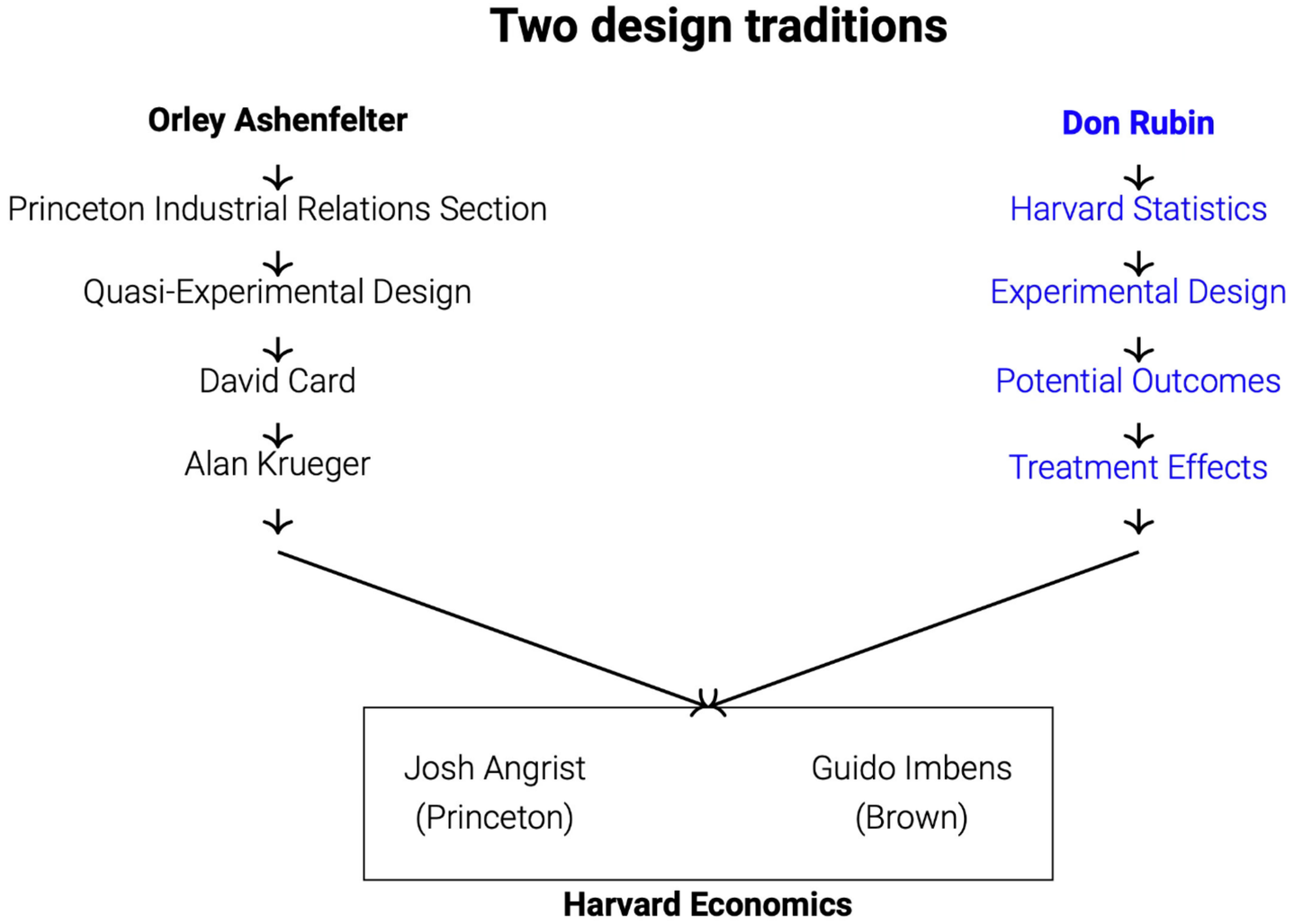
NATURAL EXPERIMENTS IN NLP AND WHERE TO FIND THEM

Pietro Lesci
University of Cambridge

X: @pietro_lesci
LinkedIn: /pietrolesci
Page: pietrolesci.github.io
Mail: pietrolesci@outlook.com

There is not one “causal inference”

There is not one “causal inference”



Source: “This and That Causal Inference”, Scott Cunningham, 2024.

CAUSAL ESTIMATION OF MEMORISATION PROFILES

Pietro Lesci, Clara Meister, Thomas Hofmann, Andreas Vlachos, Tiago Pimentel

Why study memorisation?

Why study memorisation?

Language models can reproduce entire sequences from their training set verbatim (Carlini, 2021; 2023).

Why study memorisation?

Language models can reproduce entire sequences from their training set verbatim (Carlini, 2021; 2023).

Memorisation has implications for:

- Copyright and data protection
- How models encode factual information
- Our understanding of models' training dynamics

Why study memorisation?

Language models can reproduce entire sequences from their training set verbatim (Carlini, 2021; 2023).

Memorisation has implications for:

- Copyright and data protection
- How models encode factual information
- Our understanding of models' training dynamics

In order to be able to study memorisation we need methods to quickly, cheaply, and accurately measure it!

Tl;dr: Estimate memorisation directly from the data!

Tl;dr: Estimate memorisation directly from the data!

We want to estimate **counterfactual memorisation**
(as defined in Feldman, 2020):

Tl;dr: Estimate memorisation directly from the data!

We want to estimate **counterfactual memorisation**
(as defined in Feldman, 2020):

"The causal effect of observing an instance during training on a model's ability to correctly predict that instance"

Tl;dr: Estimate memorisation directly from the data!

We want to estimate **counterfactual memorisation**
(as defined in Feldman, 2020):

"The causal effect of observing an instance during training on a model's ability to correctly predict that instance"

Tl;dr: Estimate memorisation directly from the data!

We want to estimate **counterfactual memorisation**
(as defined in Feldman, 2020):

"The causal effect of observing an instance during training on a model's ability to correctly predict that instance"

🤔 Problem: Current methods require re-training the model multiple times, which is infeasible for recent language models and datasets

Tl;dr: Estimate memorisation directly from the data!

We want to estimate **counterfactual memorisation**
(as defined in Feldman, 2020):

"The causal effect of observing an instance during training on a model's ability to correctly predict that instance"

🤔 Problem: Current methods require re-training the model multiple times, which is infeasible for recent language models and datasets

😃 Solution: Use econometrics to estimate causal effects directly from data (i.e., model evaluations)!

Tl;dr: Estimate memorisation directly from the data!

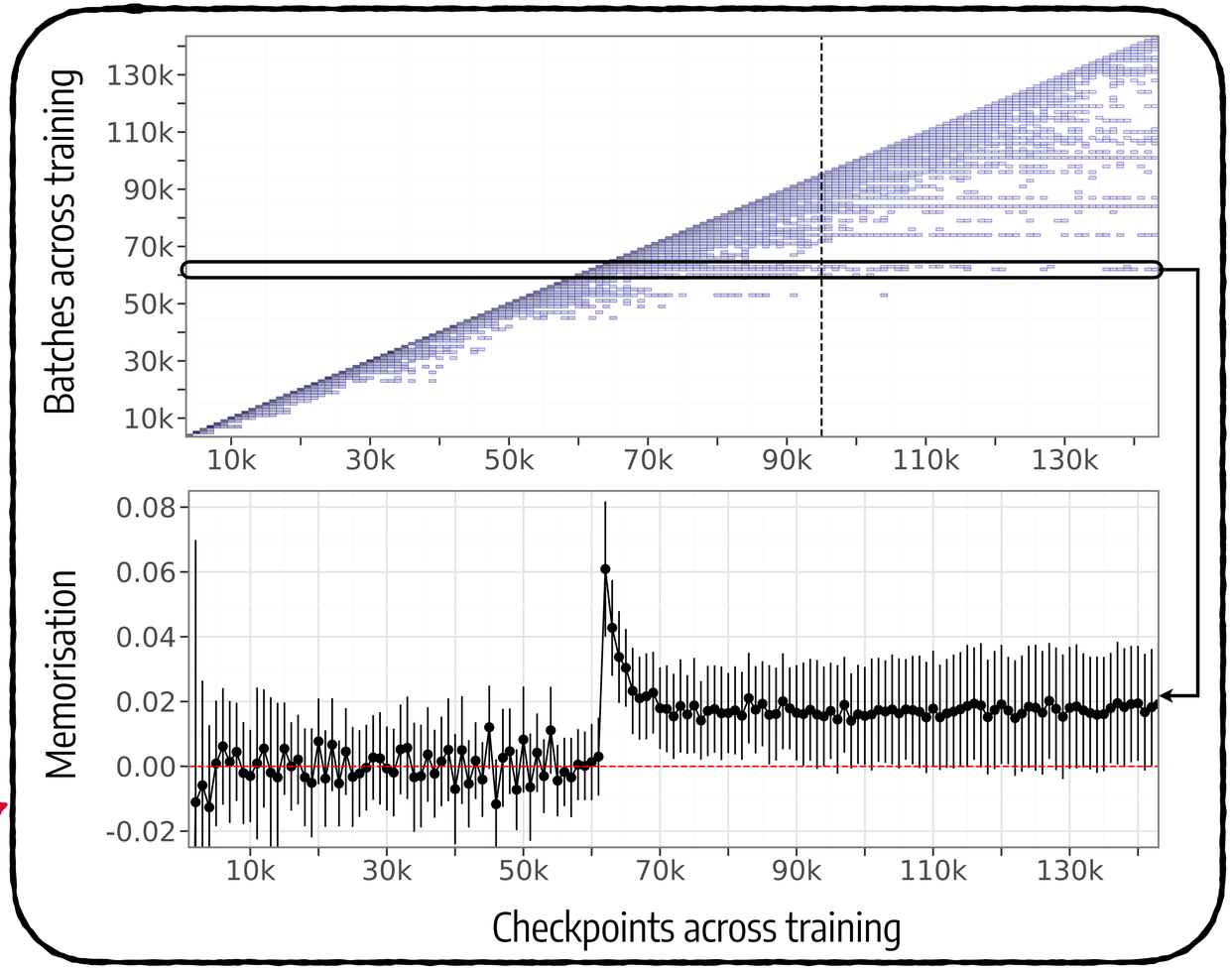
We want to estimate **counterfactual memorisation** (as defined in Feldman, 2020):

"The causal effect of observing an instance during training on a model's ability to correctly predict that instance"

🤔 Problem: Current methods require re-training the model multiple times, which is infeasible for recent language models and datasets

😀 Solution: Use econometrics to estimate causal effects directly from data (i.e., model evaluations)!

