Modelling the Hemodynamic Response Function (HRF) by Double Machine Learning

Presented by: Hongbiao Chen

Supervisor: Dr. Michel Besserve

Second Reader: Prof. Dr. Nikos Logothetis

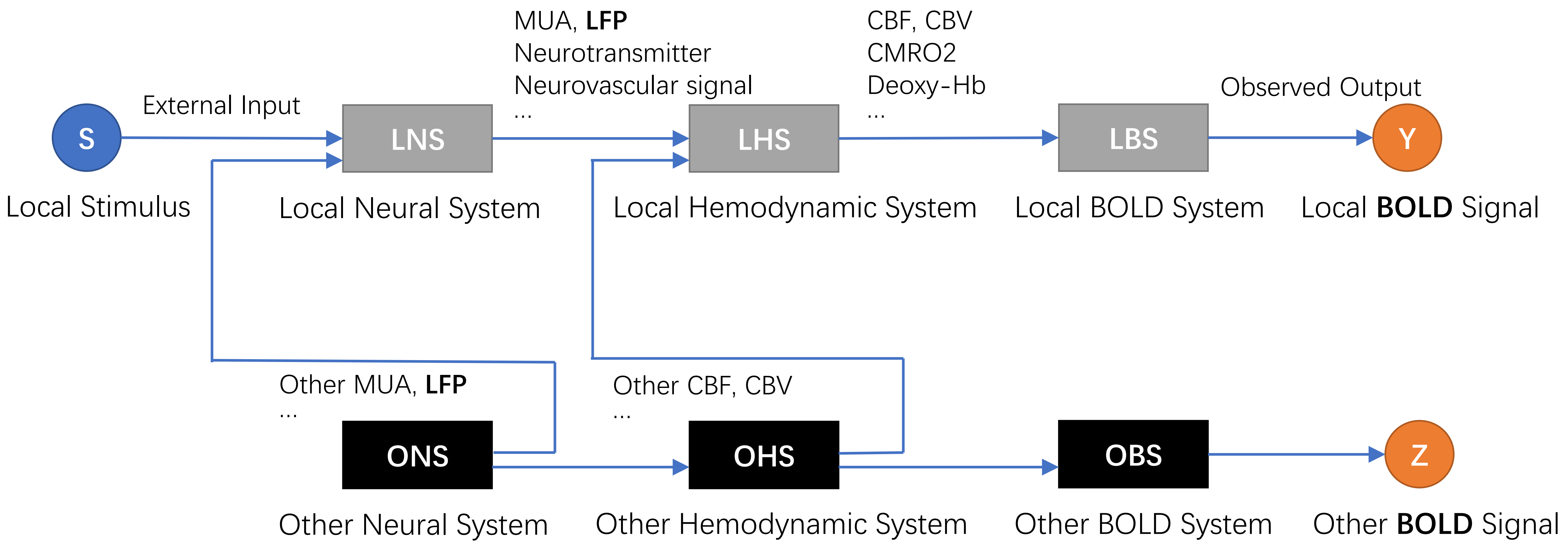
Overview

The **global LFP** would affect the **local LFP** and the **local BOLD** signal, which acts as a high-dimensional **confounder** for the **local HRF** estimation.

Double ML can remove the effect of **global LFP** for the **unbiased** estimation, compared to the **naive regression** method.

- **Introduction**
    - From LFP to BOLD
    - Modelling HRF
- **Method**
    - Double ML
    - EconML toolbox
- ~~**Result**~~
- ~~**Summary**~~

