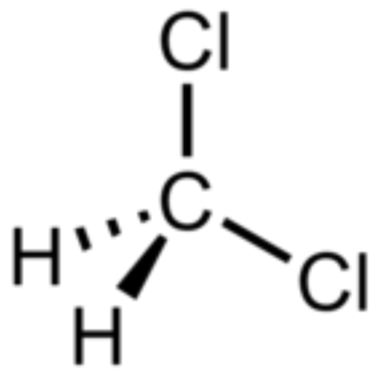
Harnessing Extra Randomness

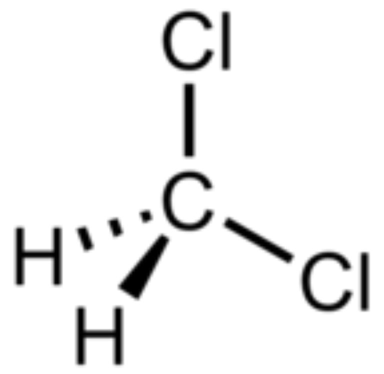
Replicability, Flexibility & Causality

F. Richard Guo
Statistical Laboratory, University of Cambridge

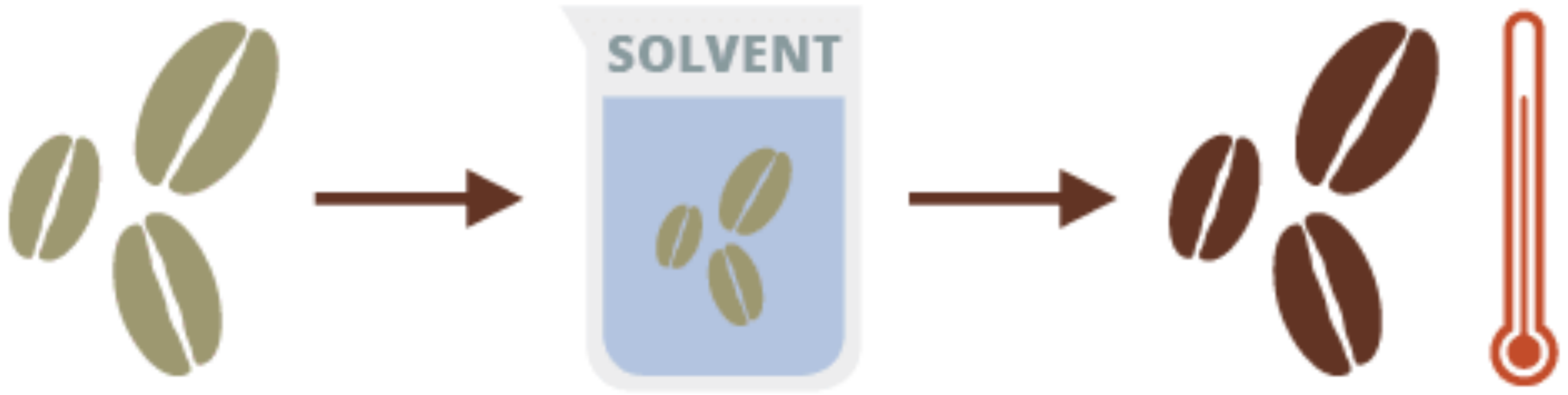
Feb, 2023
Based on joint work w/ Rajen Shah

Organic solvents such as  are widely used in the food industry.

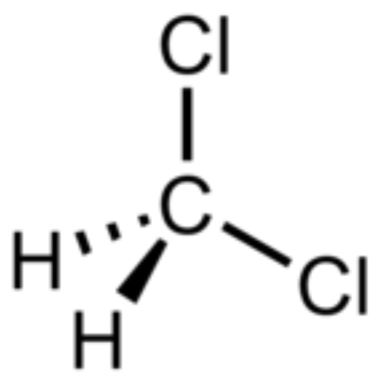
Methylene Chloride

Organic solvents such as  are widely used in the food industry.

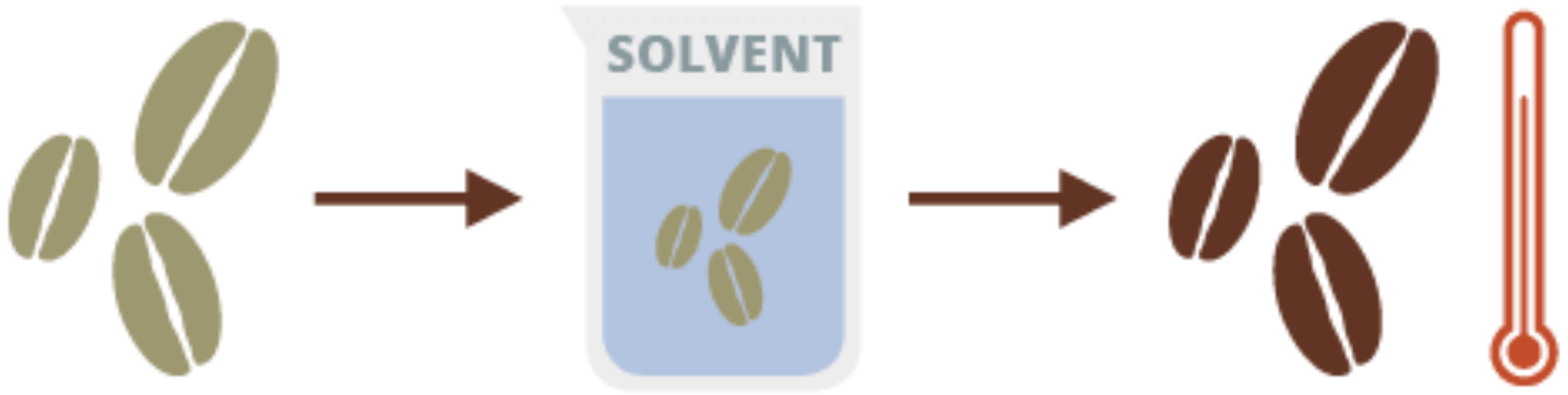
Methylene Chloride



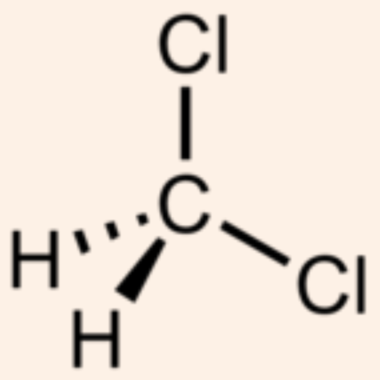
© www.compoundchem.com

Organic solvents such as  are widely used in the food industry.

Methylene Chloride

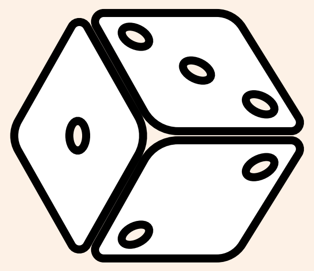


© www.compoundchem.com



Methylene Chloride

=



Extra Randomness

Randomized procedures

Randomized procedures

👉 Output of the procedure is a random function of data.

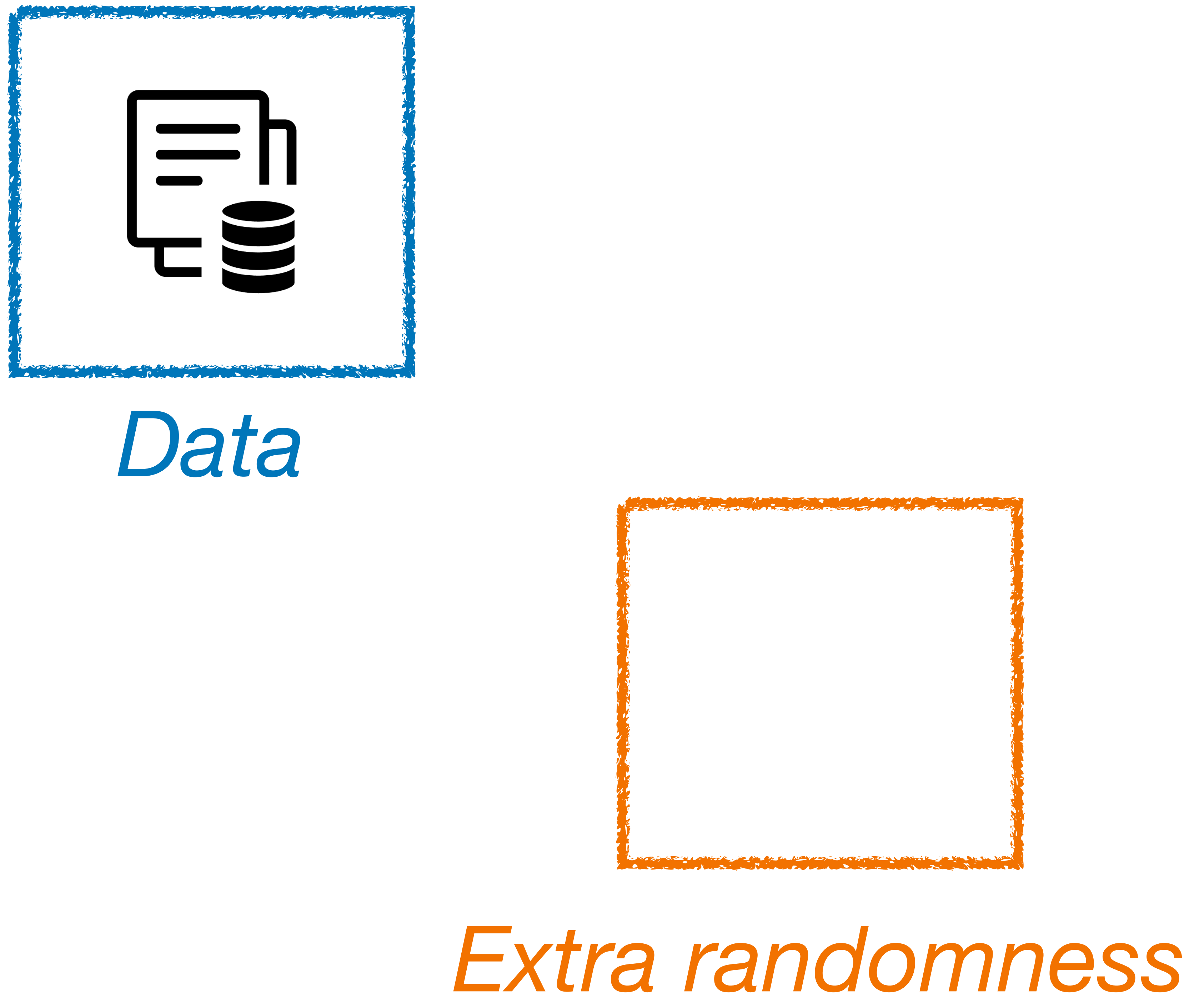
Randomized procedures

👉 Output of the procedure is a random function of data.

Data

Randomized procedures

👉 Output of the procedure is a random function of data.

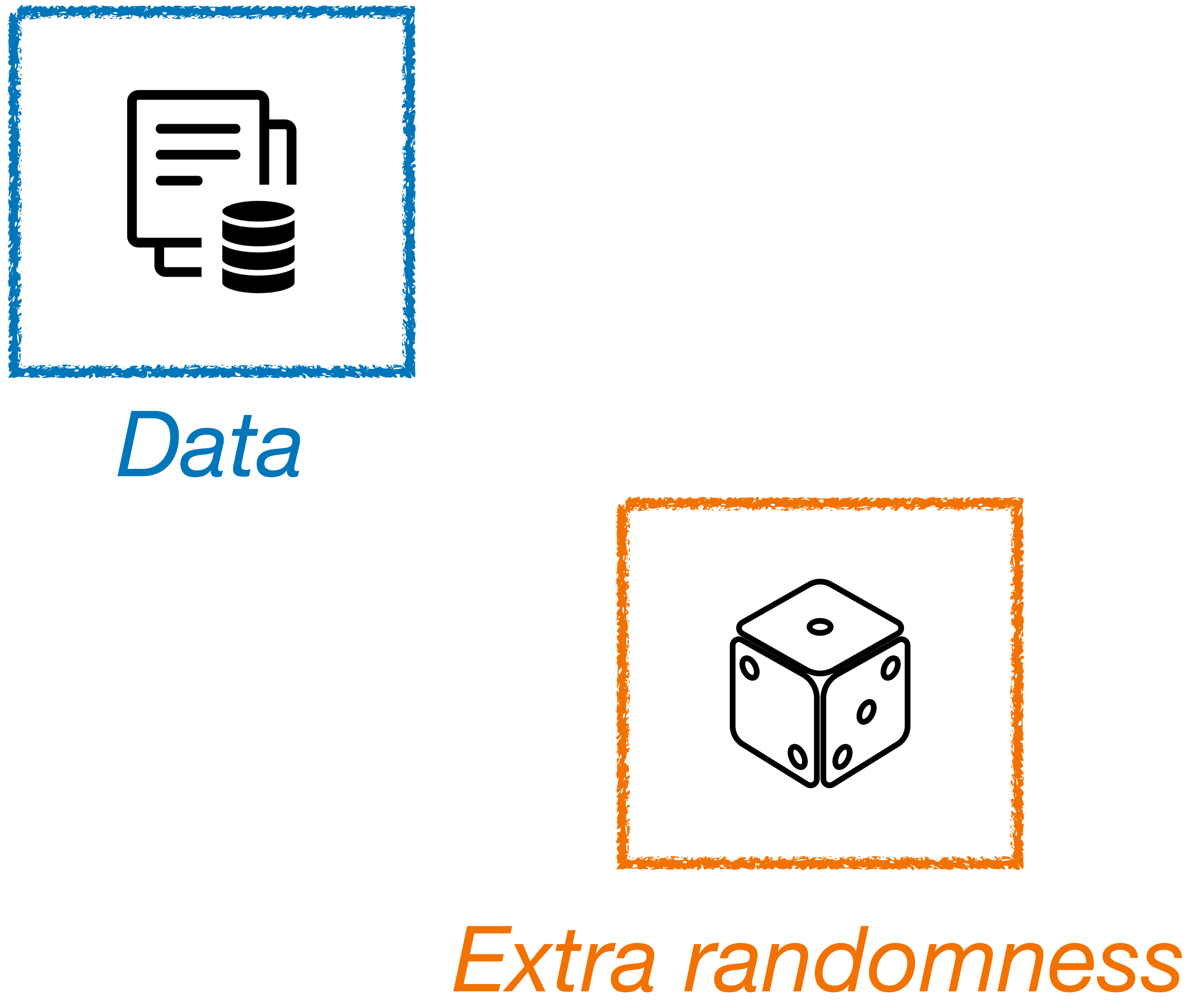


Data

Extra randomness

Randomized procedures

👉 Output of the procedure is a random function of data.

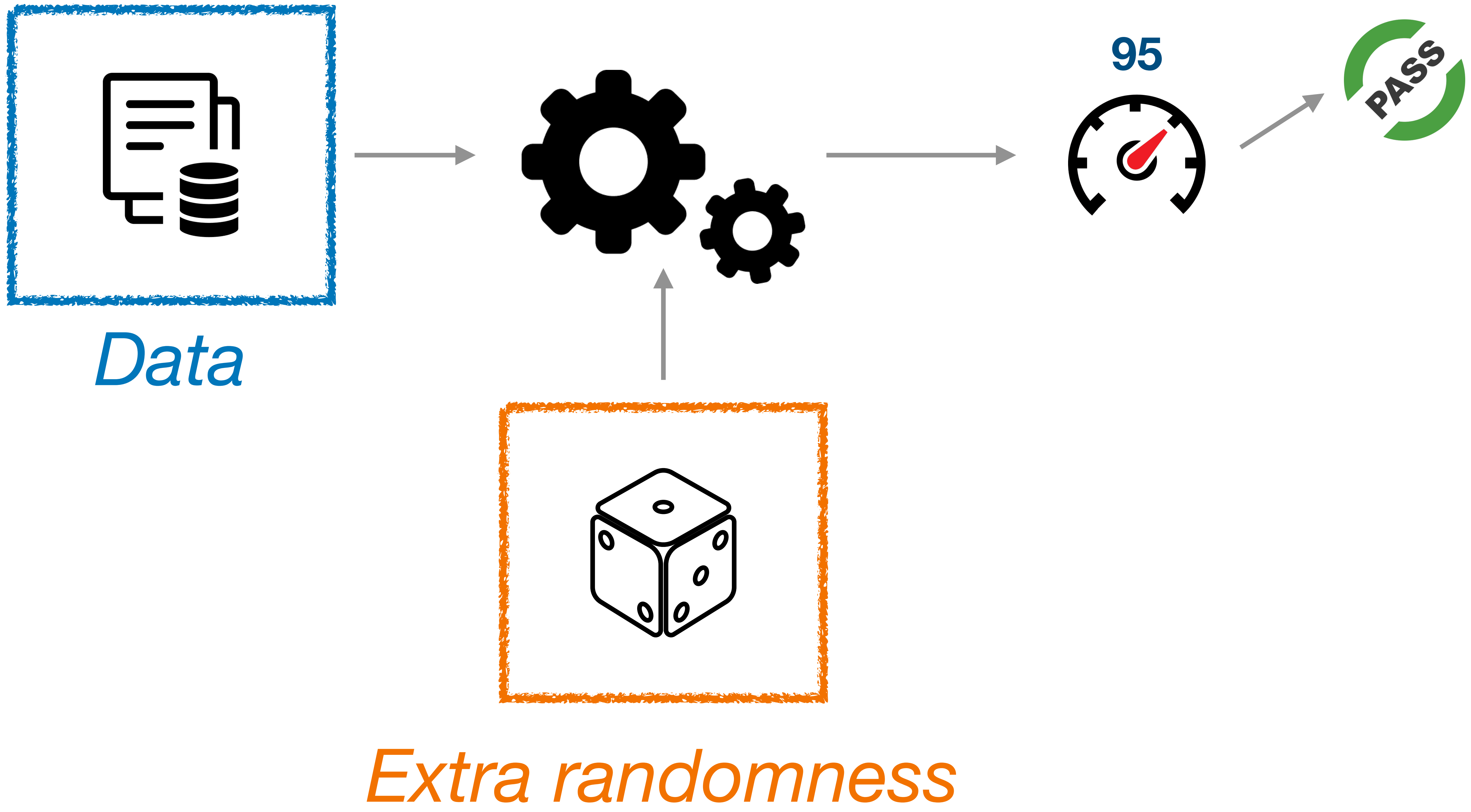


Data

Extra randomness

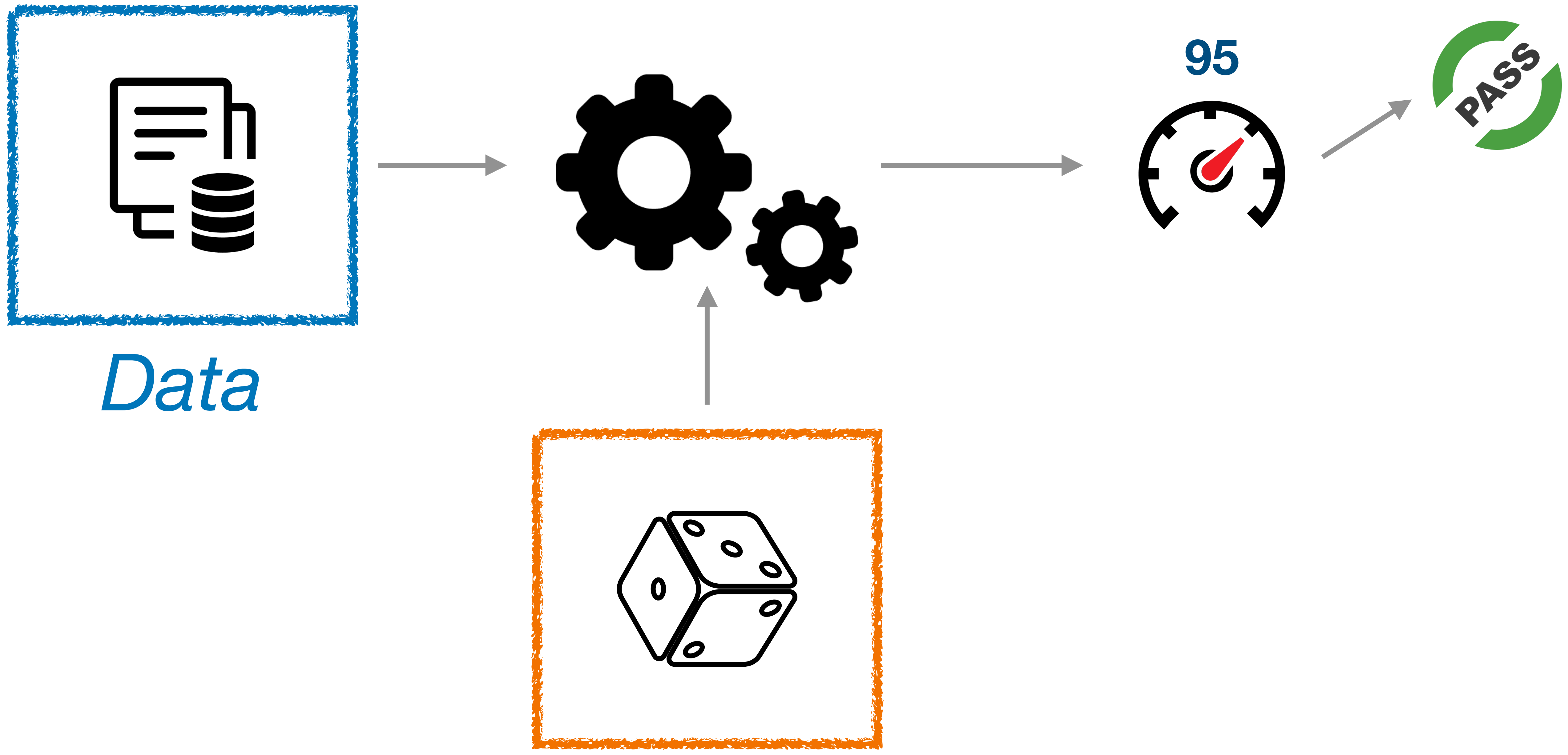
Randomized procedures

👉 Output of the procedure is a random function of data.



Randomized procedures

👉 Output of the procedure is a random function of data.

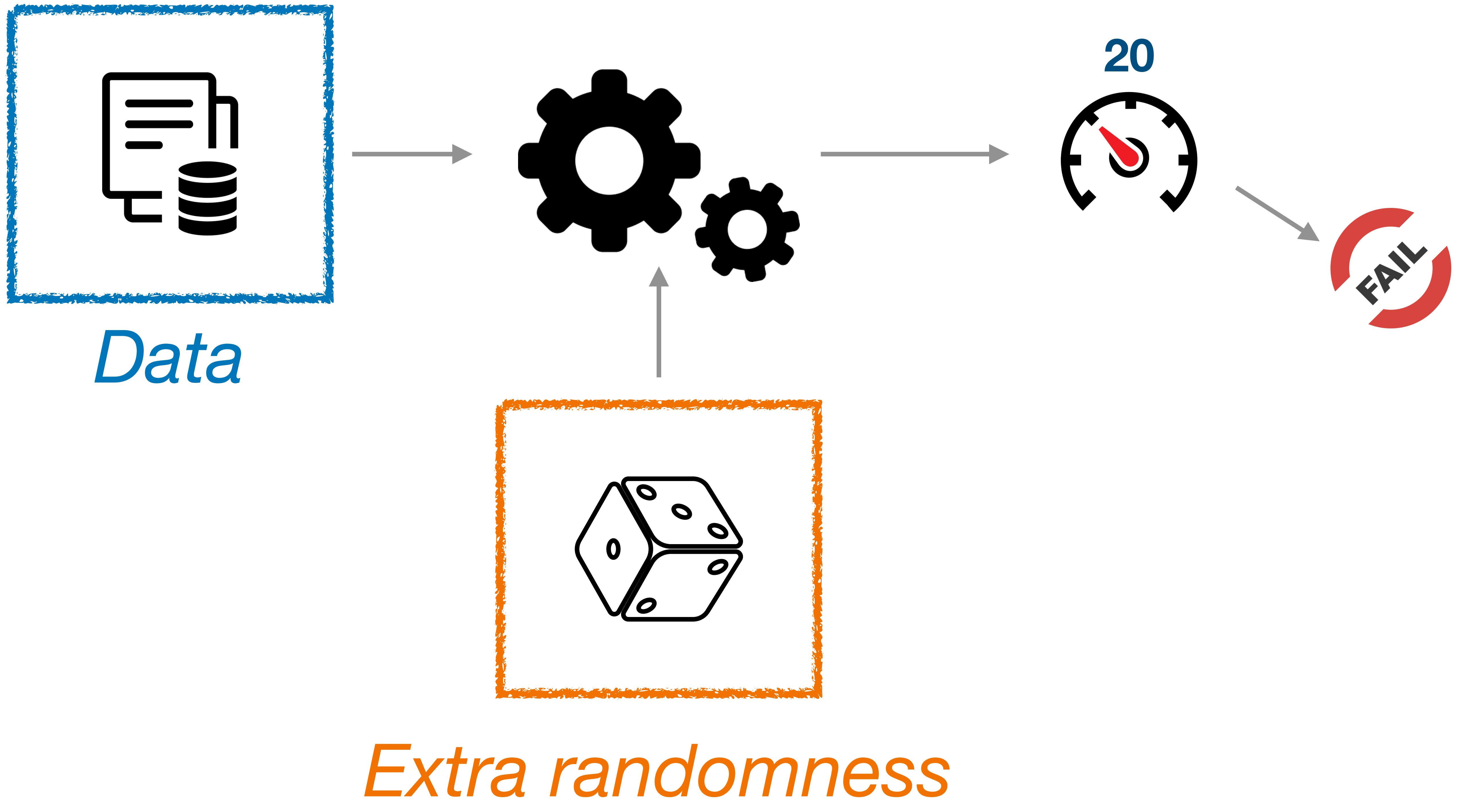


Data

Extra randomness

Randomized procedures

👉 Output of the procedure is a random function of data.



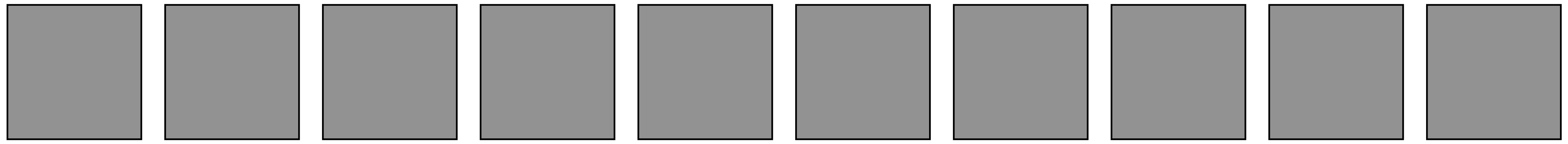
- **Data splitting**

  Randomly divide iid data into several parts for different purposes.

- **Data splitting**

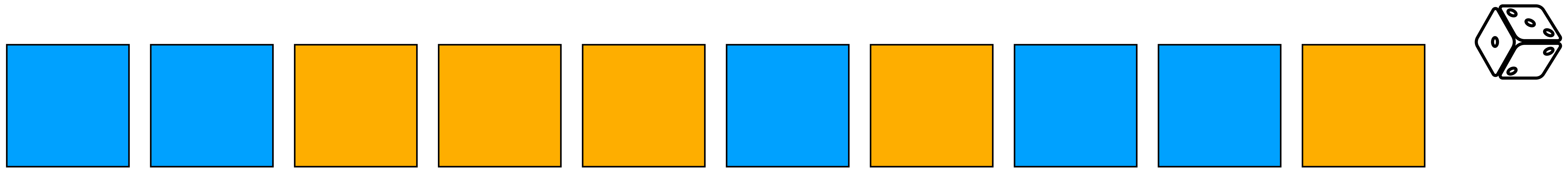
  Randomly divide iid data into several parts for different purposes.

iid data points



- **Data splitting**

  Randomly divide iid data into several parts for different purposes.