The output of our method: the Memorisation Profile



Memorisation as a difference of potential outcomes

Memorisation as a difference of potential outcomes

$$\tau_{\boldsymbol{x},c} \overset{\text{def}}{=} \underbrace{Y_c(\boldsymbol{x}; g)} - \underbrace{Y_c(\boldsymbol{x}; \infty)}$$

Memorisation as a difference of potential outcomes

$$\tau_{\boldsymbol{x},c} \overset{\text{def}}{=} \underbrace{Y_c(\boldsymbol{x}; g)}_{\downarrow} - \underbrace{Y_c(\boldsymbol{x}; \infty)}_{}$$

Memorisation as a difference of potential outcomes

$$\tau_{\boldsymbol{x},c} \overset{\text{def}}{=} \underbrace{Y_c(\boldsymbol{x}; g)}_{\downarrow} - \underbrace{Y_c(\boldsymbol{x}; \infty)}_{}$$

Memorisation as a difference of potential outcomes

$$\tau_{\boldsymbol{x},c} \overset{\text{def}}{=} \underbrace{Y_c(\boldsymbol{x}; g)}_{\text{Performance of model on } \boldsymbol{x} \text{ when trained with } \boldsymbol{x} \text{ at step } g} - \underbrace{Y_c(\boldsymbol{x}; \infty)}$$

Memorisation as a difference of potential outcomes

$$\tau_{\boldsymbol{x},c} \overset{\text{def}}{=} \underbrace{Y_c(\boldsymbol{x}; g)}_{\text{Performance of model on } \boldsymbol{x} \text{ when trained with } \boldsymbol{x} \text{ at step } g} - \underbrace{Y_c(\boldsymbol{x}; \infty)}_{\downarrow}$$

Memorisation as a difference of potential outcomes

$$\tau_{\boldsymbol{x},c} \overset{\text{def}}{=} \underbrace{Y_c(\boldsymbol{x}; g)}_{\text{Performance of model on } \boldsymbol{x} \text{ when trained with } \boldsymbol{x} \text{ at step } g} - \underbrace{Y_c(\boldsymbol{x}; \infty)}_{\text{Performance of model on } \boldsymbol{x} \text{ when } \textbf{not} \text{ trained with } \boldsymbol{x}}$$

Memorisation as a difference of potential outcomes

$$\tau_{\boldsymbol{x},c} \overset{\text{def}}{=} \underbrace{Y_c(\boldsymbol{x}; g)}_{\text{Performance of model on } \boldsymbol{x} \text{ when trained with } \boldsymbol{x} \text{ at step } g} - \underbrace{Y_c(\boldsymbol{x}; \infty)}_{\text{Performance of model on } \boldsymbol{x} \text{ when } \textbf{not} \text{ trained with } \boldsymbol{x}}$$

Memorisation as a difference of potential outcomes

$$\tau_{\boldsymbol{x},c} \overset{\text{def}}{=} \underbrace{Y_c(\boldsymbol{x}; g)}_{\text{Performance of model on } \boldsymbol{x} \text{ when trained with } \boldsymbol{x} \text{ at step } g} - \underbrace{Y_c(\boldsymbol{x}; \infty)}_{\text{Performance of model on } \boldsymbol{x} \text{ when } \textbf{not} \text{ trained with } \boldsymbol{x}}$$

For each training run, we observe only one of the two potential outcomes!

Estimating the counterfactual outcome $Y_c(\boldsymbol{x}; \infty)$

Estimating the counterfactual outcome $Y_c(x; \infty)$

(k,l)-extractable memorisation (Carlini et al., 2023):

Assume the counterfactual performance is negligible in the absence of training

Estimating the counterfactual outcome $Y_c(\boldsymbol{x}; \infty)$

(k,l)-extractable memorisation (Carlini et al., 2023):

Assume the counterfactual performance is negligible in the absence of training

$$\tau^{\texttt{extr}}_{\boldsymbol{x},c} = Y_c(\boldsymbol{x}; g)$$

Estimating the counterfactual outcome $Y_c(\boldsymbol{x}; \infty)$

(k,l)-extractable memorisation (Carlini et al., 2023):

Assume the counterfactual performance is negligible in the absence of training

$$\tau^{\texttt{extr}}_{\boldsymbol{x},c} = Y_c(\boldsymbol{x}; g)$$

- Strong assumption
- Does not account for a “baseline” predictability

Estimating the counterfactual outcome $Y_c(\boldsymbol{x}; \infty)$

(k,l)-extractable memorisation (Carlini et al., 2023):

Assume the counterfactual performance is negligible in the absence of training

$$\tau^{\texttt{extr}}_{\boldsymbol{x},c} = Y_c(\boldsymbol{x}; g)$$

- Strong assumption
- Does not account for a “baseline” predictability

Architectural memorisation (Feldman, 2020; and later work):

Re-train the model multiple times and average across runs

Estimating the counterfactual outcome $Y_c(\boldsymbol{x}; \infty)$

(k,l)-extractable memorisation (Carlini et al., 2023):

Assume the counterfactual performance is negligible in the absence of training

$$\tau^{\texttt{extr}}_{\boldsymbol{x},c} = Y_c(\boldsymbol{x}; g)$$

- Strong assumption
- Does not account for a “baseline” predictability

Architectural memorisation (Feldman, 2020; and later work):

Re-train the model multiple times and average across runs

$$\tau^{\texttt{arch}}_{\boldsymbol{x},p(\psi)} = \mathop{\mathbb{E}}_{\boldsymbol{\psi}}\left[Y_T(\boldsymbol{x}; G(\boldsymbol{x})) \mid G(\boldsymbol{x}) \neq \infty\right] - \mathop{\mathbb{E}}_{\boldsymbol{\psi}}\left[Y_T(\boldsymbol{x}; \infty) \mid G(\boldsymbol{x}) = \infty\right]$$

Estimating the counterfactual outcome $Y_c(\boldsymbol{x}; \infty)$

(k,l)-extractable memorisation (Carlini et al., 2023):

Assume the counterfactual performance is negligible in the absence of training

$$\tau^{\texttt{extr}}_{\boldsymbol{x},c} = Y_c(\boldsymbol{x}; g)$$

- Strong assumption
- Does not account for a "baseline" predictability

Architectural memorisation (Feldman, 2020; and later work):

Re-train the model multiple times and average across runs

$$\tau^{\texttt{arch}}_{\boldsymbol{x},p(\psi)} = \mathop{\mathbb{E}}_{\boldsymbol{\psi}}\left[Y_T(\boldsymbol{x}; G(\boldsymbol{x})) \mid G(\boldsymbol{x}) \neq \infty\right] - \mathop{\mathbb{E}}_{\boldsymbol{\psi}}\left[Y_T(\boldsymbol{x}; \infty) \mid G(\boldsymbol{x}) = \infty\right]$$

- Computationally expensive
- Estimates memorisation for a model architecture rather than a specific model
- Does not account for data order
- No insights on the training dynamics

From instance-level to batch-level memorisation

$$\tau_{\boldsymbol{x},c} \overset{\text{def}}{=} Y_c(\boldsymbol{x}; g) - Y_c(\boldsymbol{x}; \infty)$$

From instance-level to batch-level memorisation

$$\tau_{g,c} \overset{\text{def}}{=} \underbrace{\mathop{\mathbb{E}}_{\boldsymbol{x}}\left[Y_c(\boldsymbol{x}; g) \mid G(\boldsymbol{x}) = g\right]} - \underbrace{\mathop{\mathbb{E}}_{\boldsymbol{x}}\left[Y_c(\boldsymbol{x}; \infty) \mid G(\boldsymbol{x}) = g\right]}$$

From instance-level to batch-level memorisation

$$\tau_{g,c} \overset{\text{def}}{=} \underbrace{\mathop{\mathbb{E}}_{\boldsymbol{x}}\left[Y_c(\boldsymbol{x}; g) \mid G(\boldsymbol{x}) = g\right]}_{\text{Average performance on batch when trained on batch at step } g} - \underbrace{\mathop{\mathbb{E}}_{\boldsymbol{x}}\left[Y_c(\boldsymbol{x}; \infty) \mid G(\boldsymbol{x}) = g\right]}$$

From instance-level to batch-level memorisation

$$\tau_{g,c} \overset{\text{def}}{=} \underbrace{\mathop{\mathbb{E}}_{\boldsymbol{x}}\left[Y_c(\boldsymbol{x}; g) \mid G(\boldsymbol{x}) = g\right]}_{\text{Average performance on batch when trained on batch at step } g} - \underbrace{\mathop{\mathbb{E}}_{\boldsymbol{x}}\left[Y_c(\boldsymbol{x}; \infty) \mid G(\boldsymbol{x}) = g\right]}_{\text{Average performance on batch when \textbf{not} trained on batch given it is selected for training at step } g}$$

From instance-level to batch-level memorisation

Training Batch

Validation Batch

$$\tau_{g,c} \overset{\text{def}}{=} \underbrace{\mathop{\mathbb{E}}_{\boldsymbol{x}}\left[Y_c(\boldsymbol{x}; g) \mid G(\boldsymbol{x}) = g\right]}_{\text{Average performance on batch when trained on batch at step } g} - \underbrace{\mathop{\mathbb{E}}_{\boldsymbol{x}}\left[Y_c(\boldsymbol{x}; \infty) \mid G(\boldsymbol{x}) = g\right]}_{\text{Average performance on batch when } \textbf{not} \text{ trained on batch given it is selected for training at step } g}$$

From instance-level to batch-level memorisation

Training Batch Validation Batch

$$\tau_{g,c} \overset{\text{def}}{=} \underbrace{\mathop{\mathbb{E}}_{\boldsymbol{x}}\left[Y_c(\boldsymbol{x}; g) \mid \boxed{}\right]}_{\text{Average performance on batch when trained on batch at step } g} - \underbrace{\mathop{\mathbb{E}}_{\boldsymbol{x}}\left[Y_c(\boldsymbol{x}; \infty) \mid \boxed{}\right]}_{\text{Average performance on batch when \textbf{not} trained on batch given it is selected for training at step } g}$$

From instance-level to batch-level memorisation

Training Batch

Validation Batch

✅ g and 🟨 = Observable

∞ and 🟦 = Observable

❌ g and 🟦 = Not Observable

∞ and 🟨 = Not Observable

$$\tau_{g,c} \overset{\text{def}}{=} \underbrace{\mathop{\mathbb{E}}_{\boldsymbol{x}}\left[Y_c(\boldsymbol{x}; g) \mid \square\right]}_{\text{Average performance on batch when trained on batch at step } g} - \underbrace{\mathop{\mathbb{E}}_{\boldsymbol{x}}\left[Y_c(\boldsymbol{x}; \infty) \mid \square\right]}_{\text{Average performance on batch when \textbf{not} trained on batch given it is selected for training at step } g}$$