Introduction: From LFP to BOLD



S

External Input

Local Stimulus

LNS

Local Neural System

MUA, **LFP**
Neurotransmitter
Neurovascular signal
…

LHS

Local Hemodynamic System

CBF, CBV
CMRO2
Deoxy-Hb
…

LBS

Local BOLD System

Observed Output

Y

Local **BOLD** Signal

Other MUA, **LFP**
…

ONS

Other Neural System

Other CBF, CBV
…

OHS

Other Hemodynamic System

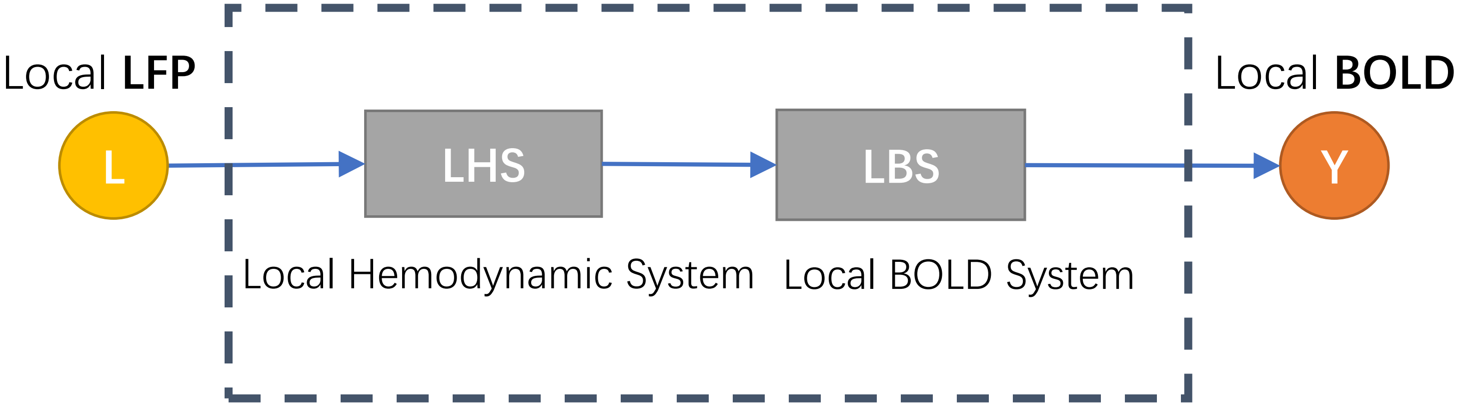
OBS

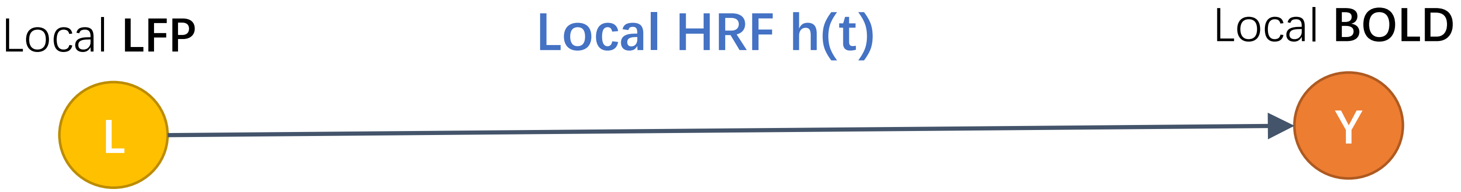
Other BOLD System

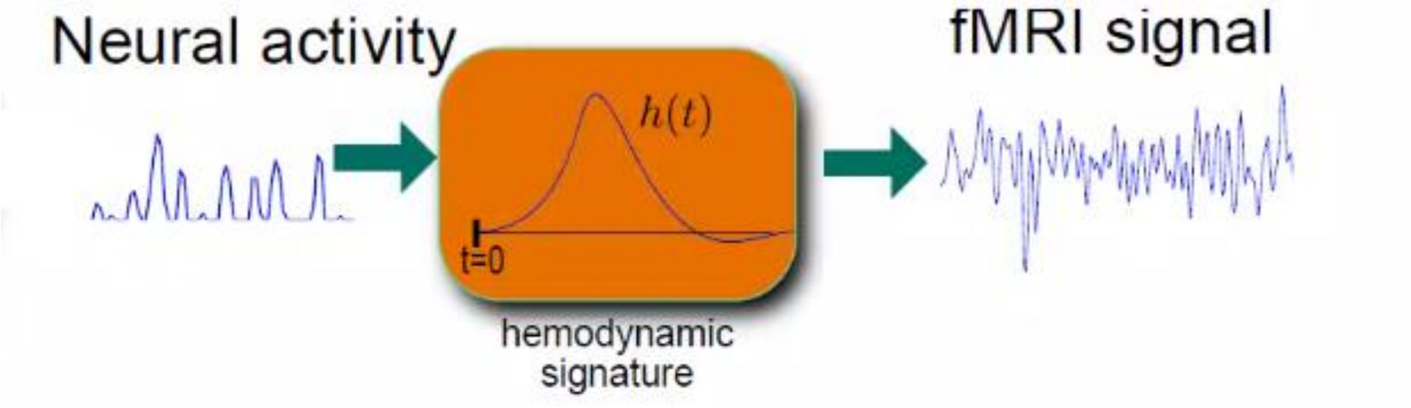
Z

Other **BOLD** Signal

Introduction: Modelling HRF







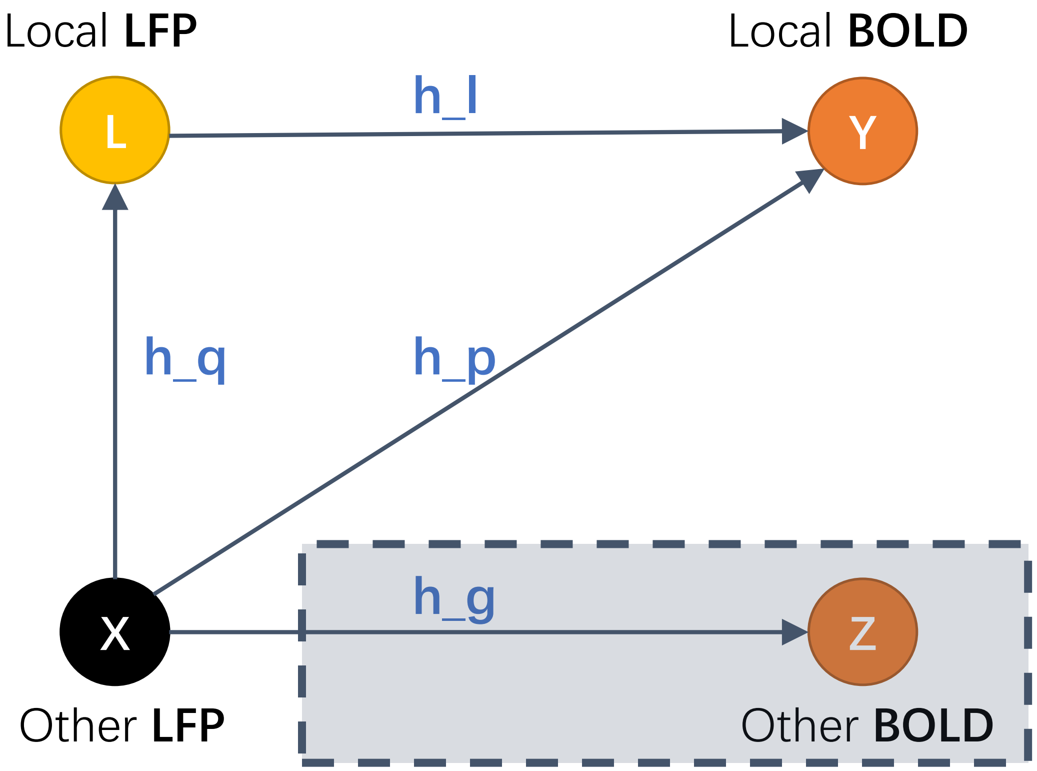
$$y(t) = (s*h)(t).$$

$$h(t) = \sum_{i=1}^{B} \beta_i g_i(t)$$

y(t): local BOLD at time t
s(t): local LFP at time t
h(t): hemodynamic response at time t

β_i: regression coefficients
g_i: basis functions
Assumed **B** basis function in total.

Introduction: In the Present of Confounder

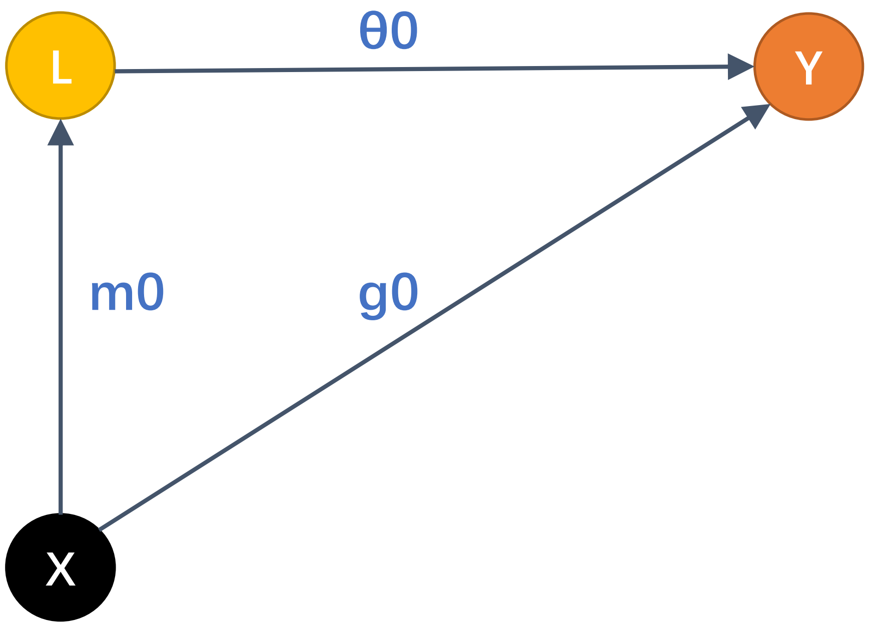


Y(t) = **h_p** ⋆ (X(t) . a^T) + **h_l** ⋆ (**L(t)**) + noise_1
L(t) = **h_q** ⋆ (X(t) . b^T) + noise_2

a^T: coefficients for h_p link
b^T: coefficients for h_q link

noise_1: additional noise of local BOLD
noise_2: additional noise of local LFP

Introduction: Partially Linear Regression (PLR)



Y = **θ0**(**L**) + **g0**(**X**) + U

L = **m0**(**X**) + V

E [U | X, L] = 0

E [V | X] = 0

Y: outcome variable
L: treatment variable (low-dimension)
X: confounder (high-dimension)

U: noise for Y
V: noise for L

θ0: interested parameters
g0 and **m0**: nuisance parameters
Some hidden functions, could be constant or nonlinear

Method: Naive Regression

Regress **Y** by using L and **X**, and get the predicted Y:

$$\hat{\theta}_0(\mathrm{L}) + \hat{g}_0(\mathrm{X})$$

For example, using alternating minimization:

(1) Given an initial guess of $\hat{\theta}_0()$

(2) Regress Y-$\hat{\theta}_0$(L) on **X** to fit $\hat{g}_0()$ by random forest

(3) Regress Y-first_fitted_$\hat{g}_0$(X) on **L** to fit updated $\hat{\theta}_0()$

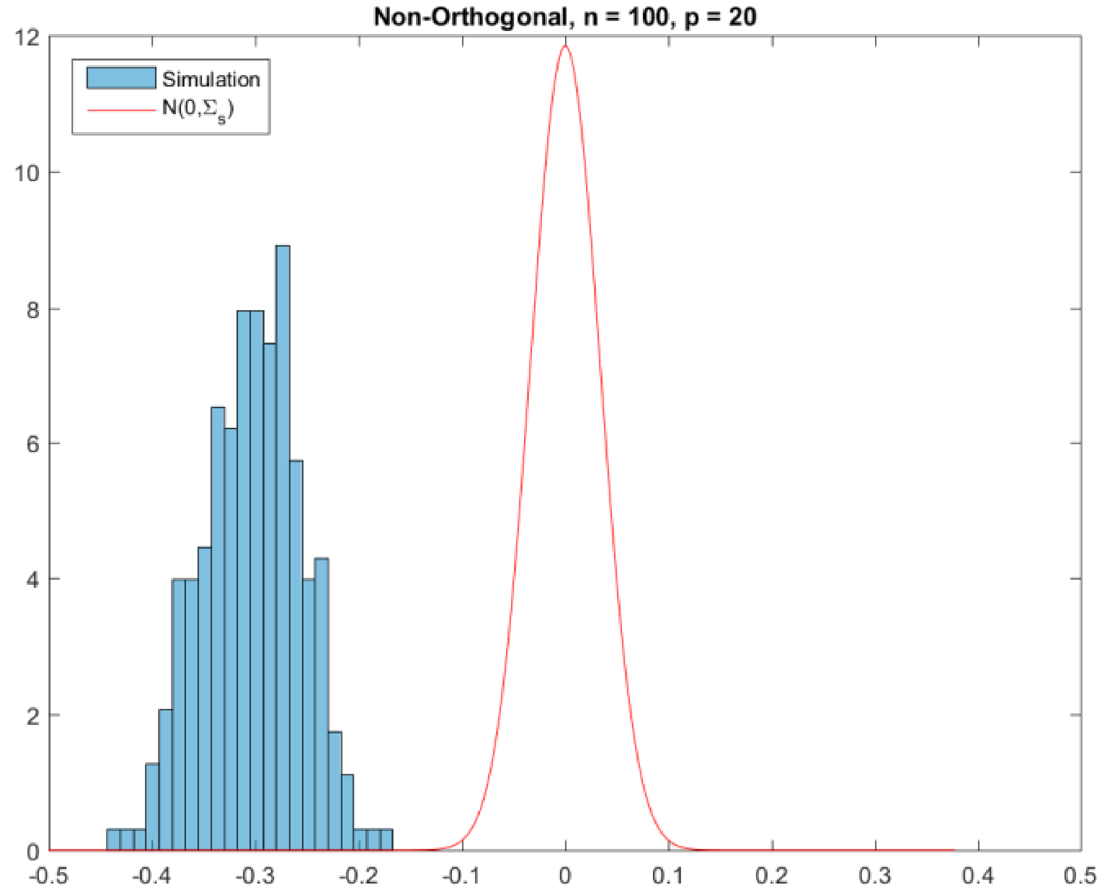
(4) Repeat (2) and (3) until convergence to get final $\hat{\theta}_0()$ and $\hat{g}_0()$

