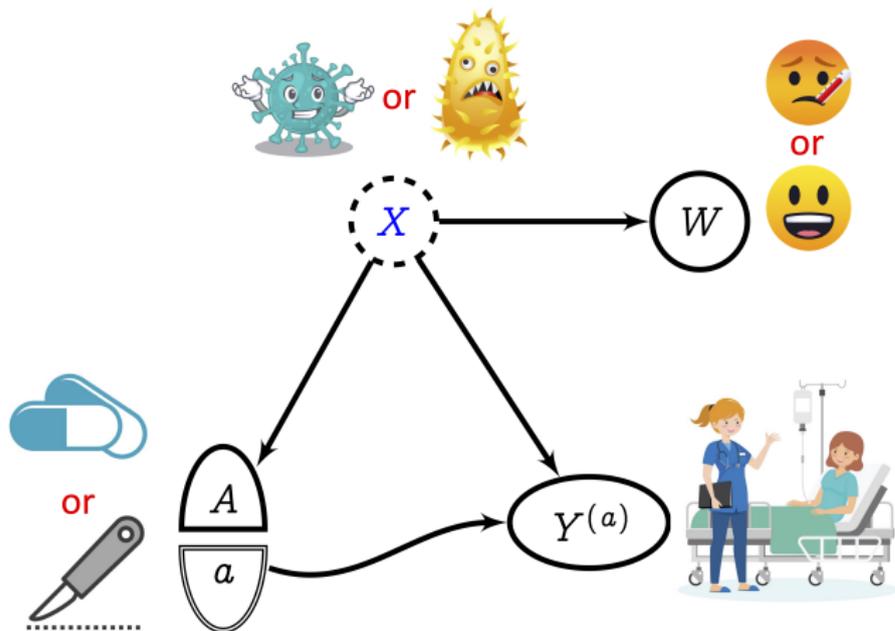
...but seriously, what if there are hidden confounders?

## We record symptom $W$ , not disease $X$



- $P(W = \text{fever} | X = \text{mild}) = 0.2$
- $P(W = \text{fever} | X = \text{severe}) = 0.8$

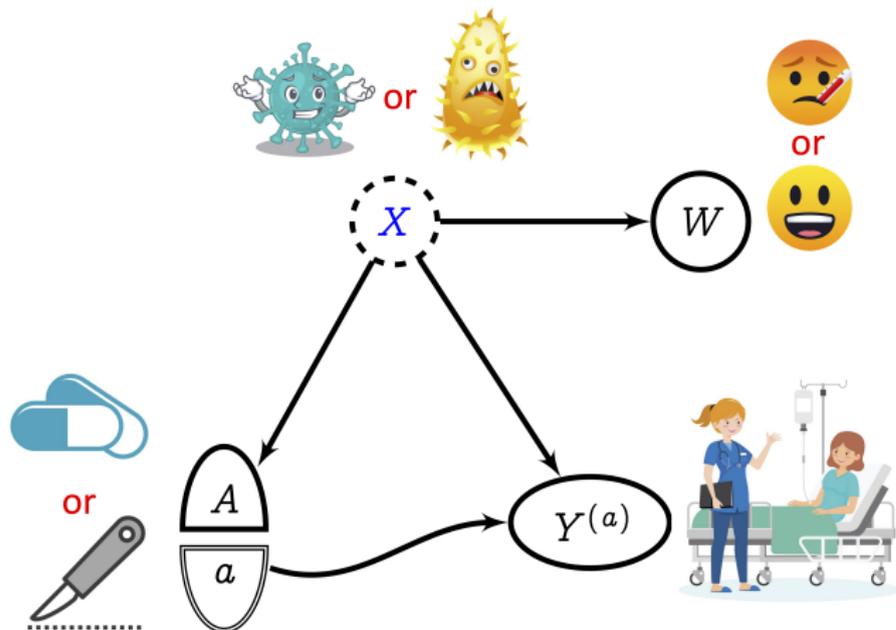
## We record symptom $W$ , not disease $X$



- $P(W = \text{fever} | X = \text{mild}) = 0.2$
- $P(W = \text{fever} | X = \text{severe}) = 0.8$

Could we just write:  $P(Y^{(a)}) \stackrel{?}{=} \sum_{w \in \{0,1\}} \mathbb{E}[Y | a, w] p(w)$

## We record symptom $W$ , not disease $X$



Wrong recommendation made:

- $\sum_{w \in \{0,1\}} \mathbb{E}[\text{cured} | \text{pills}, w] p(w) = 0.8 \quad (\neq 0.64)$
- $\sum_{w \in \{0,1\}} \mathbb{E}[\text{cured} | \text{surgery}, w] p(w) = 0.73 \quad (\neq 0.75)$

Correct answer **impossible** without observing  $X$

## Proxy causal learning (negative controls)

Causal effect estimation, with hidden covariates  $X$ :

- Use proxy variables (negative controls)

Applications: effect of actions under

- privacy constraints (email, ads, DMA)
- data gathering constraints (edge computing)
- fundamental limitations (preferences, state of mind)

## Proxy causal learning (negative controls)

Causal effect estimation, with hidden covariates  $X$ :

- Use proxy variables (negative controls)

Applications: effect of actions under

- privacy constraints (email, ads, DMA)
- data gathering constraints (edge computing)
- fundamental limitations (preferences, state of mind)

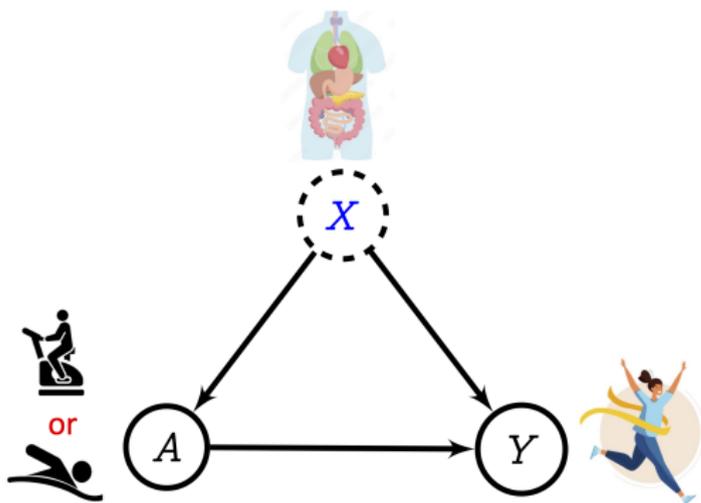
Don't ~~meet your heroes~~ model your hidden variables!