From instance-level to batch-level memorisation

Training Batch

Validation Batch

✅ g and 🟨 = Observable

∞ and 🟦 = Observable

❌ g and 🟦 = Not Observable

∞ and 🟨 = Not Observable

$$\tau_{g,c} \overset{\text{def}}{=} \underbrace{\mathop{\mathbb{E}}_{x}\left[Y_c(x; g) \mid \text{🟨}\right]}_{\text{Average performance on batch when trained on batch at step } g} - \underbrace{\mathop{\mathbb{E}}_{x}\left[Y_c(x; \infty) \mid \text{🟨}\right]}_{\text{Average performance on batch when \textbf{not} trained on batch given it is selected for training at step } g}$$

From instance-level to batch-level memorisation

Training Batch

Validation Batch

✅ g and 🟨 = Observable

∞ and 🟦 = Observable

❌ g and 🟦 = Not Observable

∞ and 🟨 = Not Observable

$$\tau_{g,c} \overset{\text{def}}{=} \underbrace{\mathop{\mathbb{E}}_{x}\left[Y_c(x; g) \mid \text{🟨} \text{✅}\right]}_{\text{Average performance on batch when trained on batch at step } g} - \underbrace{\mathop{\mathbb{E}}_{x}\left[Y_c(x; \infty) \mid \text{🟨} \text{❌}\right]}_{\text{Average performance on batch when \textbf{not} trained on batch given it is selected for training at step } g}$$

On the way to estimating the counterfactual

Training Batch

Validation Batch

✅ g and = Observable

∞ and = Observable

❌ g and = Not Observable

∞ and = Not Observable

$$\tau_{g,c} = \mathop{\mathbb{E}}_{\boldsymbol{x}}\left[Y_c(\boldsymbol{x}; g) \mid \square\right] - \mathop{\mathbb{E}}_{\boldsymbol{x}}\left[Y_c(\boldsymbol{x}; \infty) \mid \square\right]$$

On the way to estimating the counterfactual

Training Batch

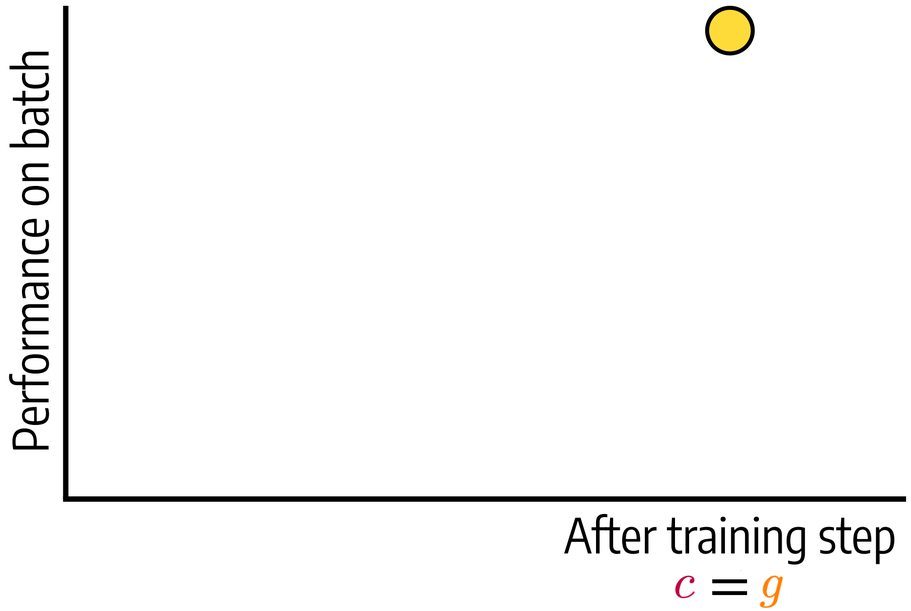
Validation Batch

✅ g and 🟨 = Observable

✅ ∞ and 🟦 = Observable

❌ g and 🟦 = Not Observable

❌ ∞ and 🟨 = Not Observable



$$\mathop{\mathbb{E}}_{\boldsymbol{x}}\left[Y_c(\boldsymbol{x}; g) \mid \text{🟨}\right]$$

$$\tau_{g,c} = \mathop{\mathbb{E}}_{\boldsymbol{x}}\left[Y_c(\boldsymbol{x}; g) \mid \text{🟨}\right] - \mathop{\mathbb{E}}_{\boldsymbol{x}}\left[Y_c(\boldsymbol{x}; \infty) \mid \text{🟨}\right]$$

On the way to estimating the counterfactual

Training Batch

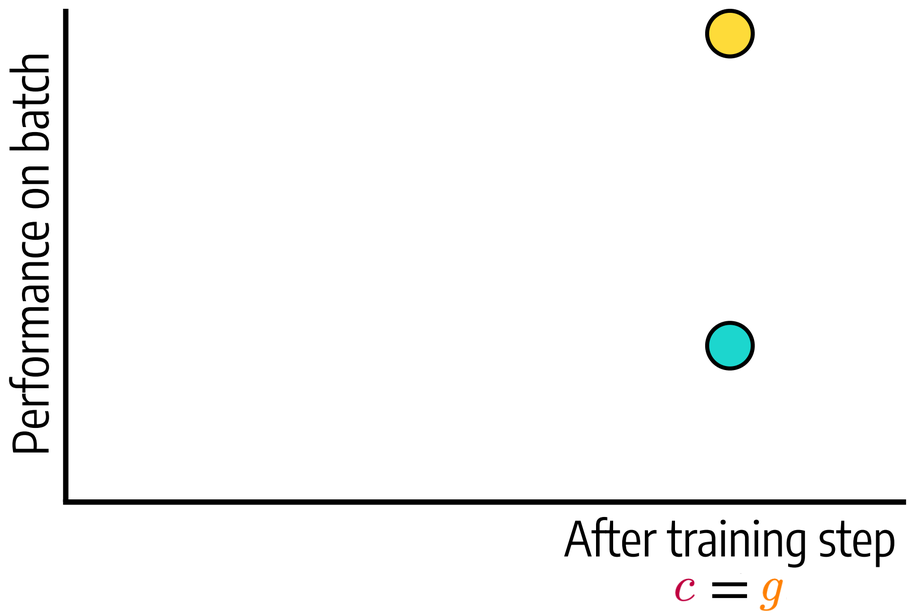
Validation Batch

✅ g and 🟨 = Observable

✅ ∞ and 🟦 = Observable

❌ g and 🟦 = Not Observable

❌ ∞ and 🟨 = Not Observable



$$\mathop{\mathbb{E}}_{x}\left[Y_c(x; g) \mid \text{🟨}\right]$$

$$\mathop{\mathbb{E}}_{x}\left[Y_c(x; \infty) \mid \text{🟦}\right]$$

$$\tau_{g,c} = \mathop{\mathbb{E}}_{x}\left[Y_c(x; g) \mid \text{🟨}\right] - \mathop{\mathbb{E}}_{x}\left[Y_c(x; \infty) \mid \text{🟨}\right]$$

On the way to estimating the counterfactual

Training Batch

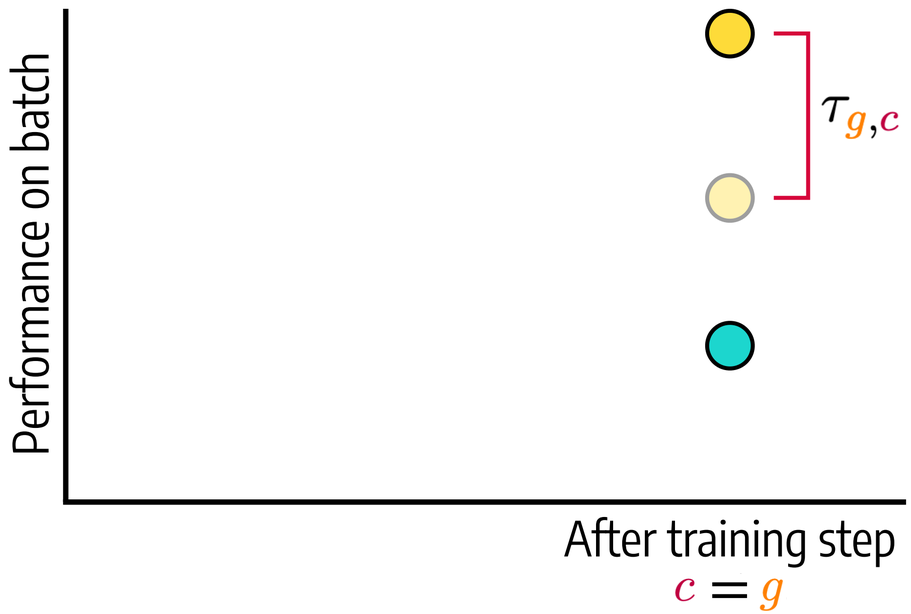
Validation Batch

g and = Observable

∞ and = Observable

g and = Not Observable

∞ and = Not Observable



$$\mathop{\mathbb{E}}_{x}\left[Y_c(x; g) \mid \right]$$

$$\mathop{\mathbb{E}}_{x}\left[Y_c(x; \infty) \mid \right]$$

$$\mathop{\mathbb{E}}_{x}\left[Y_c(x; \infty) \mid \right]$$

$$\tau_{g,c} = \mathop{\mathbb{E}}_{x}\left[Y_c(x; g) \mid \right] - \mathop{\mathbb{E}}_{x}\left[Y_c(x; \infty) \mid \right]$$