Didn't use the information from **L** = **m0**(**X**) + V;

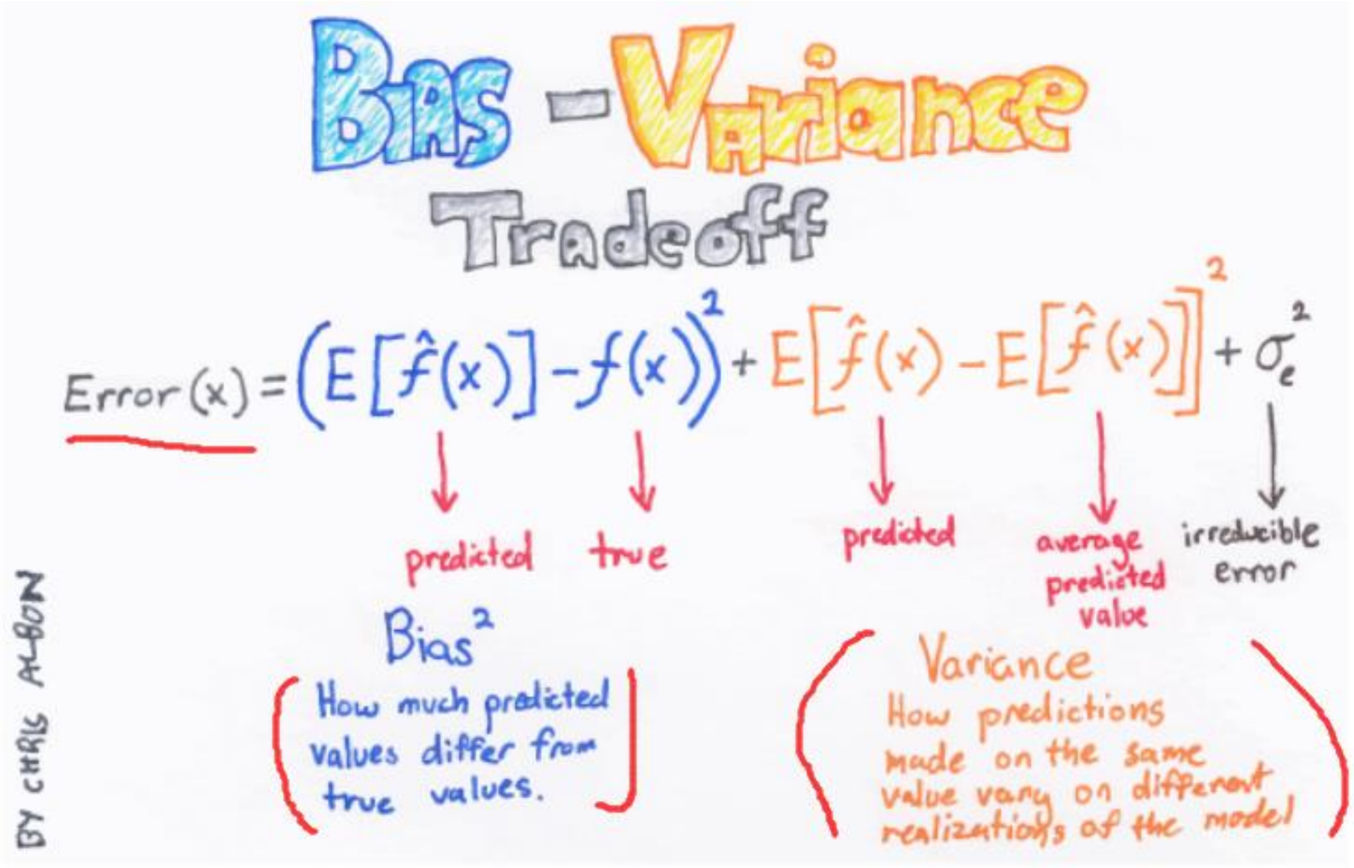
Based on the moment condition: E[(**Y-θ0**(**L**)-**g0**(**X**))L] = 0.

Method: Naive Regression is Bad

Good **Prediction Performance**, but Bad Causal **Parameter $\boldsymbol{\theta_0}$ Estimation**.
The distribution of $\hat{\theta}_0$-θ_0 looks like this:



Method: Two Sources of Bias from ML


Bias - Variance
Tradeoff
$\text{Error}(x) = \left(E[\hat{f}(x)] - f(x)\right)^2 + E\left[\hat{f}(x) - E[\hat{f}(x)]\right]^2 + \sigma_e^2$
predicted
true
predicted
average predicted value
irreducible error
Bias^2
How much predicted values differ from true values.
Variance
How predictions made on the same value vary on different realizations of the model
BY CHRIS ALBON

Machine Learning Flashcard by Chris Albon

(1) Bias from Regularization
(2) Bias from Overfitting

Method: Double ML

Regress Y and L by using **X**, and get the predicted Y and L:

$$\hat{l}_0(X) = \hat{E}[Y|X]$$
$$\hat{m}_0(X) = \hat{E}[L|X]$$

Frisch-Waugh-Lovell (1930s) style

(1) Get $\hat{E}[Y|X]$ and $\hat{E}[L|X]$ by using random forest

(2) Compute the residuals: $\hat{W}$ = Y - $\hat{E}[Y|X]$ and $\hat{V}$ = L - $\hat{E}[L|X]$

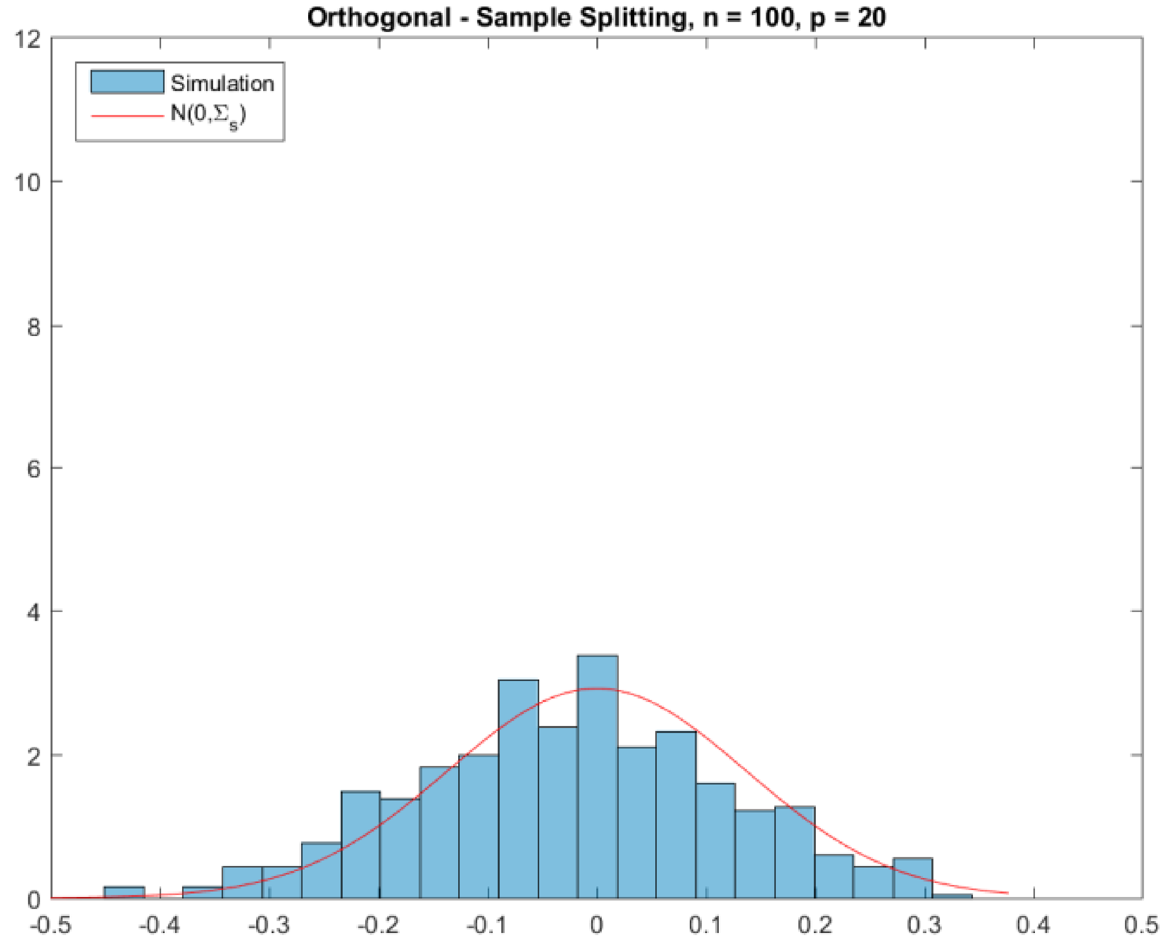
(3) Regress $\hat{W}$ on $\hat{V}$ to fit the $\hat{\theta}_0()$