# What are proxies, and when are they useful?

Unobserved  $X$  with (possibly) complex nonlinear effects on  $A$ ,  $Y$

In this example:

- $X$ : true physical status
- $A$ : exercise regimes
- $Y$ : fitness goal

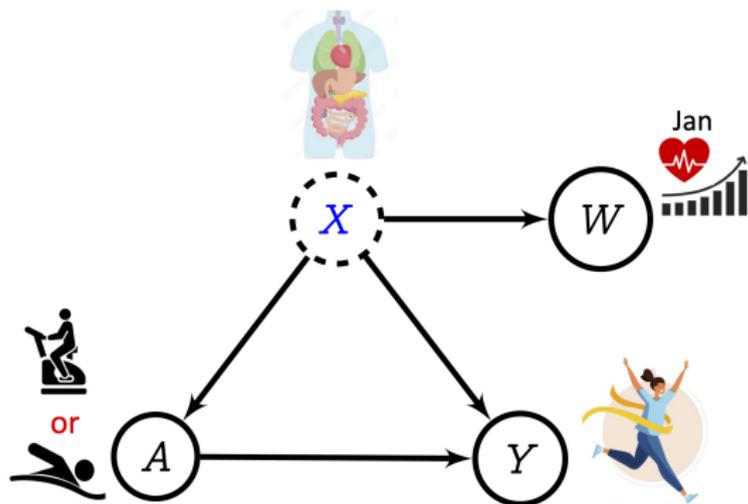


# What are proxies, and when are they useful?

Unobserved  $X$  with (possibly) complex nonlinear effects on  $A$ ,  $Y$

In this example:

- $X$ : true physical status
- $A$ : exercise regimes
- $Y$ : fitness goal
- $W$ : health readings before  $A$

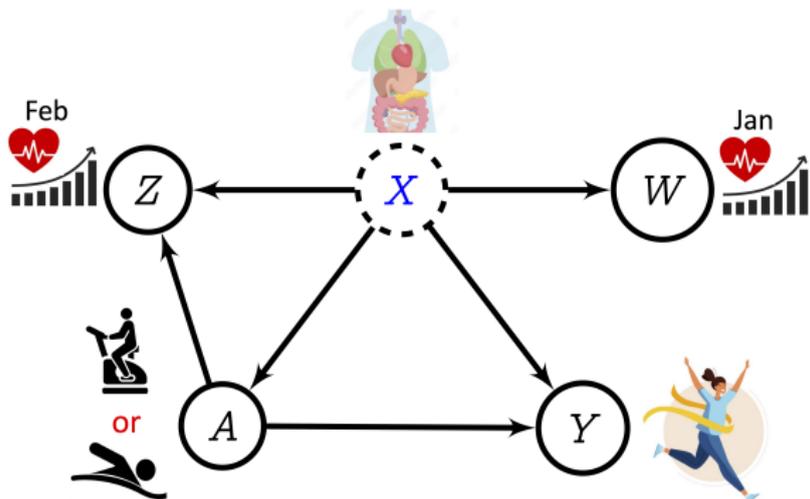


# What are proxies, and when are they useful?

Unobserved  $X$  with (possibly) complex nonlinear effects on  $A$ ,  $Y$

In this example:

- $X$ : true physical status
- $A$ : exercise regimes
- $Y$ : fitness goal
- $W$ : health readings before  $A$
- $Z$ : health readings after  $A$

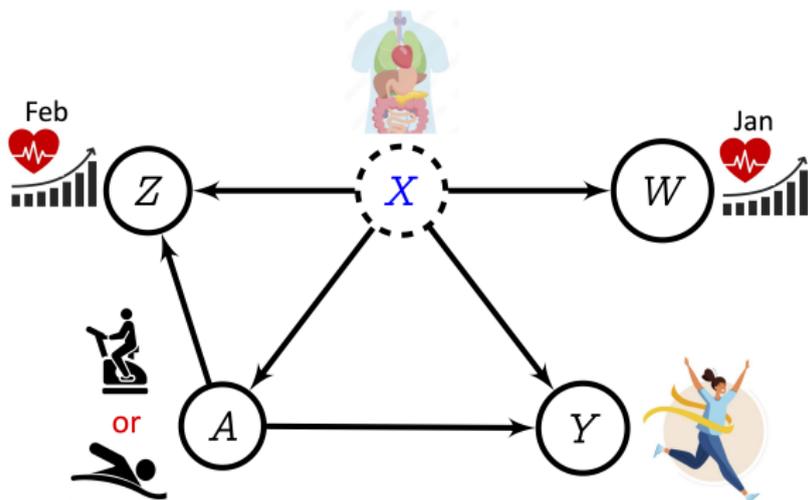


# What are proxies, and when are they useful?

Unobserved  $X$  with (possibly) complex nonlinear effects on  $A$ ,  $Y$

In this example:

- $X$ : true physical status
- $A$ : exercise regimes
- $Y$ : fitness goal
- $W$ : health readings before  $A$
- $Z$ : health readings after  $A$



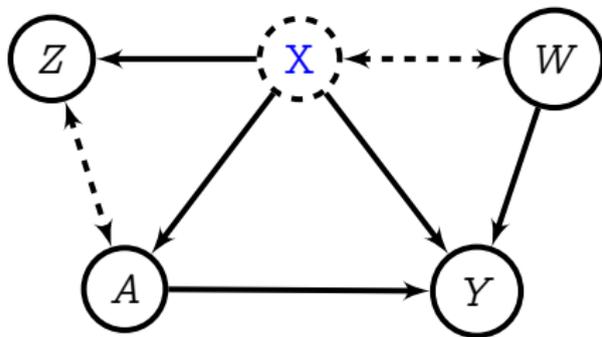
$\Rightarrow$  Can recover  $\mathbb{E}(Y^{(a)})$  from observational data

## Proxy variables: general setting

Unobserved  $X$  with (possibly) complex nonlinear effects on  $A$ ,  $Y$

The definitions are:

- $X$ : unobserved confounder.
- $A$ : treatment
- $Y$ : outcome
- $Z$ : treatment proxy
- $W$  outcome proxy

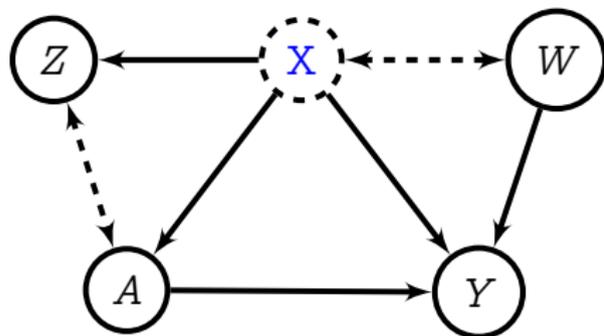


## Proxy variables: general setting

Unobserved  $X$  with (possibly) complex nonlinear effects on  $A$ ,  $Y$

The definitions are:

- $X$ : unobserved confounder.
- $A$ : treatment
- $Y$ : outcome
- $Z$ : treatment proxy
- $W$  outcome proxy



Structural assumptions:

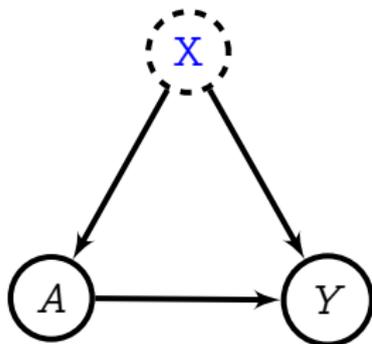
$$W \perp\!\!\!\perp (Z, A) | X$$

$$Y \perp\!\!\!\perp Z | (A, X)$$

## Why proxy variables? A simple proof

The definitions are:

- $X$ : unobserved confounder.
- $A$ : treatment
- $Y$ : outcome



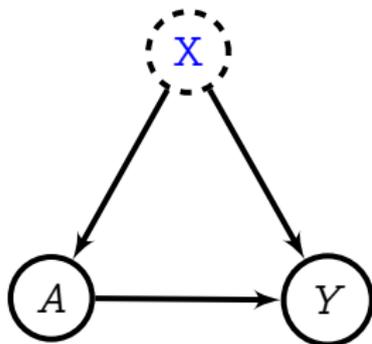
If  $X$  were observed,

$$\underbrace{P(Y^{(a)})}_{d_y \times 1} := \sum_{i=1}^{d_x} P(Y | \mathbf{x}_i, a) P(\mathbf{x}_i)$$

## Why proxy variables? A simple proof

The definitions are:

- $X$ : unobserved confounder.
- $A$ : treatment
- $Y$ : outcome



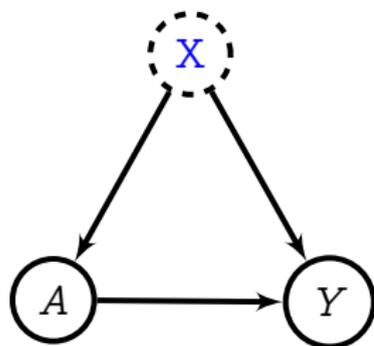
If  $X$  were observed,

$$\underbrace{P(Y^{(a)})}_{d_y \times 1} := \sum_{i=1}^{d_x} P(Y | \mathbf{x}_i, a) P(\mathbf{x}_i) = \underbrace{P(Y | X, a)}_{d_y \times d_x} \underbrace{P(X)}_{d_x \times 1}$$

## Why proxy variables? A simple proof

The definitions are:

- $X$ : unobserved confounder.
- $A$ : treatment
- $Y$ : outcome



If  $X$  were observed,

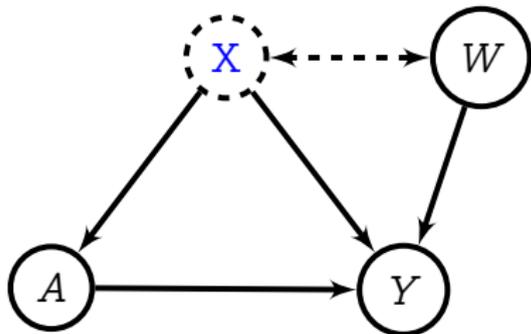
$$\underbrace{P(Y^{(a)})}_{d_y \times 1} := \sum_{i=1}^{d_x} P(Y | \mathbf{x}_i, a) P(\mathbf{x}_i) = \underbrace{P(Y | X, a)}_{d_y \times d_x} \underbrace{P(X)}_{d_x \times 1}$$

Goal: “get rid of the blue”  $X$

## ...add the outcome proxy $W$

The definitions are:

- $X$ : unobserved confounder.
- $A$ : treatment
- $Y$ : outcome
- $W$ : outcome proxy



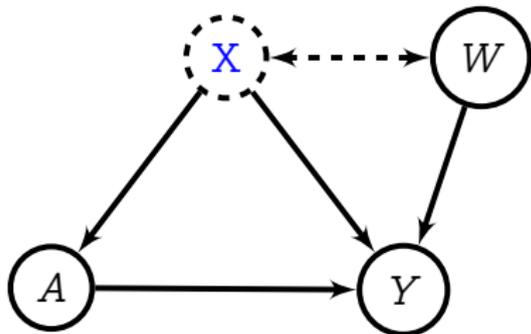
For each  $a$ , if we could solve:

$$\underbrace{P(Y|X, a)}_{d_y \times d_x} = \underbrace{H_{w,a}}_{d_y \times d_w} \underbrace{P(W|X)}_{d_w \times d_x}$$

## ...add the outcome proxy $W$

The definitions are:

- $X$ : unobserved confounder.
- $A$ : treatment
- $Y$ : outcome
- $W$ : outcome proxy



For each  $a$ , if we could solve:

$$\underbrace{P(Y|X, a)}_{d_y \times d_x} = \underbrace{H_{w,a}}_{d_y \times d_w} \underbrace{P(W|X)}_{d_w \times d_x}$$