On the way to estimating the counterfactual

Training Batch

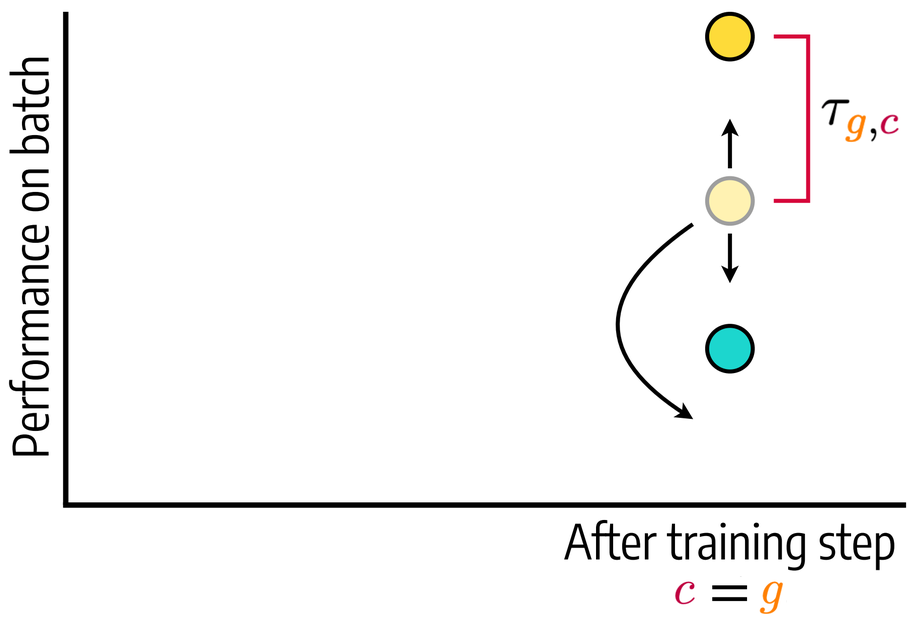
Validation Batch

✅ g and 🟨 = Observable

∞ and 🟦 = Observable

❌ g and 🟦 = Not Observable

∞ and 🟨 = Not Observable



$$\mathop{\mathbb{E}}_{x}\left[Y_c(x; g) \mid \text{🟨} \checkmark\right]$$

$$\mathop{\mathbb{E}}_{x}\left[Y_c(x; \infty) \mid \text{🟨} \times\right]$$

$$\mathop{\mathbb{E}}_{x}\left[Y_c(x; \infty) \mid \text{🟦} \checkmark\right]$$

$$\tau_{g,c} = \mathop{\mathbb{E}}_{x}\left[Y_c(x; g) \mid \text{🟨} \checkmark\right] - \mathop{\mathbb{E}}_{x}\left[Y_c(x; \infty) \mid \text{🟨} \times\right]$$

The Difference estimator

Training Batch

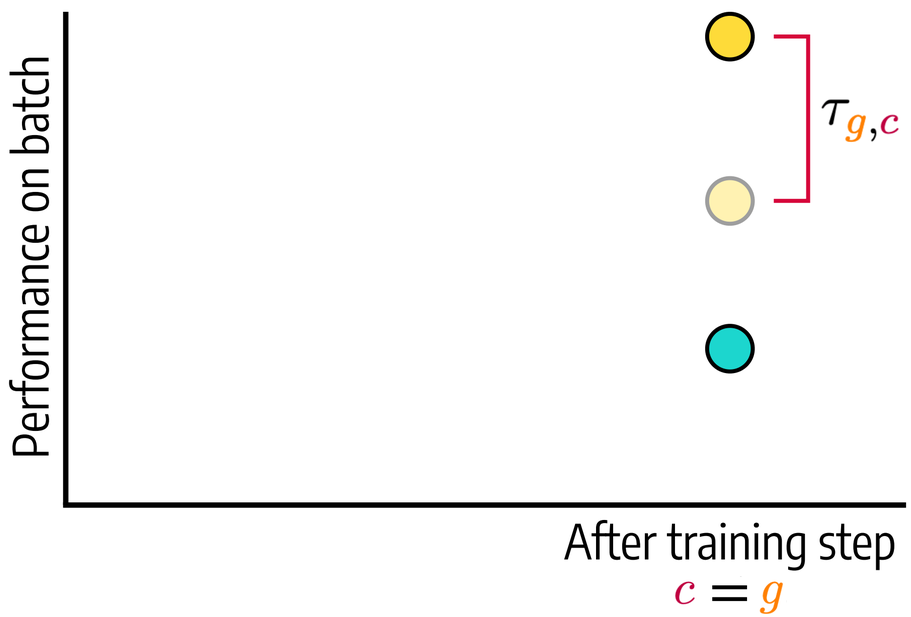
Validation Batch

✅ g and 🟨 = Observable

∞ and 🟦 = Observable

❌ g and 🟦 = Not Observable

∞ and 🟨 = Not Observable



$$\mathop{\mathbb{E}}_{\boldsymbol{x}}\left[Y_c(\boldsymbol{x}; g) \mid \text{🟨}\right] \checkmark$$

$$\mathop{\mathbb{E}}_{\boldsymbol{x}}\left[Y_c(\boldsymbol{x}; \infty) \mid \text{🟨}\right]$$

$$\mathop{\mathbb{E}}_{\boldsymbol{x}}\left[Y_c(\boldsymbol{x}; \infty) \mid \text{🟦}\right] \checkmark$$

$$\tau_{g,c} = \mathop{\mathbb{E}}_{\boldsymbol{x}}\left[Y_c(\boldsymbol{x}; g) \mid \text{🟨}\right] \checkmark - \mathop{\mathbb{E}}_{\boldsymbol{x}}\left[Y_c(\boldsymbol{x}; \infty) \mid \text{🟨}\right] \times$$

The Difference estimator

Training Batch

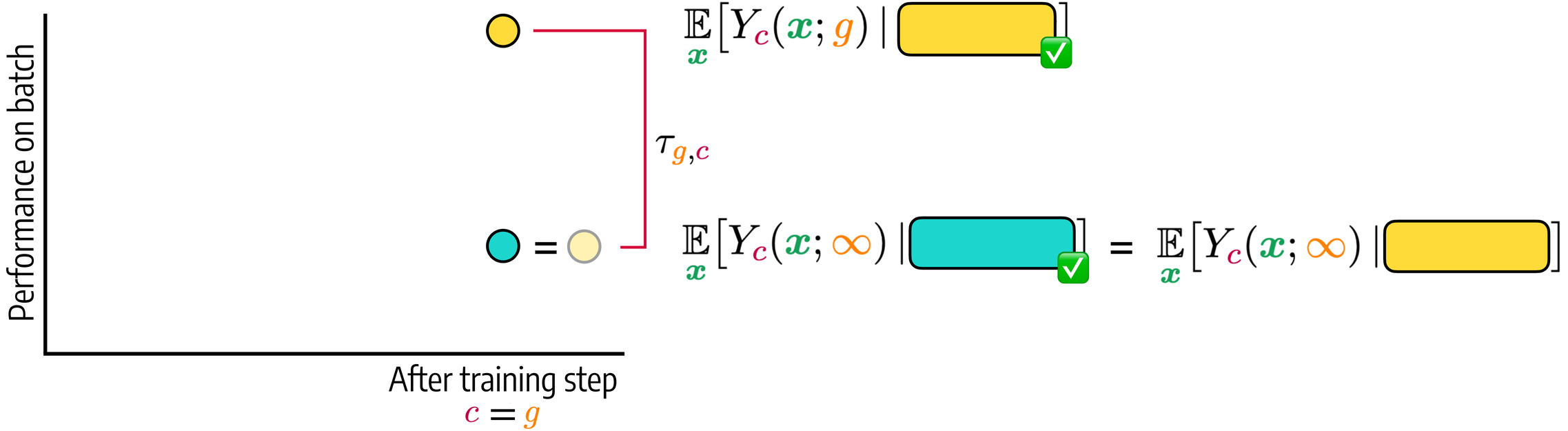
Validation Batch

✅ g and 🟨 = Observable

✅ ∞ and 🟦 = Observable

❌ g and 🟦 = Not Observable

❌ ∞ and 🟨 = Not Observable



IID Assumption

$$\tau_{g,c} = \mathbb{E}_{\boldsymbol{x}}\left[Y_c(\boldsymbol{x}; g) \mid \right] - \mathbb{E}_{\boldsymbol{x}}\left[Y_c(\boldsymbol{x}; \infty) \mid \right]$$

The Difference estimator

Training Batch

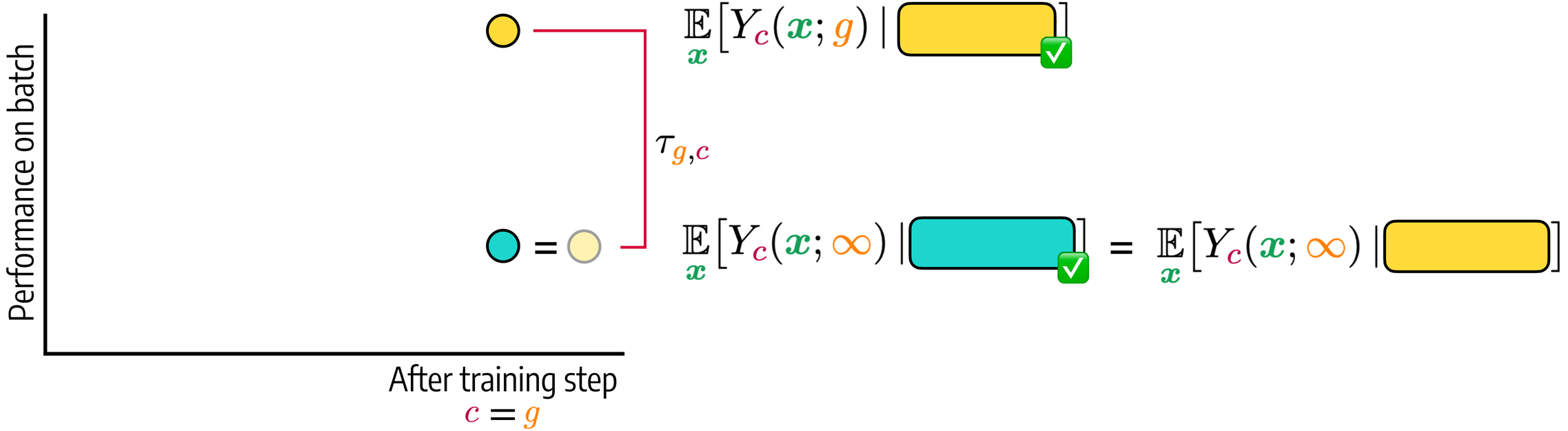
Validation Batch

✅ g and 🟨 = Observable

✅ ∞ and 🟦 = Observable

❌ g and 🟦 = Not Observable

❌ ∞ and 🟨 = Not Observable



IID Assumption

$$\tau^{\texttt{diff}}_{g,c} = \mathbb{E}_{\boldsymbol{x}}[Y_c(\boldsymbol{x}; g) \mid \text{🟨 ✅}] - \mathbb{E}_{\boldsymbol{x}}[Y_c(\boldsymbol{x}; \infty) \mid \text{🟦 ✅}]$$

The Difference-in-Differences estimator (1/2)