Use the information from **L** = **m0**(**X**) + V;

Based on the moment condition: E[((**Y-E[Y|X]**)-**θ0**(**L-E[L|X]**)) (**L-E[L|X]**)] = 0

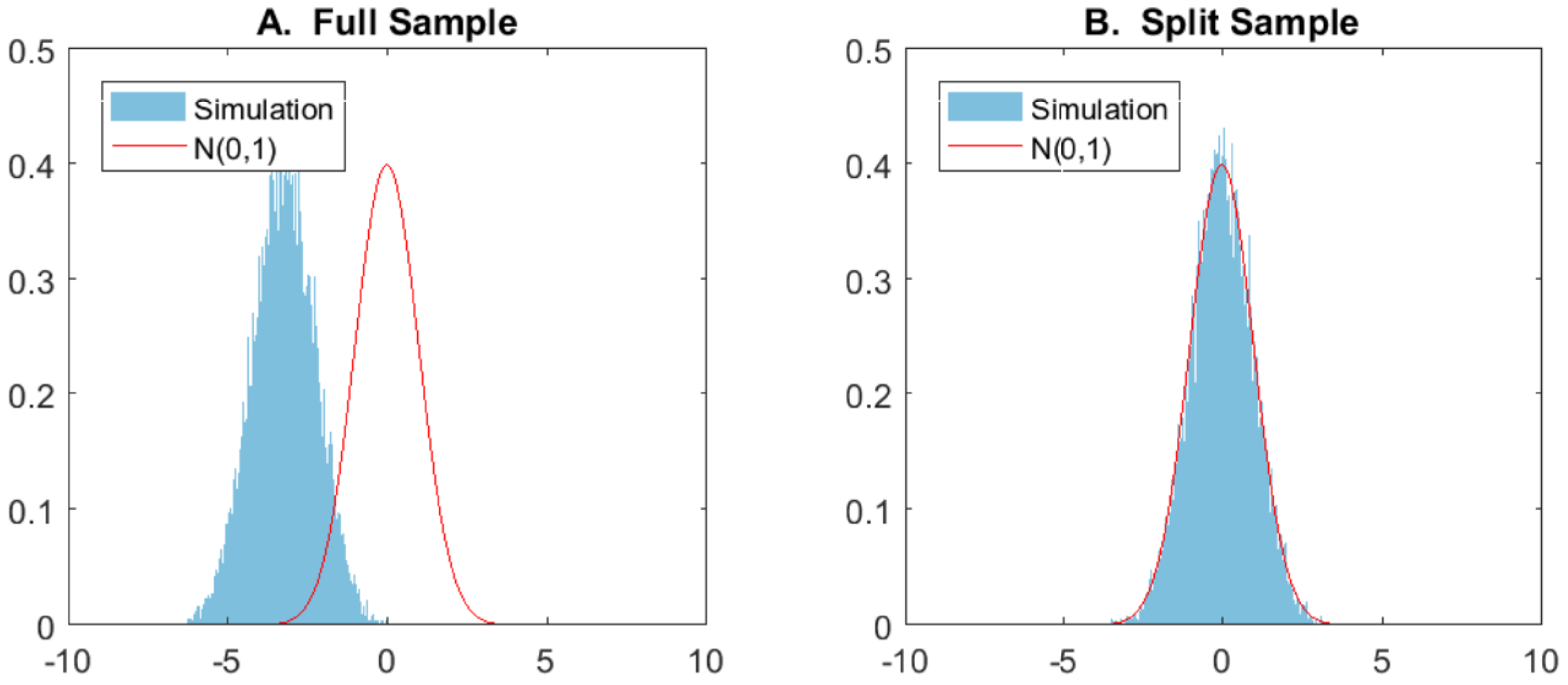
Method: Double ML is Good

The distribution of $\hat{\theta}_0$-θ_0 looks like this now:

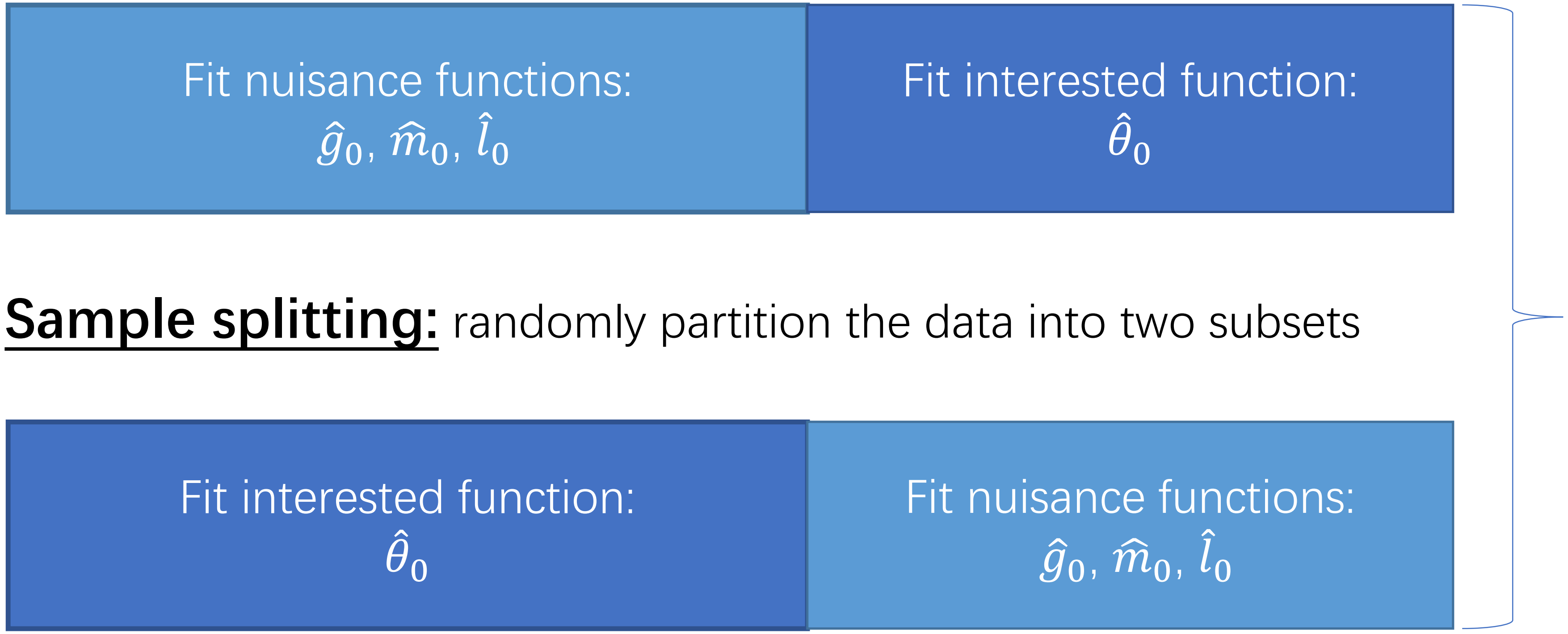


Method: Deal with Two Sources of Bias from ML

(1) Correct **Regularization** bias by **Neyman Orthogonal Structure**.
(2) Correct **Overfitting** bias by **Sample Splitting** with **cross-fitting** for efficiency.