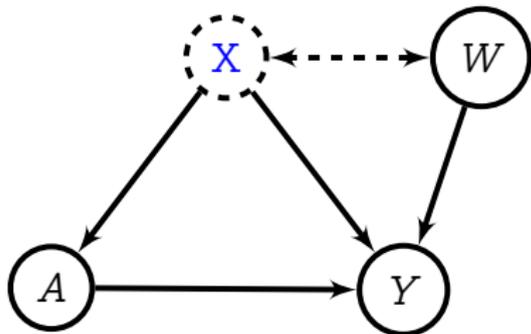
.....then

$$P(Y^{(a)}) = P(Y|X, a)P(X)$$

## ...add the outcome proxy $W$

The definitions are:

- $X$ : unobserved confounder.
- $A$ : treatment
- $Y$ : outcome
- $W$ : outcome proxy



For each  $a$ , if we could solve:

$$\underbrace{P(Y|X, a)}_{d_y \times d_x} = \underbrace{H_{w,a}}_{d_y \times d_w} \underbrace{P(W|X)}_{d_w \times d_x}$$

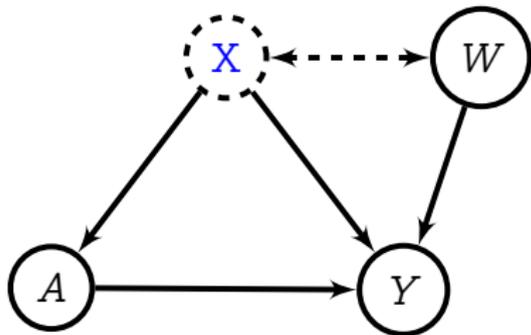
.....then

$$\begin{aligned} P(Y^{(a)}) &= P(Y|X, a)P(X) \\ &= H_{w,a}P(W|X)P(X) \end{aligned}$$

## ...add the outcome proxy $W$

The definitions are:

- $X$ : unobserved confounder.
- $A$ : treatment
- $Y$ : outcome
- $W$ : outcome proxy



For each  $a$ , if we could solve:

$$\underbrace{P(Y|X, a)}_{d_y \times d_x} = \underbrace{H_{w,a}}_{d_y \times d_w} \underbrace{P(W|X)}_{d_w \times d_x}$$

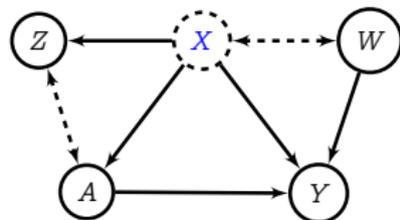
.....then

$$\begin{aligned} P(Y^{(a)}) &= P(Y|X, a)P(X) \\ &= H_{w,a}P(W|X)P(X) \\ &= H_{w,a}P(W) \end{aligned}$$

...now project onto  $p(X|Z, a)$

From last slide,

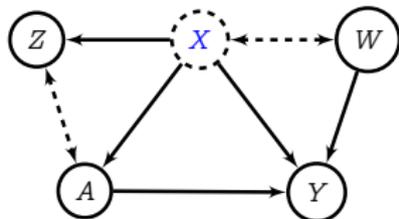
$$P(Y|X, a) = H_{w,a} P(W|X)$$



...now project onto  $p(X|Z, a)$

From last slide,

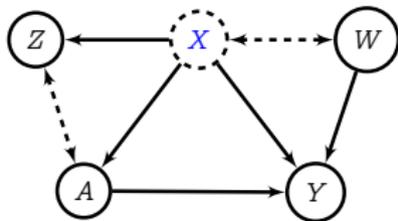
$$P(Y|X, a) \underbrace{p(X|Z, a)}_{d_x \times d_z} = H_{w,a} P(W|X) \underbrace{p(X|Z, a)}_{d_x \times d_z}$$



...now project onto  $p(X|Z, a)$

From last slide,

$$P(Y|X, a) \underbrace{p(X|Z, a)}_{d_x \times d_z} = H_{w,a} P(W|X) \underbrace{p(X|Z, a)}_{d_x \times d_z}$$



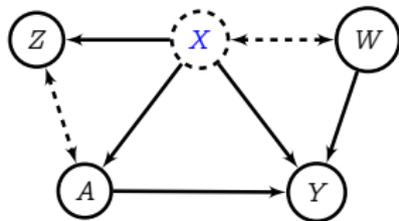
Because  $W \perp\!\!\!\perp (Z, A) | X$ ,

$$P(W|X)p(X|Z, a) = P(W|Z, a)$$

...now project onto  $p(X|Z, a)$

From last slide,

$$P(Y|X, a) \underbrace{p(X|Z, a)}_{d_x \times d_z} = H_{w,a} P(W|X) \underbrace{p(X|Z, a)}_{d_x \times d_z}$$



Because  $W \perp\!\!\!\perp (Z, A) | X$ ,

$$P(W|X)p(X|Z, a) = P(W|Z, a)$$

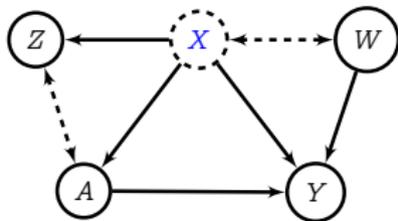
Because  $Y \perp\!\!\!\perp Z | (A, X)$ ,

$$P(Y|X, a)p(X|Z, a) = P(Y|Z, a)$$

...now project onto  $p(X|Z, a)$

From last slide,

$$P(Y|X, a) \underbrace{p(X|Z, a)}_{d_x \times d_z} = H_{w,a} P(W|X) \underbrace{p(X|Z, a)}_{d_x \times d_z}$$



Because  $W \perp\!\!\!\perp (Z, A) | X$ ,

$$P(W|X)p(X|Z, a) = P(W|Z, a)$$

Because  $Y \perp\!\!\!\perp Z | (A, X)$ ,

$$P(Y|X, a)p(X|Z, a) = P(Y|Z, a)$$

Solve for  $H_{w,a}$ :

$$P(Y|Z, a) = H_{w,a} P(W|Z, a)$$

Everything observed!