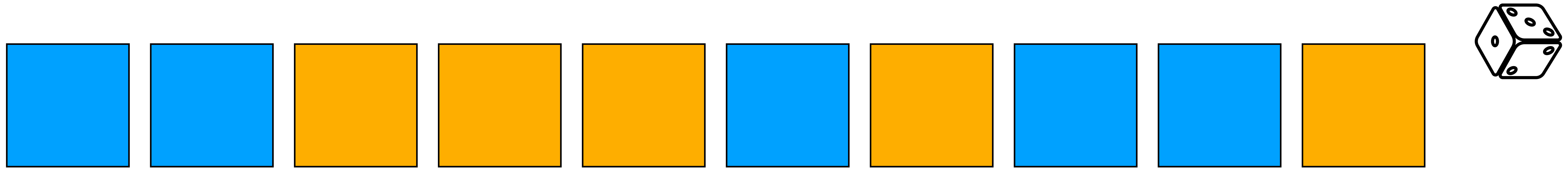


iid data points

- **Data splitting**

  Randomly divide iid data into several parts for different purposes.

iid data points



(🟧🟧🟧🟧🟧) and (🟦🟦🟦🟦🟦) are **independent.**

- **Data splitting**

  Randomly divide iid data into several parts for different purposes.

iid data points 🟦🟦🟧🟧🟧🟦🟧🟦🟦🟧 🎲

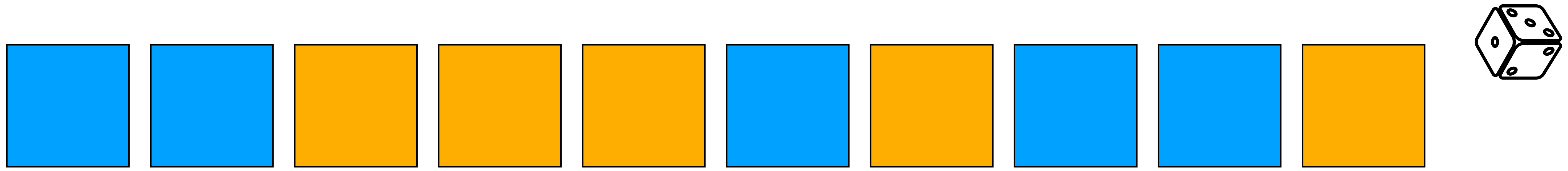
(🟧🟧🟧🟧🟧) and (🟦🟦🟦🟦🟦) are **independent.**

👉 **Control overfitting** 👉 **Prevent double dipping**

- **Data splitting**

  Randomly divide iid data into several parts for different purposes.

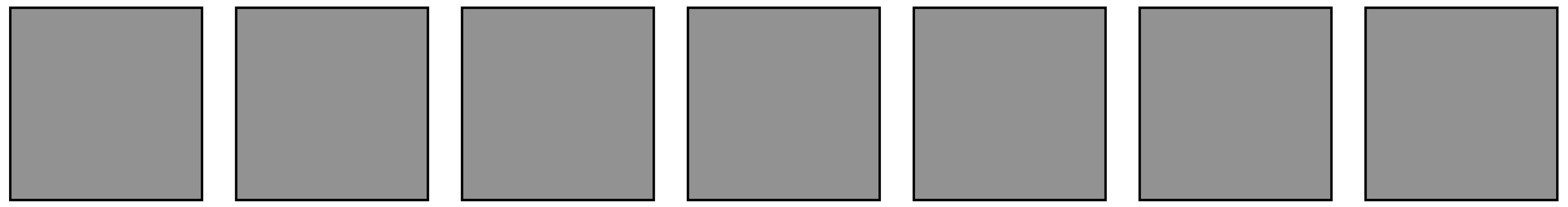
iid data points



(🟧🟧🟧🟧🟧) and (🟦🟦🟦🟦🟦) are **independent.**

👉 **Control overfitting** 👉 **Prevent double dipping**

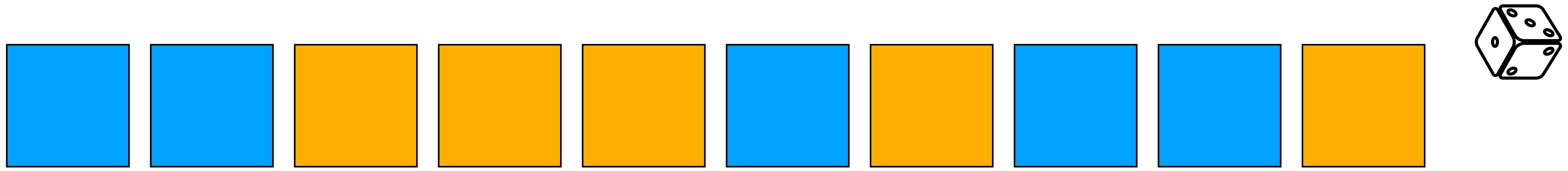
- **Sampling**



- **Data splitting**

  Randomly divide iid data into several parts for different purposes.

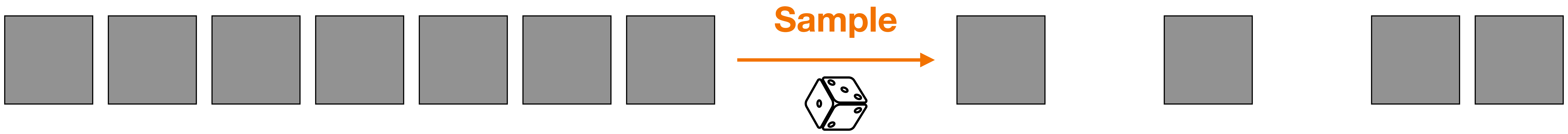
iid data points



(🟨🟨🟨🟨🟨) and (🟦🟦🟦🟦🟦) are **independent.**

👉 **Control overfitting** 👉 **Prevent double dipping**

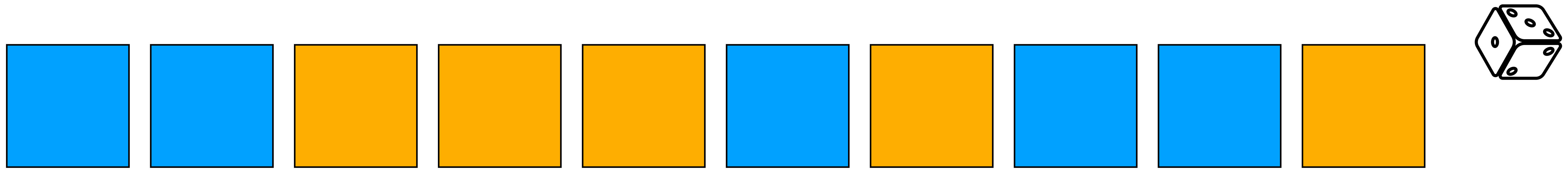
- **Sampling**



- **Data splitting**

  Randomly divide iid data into several parts for different purposes.

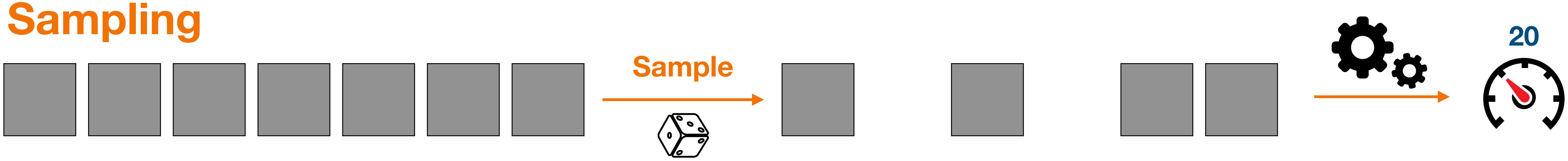
iid data points



(🟧🟧🟧🟧🟧) and (🟦🟦🟦🟦🟦) are **independent.**

👉 **Control overfitting** 👉 **Prevent double dipping**

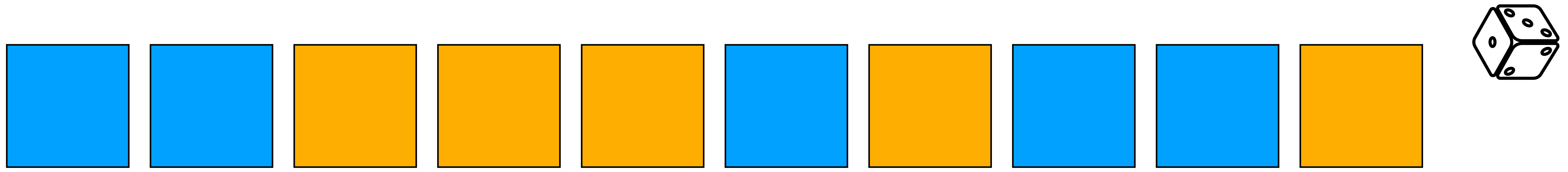
- **Sampling**



- **Data splitting**

  Randomly divide iid data into several parts for different purposes.

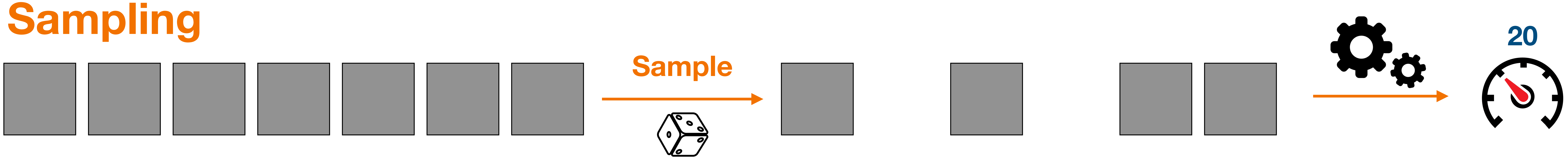
iid data points



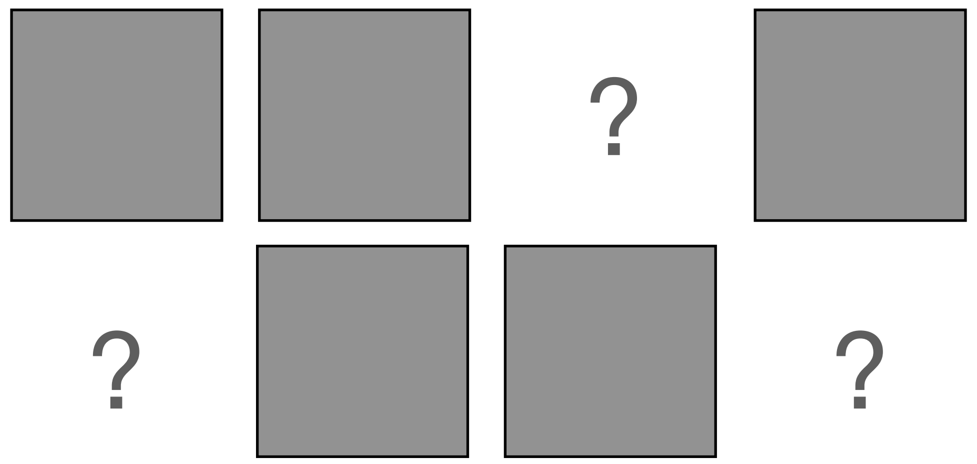
(🟧🟧🟧🟧🟧) and (🟦🟦🟦🟦🟦) are **independent.**

👉 **Control overfitting** 👉 **Prevent double dipping**

- **Sampling**

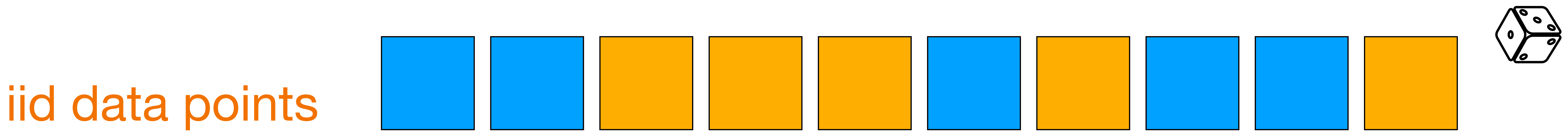


- **Random imputation**



- **Data splitting**

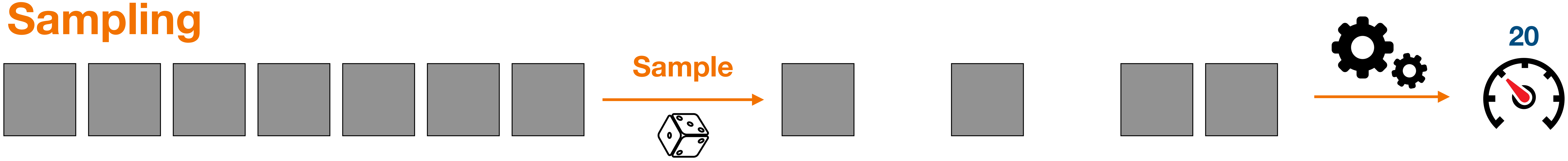
  Randomly divide iid data into several parts for different purposes.



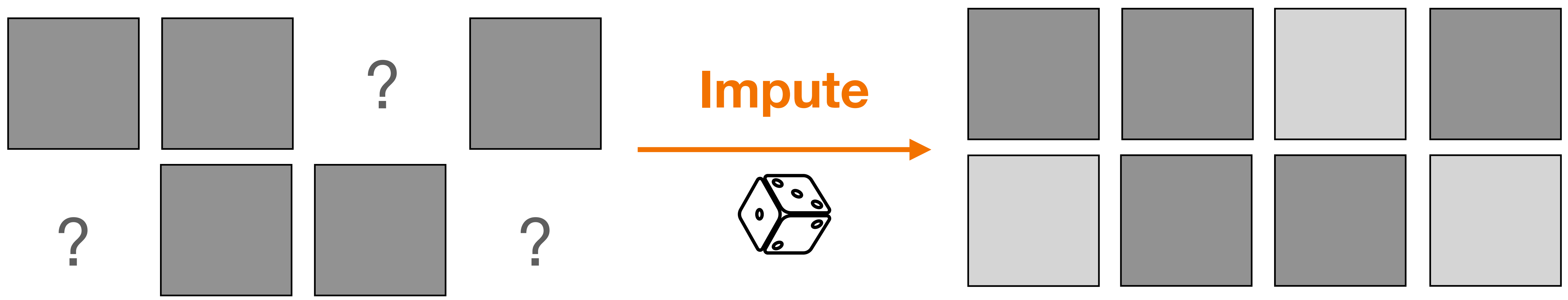
(🟧🟧🟧🟧🟧) and (🟦🟦🟦🟦🟦) are **independent.**

👉 **Control overfitting** 👉 **Prevent double dipping**

- **Sampling**

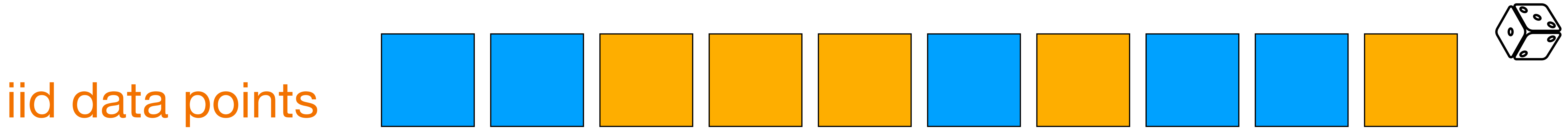


- **Random imputation**



- **Data splitting**

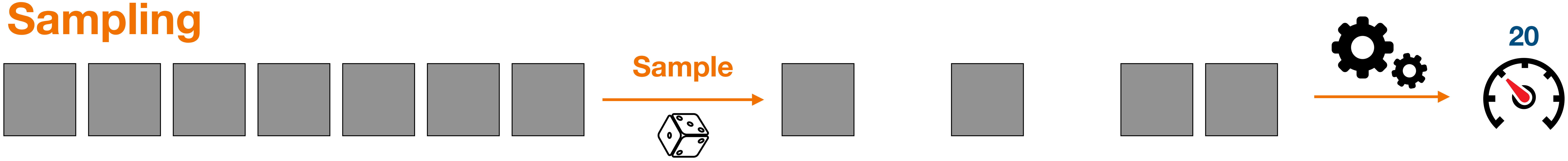
  Randomly divide iid data into several parts for different purposes.



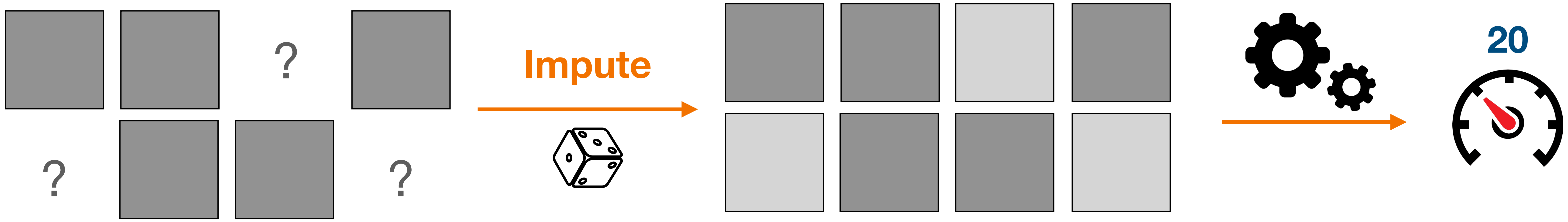
(🟧🟧🟧🟧🟧) and (🟦🟦🟦🟦🟦) are **independent.**

👉 **Control overfitting** 👉 **Prevent double dipping**

- **Sampling**



- **Random imputation**



Agenda

Agenda

1. Though useful, randomized procedures have serious drawbacks.

Agenda

1. Though useful, randomized procedures have serious drawbacks.
2. Present a general framework to resolve these drawbacks.

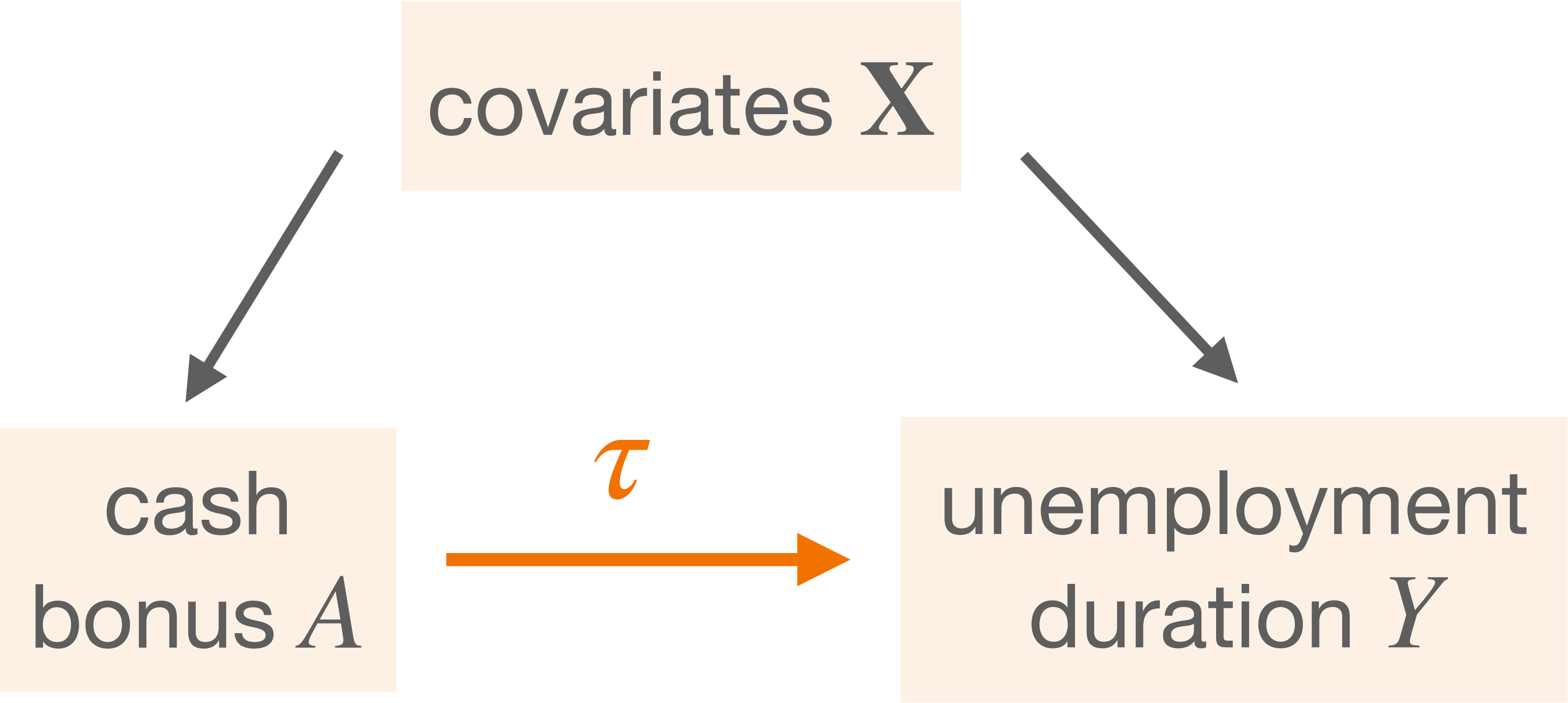
Agenda

1. Though useful, randomized procedures have serious drawbacks.
2. Present a general framework to resolve these drawbacks.
3. Harness extra randomness for many great applications!

Dilemma of data splitting

Bill, PhD, an economist

Bill, PhD, an economist



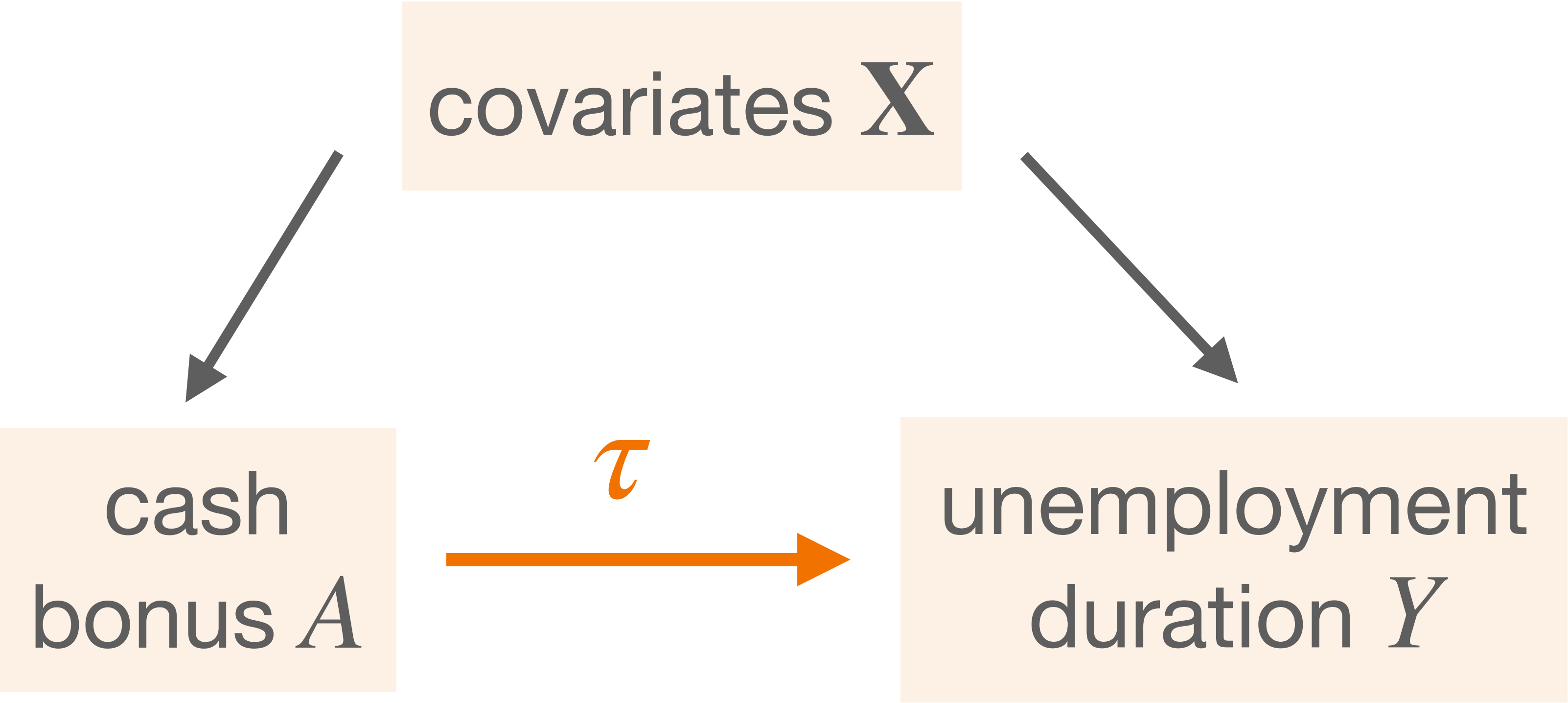
covariates $\boldsymbol{X}$

cash bonus A

τ

unemployment duration Y

Bill, PhD, an economist

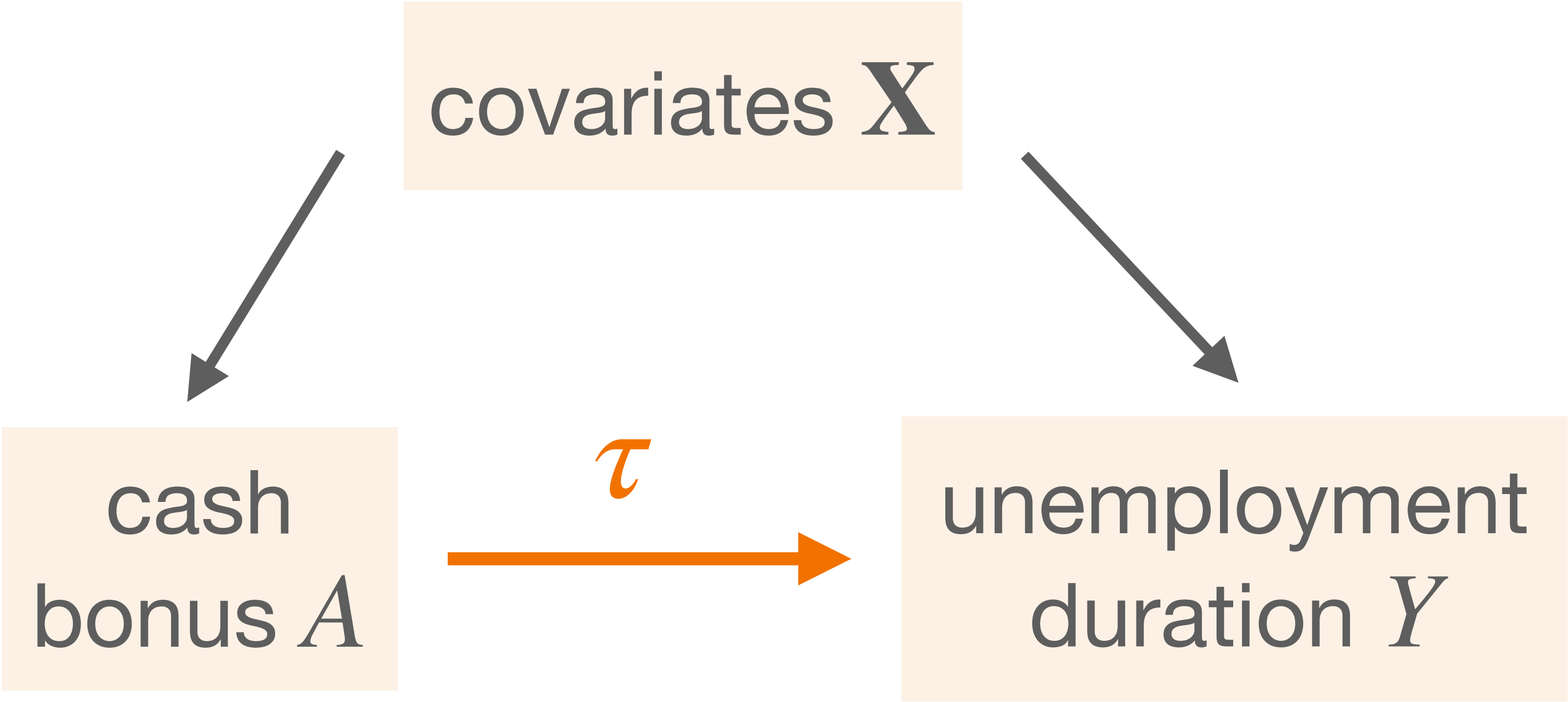


👉 Doubly robust estimation of τ requires fitting two nuisance functions:

$$\eta_1 = \mathbb{P}(A \,|\, \mathbf{X})$$

$$\eta_2 = \mathbb{E}[Y | A, \mathbf{X}]$$

Bill, PhD, an economist



👉 Doubly robust estimation of τ requires fitting two nuisance functions:

$$\eta_1 = \mathbb{P}(A \,|\, \mathbf{X})$$

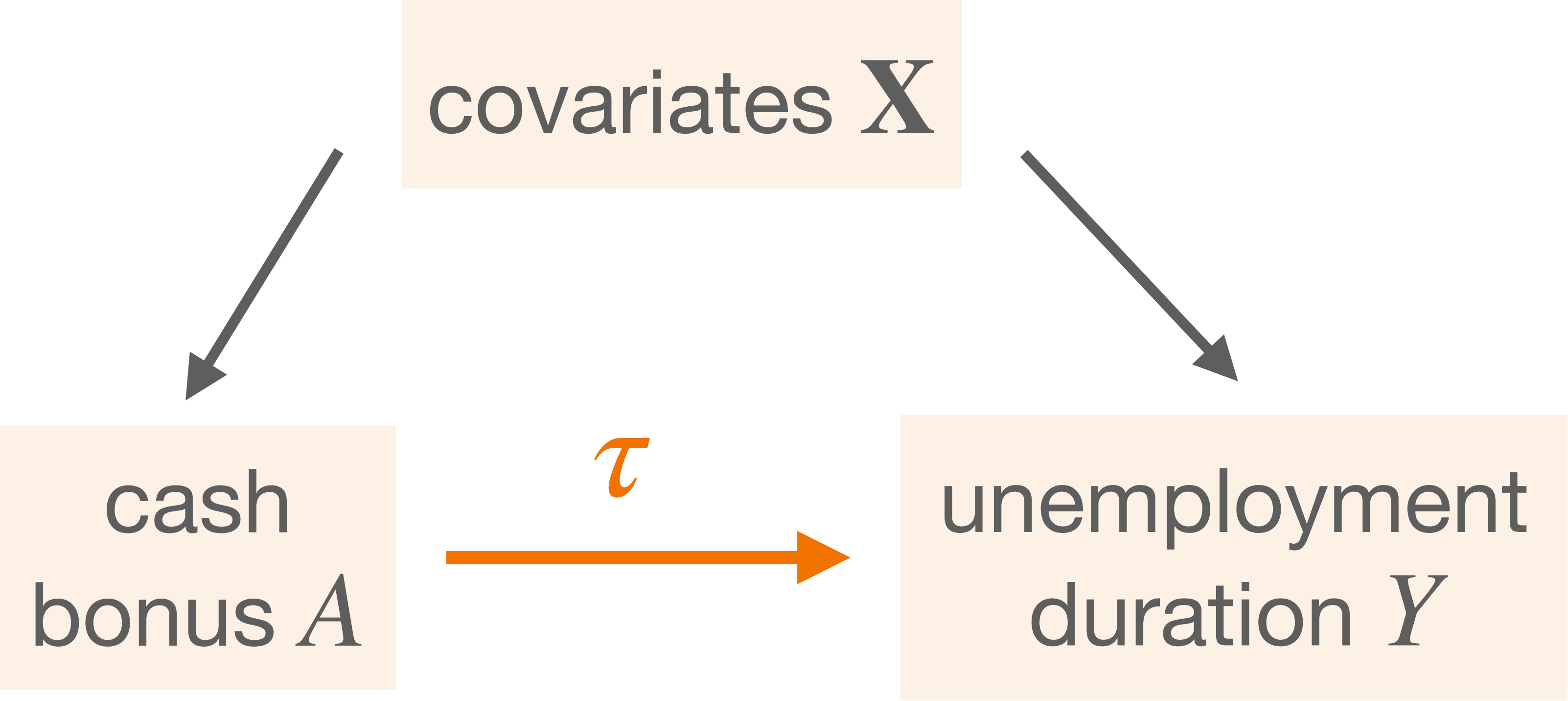
$$\eta_2 = \mathbb{E}[Y \,|\, A, \mathbf{X}]$$

Targeted / Double ML: permit using flexible ML tools to estimate η_1, η_2.

👉 Use data splitting / cross fitting to control bias from overfitting $\hat{\eta}_1, \hat{\eta}_2$.

(van der Laan & Rose, 2011; Newey and Robins, 2018; Chernozhukov et al., 2018; Díaz, 2020; Kennedy, 2022)

Bill, PhD, an economist



👉 Doubly robust estimation of τ requires fitting two nuisance functions:

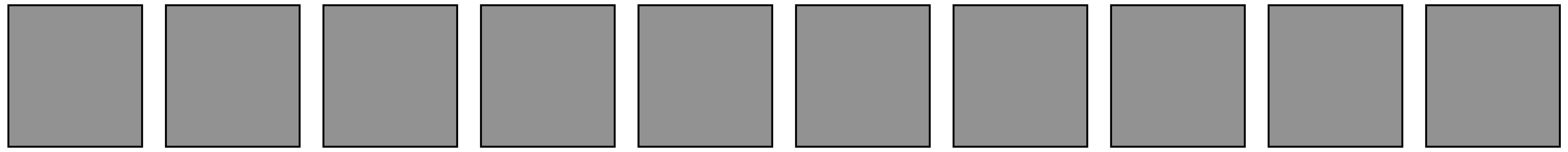
$$\eta_1 = \mathbb{P}(A \,|\, \mathbf{X})$$

$$\eta_2 = \mathbb{E}[Y \,|\, A, \mathbf{X}]$$

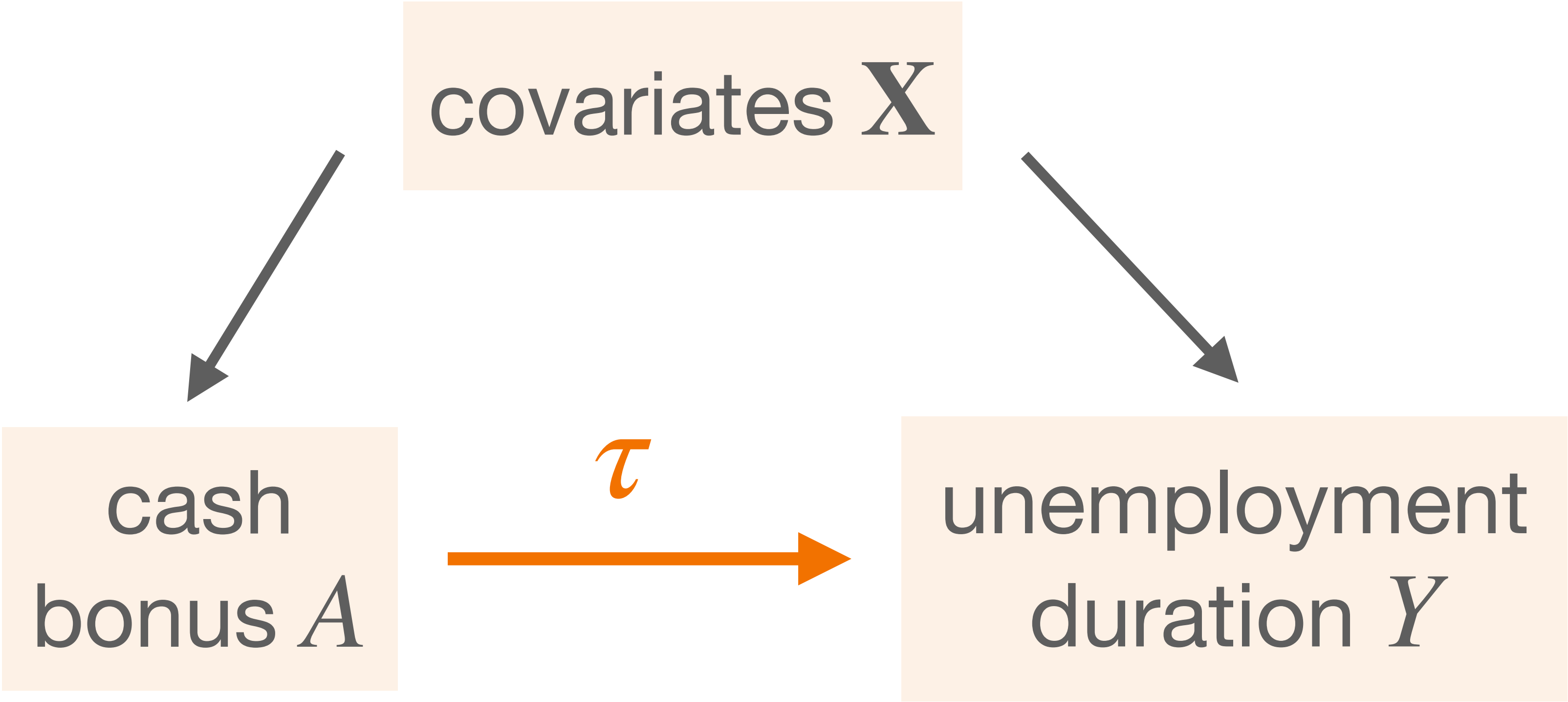
Targeted / Double ML: permit using flexible ML tools to estimate η_1, η_2.

👉 Use data splitting / cross fitting to control bias from overfitting $\hat{\eta}_1, \hat{\eta}_2$.

(van der Laan & Rose, 2011; Newey and Robins, 2018; Chernozhukov et al., 2018; Díaz, 2020; Kennedy, 2022)



Bill, PhD, an economist



covariates $\mathbf{X}$

cash bonus A

τ

unemployment duration Y

👉 Doubly robust estimation of τ requires fitting two nuisance functions:

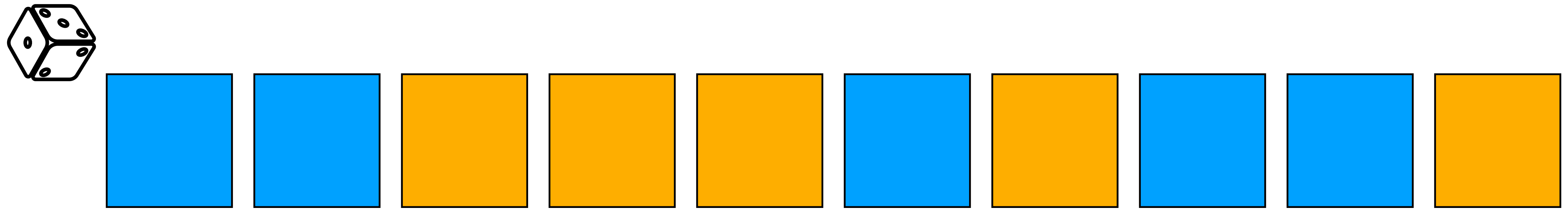
$$\eta_1 = \mathbb{P}(A \,|\, \mathbf{X})$$

$$\eta_2 = \mathbb{E}[Y \,|\, A, \mathbf{X}]$$

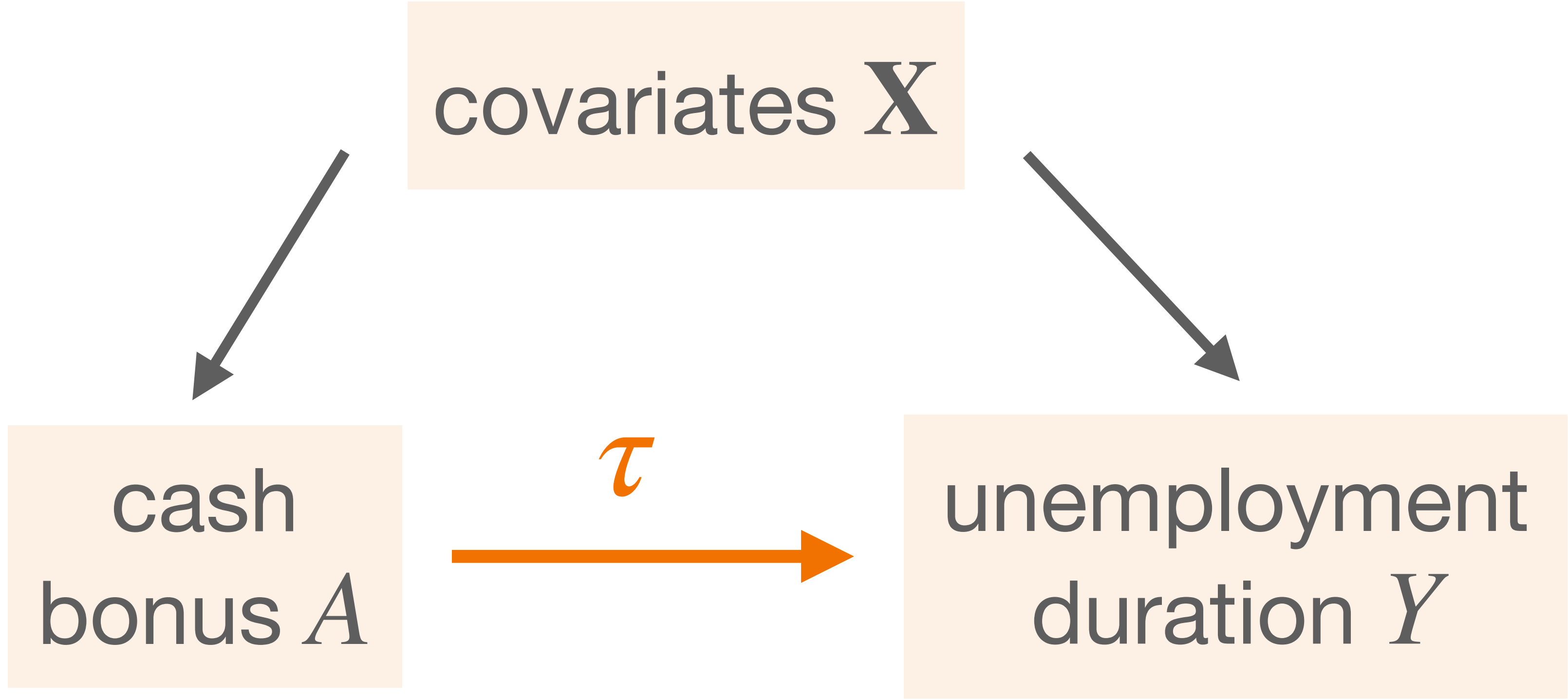
Targeted / Double ML: permit using flexible ML tools to estimate η_1, η_2.

👉 Use data splitting / cross fitting to control bias from overfitting $\hat{\eta}_1, \hat{\eta}_2$.

(van der Laan & Rose, 2011; Newey and Robins, 2018; Chernozhukov et al., 2018; Díaz, 2020; Kennedy, 2022)



Bill, PhD, an economist



👉 Doubly robust estimation of τ requires fitting two nuisance functions:

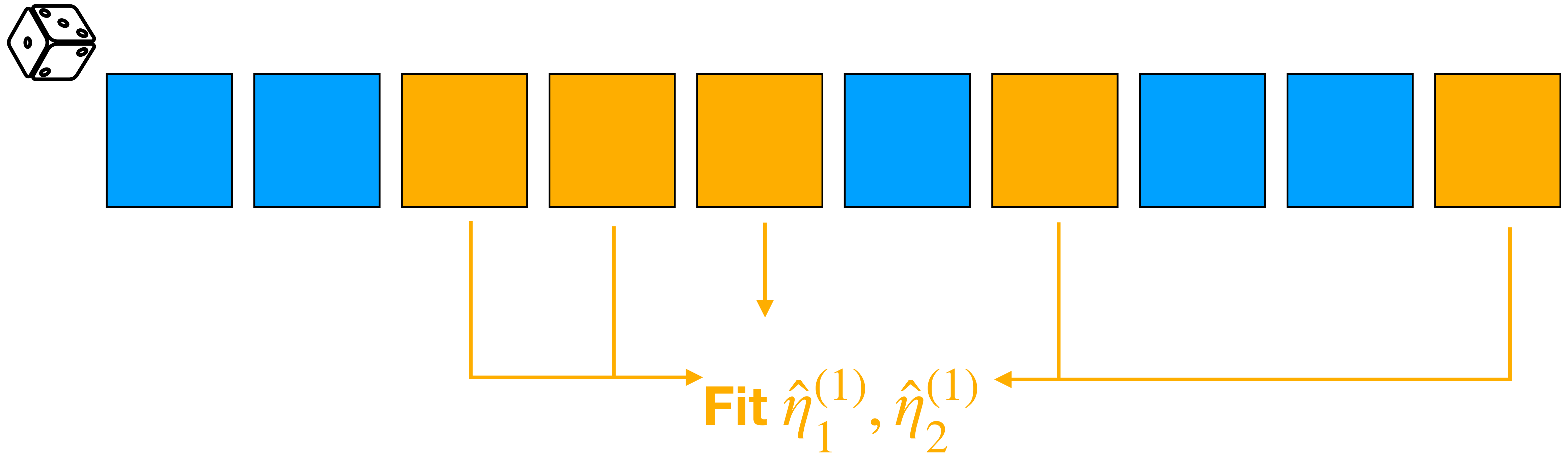
$$\eta_1 = \mathbb{P}(A \,|\, \mathbf{X})$$

$$\eta_2 = \mathbb{E}[Y \,|\, A, \mathbf{X}]$$

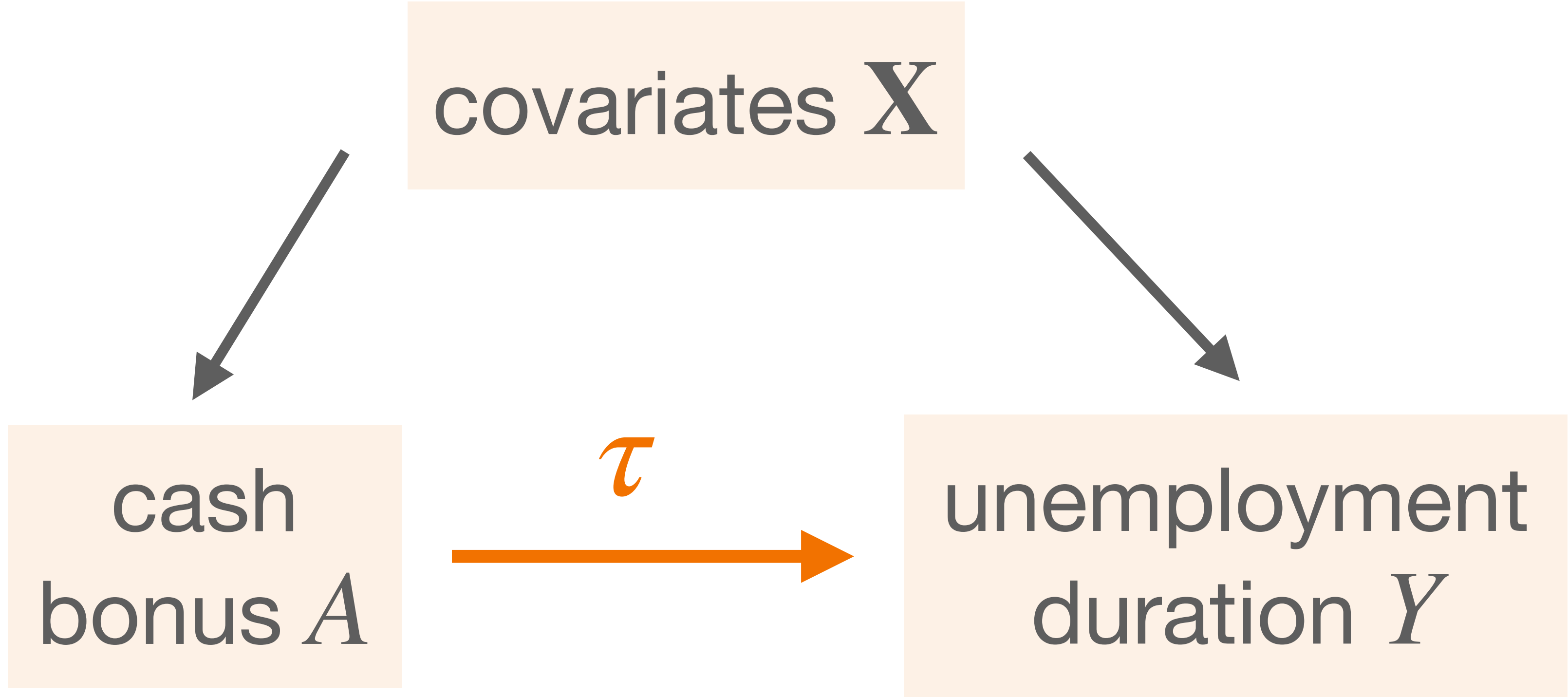
Targeted / Double ML: permit using flexible ML tools to estimate η_1, η_2.

👉 Use data splitting / cross fitting to control bias from overfitting $\hat{\eta}_1, \hat{\eta}_2$.

(van der Laan & Rose, 2011; Newey and Robins, 2018; Chernozhukov et al., 2018; Díaz, 2020; Kennedy, 2022)



Bill, PhD, an economist



👉 Doubly robust estimation of τ requires fitting two nuisance functions:

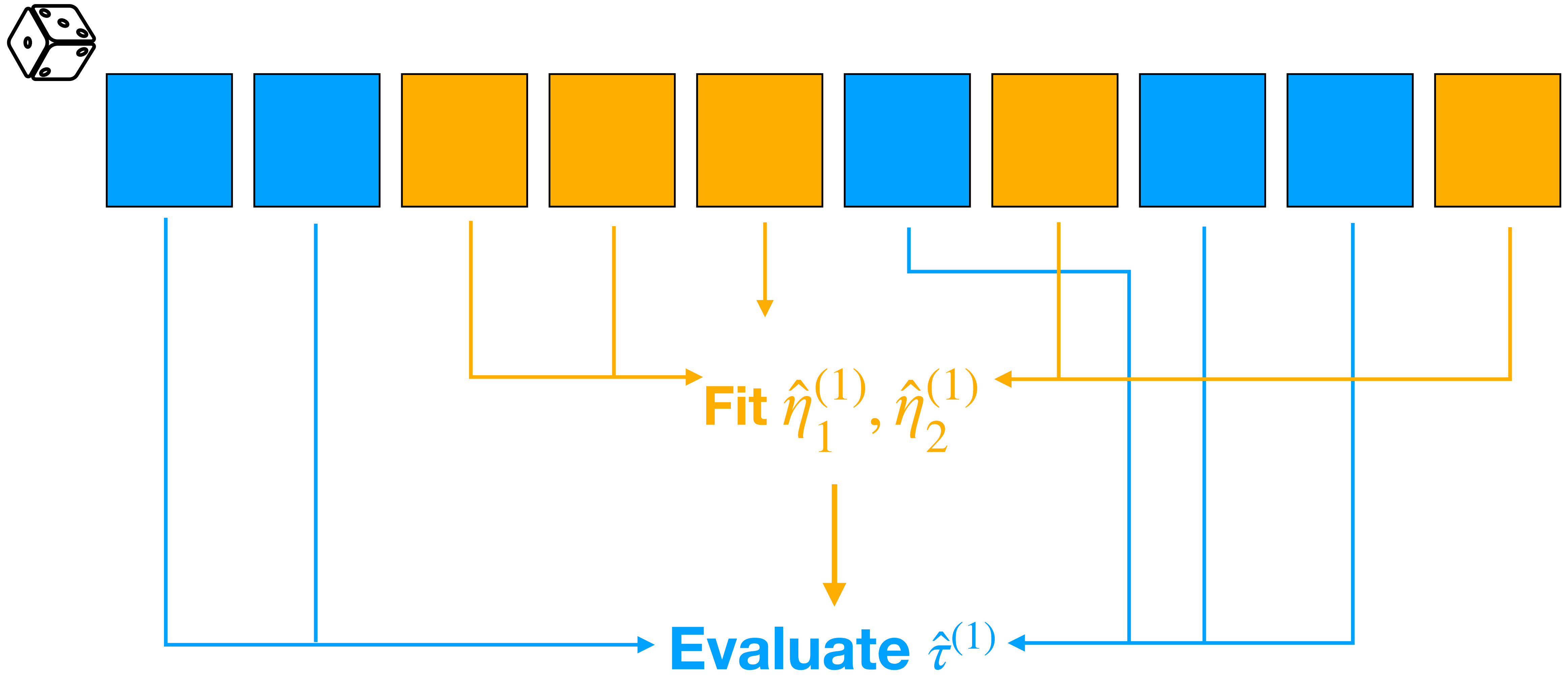
$$\eta_1 = \mathbb{P}(A \mid \mathbf{X})$$

$$\eta_2 = \mathbb{E}[Y \mid A, \mathbf{X}]$$

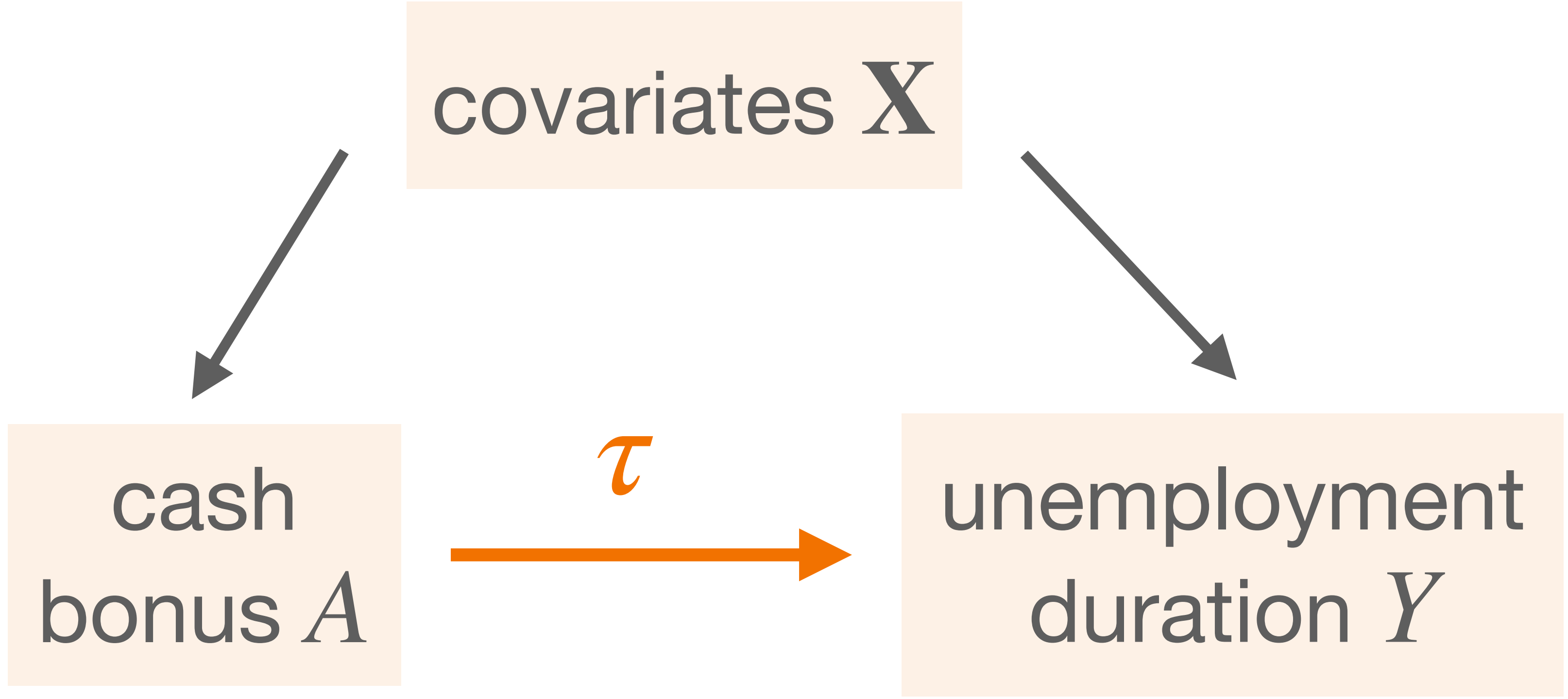
Targeted / Double ML: permit using flexible ML tools to estimate η_1, η_2.