Method: Sample Splitting with cross-fitting



Fit nuisance functions:
$\hat{g}_0, \widehat{m}_0, \hat{l}_0$

Fit interested function:
$\hat{\theta}_0$

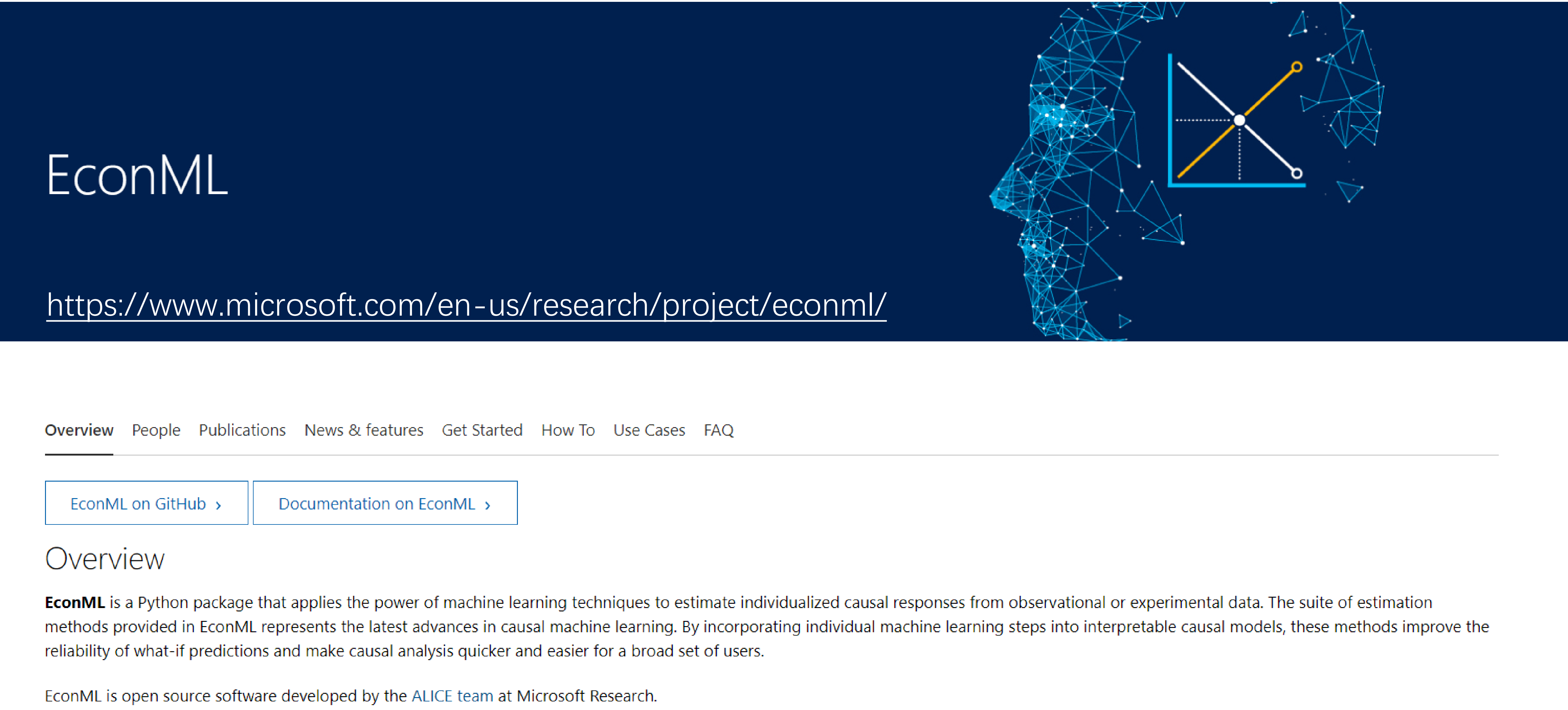
Sample splitting: randomly partition the data into two subsets

Fit interested function:
$\hat{\theta}_0$

Fit nuisance functions:
$\hat{g}_0, \widehat{m}_0, \hat{l}_0$

Cross-fitting:
average two estimated $\hat{\theta}_0$
for the final estimated $\hat{\theta}_0$

Method: EconML toolbox



EconML

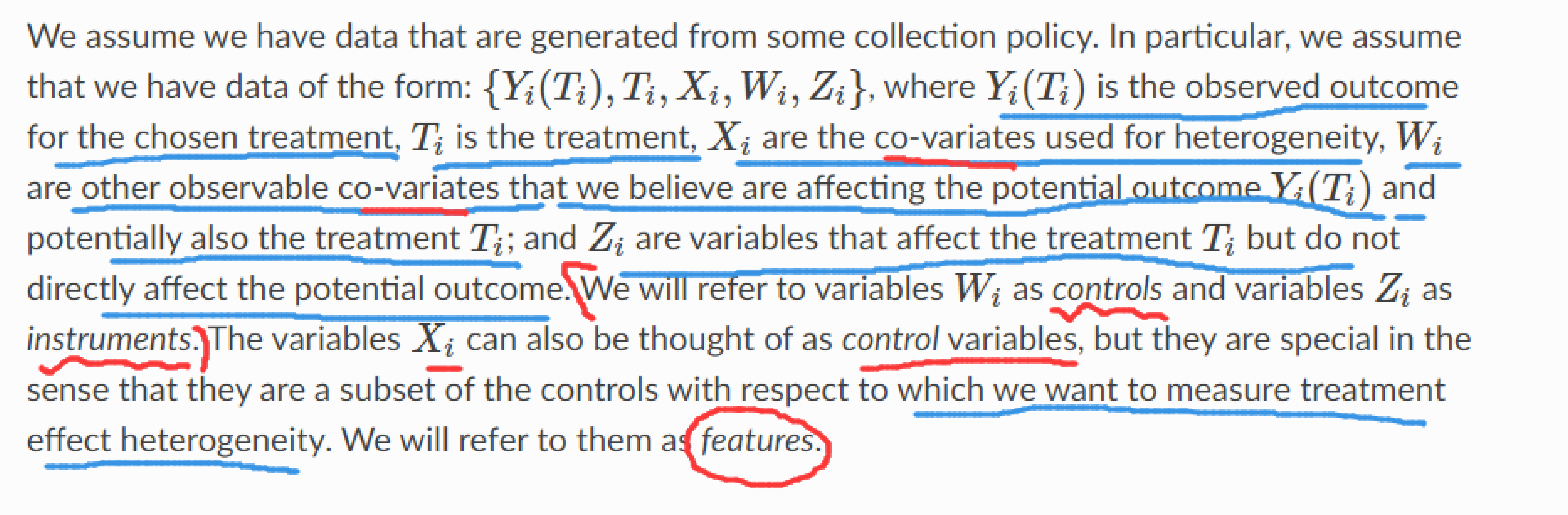
https://www.microsoft.com/en-us/research/project/econml/

Overview

EconML is a Python package that applies the power of machine learning techniques to estimate individualized causal responses from observational or experimental data. The suite of estimation methods provided in EconML represents the latest advances in causal machine learning. By incorporating individual machine learning steps into interpretable causal models, these methods improve the reliability of what-if predictions and make causal analysis quicker and easier for a broad set of users.

EconML is open source software developed by the ALICE team at Microsoft Research.