# Proxy/Negative Control Methods in the Real World

# Unobserved confounders: proxy methods

## Kernel features (ICML 2021):

arXiv.org > cs > arXiv:2105.04544

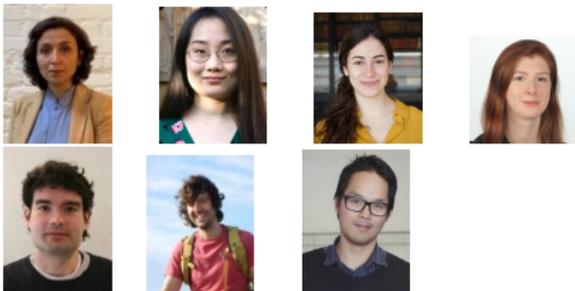
Search...  
Help | Advan

Computer Science > Machine Learning

*[Submitted on 10 May 2021 (v1), last revised 9 Oct 2021 (this version, v4)]*

### Proximal Causal Learning with Kernels: Two-Stage Estimation and Moment Restriction

Afsaneh Mastouri, Yuchen Zhu, Limor Gultchin, Anna Korba, Ricardo Silva, Matt J. Kusner, Arthur Gretton, Krikamol Muandet



## NN features (NeurIPS 2021):

arXiv.org > cs > arXiv:2106.03907

Search...  
Help | Advan

Computer Science > Machine Learning

*[Submitted on 7 Jun 2021 (v1), last revised 7 Dec 2021 (this version, v2)]*

### Deep Proxy Causal Learning and its Application to Confounded Bandit Policy Evaluation

Liyuan Xu, Heishiro Kanagawa, Arthur Gretton



Code for NN and kernel proxy methods:

<https://github.com/liyuan9988/DeepFeatureProxyVariable/>

# Unobserved confounders: proxy methods

## Kernel features (ICML 2021):

arXiv.org > cs > arXiv:2105.04544

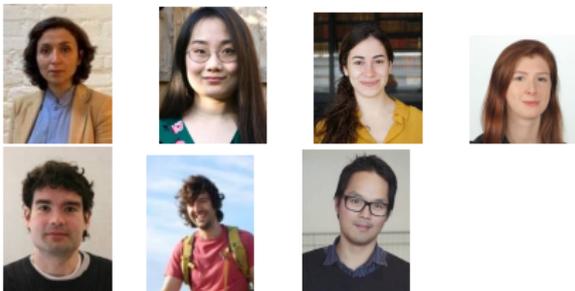
Search...  
Help | Advan

Computer Science > Machine Learning

[Submitted on 10 May 2021 (v1), last revised 9 Oct 2021 (this version, v4)]

### Proximal Causal Learning with Kernels: Two-Stage Estimation and Moment Restriction

Afsaneh Mastouri, Yuchen Zhu, Limor Gultchin, Anna Korba, Ricardo Silva, Matt J. Kusner, Arthur Gretton, Krikamol Muandet



## NN features (NeurIPS 2021):

arXiv.org > cs > arXiv:2106.03907

Search...  
Help | Advan

Computer Science > Machine Learning

[Submitted on 7 Jun 2021 (v1), last revised 7 Dec 2021 (this version, v2)]

### Deep Proxy Causal Learning and its Application to Confounded Bandit Policy Evaluation

Liyuan Xu, Heishiro Kanagawa, Arthur Gretton

Two author portraits for the paper 'Deep Proxy Causal Learning and its Application to Confounded Bandit Policy Evaluation'. The portraits are arranged in a single row. The authors are Liyuan Xu and Heishiro Kanagawa.

Code for NN and kernel proxy methods:

<https://github.com/liyuan9988/DeepFeatureProxyVariable/>

## Road map: NN proxy learning

We'll proceed as follows:

- Proxy relation for continuous variables
- Loss function for deep proxy learning
- Define **primary** (ridge) regression with this loss
- Define **secondary** (ridge) regression as input to primary

## Proxy relation, general domains

If  $X$  were observed, we would write (dose-response curve)

$$\mathbb{E}(Y^{(a)}) = \int_x \mathbb{E}(Y|a, x)p(x)dx.$$

....but we do not observe  $X$ .

## Proxy relation, general domains

If  $X$  were observed, we would write (dose-response curve)

$$\mathbb{E}(Y^{(a)}) = \int_x \mathbb{E}(Y|a, x)p(x)dx.$$

....but we do not observe  $X$ .

**Main theorem:** Assume we solved for link function:

$$\mathbb{E}(Y|a, z) = \mathbb{E}_{W|a,z} h_y(W, a)$$

- “Primary”  $\mathbb{E}(Y|a, z)$ , “secondary”  $\mathbb{E}_{W|a,z}$  linked by  $h_y$
- All variables observed,  $X$  not seen *or modeled*.

Fredholm equation of first kind. Link existence requires  $\diamond$ , identification of ATE requires  $\triangle$  (and further technical assumptions) [XKG: Assumption 2, Prop. 1, Corr. 1; Deane]

$$\mathbb{E}[f(X)|A = a, Z = z] = 0, \forall(z, a) \iff f(X) = 0, \mathbb{P}_X \text{ a.s. } \triangle$$

$$\mathbb{E}[f(X)|A = a, W = w] = 0, \forall(w, a) \iff f(X) = 0, \mathbb{P}_X \text{ a.s. } \diamond$$

## Proxy relation, general domains

If  $X$  were observed, we would write (dose-response curve)

$$\mathbb{E}(Y^{(a)}) = \int_x \mathbb{E}(Y|a, x)p(x)dx.$$

....but we do not observe  $X$ .

**Main theorem:** Assume we solved for link function:

$$\mathbb{E}(Y|a, z) = \mathbb{E}_{W|a,z} h_y(W, a)$$

- “Primary”  $\mathbb{E}(Y|a, z)$ , “secondary”  $\mathbb{E}_{W|a,z}$  linked by  $h_y$
- All variables observed,  $X$  not seen *or modeled*.

**Dose-response curve** via  $p(w)$ :

$$\mathbb{E}(Y^{(a)}) = \int_w h_y(a, w)p(w)dw$$

## Proxy relation, general domains

If  $X$  were observed, we would write (dose-response curve)

$$\mathbb{E}(Y^{(a)}) = \int_x \mathbb{E}(Y|a, x)p(x)dx.$$

....but we do not observe  $X$ .

**Main theorem:** Assume we solved for link function:

$$\mathbb{E}(Y|a, z) = \mathbb{E}_{W|a,z} h_y(W, a)$$

- “Primary”  $\mathbb{E}(Y|a, z)$ , “secondary”  $\mathbb{E}_{W|a,z}$  linked by  $h_y$
- All variables observed,  $X$  not seen *or modeled*.