👉 Use data splitting / cross fitting to control bias from overfitting $\hat{\eta}_1, \hat{\eta}_2$.

(van der Laan & Rose, 2011; Newey and Robins, 2018; Chernozhukov et al., 2018; Díaz, 2020; Kennedy, 2022)


Fit $\hat{\eta}_1^{(1)}, \hat{\eta}_2^{(1)}$
Evaluate $\hat{\tau}^{(1)}$

Bill, PhD, an economist



👉 Doubly robust estimation of τ requires fitting two nuisance functions:

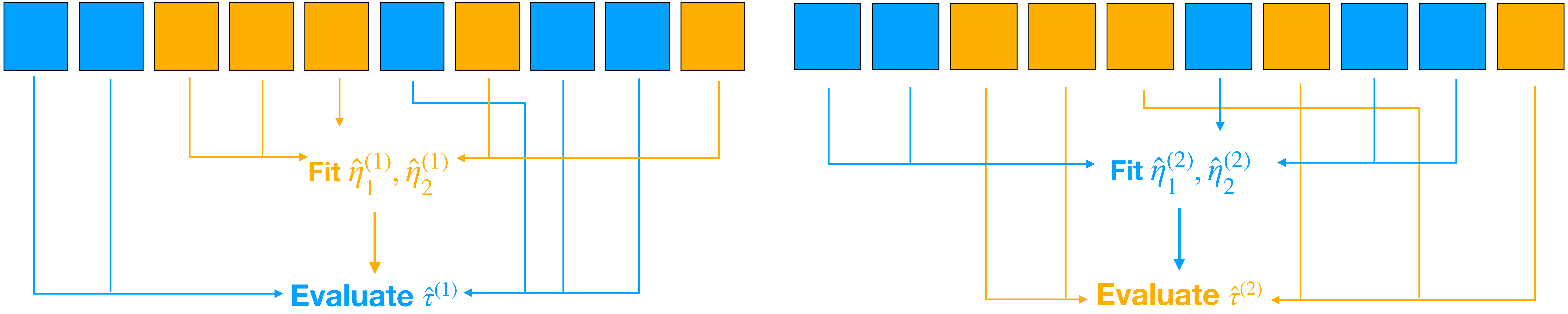
$$\eta_1 = \mathbb{P}(A \mid \mathbf{X})$$

$$\eta_2 = \mathbb{E}[Y \mid A, \mathbf{X}]$$

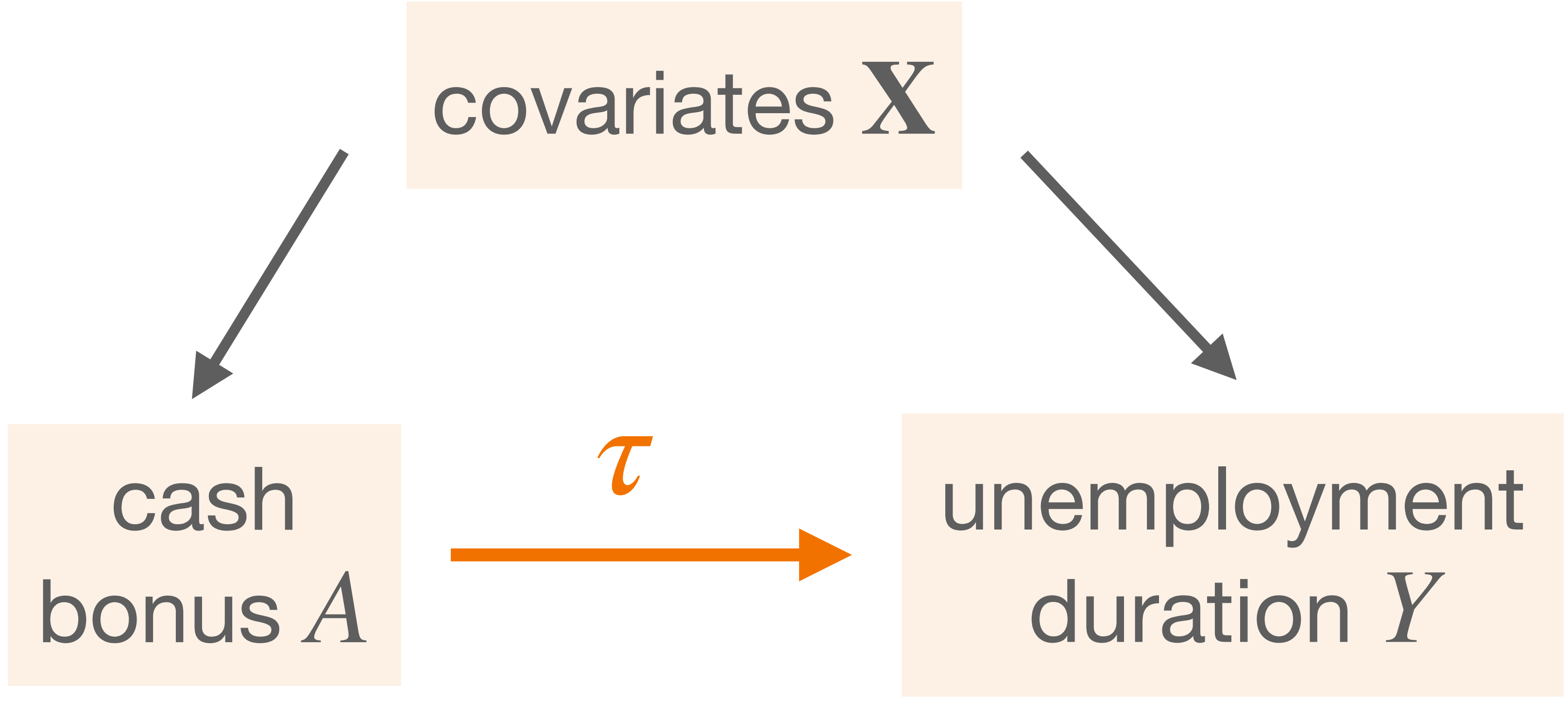
Targeted / Double ML: permit using flexible ML tools to estimate η_1, η_2.

👉 Use data splitting / cross fitting to control bias from overfitting $\hat{\eta}_1, \hat{\eta}_2$.

(van der Laan & Rose, 2011; Newey and Robins, 2018; Chernozhukov et al., 2018; Díaz, 2020; Kennedy, 2022)


Fit $\hat{\eta}_1^{(1)}, \hat{\eta}_2^{(1)}$
Evaluate $\hat{\tau}^{(1)}$
Fit $\hat{\eta}_1^{(2)}, \hat{\eta}_2^{(2)}$
Evaluate $\hat{\tau}^{(2)}$

Bill, PhD, an economist



👉 Doubly robust estimation of τ requires fitting two nuisance functions:

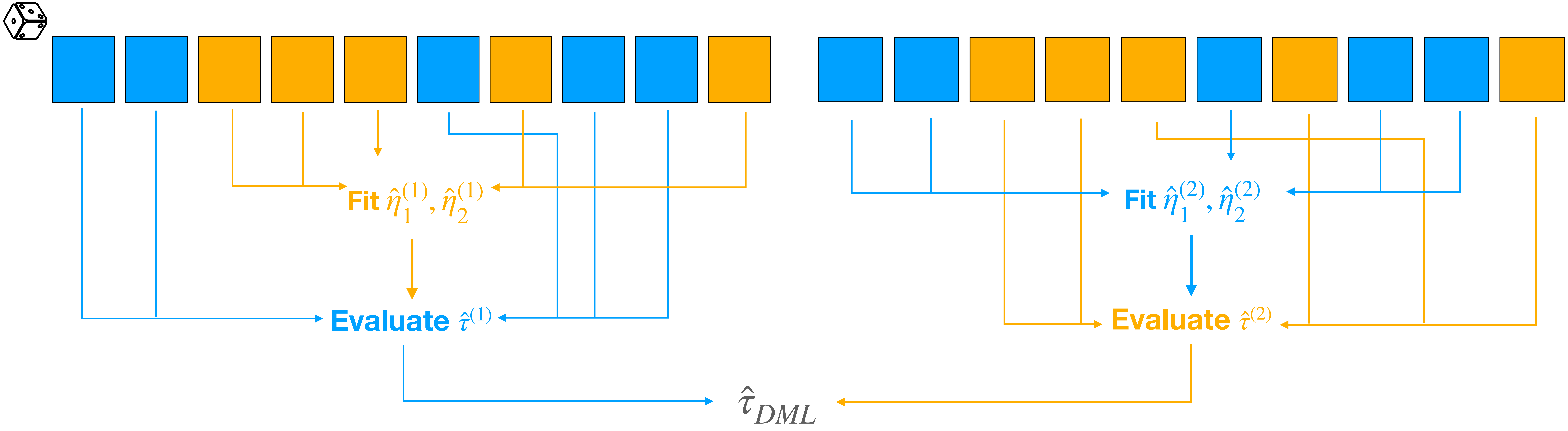
$$\eta_1 = \mathbb{P}(A \,|\, \mathbf{X})$$

$$\eta_2 = \mathbb{E}[Y \,|\, A, \mathbf{X}]$$

Targeted / Double ML: permit using flexible ML tools to estimate η_1, η_2.

👉 Use data splitting / cross fitting to control bias from overfitting $\hat{\eta}_1, \hat{\eta}_2$.

(van der Laan & Rose, 2011; Newey and Robins, 2018; Chernozhukov et al., 2018; Díaz, 2020; Kennedy, 2022)



Fit $\hat{\eta}_1^{(1)}, \hat{\eta}_2^{(1)}$
Evaluate $\hat{\tau}^{(1)}$
Fit $\hat{\eta}_1^{(2)}, \hat{\eta}_2^{(2)}$
Evaluate $\hat{\tau}^{(2)}$
$\hat{\tau}_{DML}$

```
> set.seed(42)
```

```
> set.seed(42)
> dml$fit()
```

```
     Estimate. Std. Error t value Pr(>|t|)
tau  -0.1      0.035       -2.86   0.004 **
```

```
> set.seed(42)
> dml$fit()
```

```
     Estimate. Std. Error t value Pr(>|t|)
tau  -0.1      0.035        -2.86   0.004 **
```

```
> set.seed(43)
> dml$fit()
```

```
     Estimate. Std. Error t value Pr(>|t|)
tau  -0.06     0.035        -1.71   0.08 .
```

```
> set.seed(42)
> dml$fit()
```

```
     Estimate. Std. Error t value Pr(>|t|)
tau  -0.1      0.035      -2.86   0.004 **
```

```
> set.seed(43)
> dml$fit()
```

```
     Estimate. Std. Error t value Pr(>|t|)
tau  -0.06     0.035      -1.71   0.08 .
```

```
> set.seed(44)
> dml$fit()
```

```
     Estimate. Std. Error t value Pr(>|t|)
tau  -0.07     0.037      -1.89   0.06 .
```

We find a significant negative effect* ($\hat{\tau} = -0.1$, p-value=0.004)….

* To replicate my analysis, please use “set.seed(42)” (my lucky number).

We find a significant negative effect* ($\hat{\tau} = -0.1$, p-value=0.004)….

* To replicate my analysis, please use “set.seed(42)” (my lucky number).

Reviewer:

🤔 “To replicate, why must I use your lucky number?”

> We find a significant negative effect* ($\hat{\tau} = -0.1$, p-value=0.004)….
>
> ---
>
> * To replicate my analysis, please use “set.seed(42)” (my lucky number).

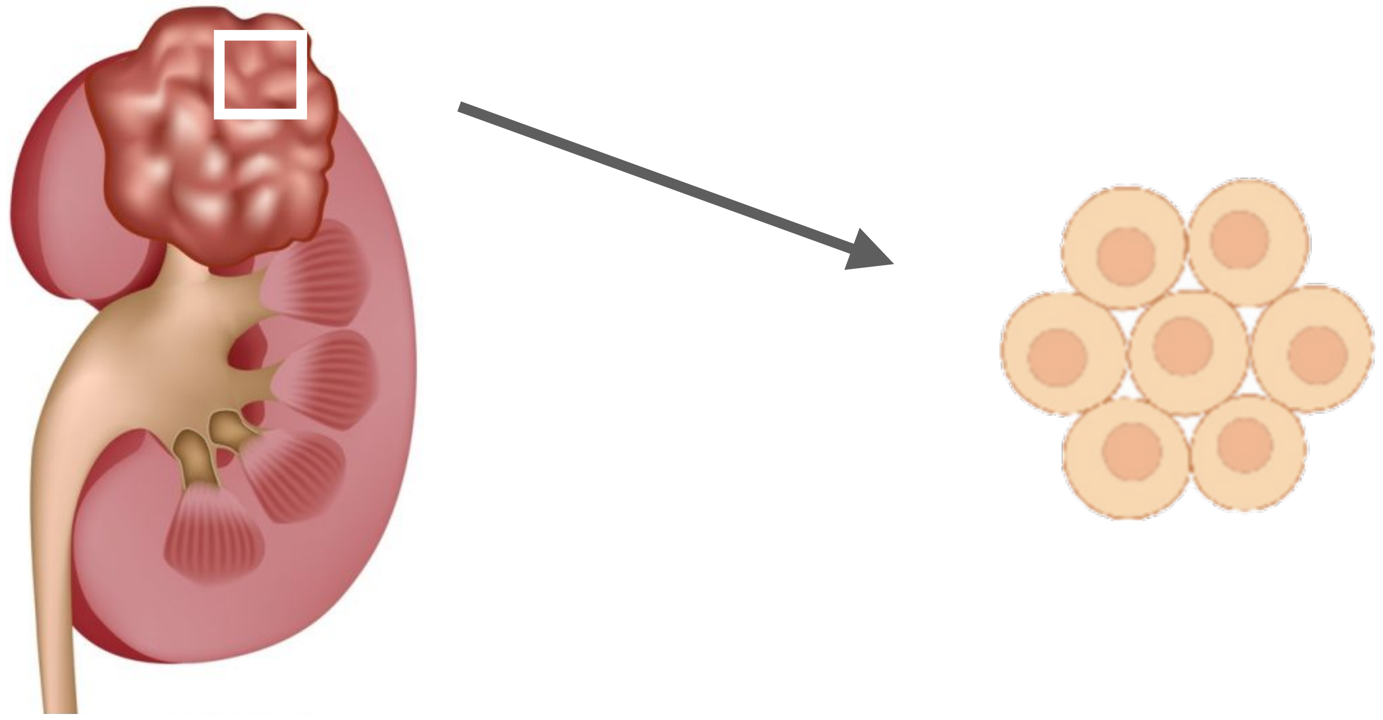
Reviewer:

🤔 “To replicate, why must I use your lucky number?”

🤔 “How do I know you did not fish for 42?”

Laura, PhD, a cancer biologist
(MS in Statistics)

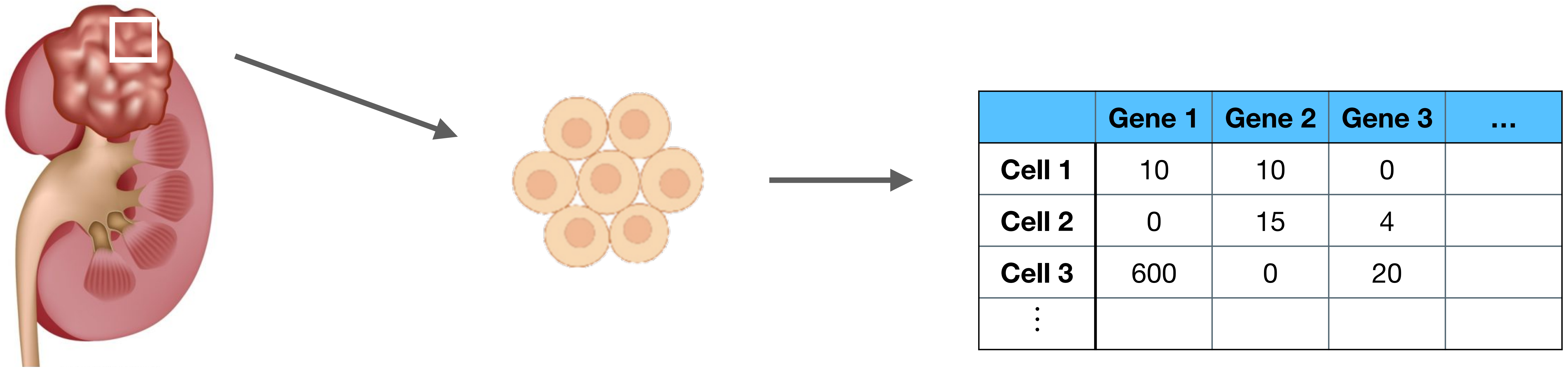
Laura, PhD, a cancer biologist
(MS in Statistics)



Kidney tumor

🤔 Is there a new subtype of kidney cancer cells?

Laura, PhD, a cancer biologist
(MS in Statistics)



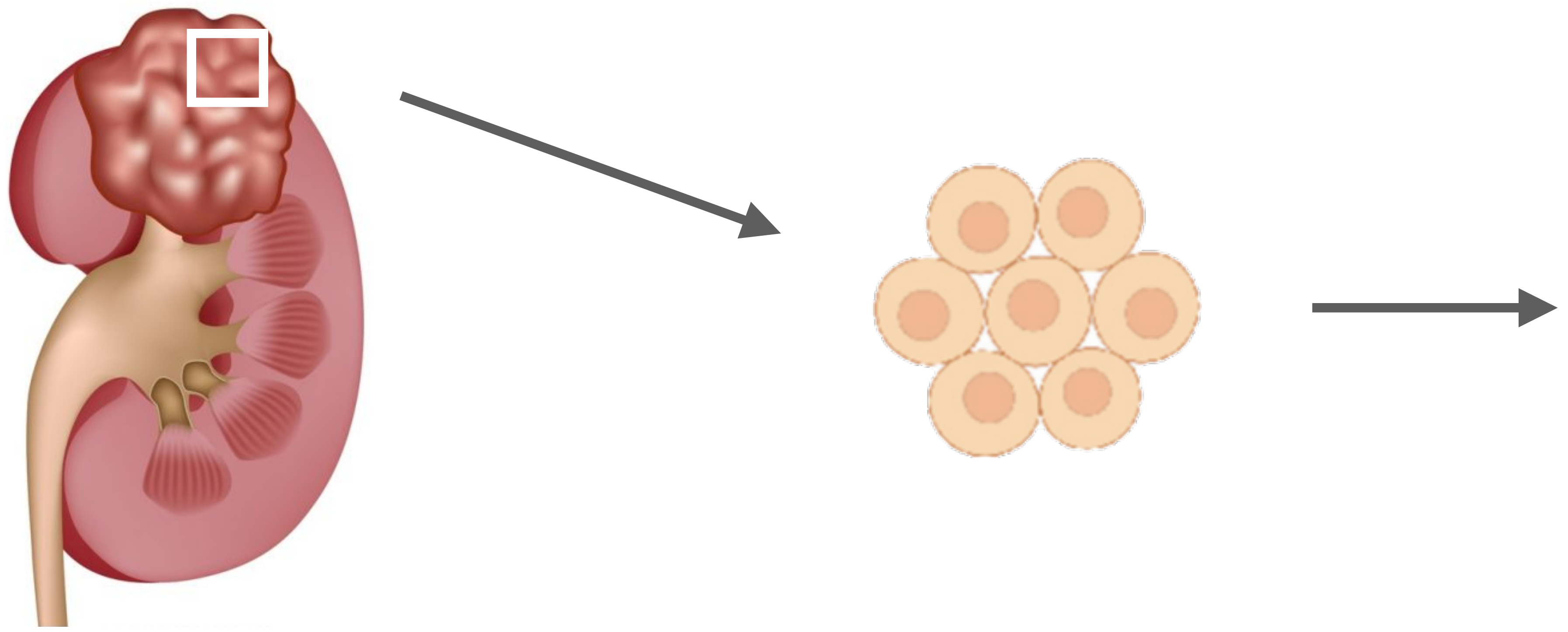
	Gene 1	Gene 2	Gene 3	...
Cell 1	10	10	0	
Cell 2	0	15	4	
Cell 3	600	0	20	
⋮				

Kidney tumor

Single-cell RNA read count

🤔 Is there a new subtype of kidney cancer cells?

Laura, PhD, a cancer biologist
(MS in Statistics)

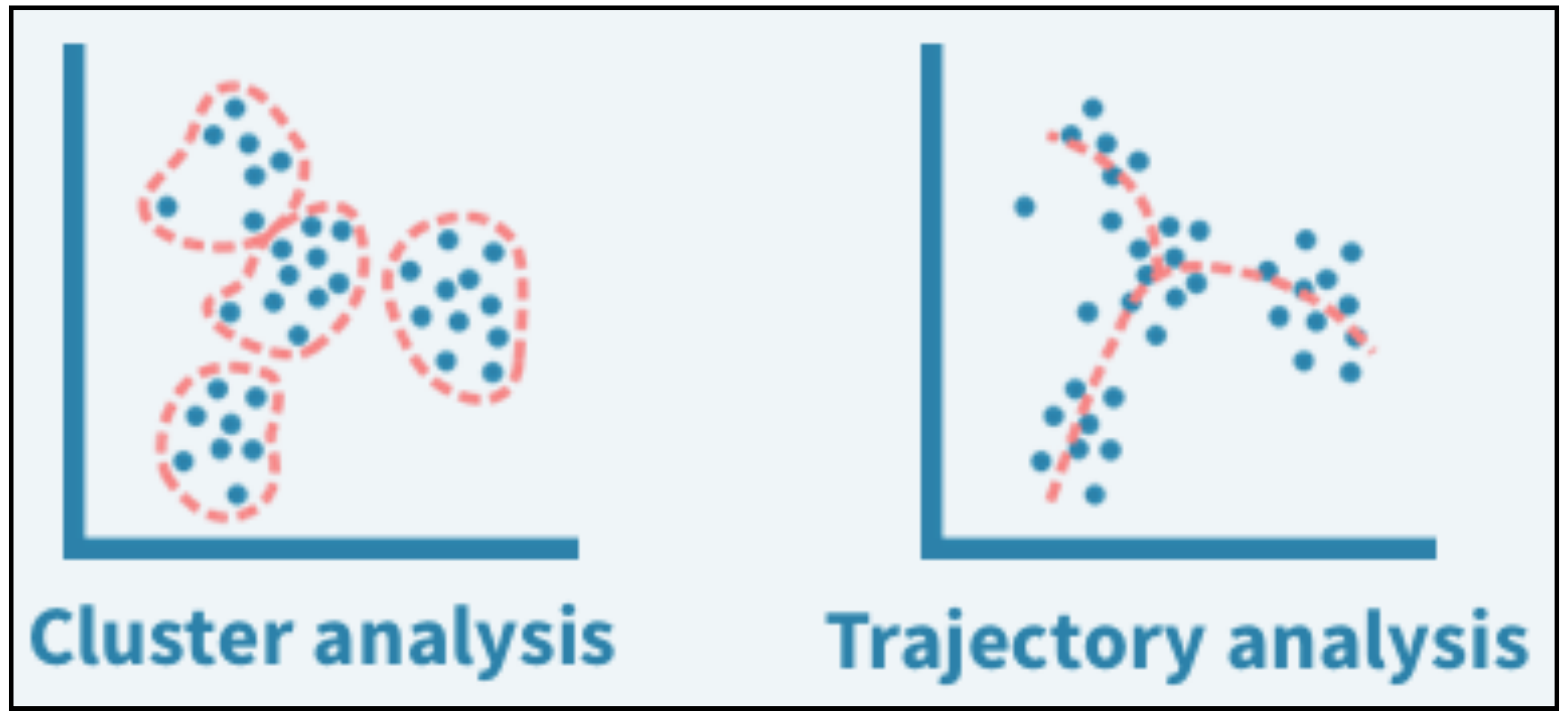


Kidney tumor

	Gene 1	Gene 2	Gene 3	...
Cell 1	10	10	0	
Cell 2	0	15	4	
Cell 3	600	0	20	
⋮				

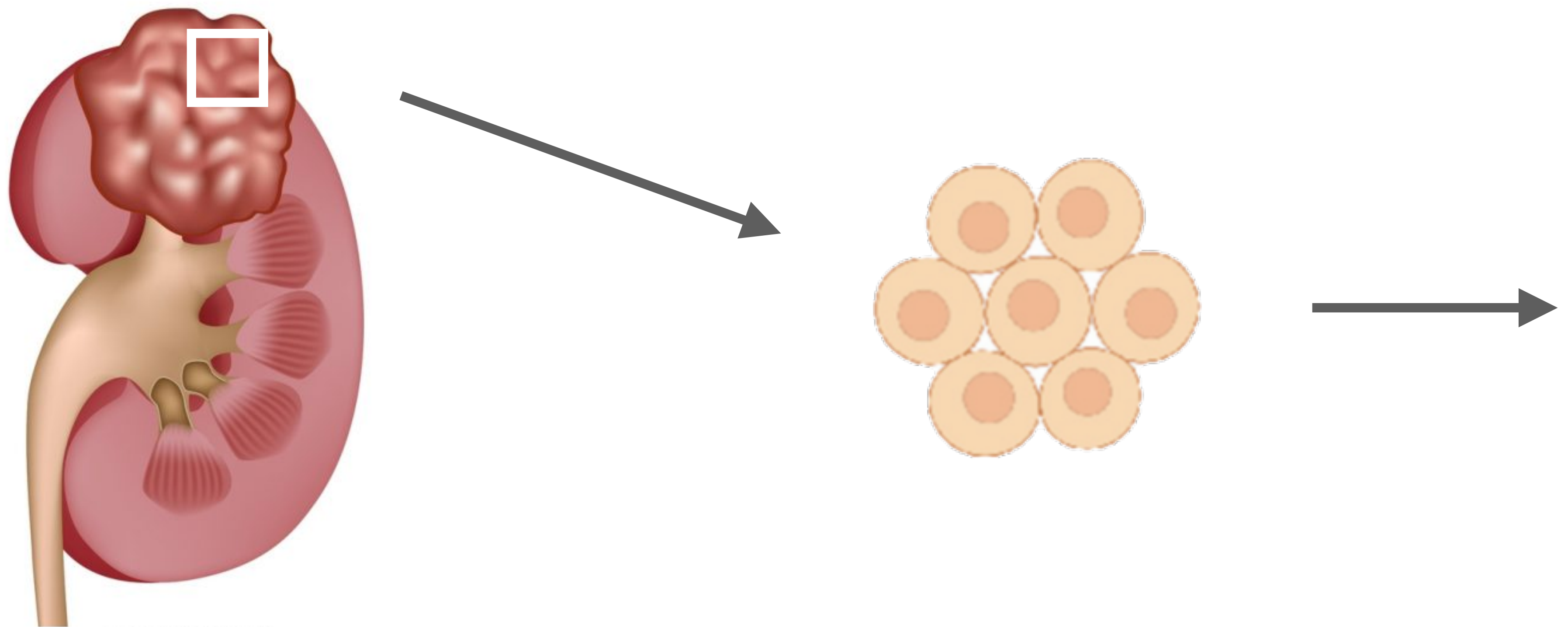
Single-cell RNA read count

🤔 Is there a new subtype of kidney cancer cells?