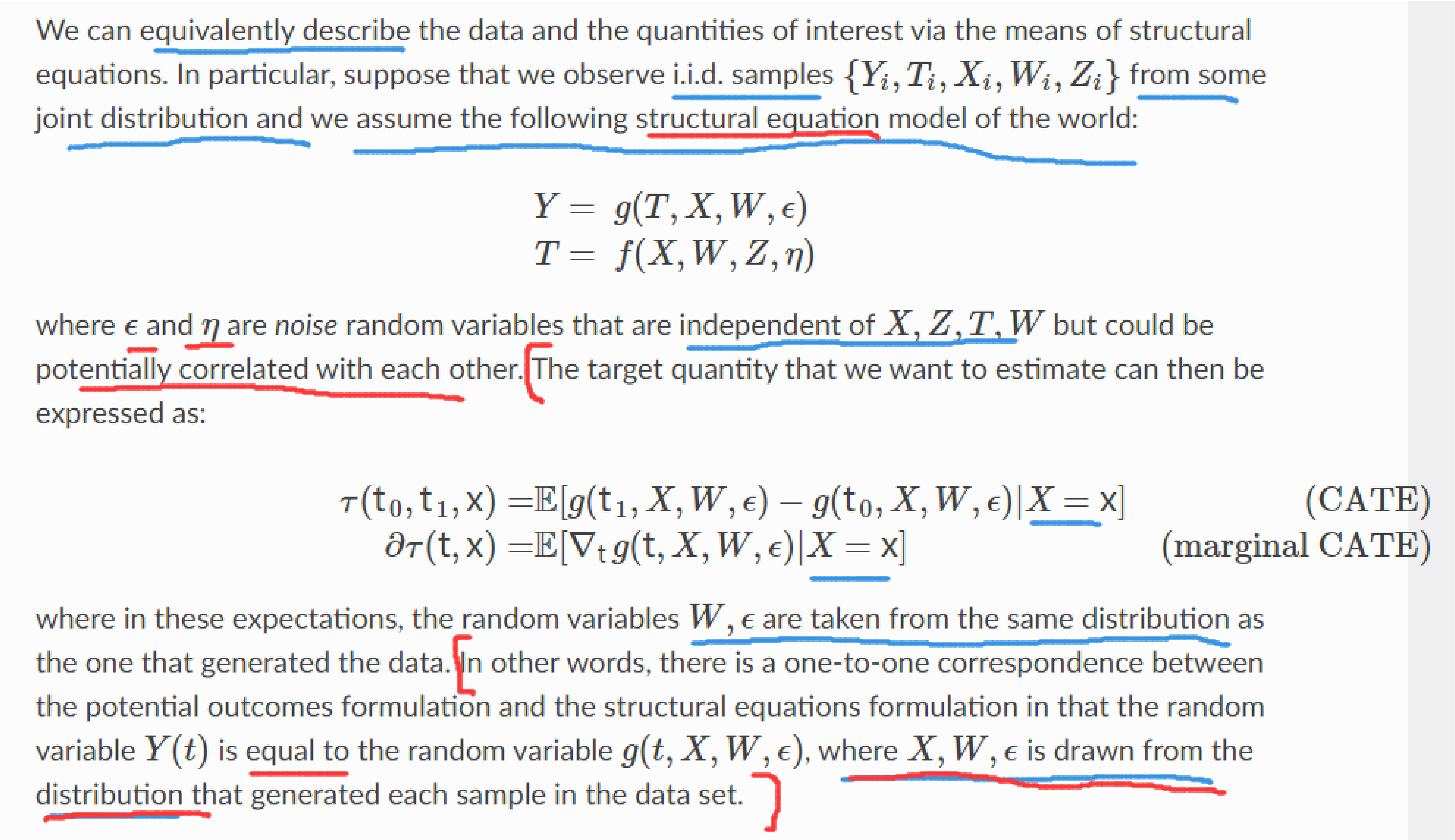
Method: Conditional Average Treatment Effect (CATE)



We assume we have data that are generated from some collection policy. In particular, we assume that we have data of the form: $\{Y_i(T_i), T_i, X_i, W_i, Z_i\}$, where $Y_i(T_i)$ is the observed outcome for the chosen treatment, T_i is the treatment, X_i are the co-variates used for heterogeneity, W_i are other observable co-variates that we believe are affecting the potential outcome $Y_i(T_i)$ and potentially also the treatment T_i; and Z_i are variables that affect the treatment T_i but do not directly affect the potential outcome. We will refer to variables W_i as *controls* and variables Z_i as *instruments*. The variables X_i can also be thought of as *control* variables, but they are special in the sense that they are a subset of the controls with respect to which we want to measure treatment effect heterogeneity. We will refer to them as *features*.

https://econml.azurewebsites.net/spec/api.html

Method: Conditional Average Treatment Effect (CATE)



We can equivalently describe the data and the quantities of interest via the means of structural equations. In particular, suppose that we observe i.i.d. samples $\{Y_i, T_i, X_i, W_i, Z_i\}$ from some joint distribution and we assume the following structural equation model of the world:

$$
\begin{aligned}
Y &= g(T, X, W, \epsilon) \\
T &= f(X, W, Z, \eta)
\end{aligned}
$$

where ϵ and η are *noise* random variables that are independent of X, Z, T, W but could be potentially correlated with each other. The target quantity that we want to estimate can then be expressed as:

$$
\begin{aligned}
\tau(\mathrm{t}_0, \mathrm{t}_1, \mathrm{x}) &= \mathbb{E}[g(\mathrm{t}_1, X, W, \epsilon) - g(\mathrm{t}_0, X, W, \epsilon) | X = \mathrm{x}] && \text{(CATE)} \\
\partial\tau(\mathrm{t}, \mathrm{x}) &= \mathbb{E}[\nabla_{\mathrm{t}} g(\mathrm{t}, X, W, \epsilon) | X = \mathrm{x}] && \text{(marginal CATE)}
\end{aligned}
$$

where in these expectations, the random variables W, ϵ are taken from the same distribution as the one that generated the data. In other words, there is a one-to-one correspondence between the potential outcomes formulation and the structural equations formulation in that the random variable $Y(t)$ is equal to the random variable $g(t, X, W, \epsilon)$, where X, W, ϵ is drawn from the distribution that generated each sample in the data set.

https://econml.azurewebsites.net/spec/api.html

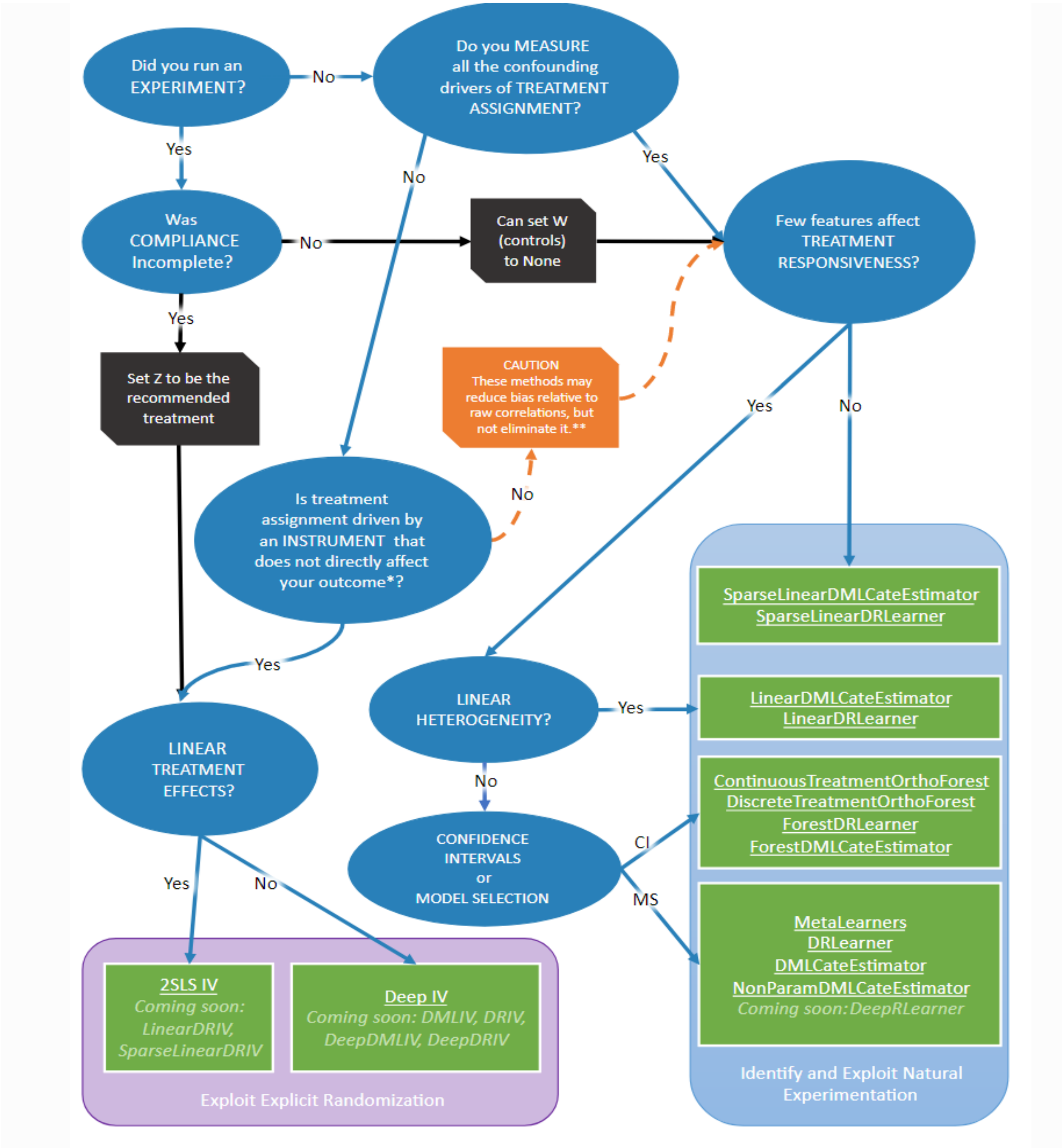
Method: Estimation Methods

Estimation:

- Double Machine Learning (DML)
- Doubly Robust Learning
- Forest Based Estimators
- Meta-Learners

Inference:

- Bootstrap Inference
- OLS Inference
- Debiased Lasso Inference
- Subsampled Honest Forest Inference



Did you run an EXPERIMENT?