Dose-response curve via  $p(w)$ :

$$\mathbb{E}(Y^{(a)}) = \int_w h_y(a, w)p(w)dw$$

**Challenge:** need a loss function for  $h_y$

## Primary loss function for $h_y(w, a)$

Goal:

$$\mathbb{E}(Y|a, z) = \mathbb{E}_{W|a, z} h_y(W, a)$$

Primary loss function:

$$\hat{h}_y = \arg \min_{h_y} \mathbb{E}_{Y, A, Z} \left( Y - \mathbb{E}_{W|A, Z} h_y(W, A) \right)^2$$

Why?

Deaner (2021).

Mastouri, Zhu, Gultchin, Korba, Silva, Kusner, G., Muandet (2021).

Xu, Kanagawa, G. (2021).

## Primary loss function for $h_y(w, a)$

Goal:

$$\mathbb{E}(Y|a, z) = \mathbb{E}_{W|a, z} h_y(W, a)$$

Primary loss function:

$$\hat{h}_y = \arg \min_{h_y} \mathbb{E}_{Y, A, Z} \left( Y - \mathbb{E}_{W|A, Z} h_y(W, A) \right)^2$$

Why?

$f^*(a, z) = \mathbb{E}(Y|a, z)$  solves

$$\operatorname{argmin}_f \mathbb{E}_{Y, A, Z} (Y - f(A, Z))^2$$

Deaner (2021).

Mastouri, Zhu, Gultchin, Korba, Silva, Kusner, G., Muandet (2021).

Xu, Kanagawa, G. (2021).

## Primary loss function for $h_y(w, a)$

Goal:

$$\mathbb{E}(Y|a, z) = \mathbb{E}_{W|a, z} h_y(W, a)$$

Primary loss function:

$$\hat{h}_y = \arg \min_{h_y} \mathbb{E}_{Y, A, Z} \left( Y - \mathbb{E}_{W|A, Z} h_y(W, A) \right)^2$$

Why?

$f^*(a, z) = \mathbb{E}(Y|a, z)$  solves

$$\operatorname{argmin}_f \mathbb{E}_{Y, A, Z} (Y - f(A, Z))^2$$

...and by the proxy model above,

$$\mathbb{E}(Y|a, z) = \mathbb{E}_{W|a, z} h_y(W, a)$$

Deaner (2021).

Mastouri, Zhu, Gultchin, Korba, Silva, Kusner, G., Muandet (2021).

Xu, Kanagawa, G. (2021).

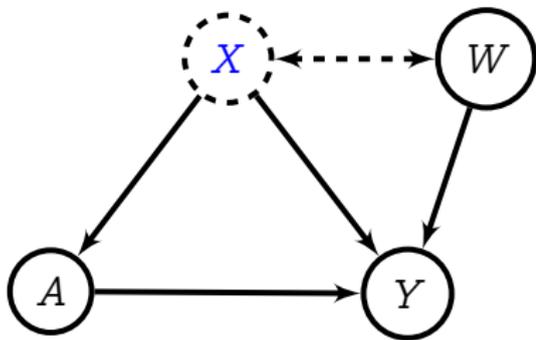
## NN for link $h_y(a, w)$

The link function is a function of **two** arguments

$$h_y(a, w) = \gamma^\top [\varphi_\theta(w) \otimes \varphi_\xi(a)] = \gamma^\top \begin{bmatrix} \varphi_{\theta,1}(w)\varphi_{\xi,1}(a) \\ \varphi_{\theta,1}(w)\varphi_{\xi,2}(a) \\ \vdots \\ \varphi_{\theta,2}(w)\varphi_{\xi,1}(a) \\ \vdots \end{bmatrix}$$

Assume we have:

- output proxy NN features  $\varphi_\theta(w)$
- treatment NN features  $\varphi_\xi(a)$
- linear final layer  $\gamma$   
(argument of feature map indicates feature space)



## NN for link $h_y(a, w)$

The **link function** is a function of **two** arguments

$$h_y(a, w) = \gamma^\top [\varphi_\theta(w) \otimes \varphi_\xi(a)]$$

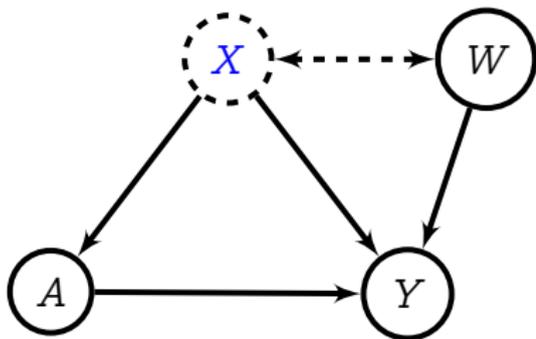
Assume we have:

- output proxy NN features  $\varphi_\theta(w)$
- treatment NN features  $\varphi_\xi(a)$
- linear final layer  $\gamma$   
(argument of feature map indicates feature space)

**Questions:**

- Why feature map  $\varphi_\theta(w) \otimes \varphi_\xi(a)$ ?
- Why final linear layer  $\gamma$ ?

**Both are necessary** (next slide)!



## NN ridge regression for $h_y(w, a)$

Goal:

$$\mathbb{E}(Y|a, z) = \mathbb{E}_{W|a, z} h_y(W, a)$$

Primary regression:

$$\hat{h}_y = \arg \min_{h_y} \mathbb{E}_{Y, A, Z} \left( Y - \mathbb{E}_{W|A, Z} h_y(W, A) \right)^2 + \lambda_2 \|\gamma\|^2$$

Deaner (2021).

Mastouri, Zhu, Gultchin, Korba, Silva, Kusner, G., Muandet (2021).

Xu, Kanagawa, G. (2021).

## NN ridge regression for $h_y(w, a)$

Goal:

$$\mathbb{E}(Y|a, z) = \mathbb{E}_{W|a, z} h_y(W, a)$$

Primary regression:

$$\hat{h}_y = \arg \min_{h_y} \mathbb{E}_{Y, A, Z} \left( Y - \mathbb{E}_{W|A, Z} h_y(W, A) \right)^2 + \lambda_2 \|\gamma\|^2$$

How to get **conditional expectation**  $\mathbb{E}_{W|a, z} h_y(W, a)$ ?

Density estimation for  $p(W|a, z)$ ? Sample from  $p(W|a, z)$ ?