Unsupervised learning

Laura, PhD, a cancer biologist
(MS in Statistics)

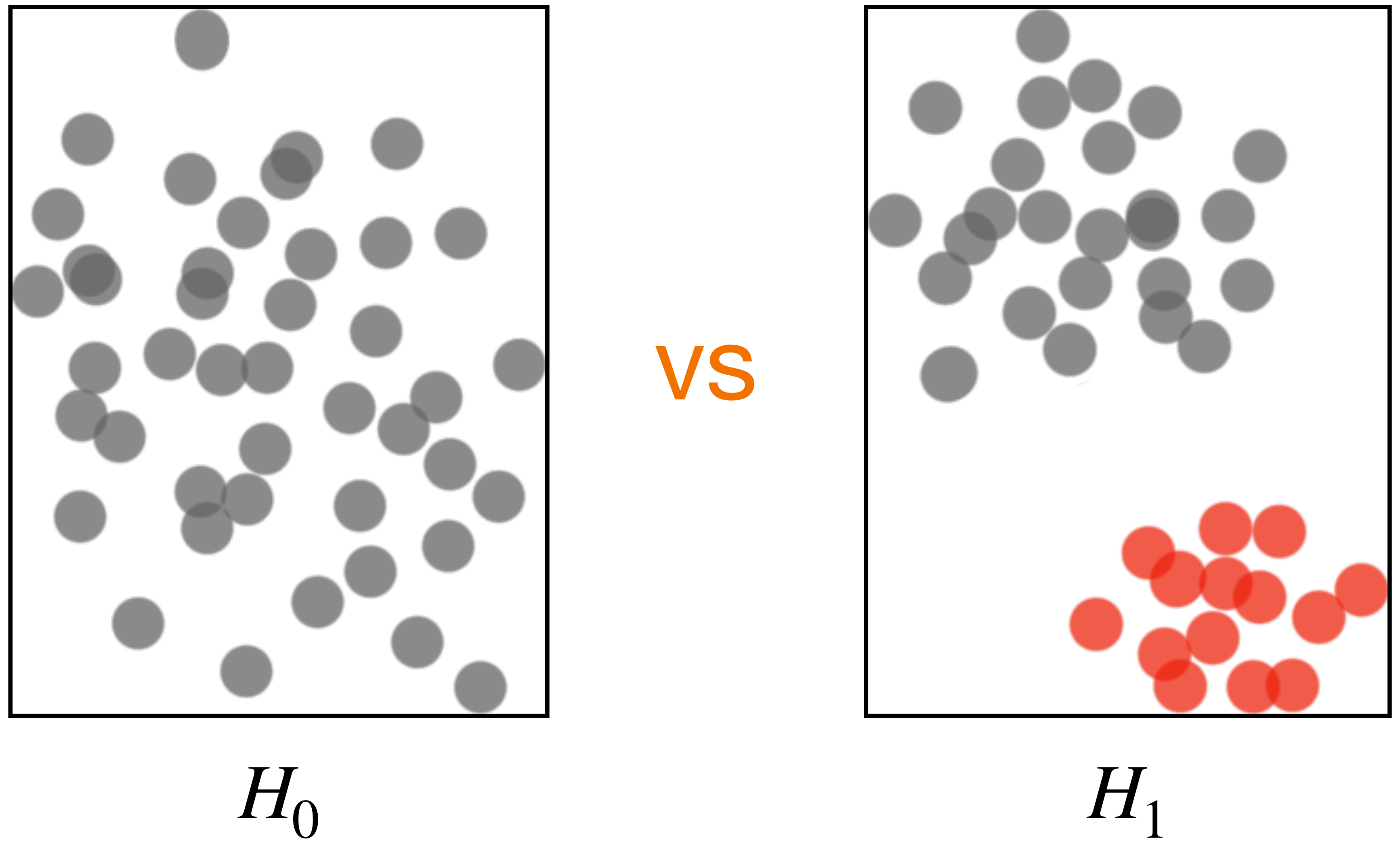


Kidney tumor

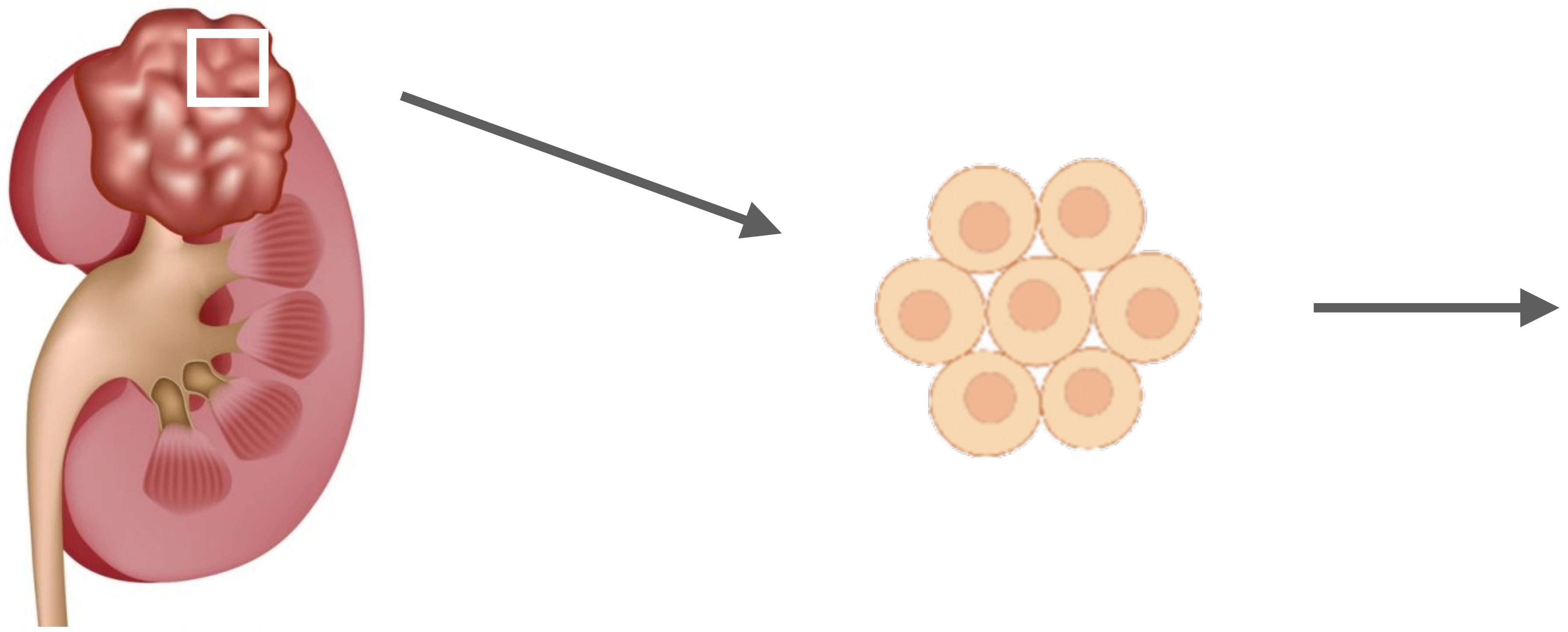
	Gene 1	Gene 2	Gene 3	...
Cell 1	10	10	0	
Cell 2	0	15	4	
Cell 3	600	0	20	
⋮				

Single-cell RNA read count

🤔 Is there a new subtype of kidney cancer cells?



Laura, PhD, a cancer biologist
(MS in Statistics)



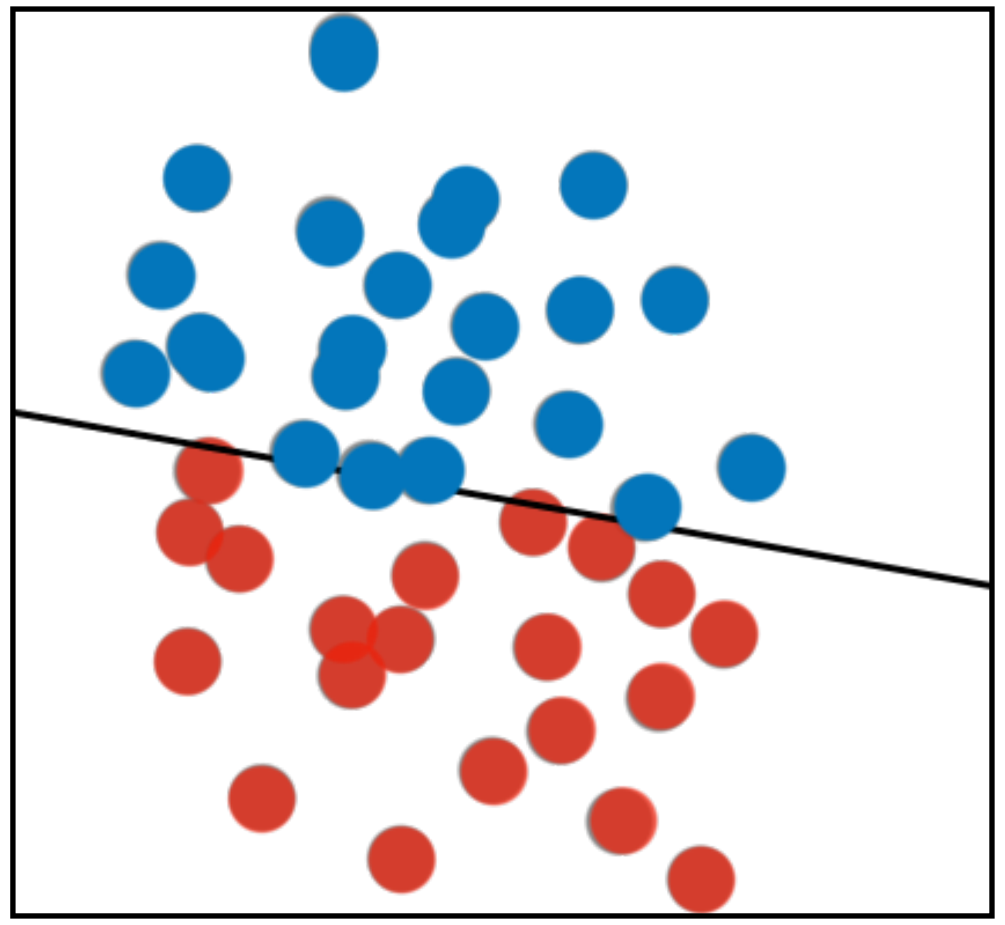
Kidney tumor

	Gene 1	Gene 2	Gene 3	...
Cell 1	10	10	0	
Cell 2	0	15	4	
Cell 3	600	0	20	
⋮				

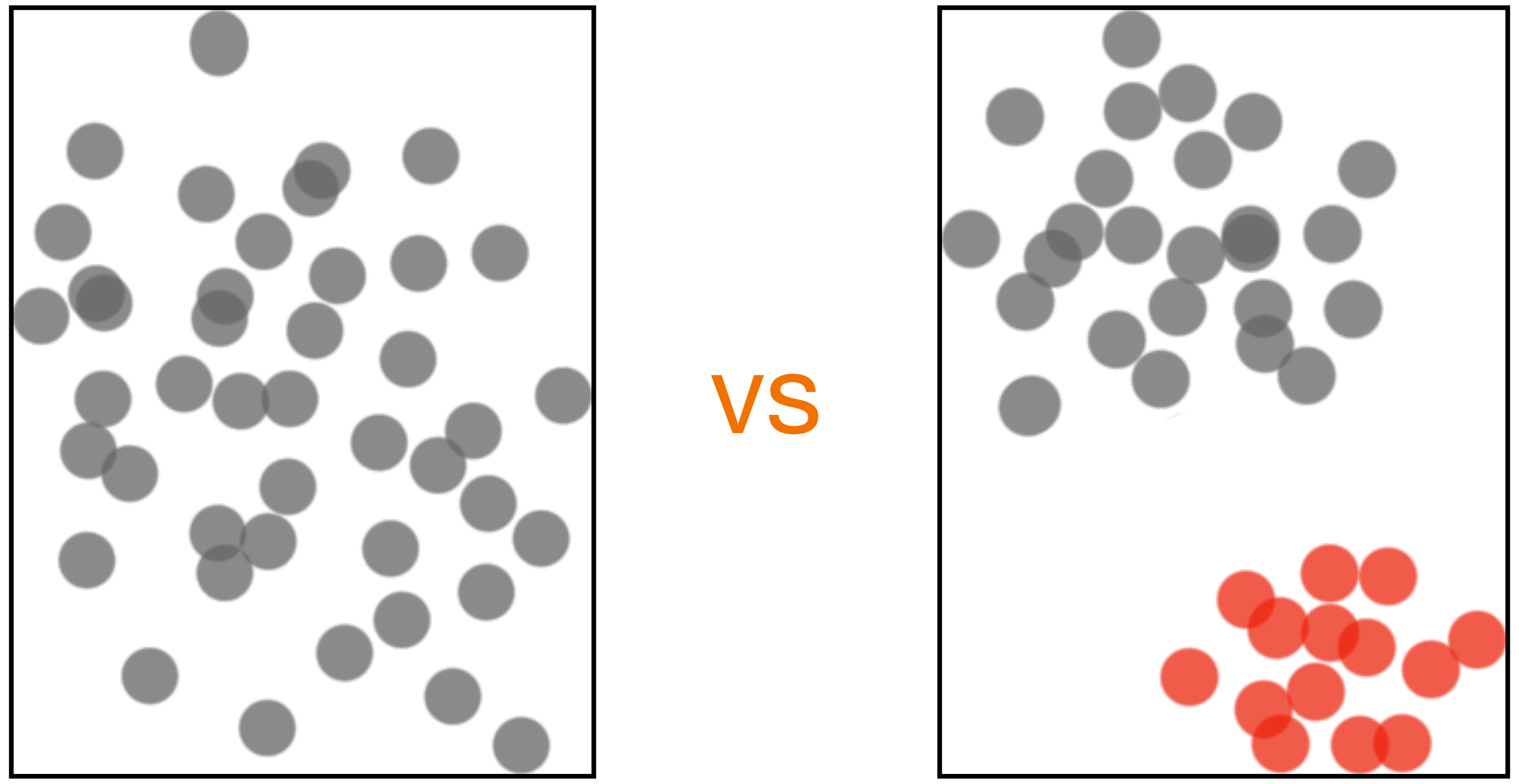
Single-cell RNA read count

🤔 Is there a new subtype of kidney cancer cells?

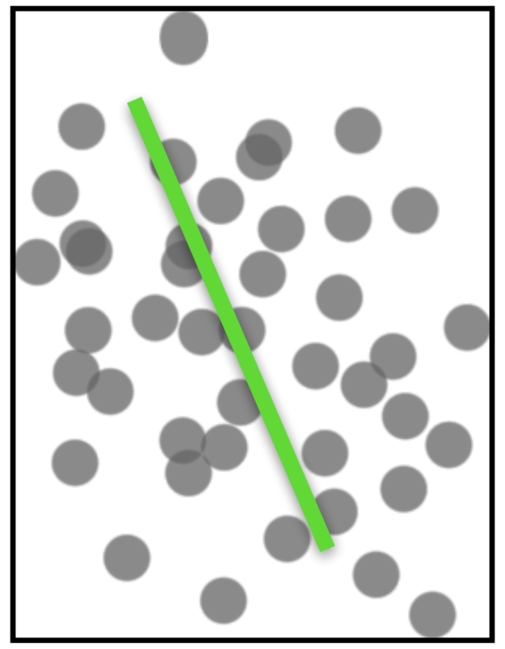
⚠️ Cannot test it with a clustering algorithm.