Training Batch

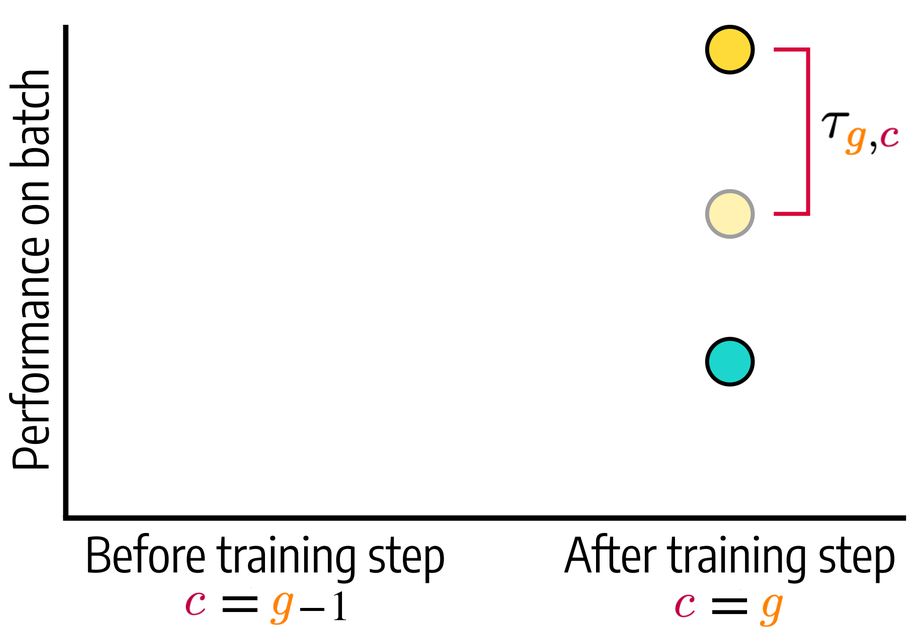
Validation Batch

✅ g and = Observable

✅ ∞ and = Observable

❌ g and = Not Observable

❌ ∞ and = Not Observable



$\mathop{\mathbb{E}}_{x}[Y_c(x; g) \mid$ ✅

$\mathop{\mathbb{E}}_{x}[Y_c(x; \infty) \mid$ ✅

The Difference-in-Differences estimator (1/2)

Training Batch

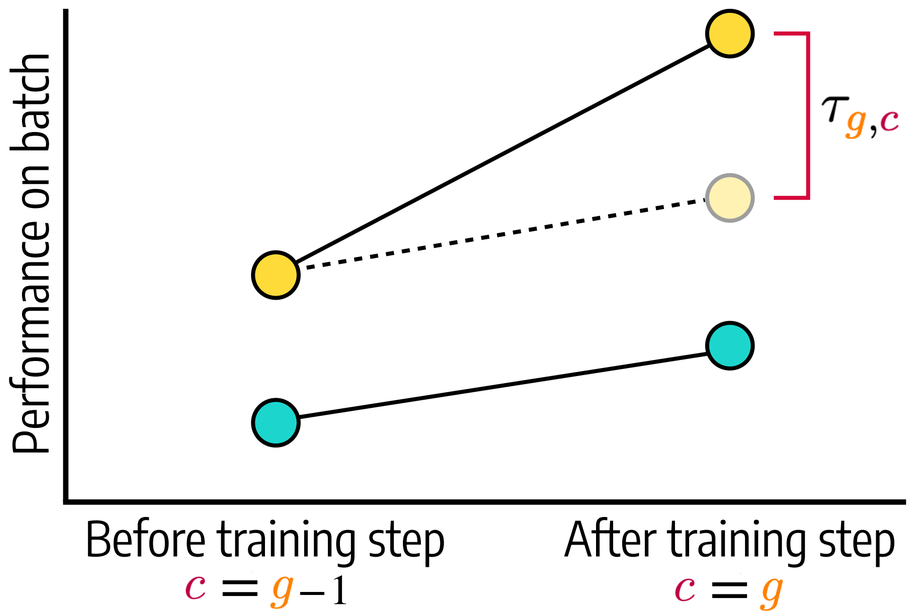
Validation Batch

✅ g and 🟨 = Observable

✅ ∞ and 🟦 = Observable

❌ g and 🟦 = Not Observable

❌ ∞ and 🟨 = Not Observable



$$\mathop{\mathbb{E}}_{x}\left[Y_c(x; g) \mid \text{🟨} \checkmark\right]$$

$$\mathop{\mathbb{E}}_{x}\left[Y_c(x; \infty) \mid \text{🟦} \checkmark\right]$$

The Difference-in-Differences estimator (1/2)

Training Batch

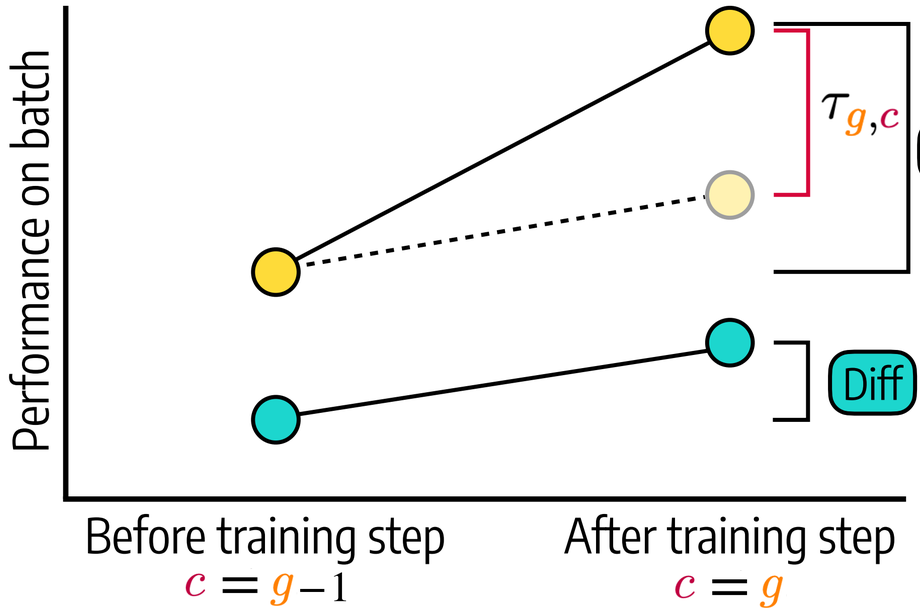
Validation Batch

✅ g and 🟨 = Observable

✅ ∞ and 🟦 = Observable

❌ g and 🟦 = Not Observable

❌ ∞ and 🟨 = Not Observable



Diff = $\mathop{\mathbb{E}}_{x}\left[Y_c(x; g) \mid \text{🟨}\right] - \mathop{\mathbb{E}}_{x}\left[Y_{g-1}(x; g) \mid \text{🟨}\right]$

Diff = $\mathop{\mathbb{E}}_{x}\left[Y_c(x; \infty) \mid \text{🟦}\right] - \mathop{\mathbb{E}}_{x}\left[Y_{g-1}(x; \infty) \mid \text{🟦}\right]$

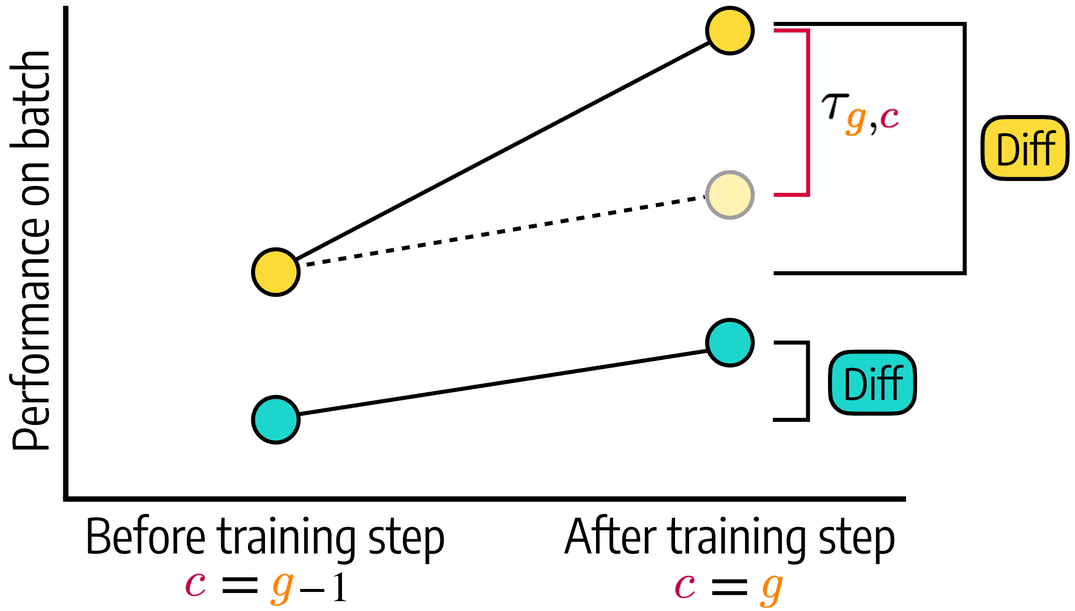
The Difference-in-Differences estimator (2/2)

Training Batch

Validation Batch

✅ g and = Observable
∞ and = Observable

❌ g and = Not Observable
∞ and = Not Observable



The Difference-in-Differences estimator (2/2)

Training Batch

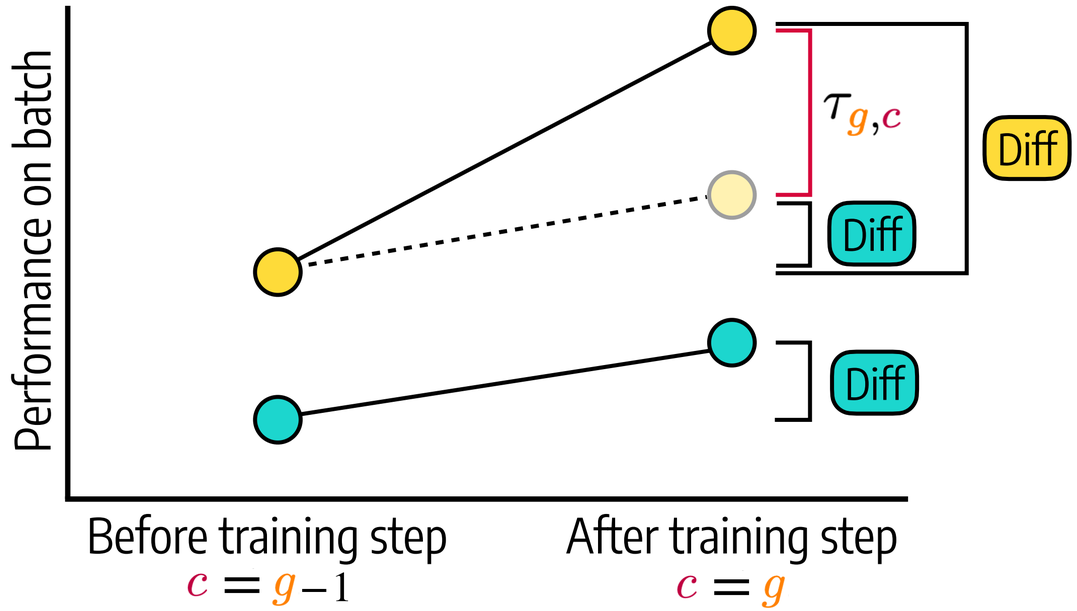
Validation Batch

✅ g and = Observable

∞ and = Observable

❌ g and = Not Observable

∞ and = Not Observable



Parallel Trend Assumption

The Difference-in-Differences estimator (2/2)