## NN ridge regression for $h_y(w, a)$

Goal:

$$\mathbb{E}(Y|a, z) = \mathbb{E}_{W|a, z} h_y(W, a)$$

Primary regression:

$$\hat{h}_y = \arg \min_{h_y} \mathbb{E}_{Y, A, Z} \left( Y - \mathbb{E}_{W|A, Z} h_y(W, A) \right)^2 + \lambda_2 \|\gamma\|^2$$

Recall link function

$$h_y(W, a) = \left[ \gamma^\top (\varphi_\theta(W) \otimes \varphi_\xi(a)) \right]$$

Deaner (2021).

Mastouri, Zhu, Gultchin, Korba, Silva, Kusner, G., Muandet (2021).

Xu, Kanagawa, G. (2021).

## NN ridge regression for $h_y(w, a)$

Goal:

$$\mathbb{E}(Y|a, z) = \mathbb{E}_{W|a, z} h_y(W, a)$$

Primary regression:

$$\hat{h}_y = \arg \min_{h_y} \mathbb{E}_{Y, A, Z} \left( Y - \mathbb{E}_{W|A, Z} h_y(W, A) \right)^2 + \lambda_2 \|\gamma\|^2$$

Recall link function

$$\mathbb{E}_{W|a, z} h_y(W, a) = \mathbb{E}_{W|a, z} \left[ \gamma^\top (\varphi_\theta(W) \otimes \varphi_\xi(a)) \right]$$

Deaner (2021).

Mastouri, Zhu, Gultchin, Korba, Silva, Kusner, G., Muandet (2021).

Xu, Kanagawa, G. (2021).

## NN ridge regression for $h_y(w, a)$

Goal:

$$\mathbb{E}(Y|a, z) = \mathbb{E}_{W|a, z} h_y(W, a)$$

Primary regression:

$$\hat{h}_y = \arg \min_{h_y} \mathbb{E}_{Y, A, Z} \left( Y - \mathbb{E}_{W|A, Z} h_y(W, A) \right)^2 + \lambda_2 \|\gamma\|^2$$

Recall link function

$$\begin{aligned} \mathbb{E}_{W|a, z} h_y(W, a) &= \mathbb{E}_{W|a, z} \left[ \gamma^\top (\varphi_\theta(W) \otimes \varphi_\xi(a)) \right] \\ &= \gamma^\top \left( \underbrace{\mathbb{E}_{W|a, z} [\varphi_\theta(W)]}_{\text{cond. feat. mean}} \otimes \varphi_\xi(a) \right) \end{aligned}$$

(this is why linear  $\gamma$  and feature map  $\varphi_\theta(w) \otimes \varphi_\xi(a)$ )

Deaner (2021).

Mastouri, Zhu, Gultchin, Korba, Silva, Kusner, G., Muandet (2021).

Xu, Kanagawa, G. (2021).

## NN ridge regression for $h_y(w, a)$

Goal:

$$\mathbb{E}(Y|a, z) = \mathbb{E}_{W|a, z} h_y(W, a)$$

Primary regression:

$$\hat{h}_y = \arg \min_{h_y} \mathbb{E}_{Y, A, Z} \left( Y - \mathbb{E}_{W|A, Z} h_y(W, A) \right)^2 + \lambda_2 \|\gamma\|^2$$

Recall link function

$$\begin{aligned} \mathbb{E}_{W|a, z} h_y(W, a) &= \mathbb{E}_{W|a, z} \left[ \gamma^\top (\varphi_\theta(W) \otimes \varphi_\xi(a)) \right] \\ &= \gamma^\top \left( \underbrace{\mathbb{E}_{W|a, z} [\varphi_\theta(W)]}_{\text{cond. feat. mean}} \otimes \varphi_\xi(a) \right) \end{aligned}$$

Ridge regression (again!)

$$\mathbb{E}_{W|a, z} \varphi_\theta(W) = \hat{F}_{\theta, \zeta} \varphi_\zeta(a, z)$$

Deaner (2021).

Mastouri, Zhu, Gultchin, Korba, Silva, Kusner, G., Muandet (2021).

Xu, Kanagawa, G. (2021).

## NN ridge regression for $\mathbb{E}_{W|a,z} \varphi_{\theta}(W)$

Secondary regression: learn NN features  $\varphi_{\zeta}(Z)$  and linear layer  $F$ :

$$\mathbb{E}_{W|a,z} \varphi_{\theta}(W) = \hat{F}_{\theta,\zeta} \varphi_{\zeta}(a, z)$$

with RR loss

$$\mathbb{E}_{W,A,Z} \|\varphi_{\theta}(W) - F \varphi_{\zeta}(A, Z)\|^2 + \lambda_1 \|F\|^2$$

$\hat{F}_{\theta,\zeta}$  in closed form wrt  $\varphi_{\theta}, \varphi_{\zeta}$ .

## NN ridge regression for $\mathbb{E}_{W|a,z} \varphi_{\theta}(W)$

Secondary regression: learn NN features  $\varphi_{\zeta}(Z)$  and linear layer  $F$ :

$$\mathbb{E}_{W|a,z} \varphi_{\theta}(W) = \hat{F}_{\theta,\zeta} \varphi_{\zeta}(a, z)$$

with RR loss

$$\mathbb{E}_{W,A,Z} \|\varphi_{\theta}(W) - F \varphi_{\zeta}(A, Z)\|^2 + \lambda_1 \|F\|^2$$

$\hat{F}_{\theta,\zeta}$  in closed form wrt  $\varphi_{\theta}, \varphi_{\zeta}$ .

Plug  $\hat{F}_{\theta,\zeta}$  into S1 loss, backprop through Cholesky for  $\zeta$   
(...not  $\theta$ ...why not?)