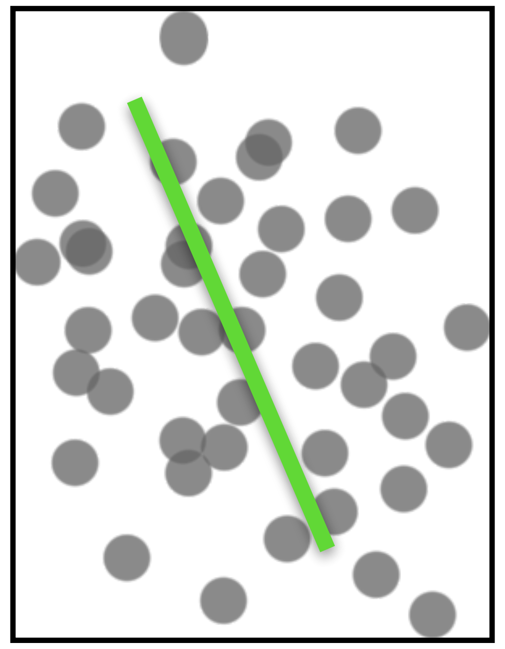
Spurious clusters



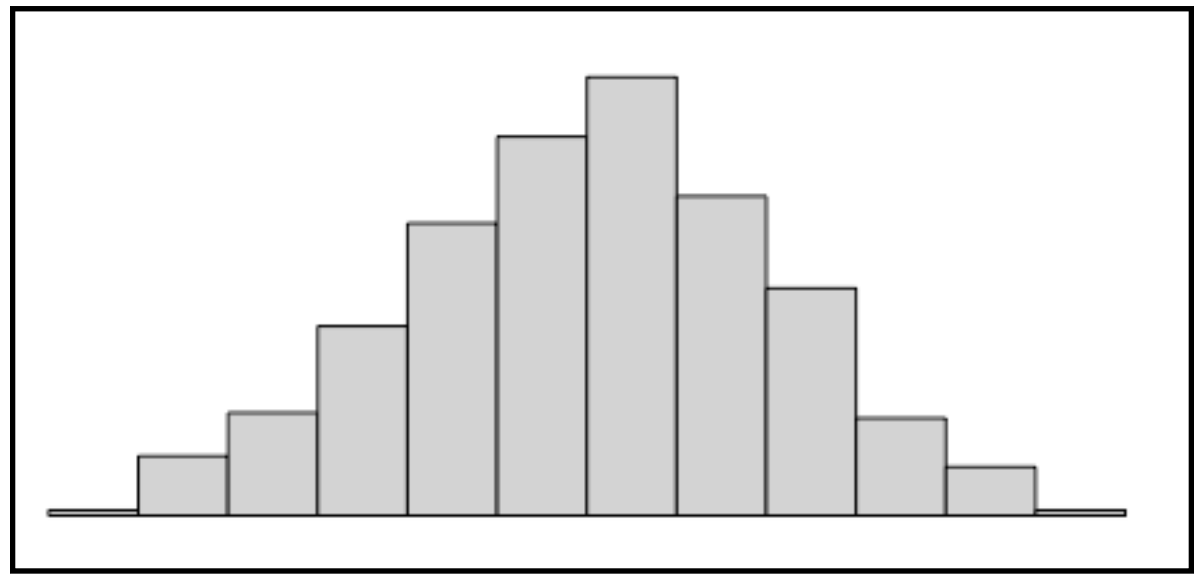
H_0 H_1



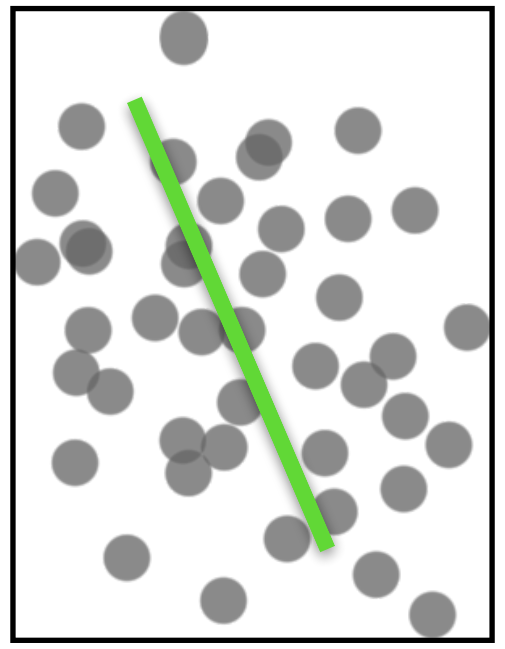
H_0



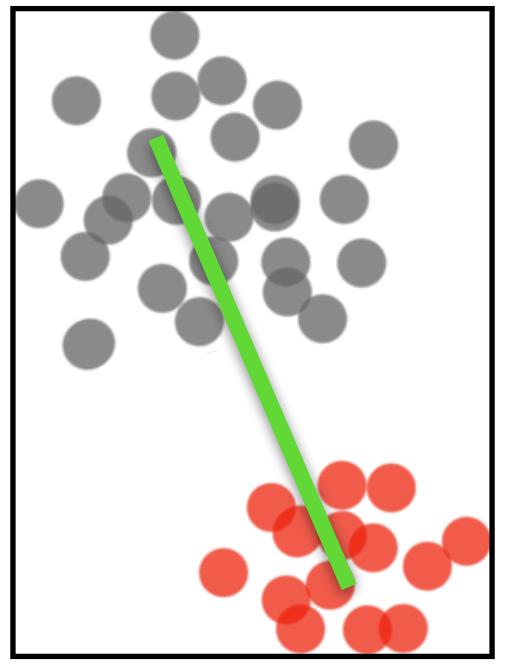
H_0



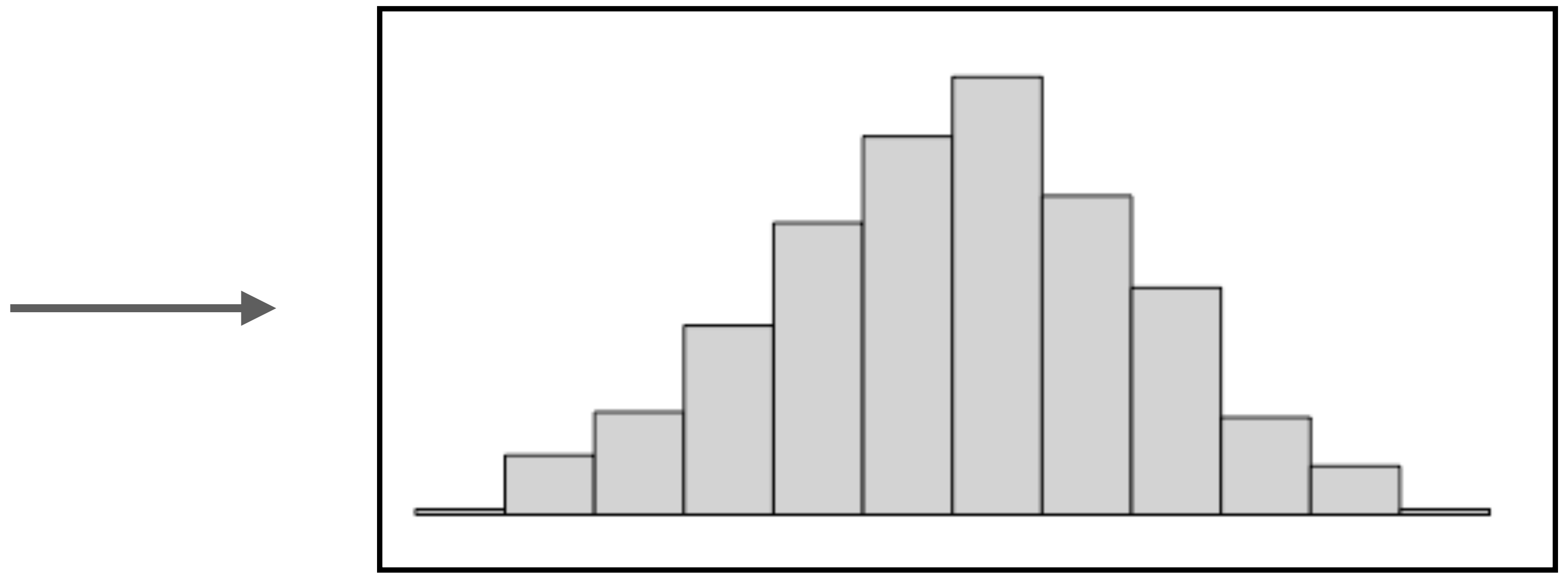
H_0



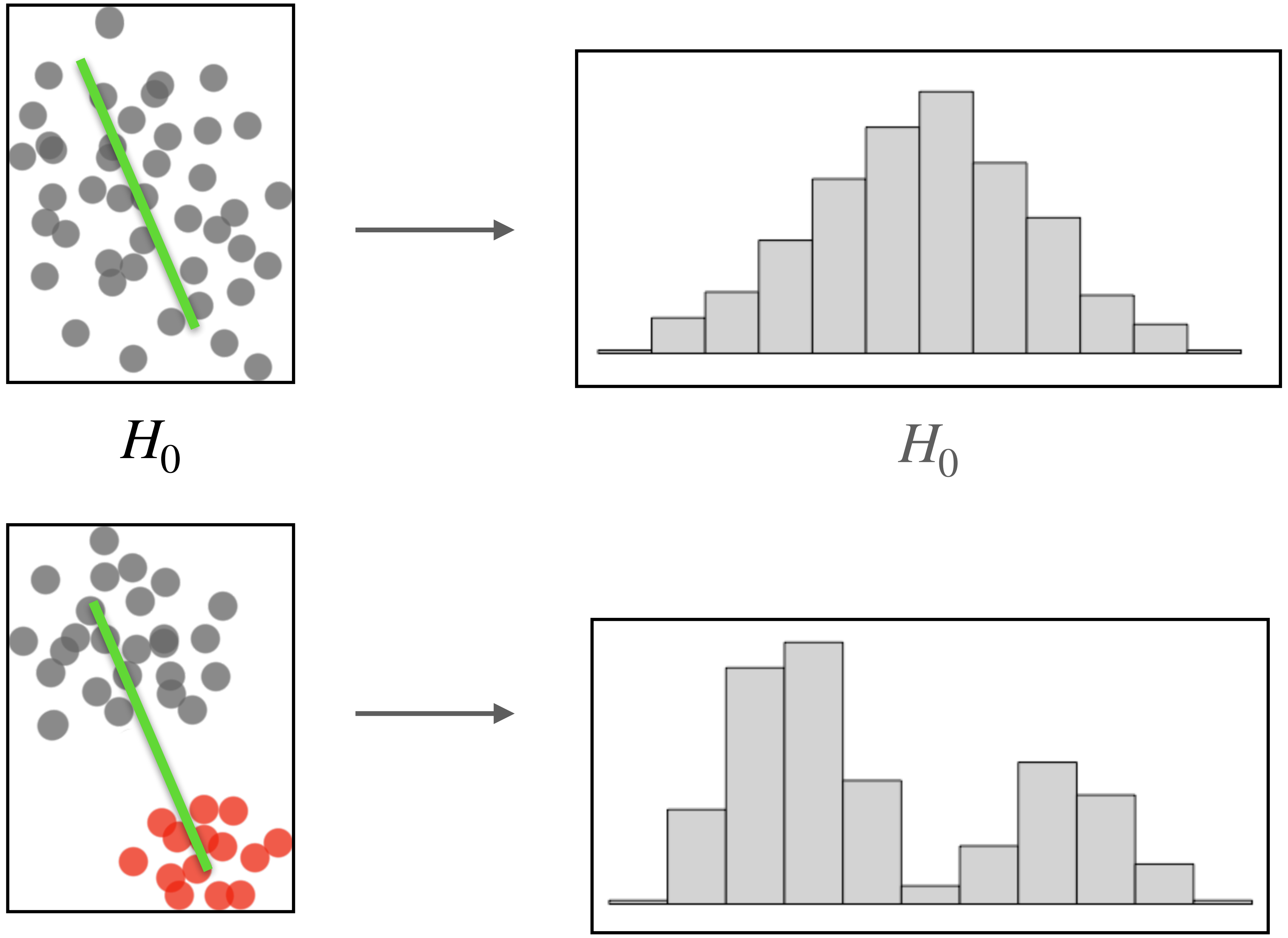
H_0



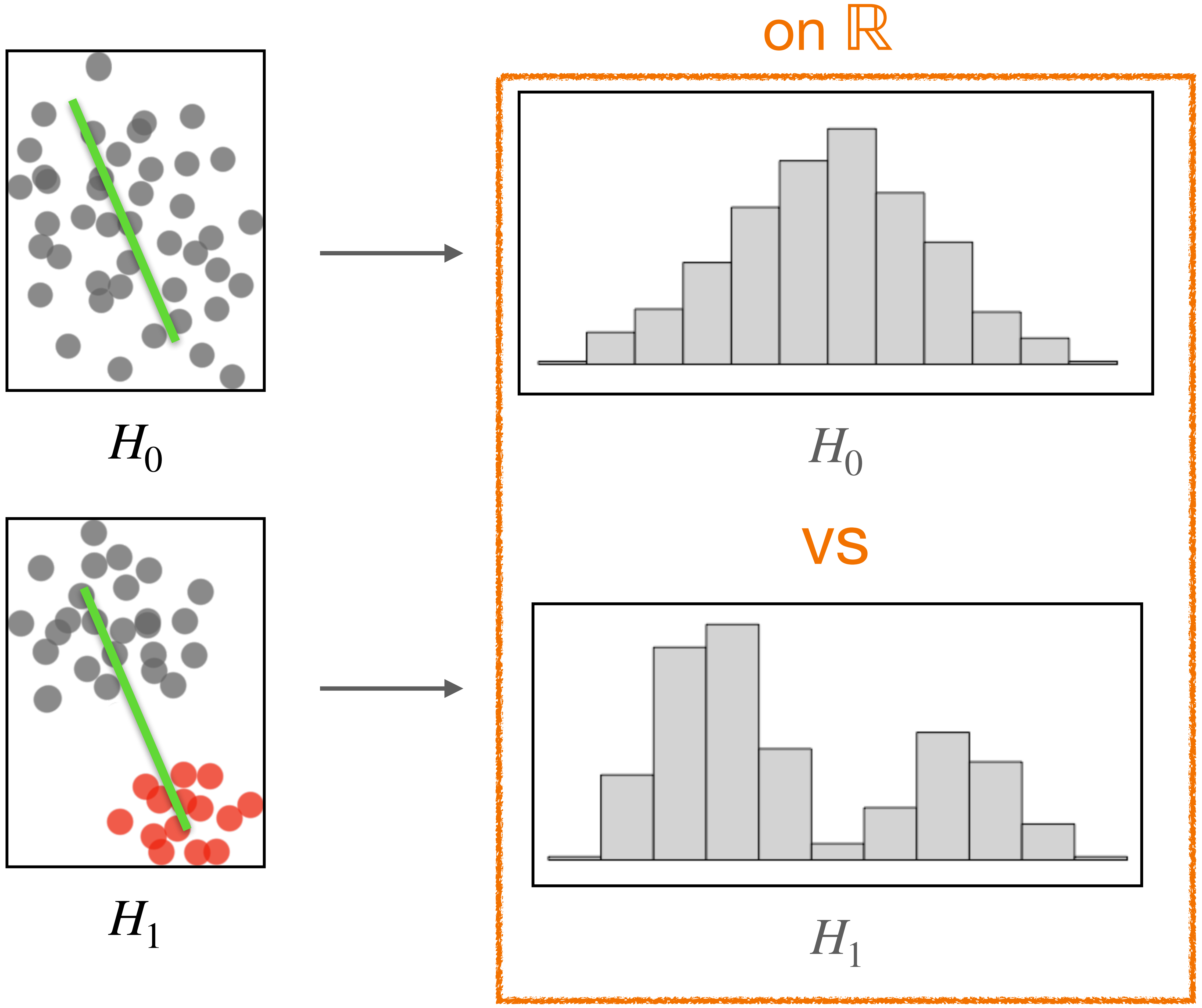
H_1

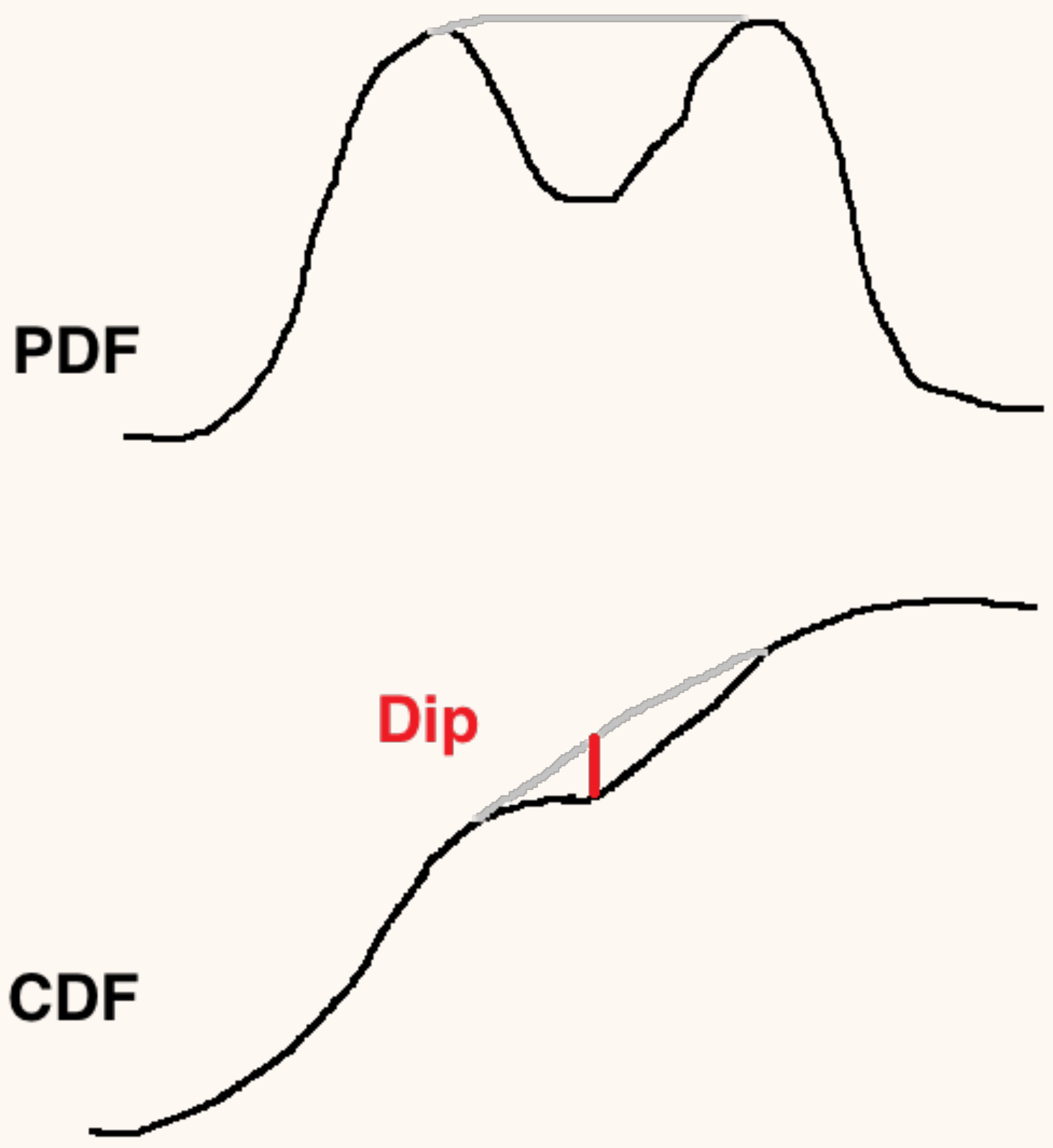


H_0



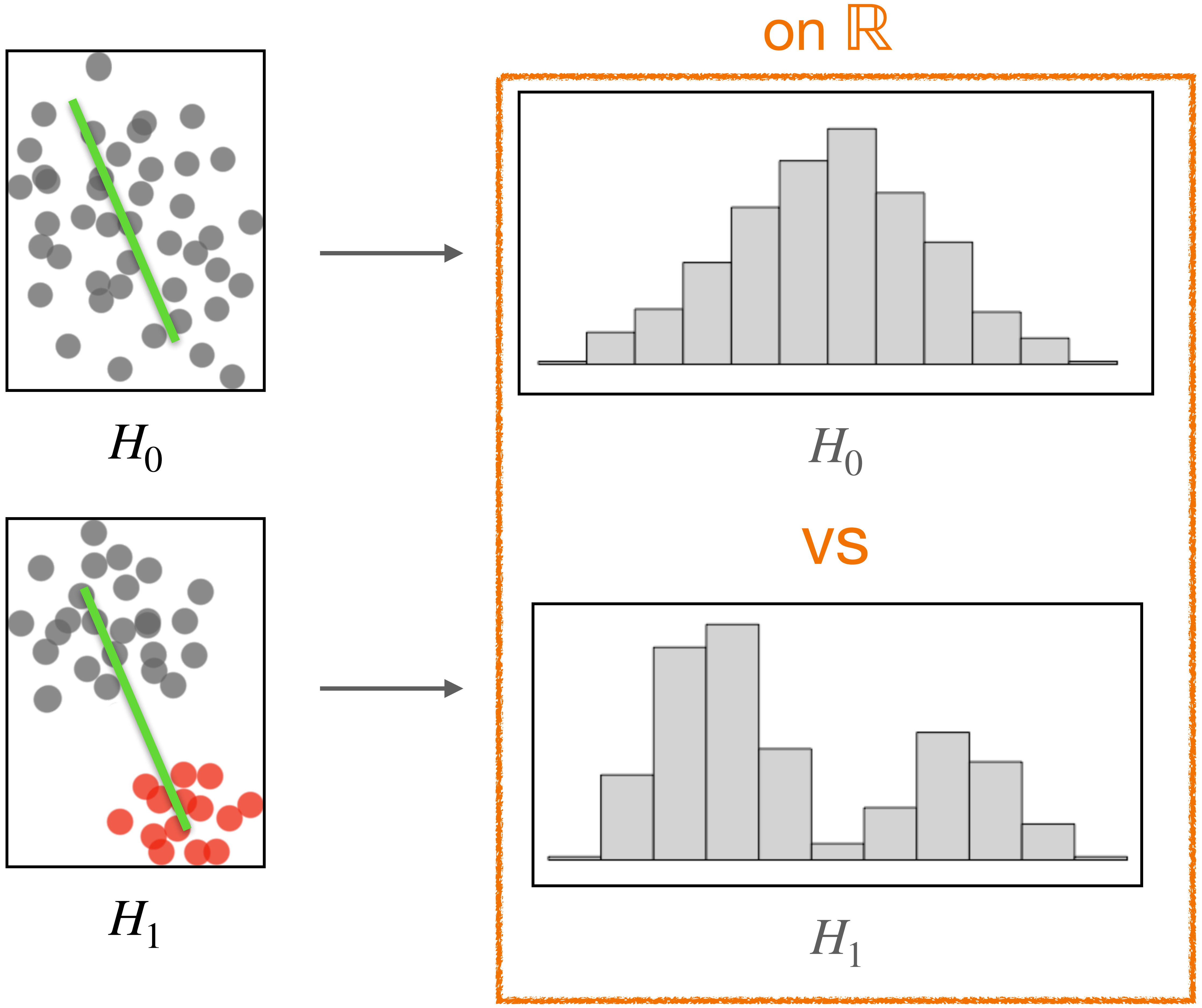
H_0
H_0
H_1
H_1

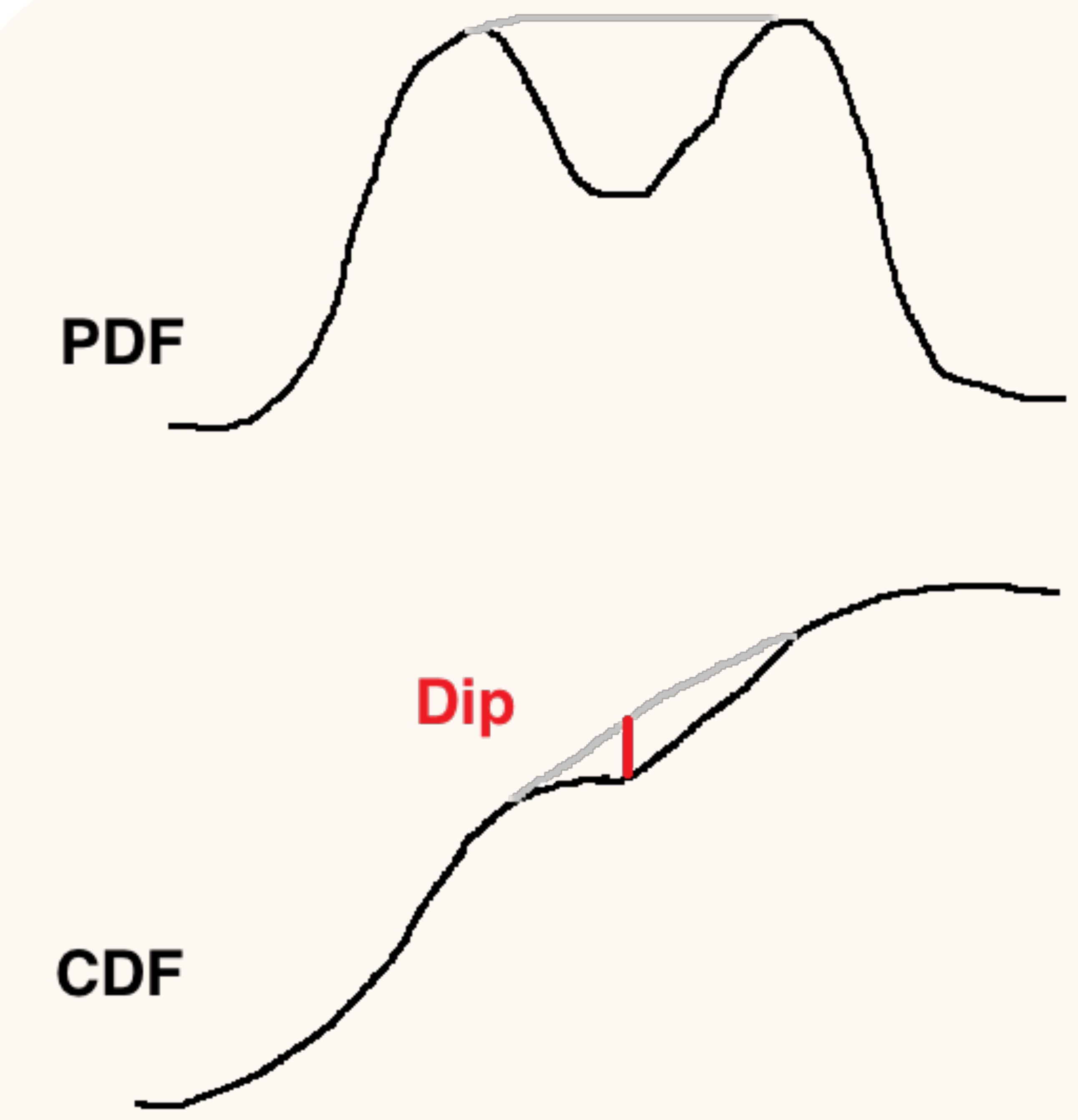




© Edward Ross

J. A. Hartigan & P. M. Hartigan.
The Dip Test of Unimodality.
Annals of Statistics (1985)

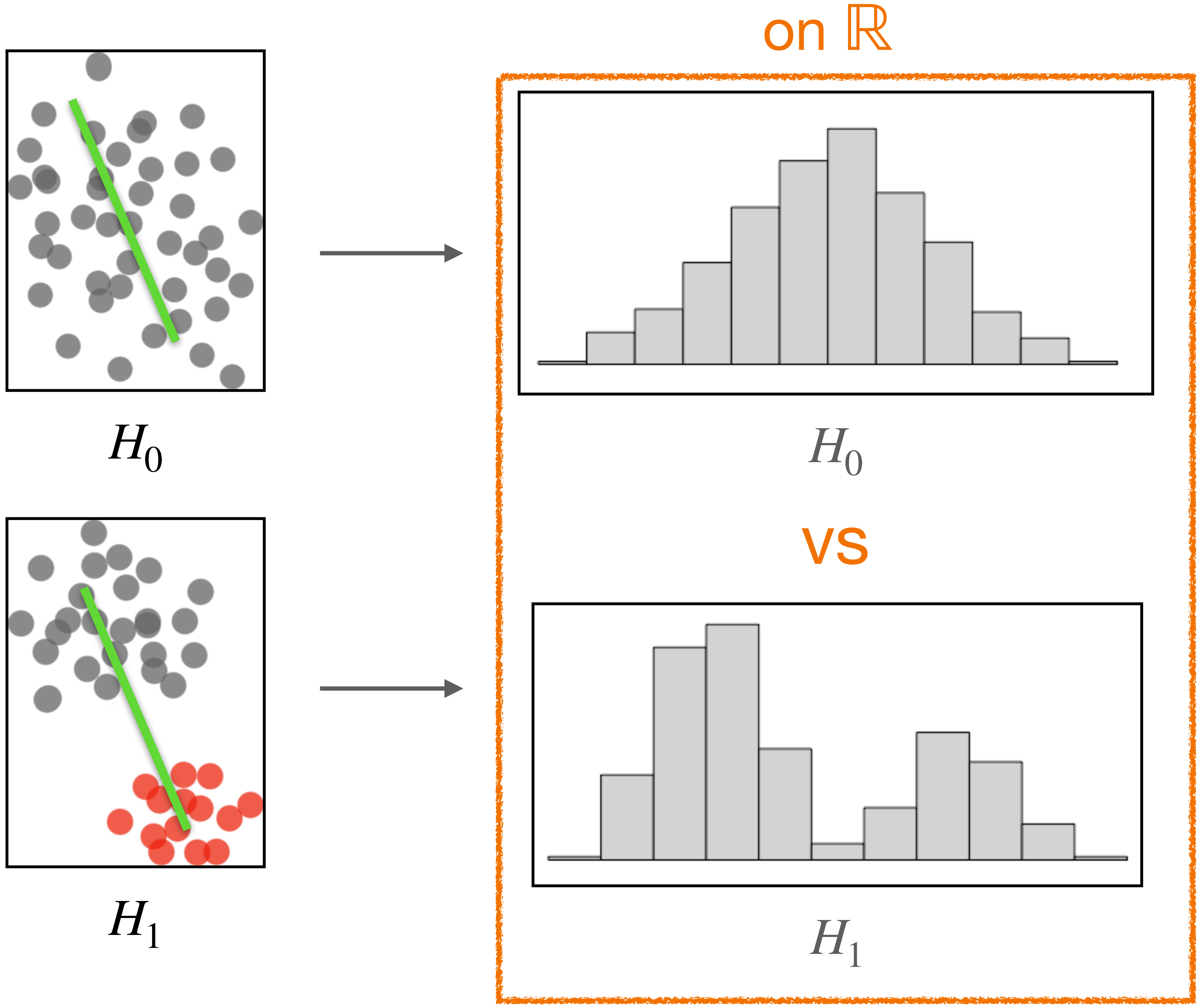


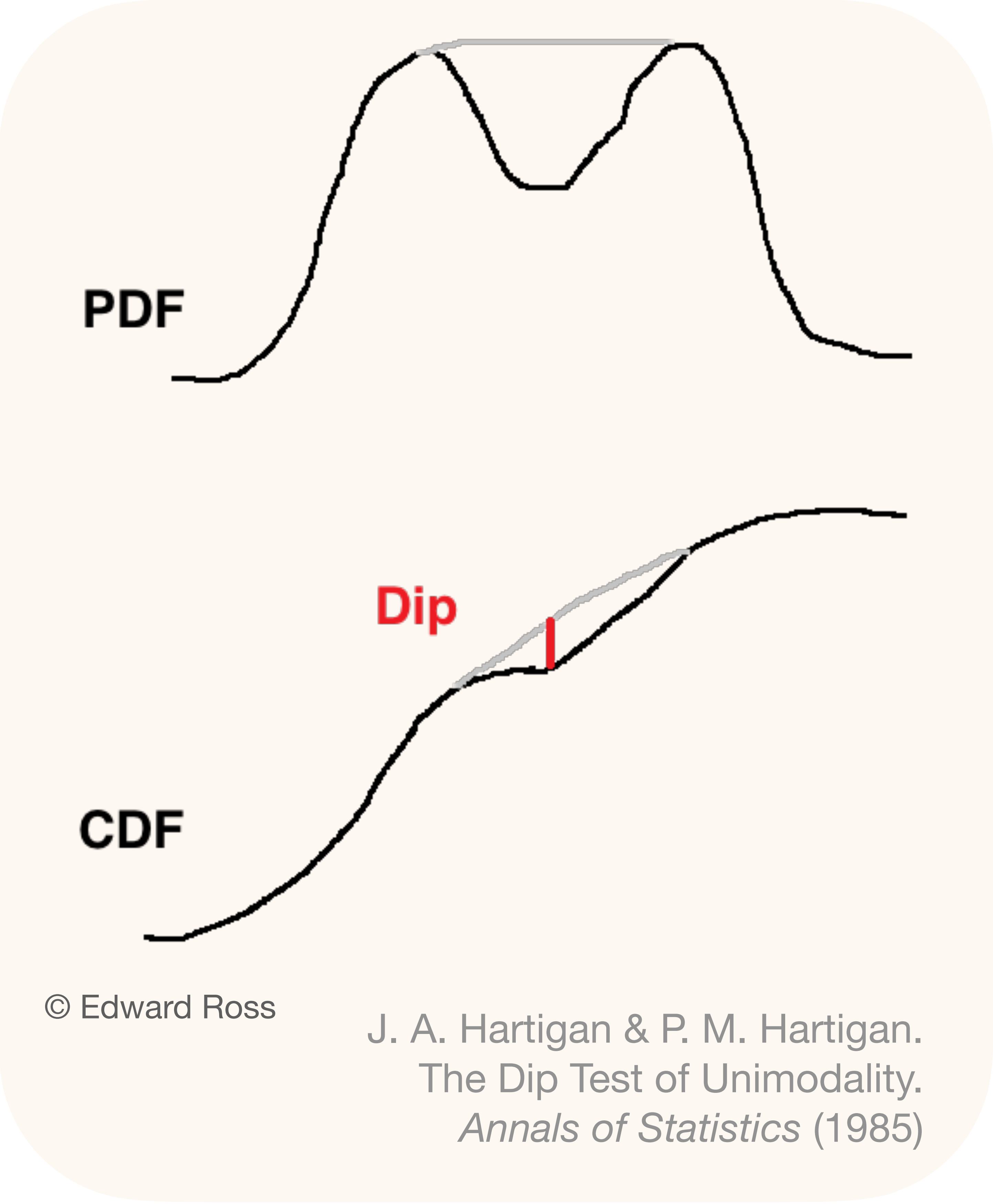


© Edward Ross

J. A. Hartigan & P. M. Hartigan.
The Dip Test of Unimodality.
Annals of Statistics (1985)

💡 Use clustering (e.g., k-means) to find the direction!

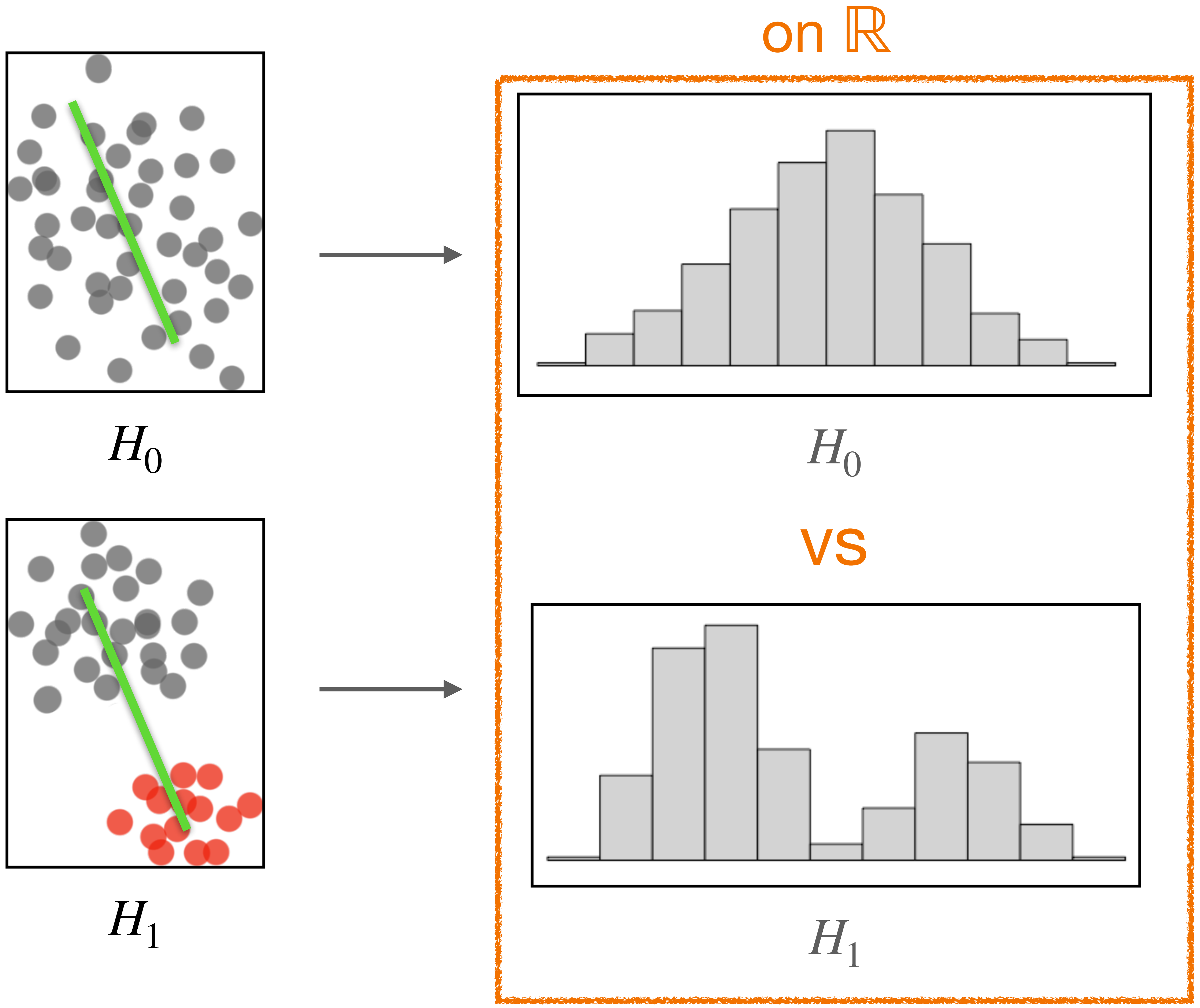


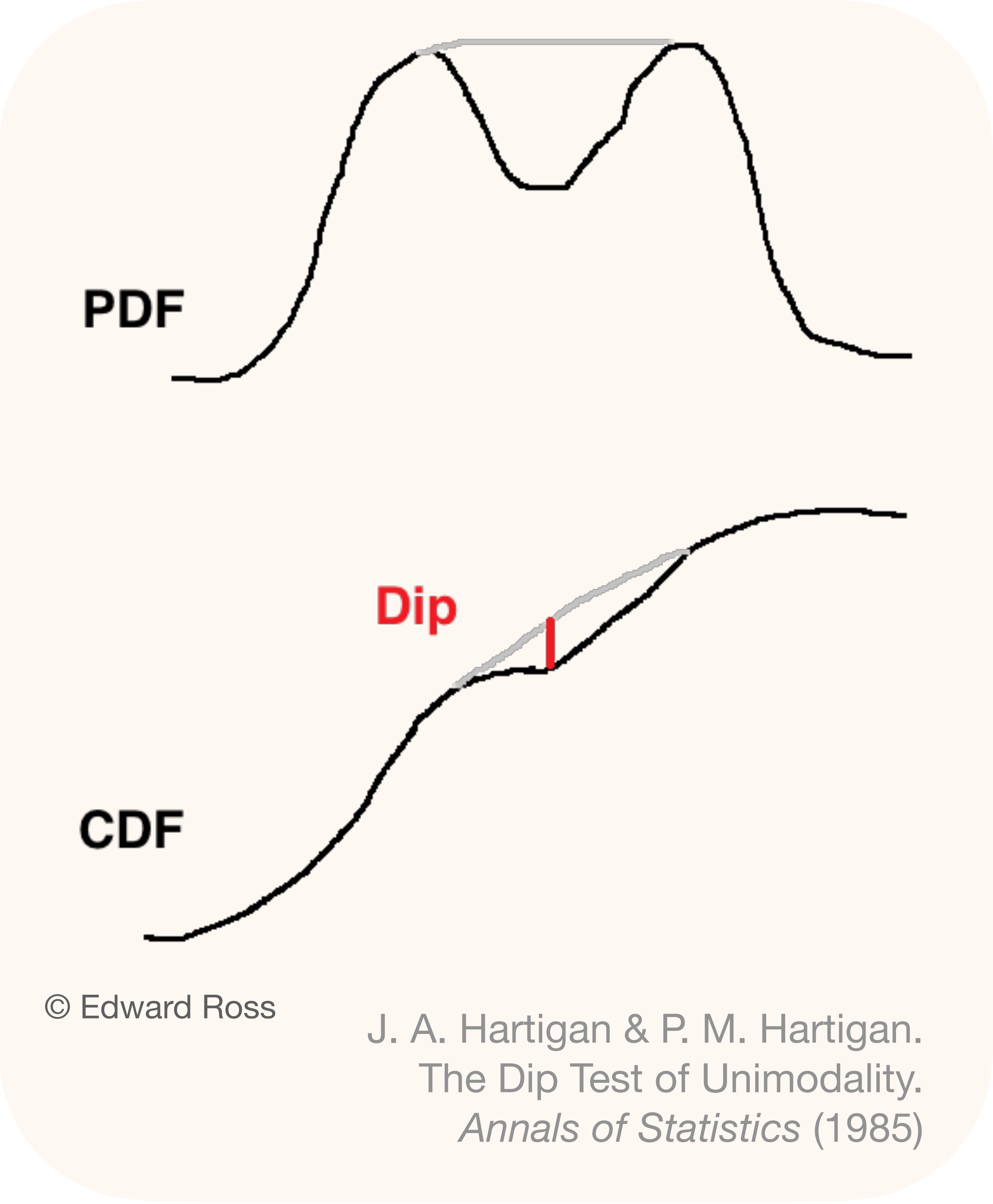


J. A. Hartigan & P. M. Hartigan. The Dip Test of Unimodality. *Annals of Statistics* (1985)

💡 Use clustering (e.g., k-means) to find the direction!