Training Batch

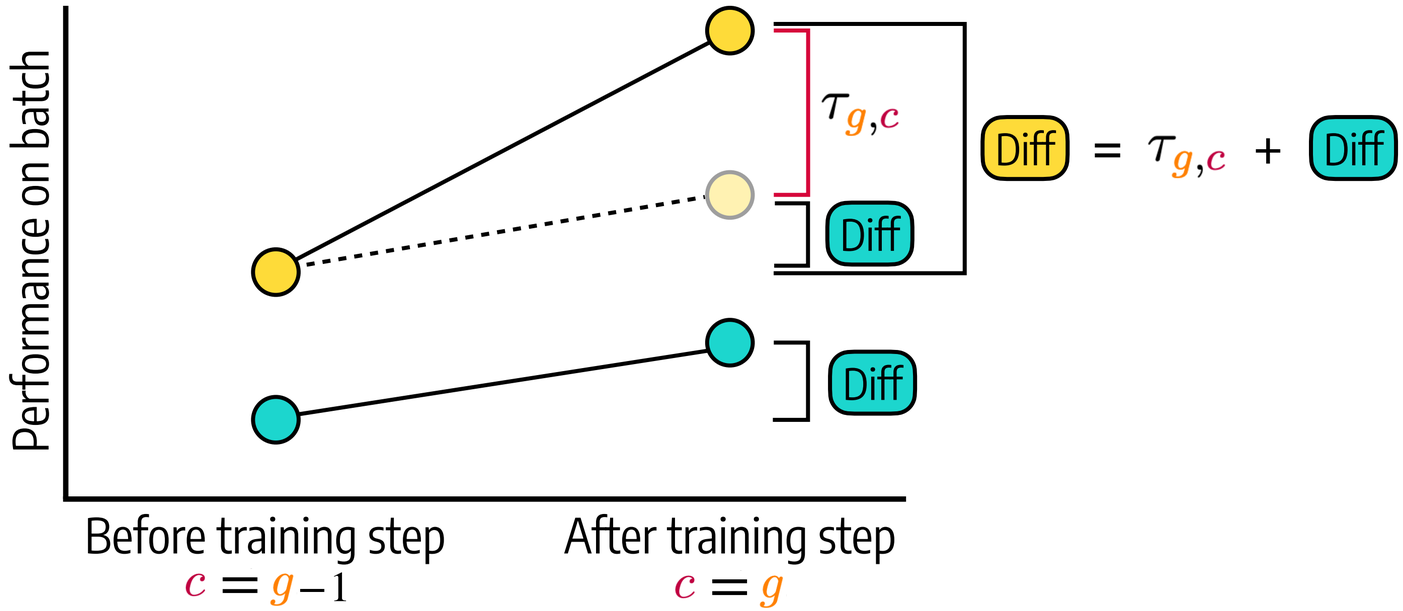
Validation Batch

✅ g and 🟨 = Observable

✅ ∞ and 🟦 = Observable

❌ g and 🟦 = Not Observable

❌ ∞ and 🟨 = Not Observable



Diff = $\tau_{g,c}$ + Diff

Parallel Trend Assumption

The Difference-in-Differences estimator (2/2)

Training Batch

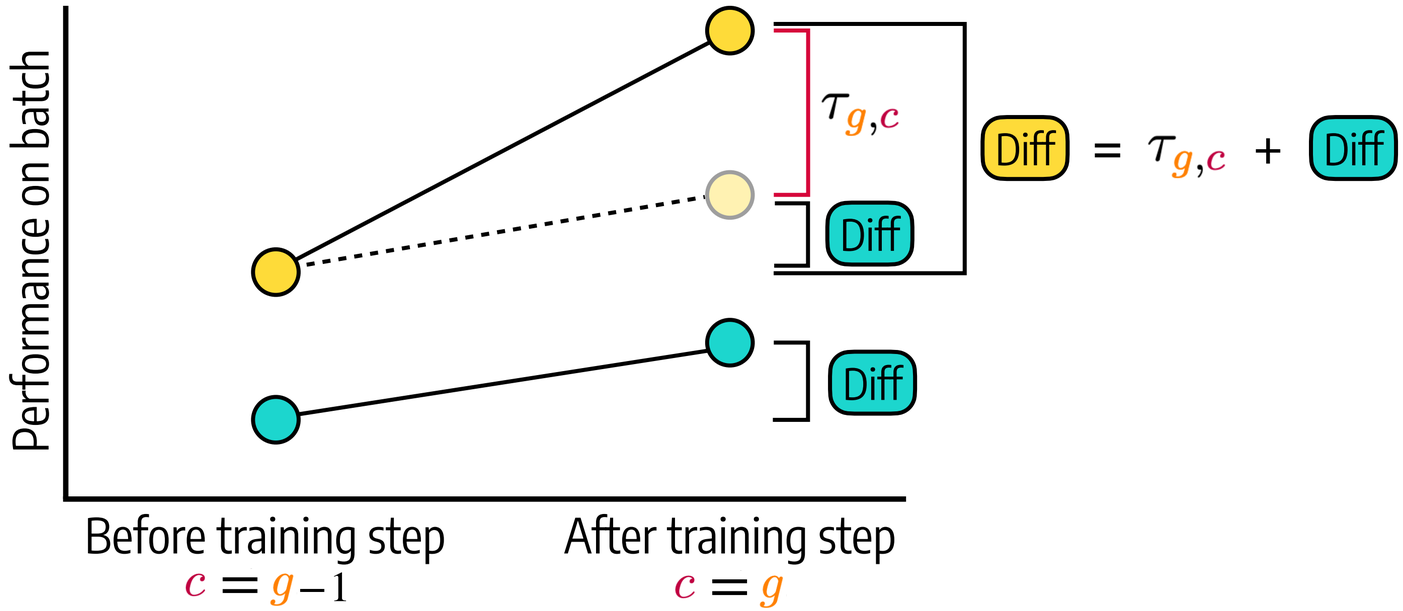
Validation Batch

✅ g and 🟨 = Observable

✅ ∞ and 🟦 = Observable

❌ g and 🟦 = Not Observable

❌ ∞ and 🟨 = Not Observable



Diff = $\tau_{g,c}$ + Diff

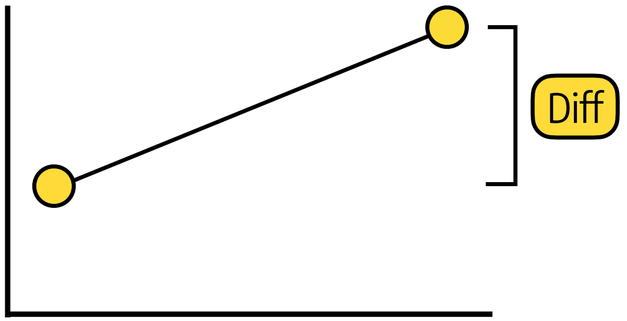
Parallel Trend Assumption

$$\tau^{\texttt{did}}_{g,c} = \text{Diff} - \text{Diff}$$

Difference-in-Differences in practice

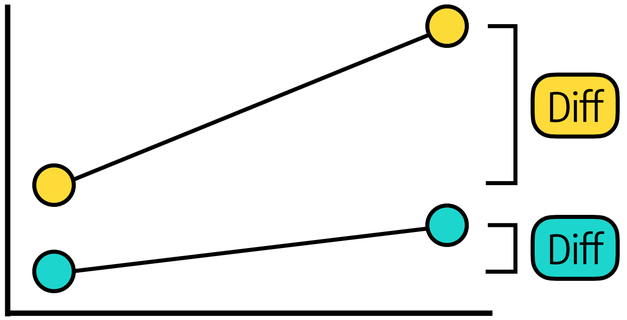
Difference-in-Differences in practice

$$\tau^{\texttt{did}}_{g,c} = \mathop{\mathbb{E}}_{\boldsymbol{x}}[Y_c(\boldsymbol{x}; g) - Y_{g-1}(\boldsymbol{x}; g) \,|\, \square]$$ ✅



Difference-in-Differences in practice

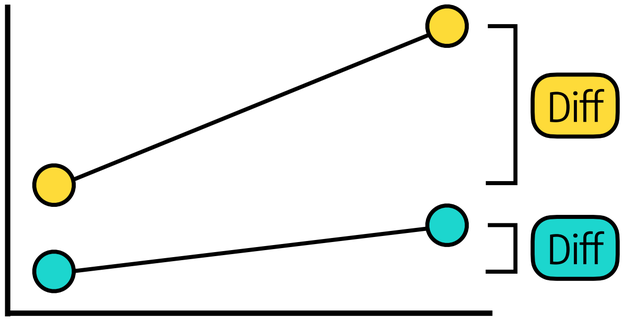
$$\tau^{\texttt{did}}_{g,c} = \mathop{\mathbb{E}}_{\boldsymbol{x}}[Y_c(\boldsymbol{x}; g) - Y_{g-1}(\boldsymbol{x}; g) \mid \boxed{}] - \mathop{\mathbb{E}}_{\boldsymbol{x}}[Y_c(\boldsymbol{x}; \infty) - Y_{g-1}(\boldsymbol{x}; \infty) \mid \boxed{}]$$



Difference-in-Differences in practice

$$\tau^{\texttt{did}}_{g,c} = \mathop{\mathbb{E}}_{\boldsymbol{x}}[Y_c(\boldsymbol{x}; g) - Y_{g-1}(\boldsymbol{x}; g) \mid \boxed{}] - \mathop{\mathbb{E}}_{\boldsymbol{x}}[Y_c(\boldsymbol{x}; \infty) - Y_{g-1}(\boldsymbol{x}; \infty) \mid \boxed{}]$$