## Final algorithm

Solve for  $\theta, \xi, \zeta$ :

Repeat until convergence:

- **Secondary:** Solve for  $\hat{F}_{\theta, \zeta}$ , then gradient steps on  $\zeta$  (backprop through Cholesky)

## Final algorithm

Solve for  $\theta, \xi, \zeta$ :

Repeat until convergence:

- **Secondary:** Solve for  $\hat{F}_{\theta, \zeta}$ , then gradient steps on  $\zeta$  (backprop through Cholesky)
- **Primary:** Solve for  $\hat{\gamma}$  in terms of  $\hat{F}_{\theta, \zeta} \varphi_{\zeta}(A, Z)$  and  $\varphi_{\xi}(A)$

## Final algorithm

Solve for  $\theta, \xi, \zeta$ :

Repeat until convergence:

- **Secondary:** Solve for  $\hat{F}_{\theta, \zeta}$ , then gradient steps on  $\zeta$  (backprop through Cholesky)
- **Primary:** Solve for  $\hat{\gamma}$  in terms of  $\hat{F}_{\theta, \zeta} \varphi_{\zeta}(A, Z)$  and  $\varphi_{\xi}(A)$
- **Primary:** Gradient steps on  $\theta, \xi$  (backprop through Cholesky)
  - $\hat{F}_{\theta, \zeta}$  remains optimal wrt current  $\varphi_{\theta}$ .

## Final algorithm

Solve for  $\theta, \xi, \zeta$ :

Repeat until convergence:

- **Secondary:** Solve for  $\hat{F}_{\theta, \zeta}$ , then gradient steps on  $\zeta$  (backprop through Cholesky)
- **Primary:** Solve for  $\hat{\gamma}$  in terms of  $\hat{F}_{\theta, \zeta} \varphi_{\zeta}(A, Z)$  and  $\varphi_{\xi}(A)$
- **Primary:** Gradient steps on  $\theta, \xi$  (backprop through Cholesky)
  - $\hat{F}_{\theta, \zeta}$  remains optimal wrt current  $\varphi_{\theta}$ .

Iterate between updates of  $\theta, \xi$  and  $\zeta$

## Final algorithm

Solve for  $\theta, \xi, \zeta$ :

Repeat until convergence:

- **Secondary:** Solve for  $\hat{F}_{\theta, \zeta}$ , then gradient steps on  $\zeta$  (backprop through Cholesky)
- **Primary:** Solve for  $\hat{\gamma}$  in terms of  $\hat{F}_{\theta, \zeta} \varphi_{\zeta}(A, Z)$  and  $\varphi_{\xi}(A)$
- **Primary:** Gradient steps on  $\theta, \xi$  (backprop through Cholesky)
  - $\hat{F}_{\theta, \zeta}$  remains optimal wrt current  $\varphi_{\theta}$ .

Iterate between updates of  $\theta, \xi$  and  $\zeta$

**Key point:** features  $\varphi_{\theta}(W)$  learned specially for:

$$\mathbb{E}(Y|a, z) = \mathbb{E}_{W|a, z} h_{\gamma}(W, a)$$

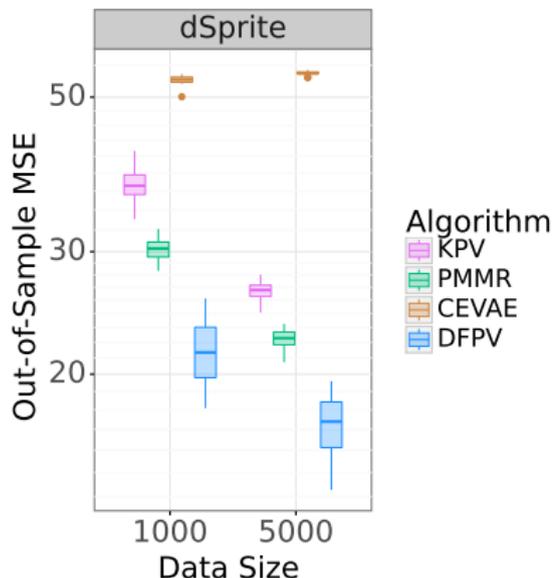
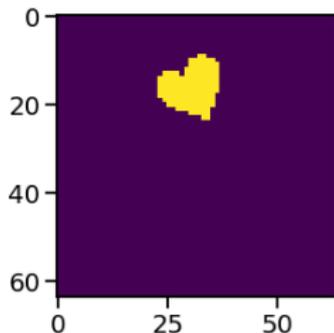
**Contrast with autoencoders/sampling:** must reconstruct/sample all of  $W$ .

# Experiments

# Synthetic experiment, adaptive neural net features

## dSprite example:

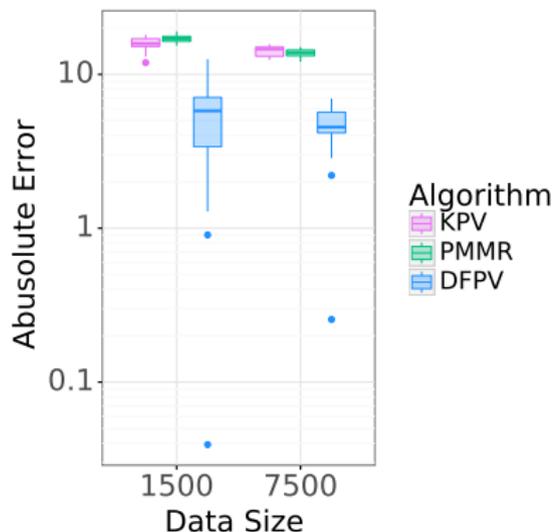
- $X = \{\text{scale, rotation, posX, posY}\}$
- Treatment  $A$  is the image generated (with Gaussian noise)
- Outcome  $Y$  is quadratic function of  $A$  with multiplicative confounding by  $\text{posY}$ .
- $Z = \{\text{scale, rotation, posX}\}$ ,  
 $W = \text{noisy image sharing posY}$
- Comparison with **CEVAE** (Louzios et al. 2017)



# Confounded offline policy evaluation

Synthetic dataset, demand prediction for flight purchase.

- Treatment  $A$  is ticket price.
- Policy  $A \sim \pi(Z)$  depends on fuel price.



# Conclusions

## Neural net and kernel solutions:

- ...for DR, HR, dynamic treatment effects
- ...even for unobserved covariates/confounders (IV and proxy methods)
- ...with treatment  $A$ , covariates  $X, V$ , proxies/instruments ( $W, Z$ ) multivariate, “complicated”
- Convergence guarantees for kernels and NN

## Key messages:

- Don't ~~meet your heroes~~ model/sample hidden variables
- “Ridge regression is all you need”

Code available for all methods

# Research support

Work supported by:

The Gatsby Charitable Foundation



Google Deepmind



# Questions?