⚠️ Double dipping!





© Edward Ross

J. A. Hartigan & P. M. Hartigan. The Dip Test of Unimodality. *Annals of Statistics* (1985)

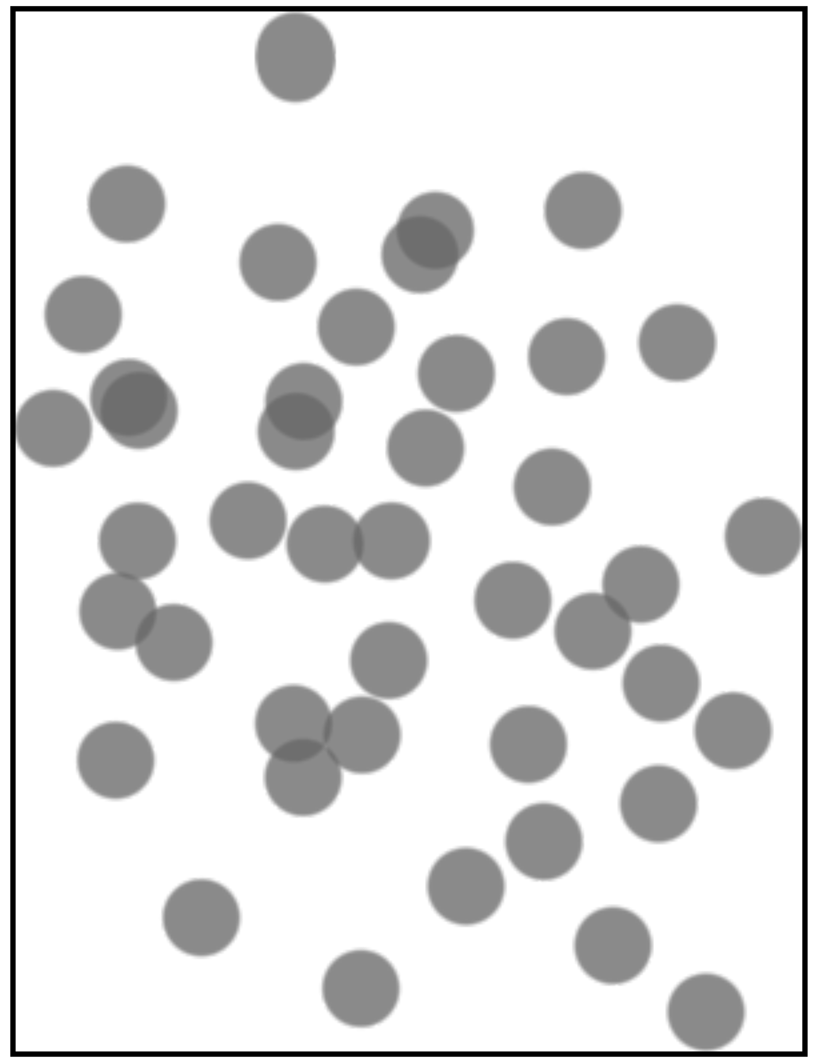
💡 Use clustering (e.g., k-means) to find the direction!

⚠️ Double dipping!

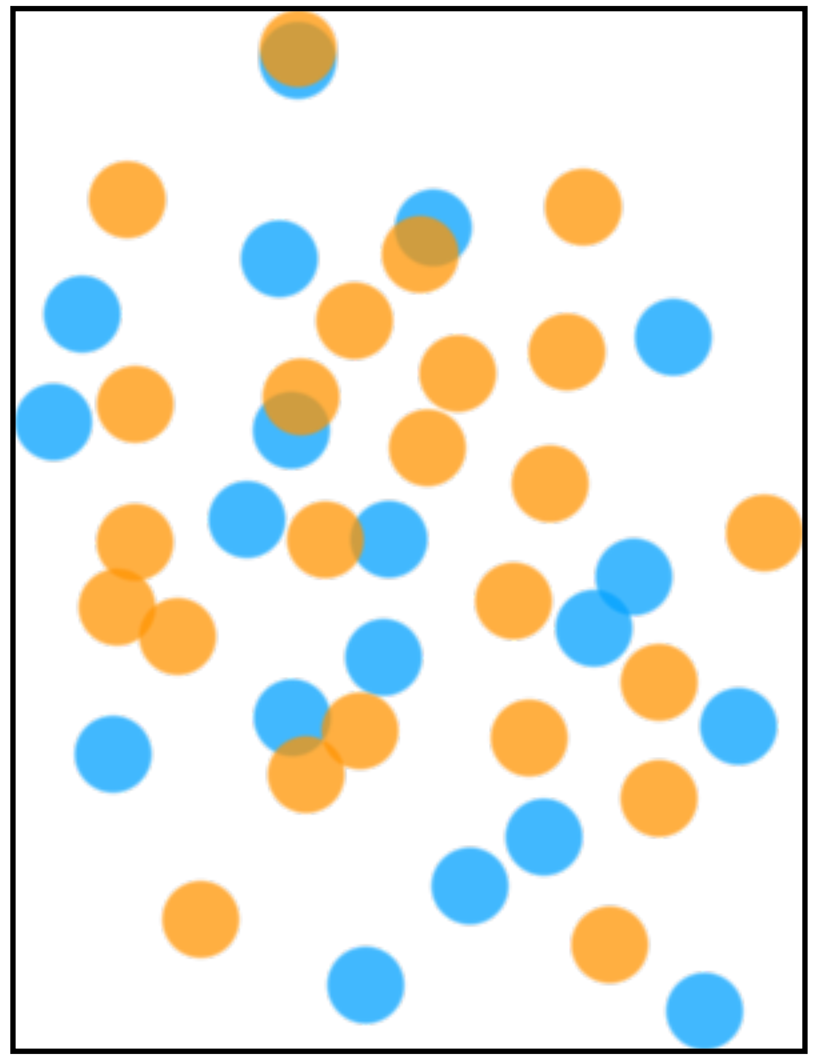
👉 Data splitting!

Hunt and test!

Hunt and test!

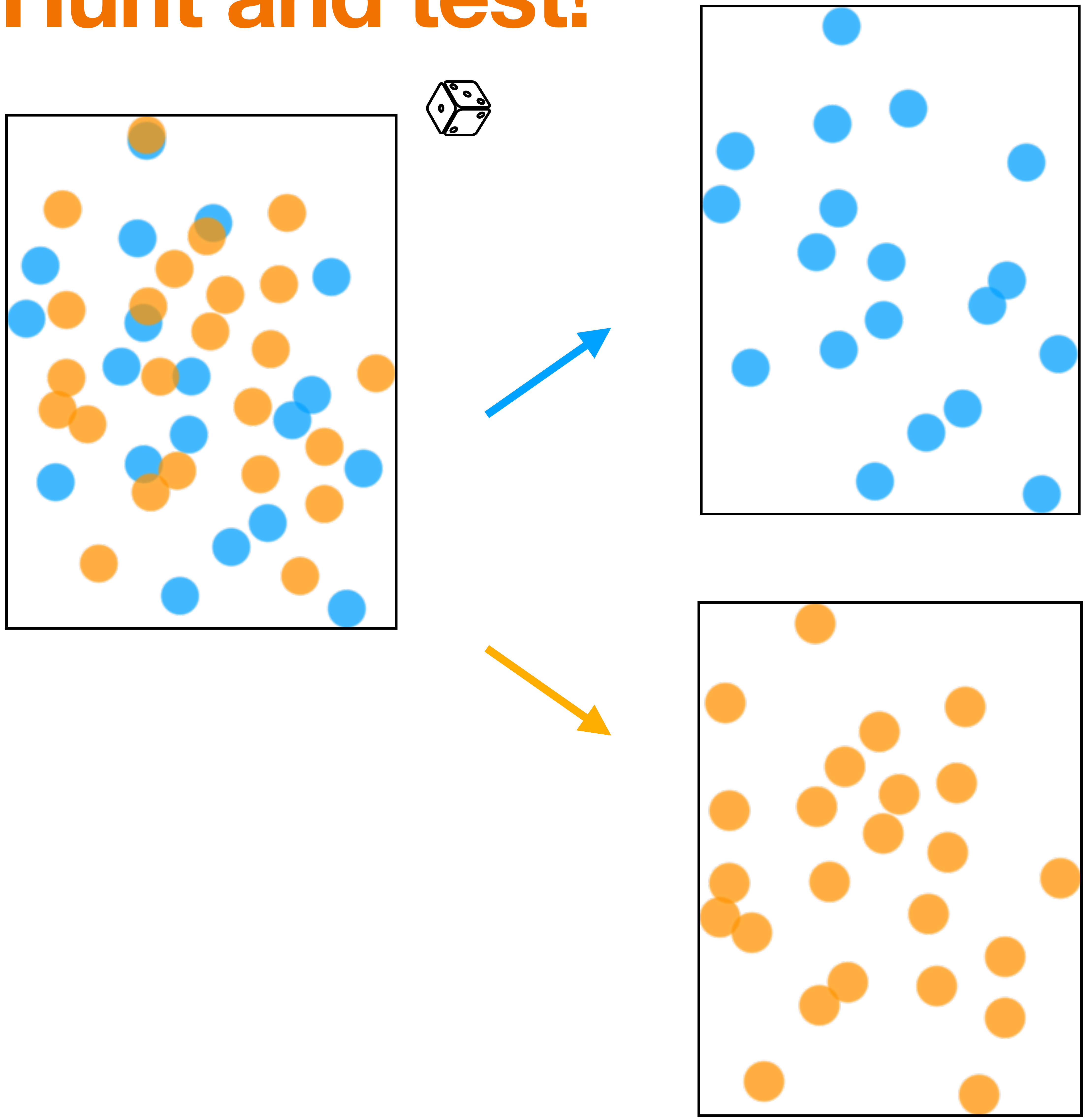


Hunt and test!

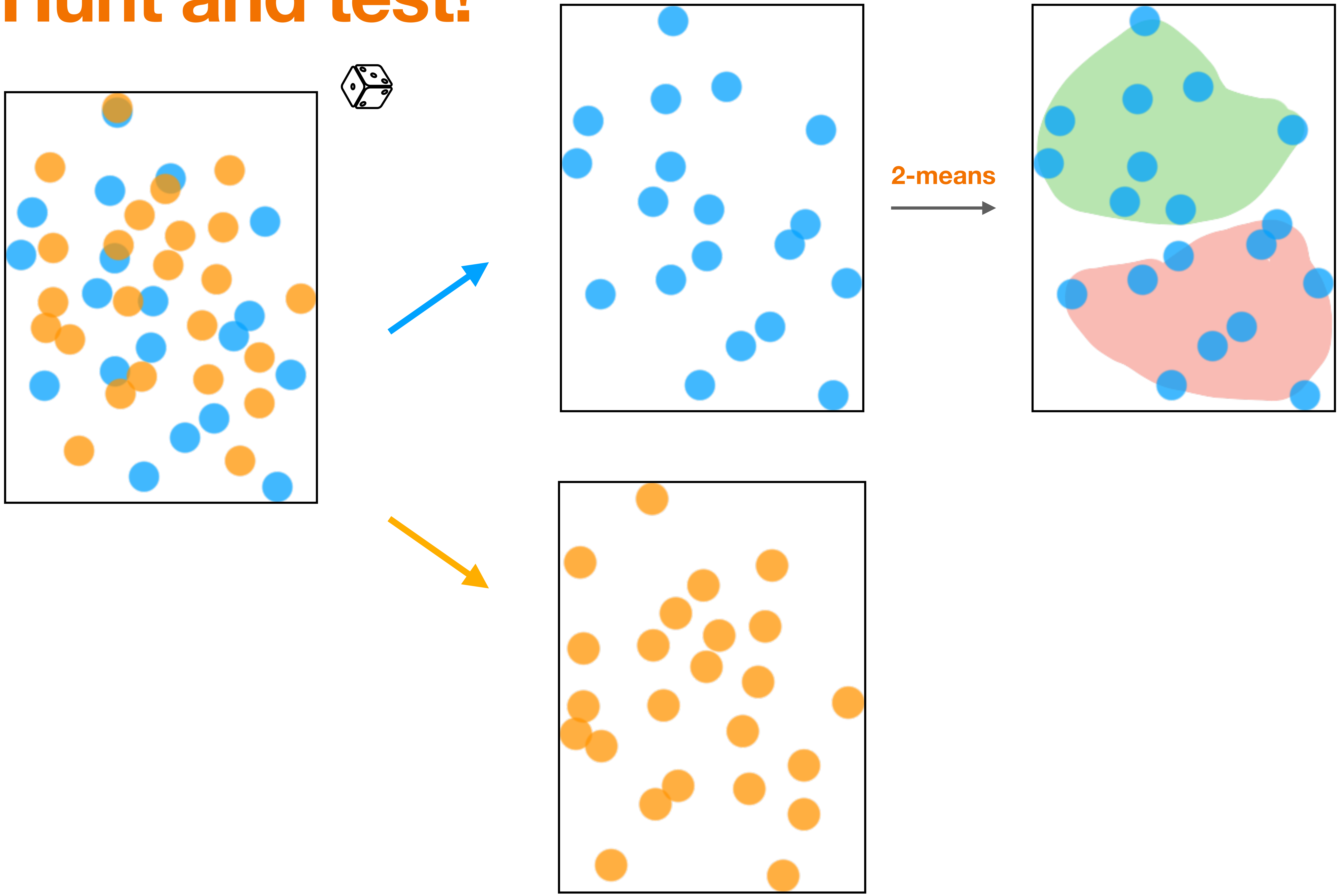


🎲

Hunt and test!

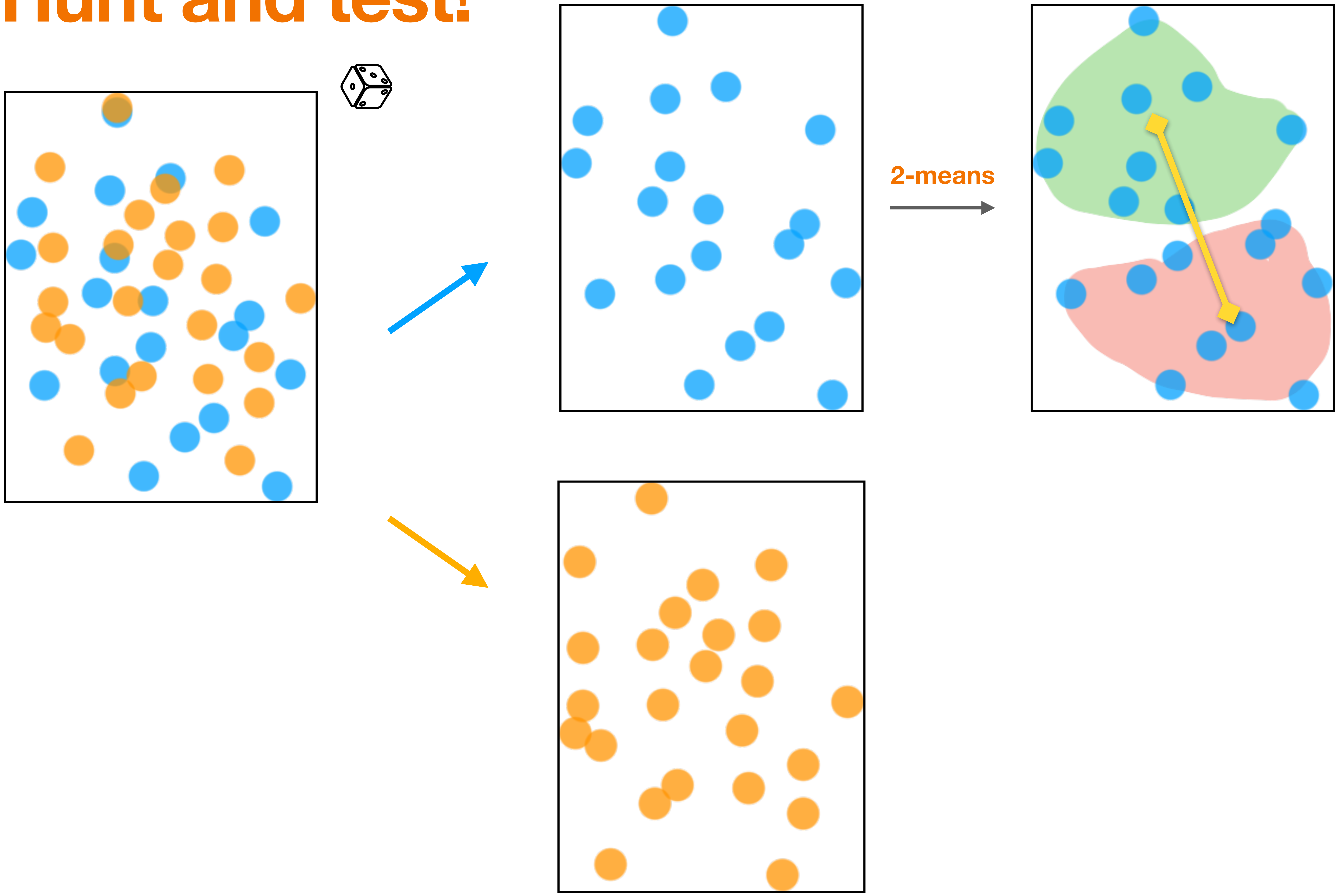


Hunt and test!

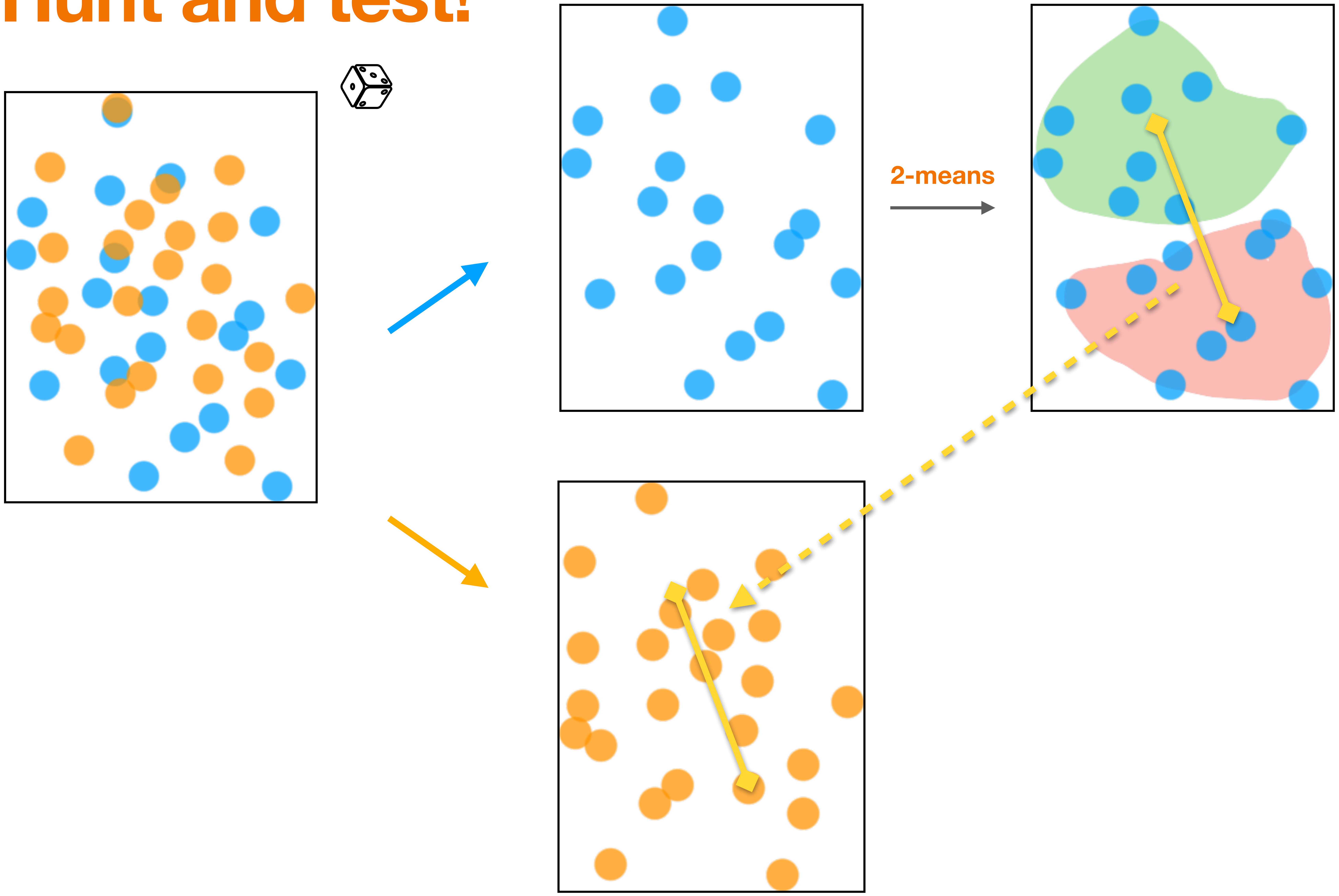


2-means

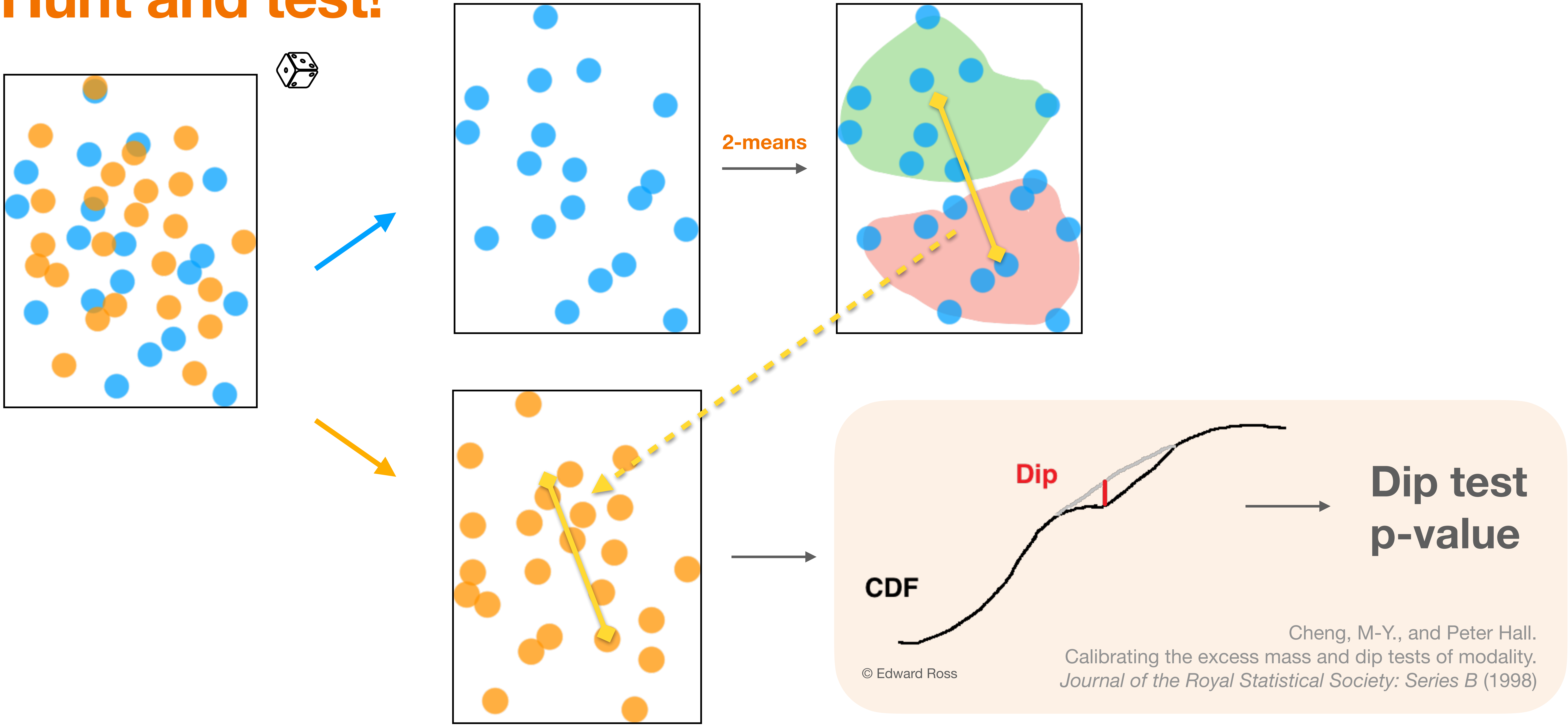
Hunt and test!



Hunt and test!



Hunt and test!