Do you MEASURE all the confounding drivers of TREATMENT ASSIGNMENT?

Was COMPLIANCE Incomplete?

Can set W (controls) to None

Few features affect TREATMENT RESPONSIVENESS?

Set Z to be the recommended treatment

CAUTION These methods may reduce bias relative to raw correlations, but not eliminate it.**

Is treatment assignment driven by an INSTRUMENT that does not directly affect your outcome*?

LINEAR HETEROGENEITY?

LINEAR TREATMENT EFFECTS?

CONFIDENCE INTERVALS or MODEL SELECTION

SparseLinearDMLCateEstimator
SparseLinearDRLearner

LinearDMLCateEstimator
LinearDRLearner

ContinuousTreatmentOrthoForest
DiscreteTreatmentOrthoForest
ForestDRLearner
ForestDMLCateEstimator

MetaLearners
DRLearner
DMLCateEstimator
NonParamDMLCateEstimator
Coming soon:DeepRLearner

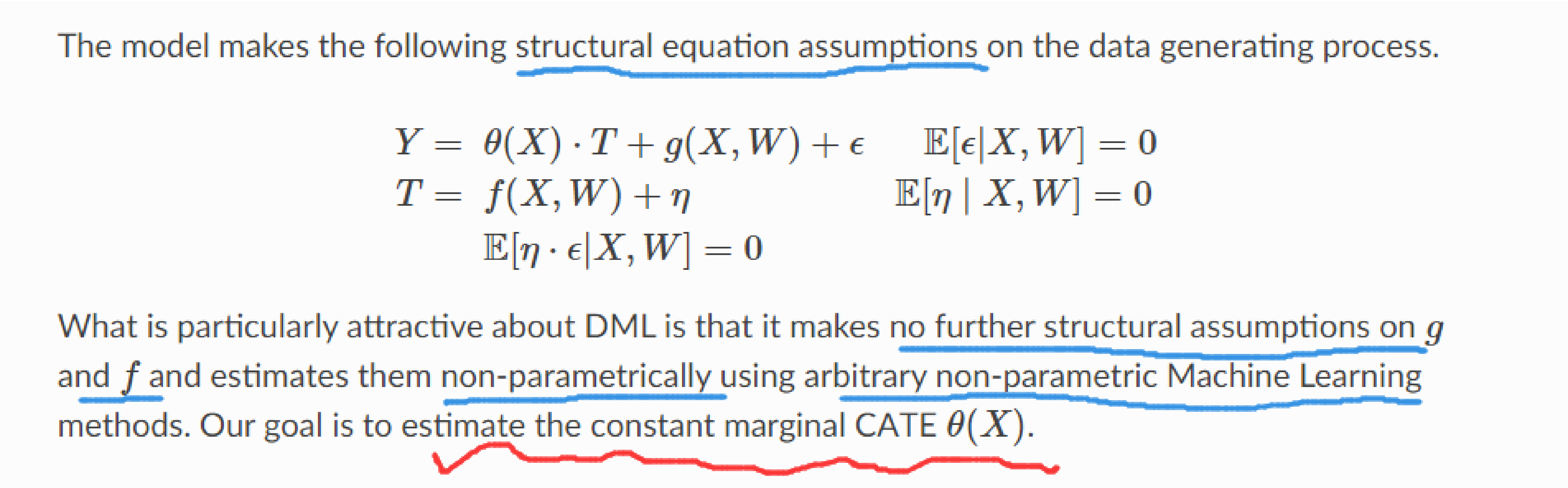
2SLS IV
Coming soon: LinearDRIV, SparseLinearDRIV

Deep IV
Coming soon: DMLIV, DRIV, DeepDMLIV, DeepDRIV

Exploit Explicit Randomization

Identify and Exploit Natural Experimentation

Method: Double Machine Learning (DML)

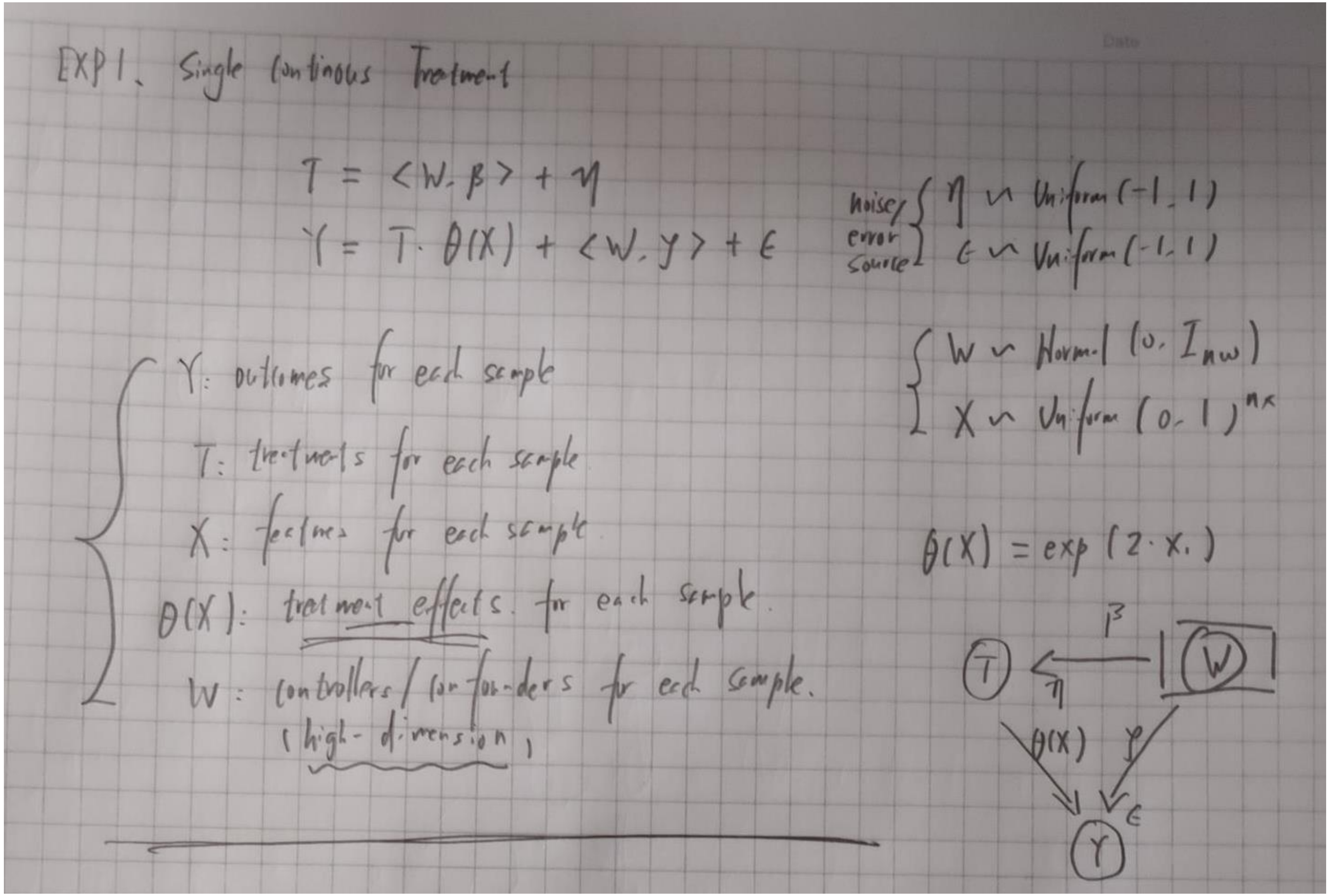


The model makes the following structural equation assumptions on the data generating process.

$$\begin{aligned} Y &= \theta(X) \cdot T + g(X, W) + \epsilon & \mathbb{E}[\epsilon | X, W] = 0 \\ T &= f(X, W) + \eta & \mathbb{E}[\eta \mid X, W] = 0 \\ & \mathbb{E}[\eta \cdot \epsilon | X, W] = 0 \end{aligned}$$

What is particularly attractive about DML is that it makes no further structural assumptions on g and f and estimates them non-parametrically using arbitrary non-parametric Machine Learning methods. Our goal is to estimate the constant marginal CATE $\theta(X)$.