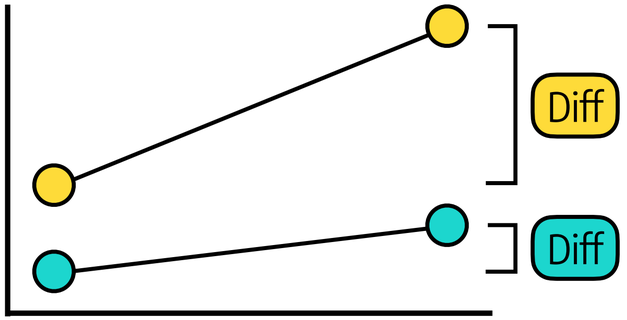
$$\widehat{\tau}^{\texttt{did}}_{g,c} = \underbrace{\left(\overline{Y}_c(g) - \overline{Y}_{g-1}(g)\right)}_{\text{Performance difference on training batches}}$$



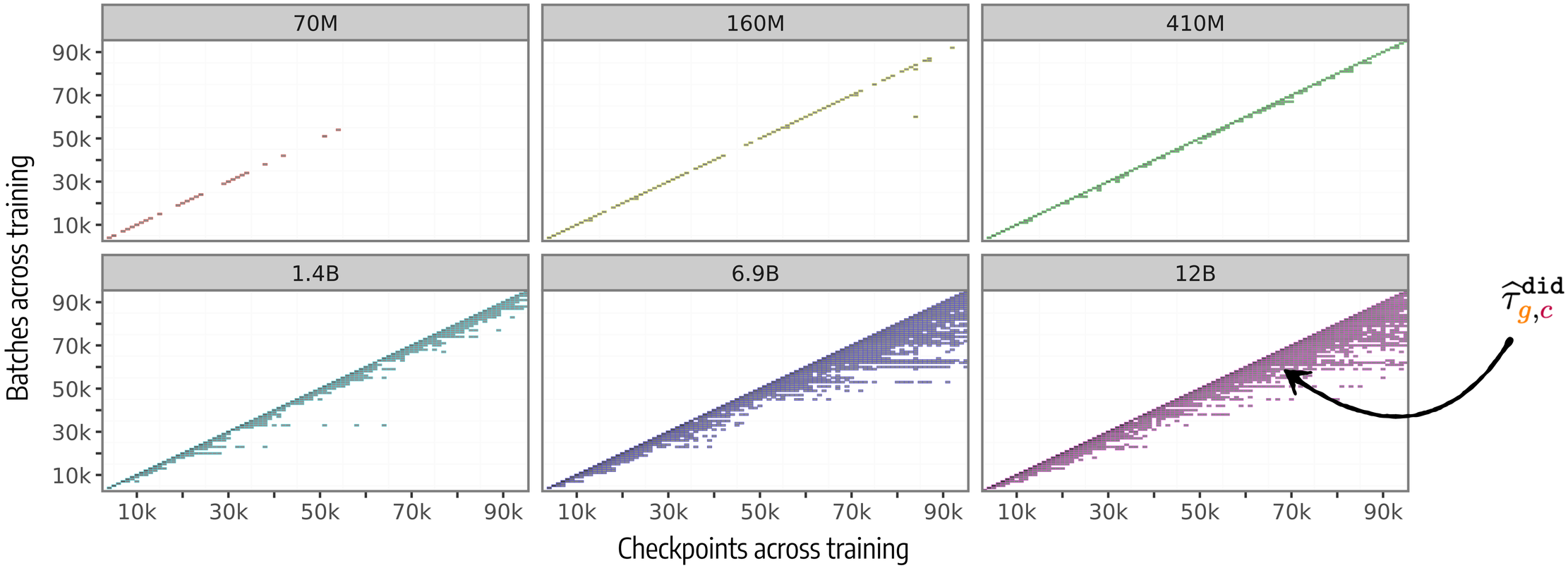
Difference-in-Differences in practice

$$\tau^{\texttt{did}}_{g,c} = \mathop{\mathbb{E}}_{\boldsymbol{x}}[Y_c(\boldsymbol{x}; g) - Y_{g-1}(\boldsymbol{x}; g) \mid \boxed{}] - \mathop{\mathbb{E}}_{\boldsymbol{x}}[Y_c(\boldsymbol{x}; \infty) - Y_{g-1}(\boldsymbol{x}; \infty) \mid \boxed{}]$$

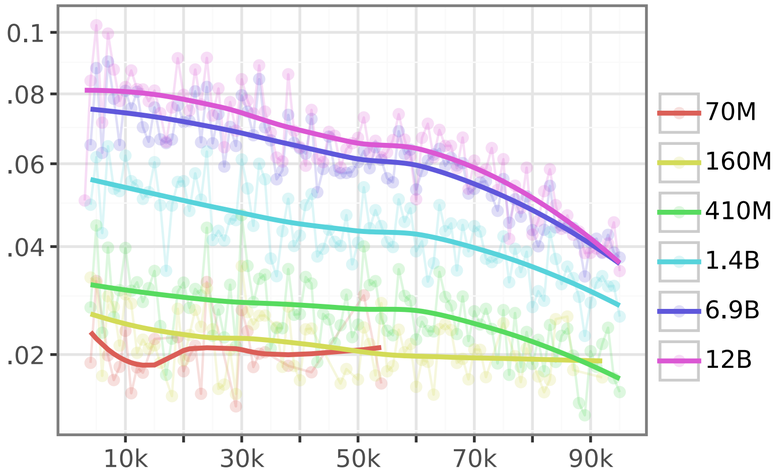
$$\widehat{\tau}^{\texttt{did}}_{g,c} = \underbrace{\left(\overline{Y}_c(g) - \overline{Y}_{g-1}(g)\right)}_{\text{Performance difference on training batches}} - \underbrace{\left(\overline{Y}_c(\infty) - \overline{Y}_{g-1}(\infty)\right)}_{\text{Performance difference on validation batches}}$$



Constructing the memorisation profile

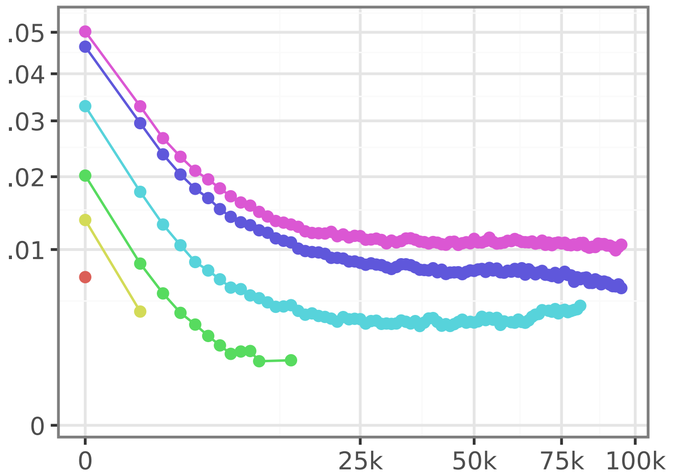


Aggregating the memorisation profile



Checkpoints/Batches across training ($g = c$)

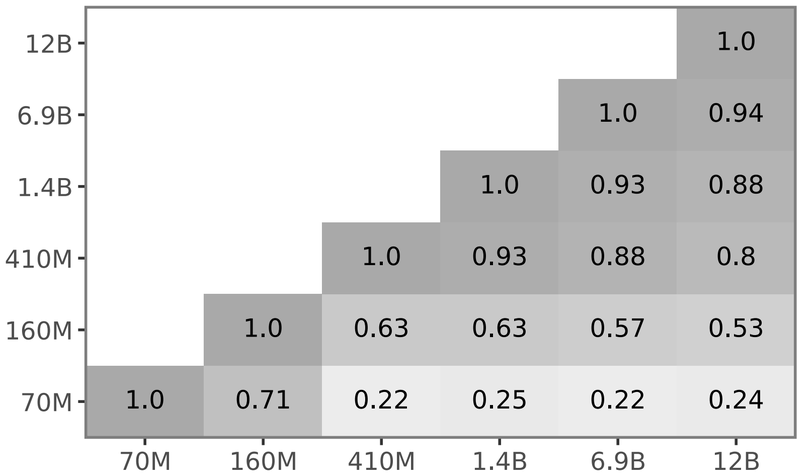
Instantaneous memorisation of a batch at the checkpoint is first seen



Checkpoints after batch is seen ($g > c$)

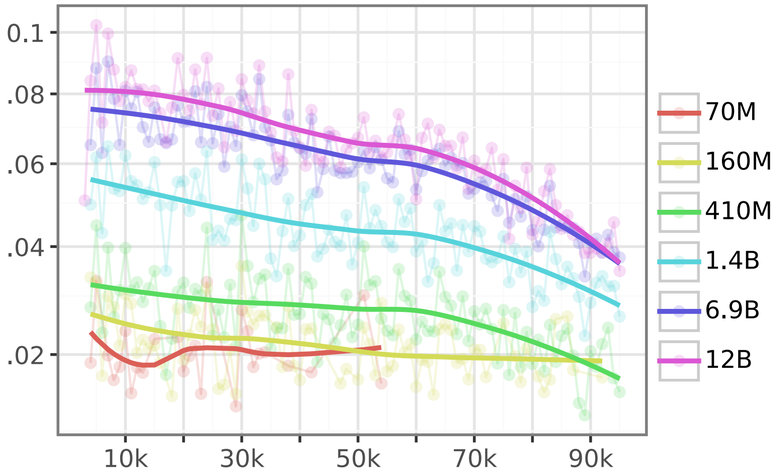
Average persistent memorisation of a batch per step after it has been seen*

*(only averaging across batches that have been trained that much)



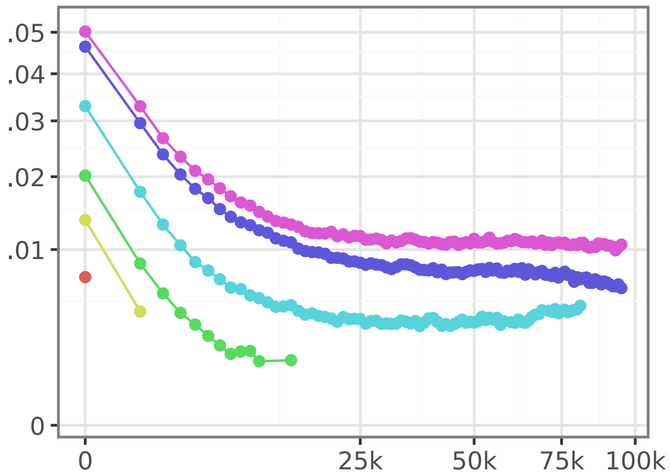
Pearson correlation between the memorisation profiles of different model sizes

Aggregating the memorisation profile



Checkpoints/Batches across training ($g = c$)

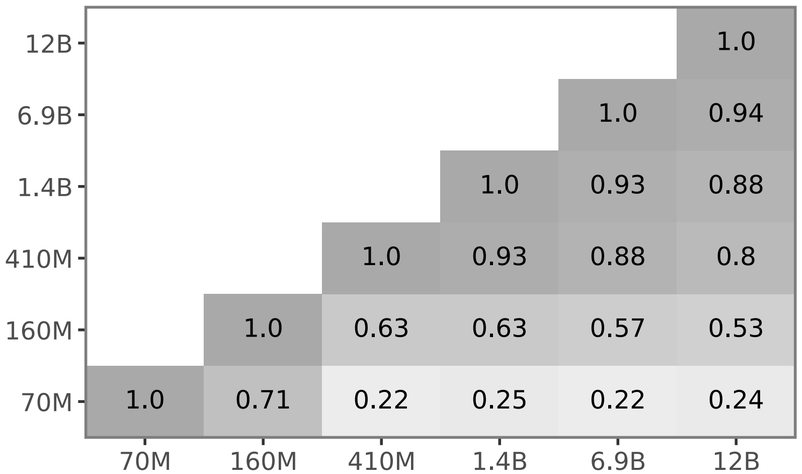
Instantaneous memorisation of a batch at the checkpoint is first seen



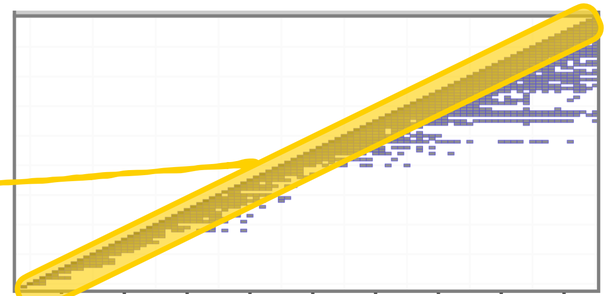
Checkpoints after batch is seen ($g > c$)

Average persistent memorisation of a batch per step after it has been seen*

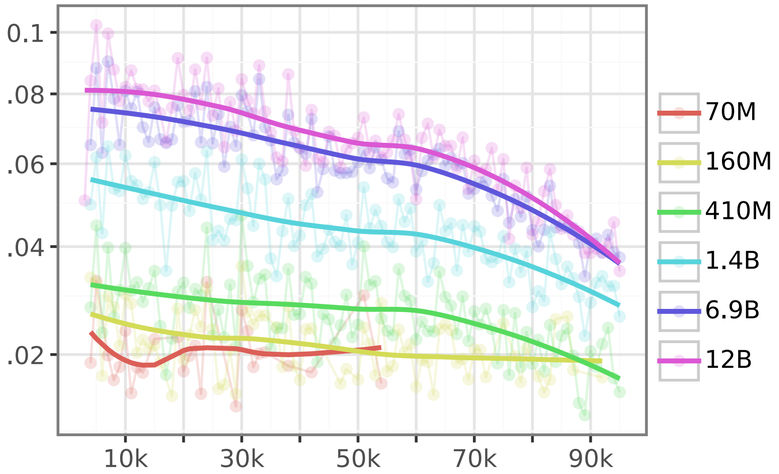
*(only averaging across batches that have been trained that much)



Pearson correlation between the memorisation profiles of different model sizes

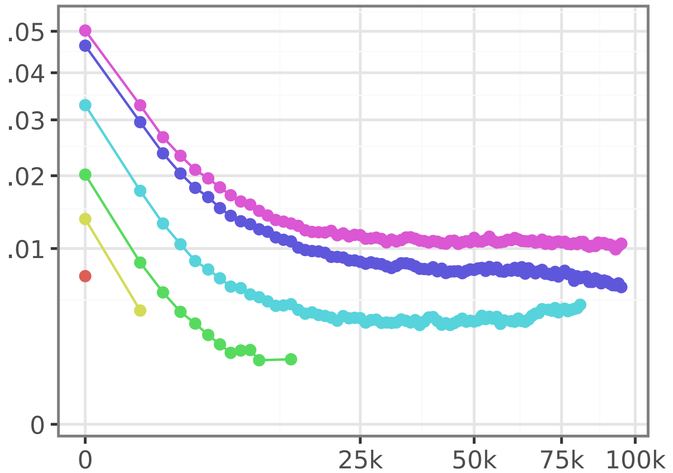


Aggregating the memorisation profile



Checkpoints/Batches across training ($g = c$)

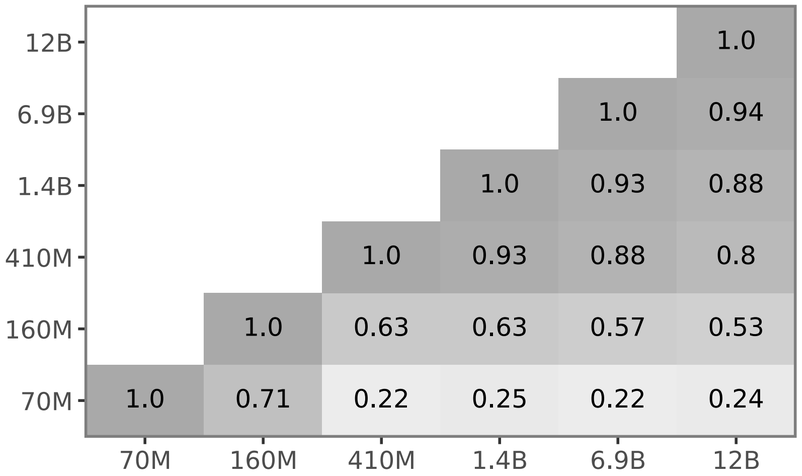
Instantaneous memorisation of a batch at the checkpoint is first seen



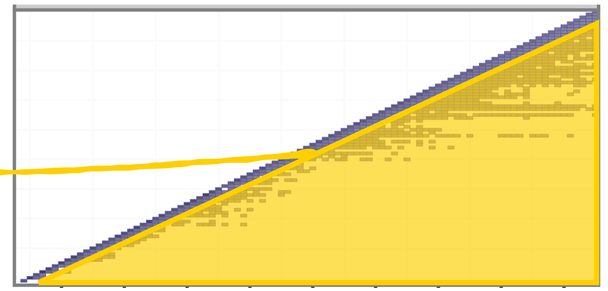
Checkpoints after batch is seen ($g > c$)

Average persistent memorisation of a batch per step after it has been seen*

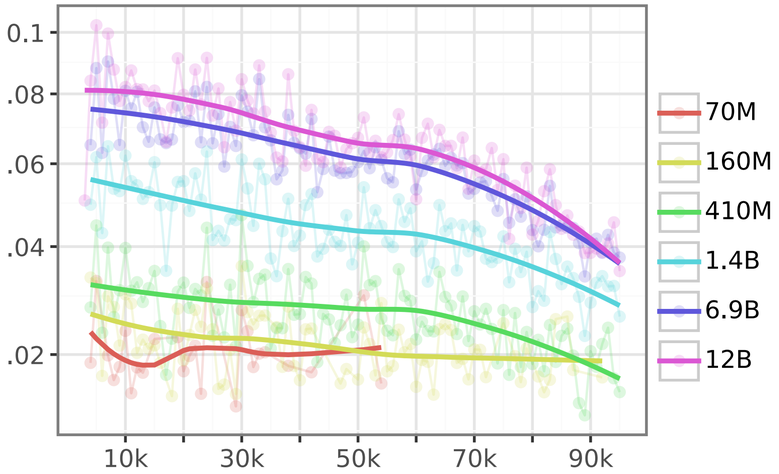
*(only averaging across batches that have been trained that much)



Pearson correlation between the memorisation profiles of different model sizes

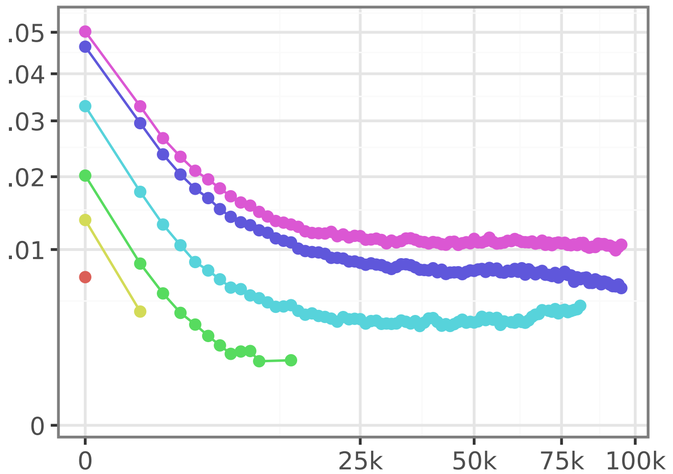


Aggregating the memorisation profile



Checkpoints/Batches across training ($g = c$)

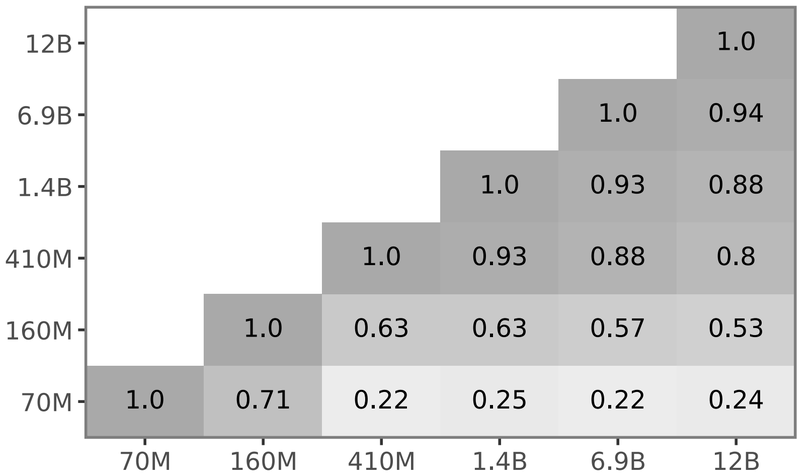
Instantaneous memorisation of a batch at the checkpoint is first seen



Checkpoints after batch is seen ($g > c$)

Average persistent memorisation of a batch per step after it has been seen*

*(only averaging across batches that have been trained that much)



Pearson correlation between the memorisation profiles of different model sizes

FINE

Pietro Lesci
University of Cambridge

X: @pietro_lesci
LinkedIn: /pietrolesci
Page: pietrolesci.github.io
Mail: pietrolesci@outlook.com