Method: DML Example



EXP1、Single Continous Treatment

$$T = \langle W, \beta \rangle + \eta$$

$$Y = T \cdot \theta(X) + \langle W, \gamma \rangle + \epsilon$$

noise/ error source $\begin{cases} \eta \sim \text{Uniform}(-1, 1) \\ \epsilon \sim \text{Uniform}(-1, 1) \end{cases}$

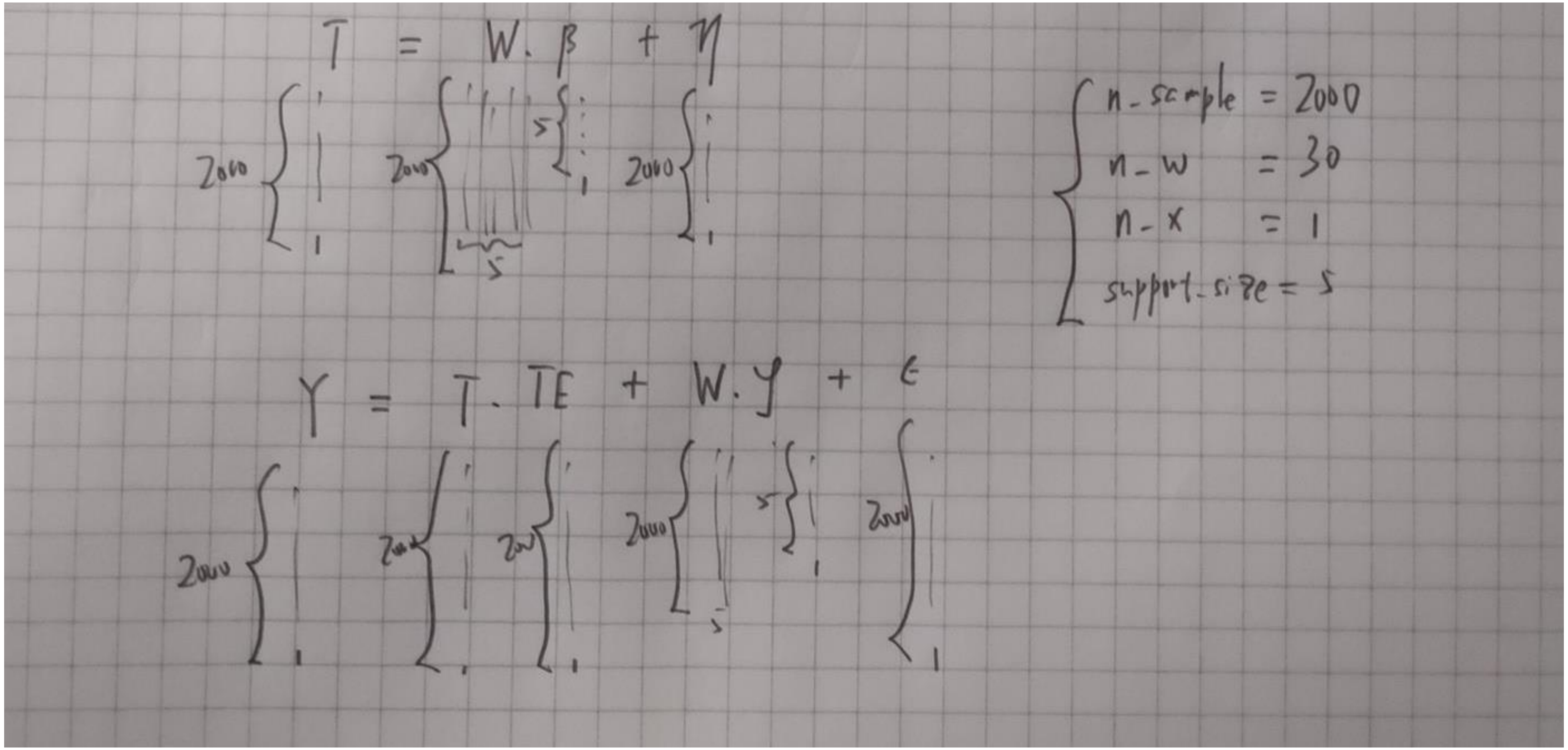
$\begin{cases} W \sim \text{Normal}(0, I_{nw}) \\ X \sim \text{Uniform}(0, 1)^{nx} \end{cases}$

$\theta(X) = \exp(2 \cdot x_1)$

- Y: outcomes for each sample
- T: treatments for each sample
- X: features for each sample
- θ(X): treatment effects for each sample
- W: controllers / confounders for each sample. (high-dimension)

https://github.com/microsoft/EconML/tree/master/notebooks

Method: DML Example



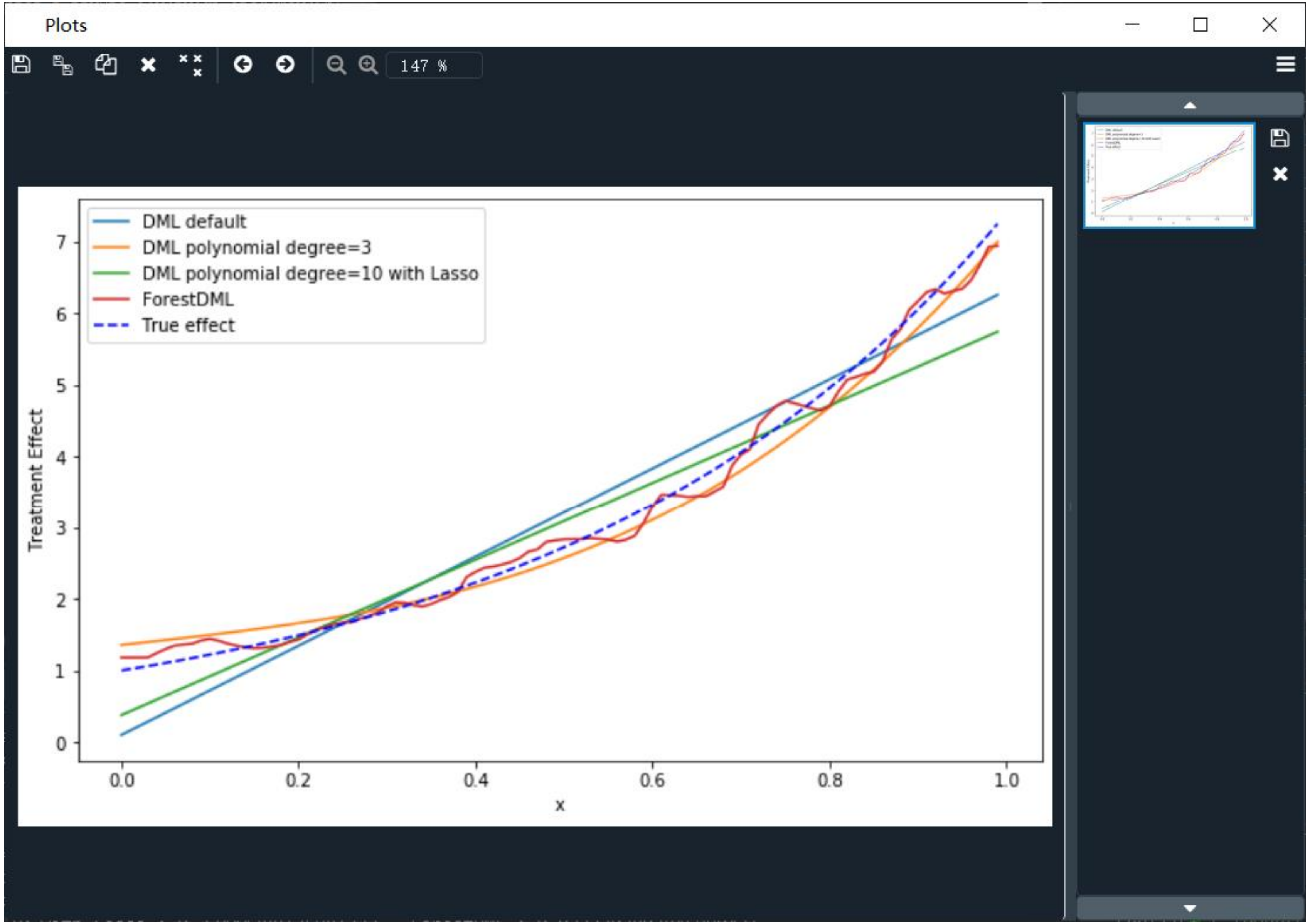
$$T = W \cdot \beta + \eta$$

$$\begin{cases} n\_sample = 2000 \\ n\_w = 30 \\ n\_x = 1 \\ support\_size = 5 \end{cases}$$

$$Y = T \cdot TE + W \cdot \gamma + \epsilon$$

2000
1
2000
5
5
1
2000
1
2000
1
2000
1
2000
1
2000
5
5
1
2000
1

Method: DML Example


Plots
147 %
DML default
DML polynomial degree=3
DML polynomial degree=10 with Lasso
ForestDML
True effect
Treatment Effect
0
1
2
3
4
5
6
7
0.0
0.2
0.4
0.6
0.8
1.0
x