Cheng, M-Y., and Peter Hall.
Calibrating the excess mass and dip tests of modality.
Journal of the Royal Statistical Society: Series B (1998)

```
> replicate(10, hunt.and.test(data))
```

```
> replicate(10, hunt.and.test(data))
```

```
p value

0.2
0.1    .
0.6
0.3
0.006  ***
0.4
0.7
0.8
0.3
0.06   .
```

```
> replicate(10, hunt.and.test(data))
```

```
p value

0.2
0.1     .
0.6
0.3
0.006  ***
0.4
0.7
0.8
0.3
0.06    .
```

🤔 *“Significant 1 out of 10 times.”*

```
> replicate(10, hunt.and.test(data))
```

```
p value

0.2
0.1      .
0.6
0.3
0.006   ***
0.4
0.7
0.8
0.3
0.06     .
```

🤔 *“Significant 1 out of 10 times.”*

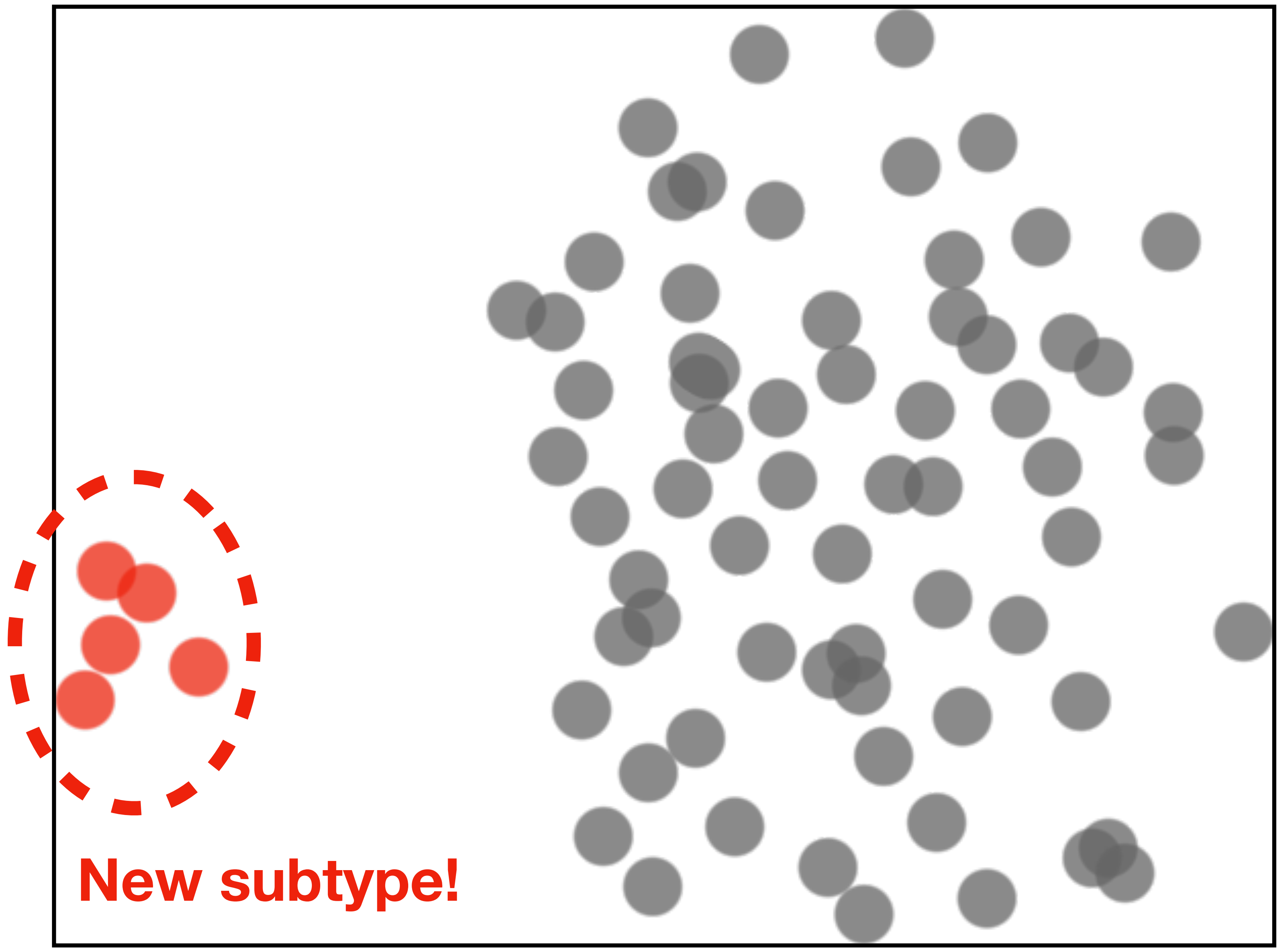
🙁 *“No evidence for a new subtype.”*

```
> replicate(10, hunt.and.test(data))
```

```
p value

0.2
0.1      .
0.6
0.3
0.006   ***
0.4
0.7
0.8
0.3
0.06     .
```

🤔 *“Significant 1 out of 10 times.”*

🙁 *“No evidence for a new subtype.”*

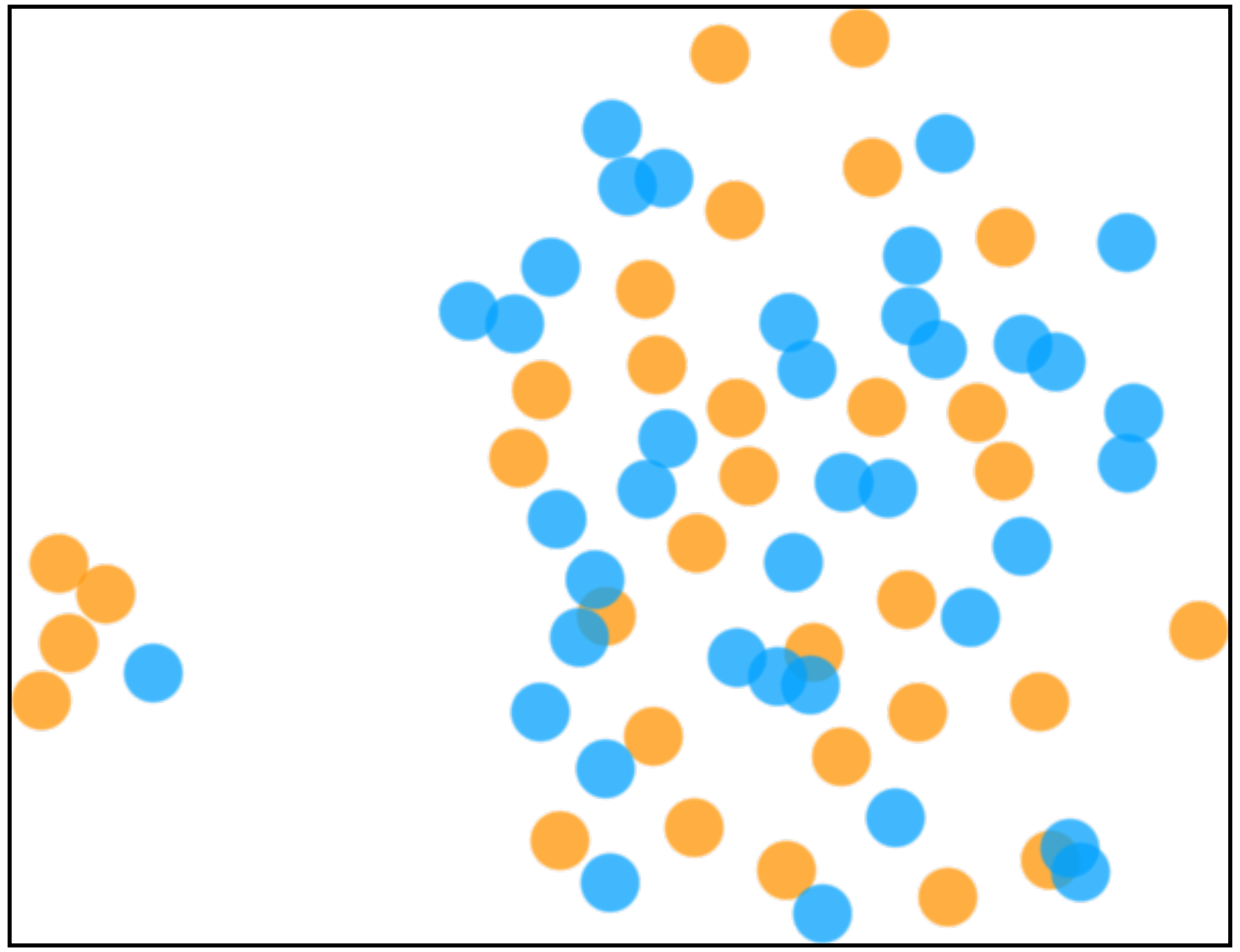


```
> replicate(10, hunt.and.test(data))
```

```
p value

0.2
0.1      .
0.6
0.3
0.006   ***
0.4
0.7
0.8
0.3
0.06     .
```

🤔 *“Significant 1 out of 10 times.”*

🙁 *“No evidence for a new subtype.”*

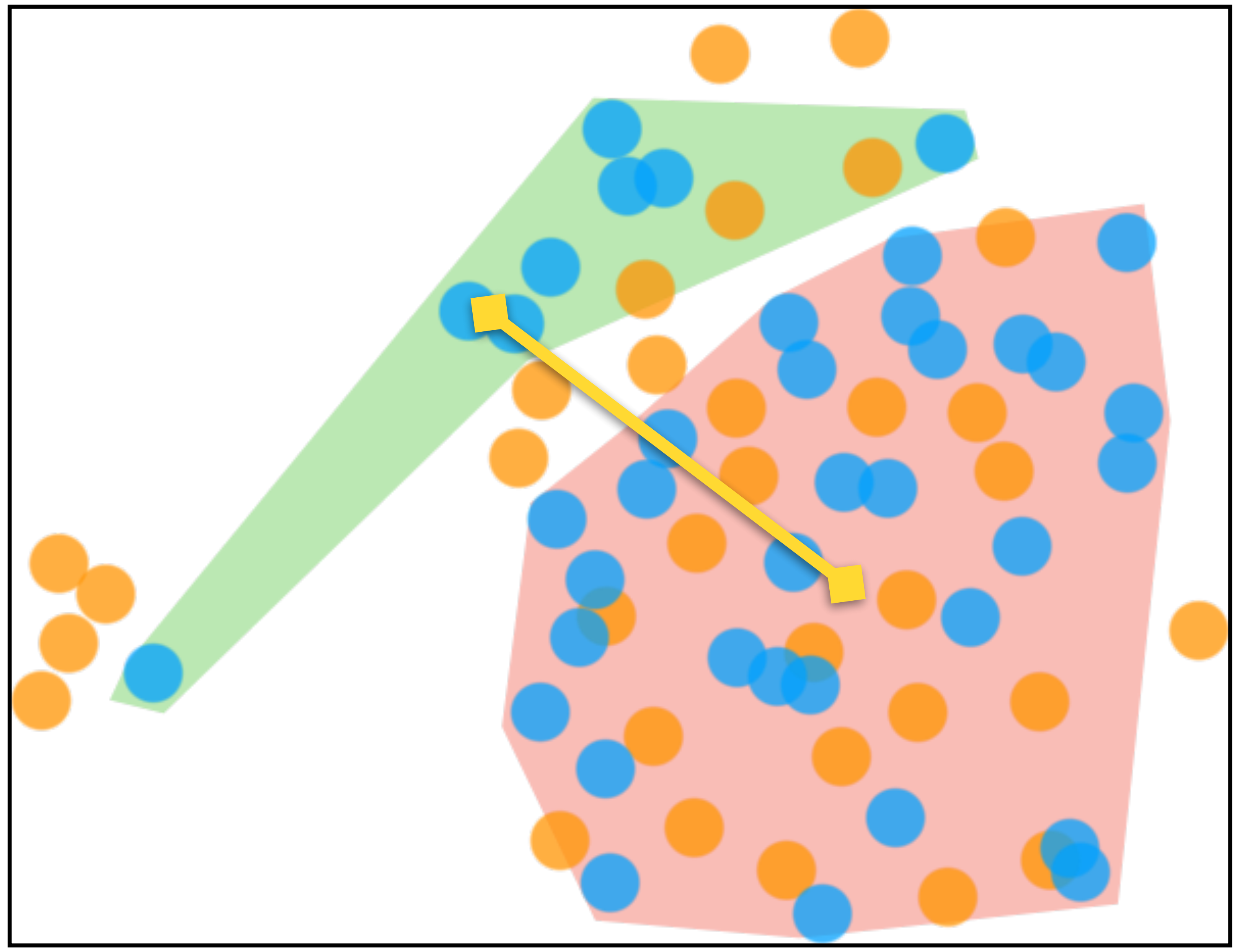


```
> replicate(10, hunt.and.test(data))
```

```
p value

0.2
0.1    .
0.6
0.3
0.006  ***
0.4
0.7
0.8
0.3
0.06   .
```

🤔 *“Significant 1 out of 10 times.”*

🙁 *“No evidence for a new subtype.”*



🎲

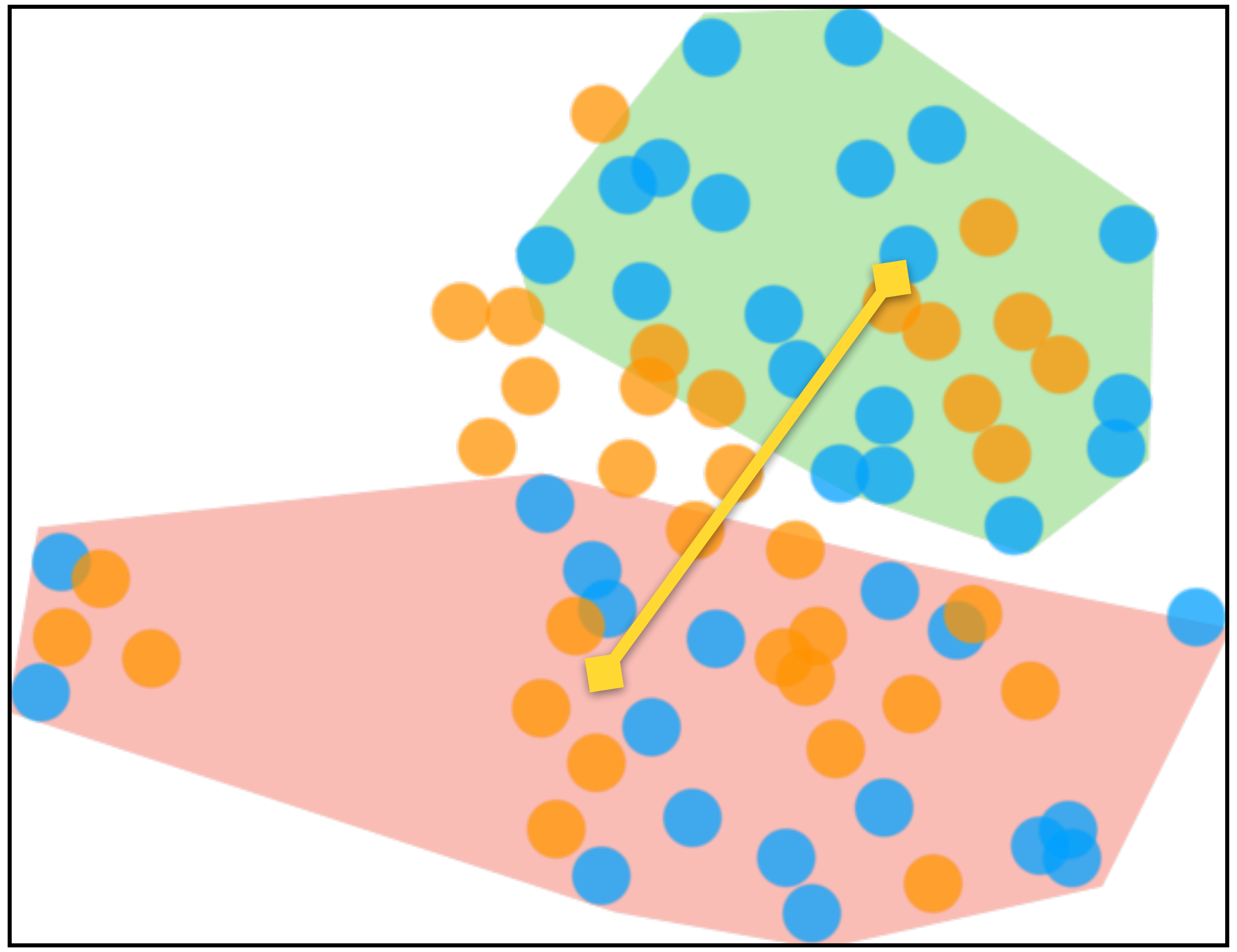
❌

```
> replicate(10, hunt.and.test(data))
```

```
p value

0.2
0.1       .
0.6
0.3
0.006   ***
0.4
0.7
0.8
0.3
0.06      .
```

🤔 *“Significant 1 out of 10 times.”*

🙁 *“No evidence for a new subtype.”*



🎲

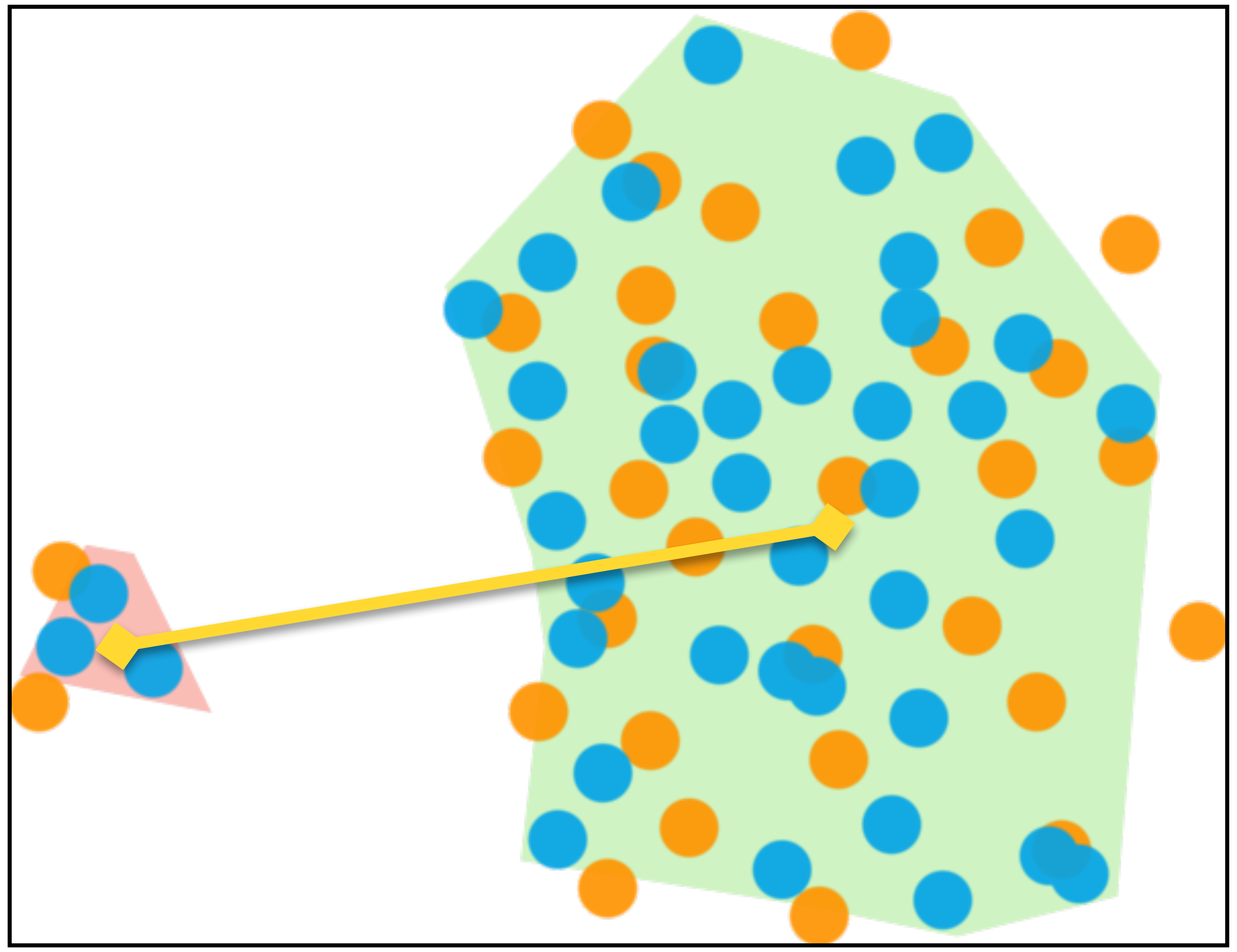
✗

```
> replicate(10, hunt.and.test(data))
```

```
p value

0.2
0.1     .
0.6
0.3
0.006   ***
0.4
0.7
0.8
0.3
0.06    .
```

🤔 *“Significant 1 out of 10 times.”*

🙁 *“No evidence for a new subtype.”*



🎲

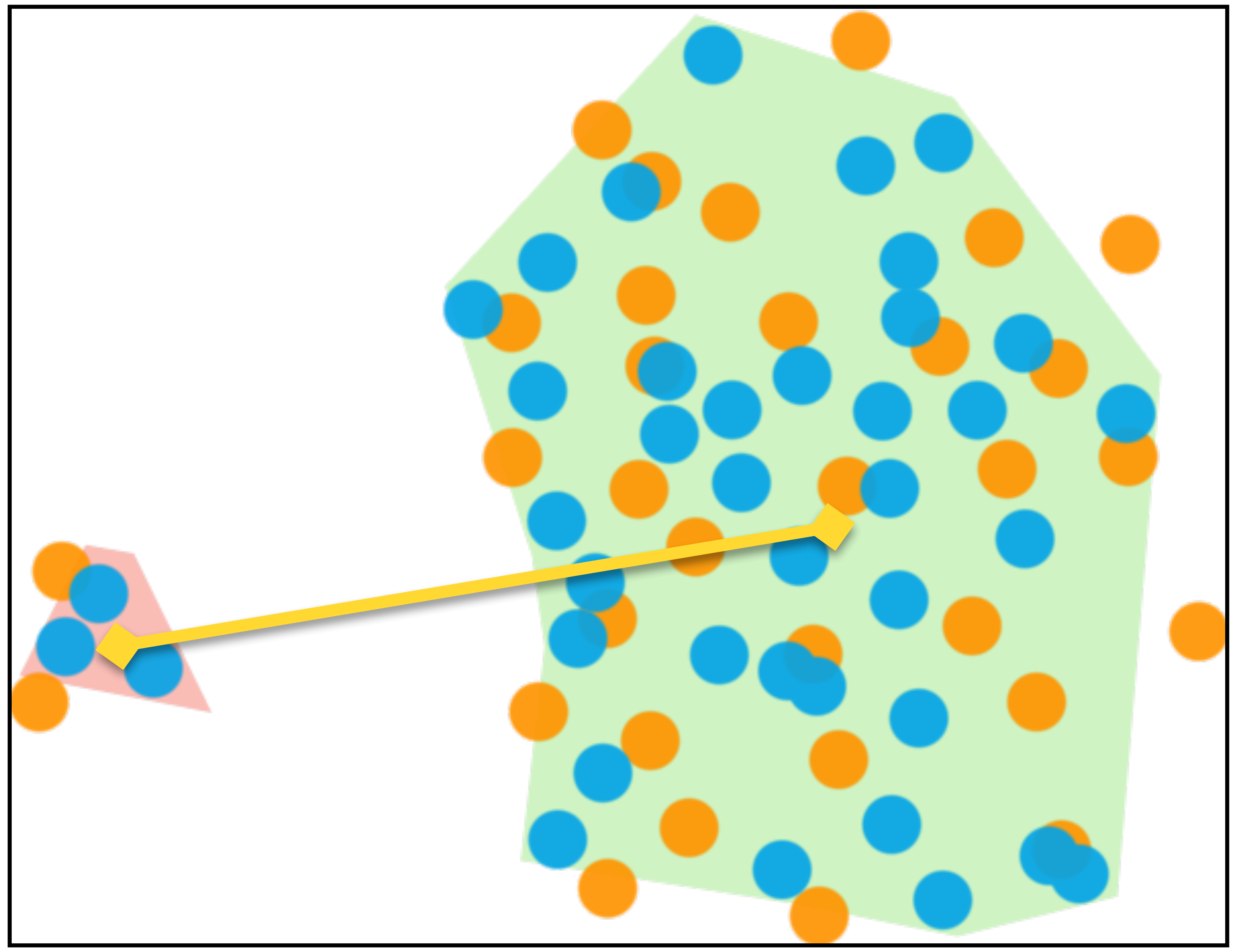
✅

```
> replicate(10, hunt.and.test(data))
```

```
p value

0.2
0.1    .
0.6
0.3
0.006  ***
0.4
0.7
0.8
0.3
0.06   .
```

🤔 *“Significant 1 out of 10 times.”*

🙁 *“No evidence for a new subtype.”*



👉 Hunted the wrong direction 9/10 times.

😔 Missed opportunity!

Dilemma of data splitting: Two drawbacks

Dilemma of data splitting: Two drawbacks

Bill

🤔 Raises concern on replicability

Laura

😔 Misses the true signal in data

Dilemma of data splitting: Two drawbacks

Bill

🤔 Raises concern on replicability

⚠️ **High variability conditional on data**

Laura

😔 Misses the true signal in data

Dilemma of data splitting: Two drawbacks

Bill

🤔 Raises concern on replicability

⚠️ **High variability conditional on data**

Laura

😔 Misses the true signal in data

⚠️ **Low power**

Dilemma of data splitting: Two drawbacks

Bill

🤔 Raises concern on replicability

⚠️ **High variability conditional on data**

Laura

😔 Misses the true signal in data

⚠️ **Low power**

💡 Use the information from multiple data splits properly!

Outline

- **Setup and main challenge**
- Method: Rank-transformed subsampling
- Applications
  - Hunt and test
  - Improving inference for double machine learning
  - Testing no direct effect in a sequentially randomized trial
- Future directions

Setup: Single split

Setup: Single split

IID Data: $X := (X_1, \ldots, X_n) \sim P^n$. Hypothesis testing: $P \in H_0$ vs $P \in H_1$.

Setup: Single split

IID Data: $X := (X_1, \ldots, X_n) \sim P^n$. Hypothesis testing: $P \in H_0$ vs $P \in H_1$.

“Single-split” statistic: $T_n(X_1, \ldots, X_n; \Omega)$, where Ω is 🎲 .

- Extra randomness $\Omega \sim P_\Omega$ independent of X.
- Ω is used to split data, perform resampling, etc.

Setup: Single split

IID Data: $X := (X_1, \ldots, X_n) \sim P^n$. Hypothesis testing: $P \in H_0$ vs $P \in H_1$.

“Single-split” statistic: $T_n(X_1, \ldots, X_n; \Omega)$, where Ω is 🎲 .

- Extra randomness $\Omega \sim P_\Omega$ independent of X.
- Ω is used to split data, perform resampling, etc.

Assumption. For $P \in H_0$, $T_n(X; \Omega) \to_d F_0$ as $n \to \infty$ unconditionally.

- “unconditionally” = over randomness of both X and Ω
- “conditionally” = over randomness of $\Omega \mid X$

Setup: Single split

IID Data: $X := (X_1, \ldots, X_n) \sim P^n$. Hypothesis testing: $P \in H_0$ vs $P \in H_1$.

"Single-split" statistic: $T_n(X_1, \ldots, X_n; \Omega)$, where Ω is 🎲 .

- Extra randomness $\Omega \sim P_\Omega$ independent of X.
- Ω is used to split data, perform resampling, etc.

Assumption. For $P \in H_0$, $T_n(X; \Omega) \to_d F_0$ as $n \to \infty$ unconditionally.

(1) $F_0 =$ unif$(0,1)$ for p-value
(2) $F_0 = \mathcal{N}(0,1)$ for Z-statistic

- "unconditionally" = over randomness of both X and Ω
- "conditionally" = over randomness of $\Omega \mid X$

Setup: Single split

IID Data: $X := (X_1, \ldots, X_n) \sim P^n$. Hypothesis testing: $P \in H_0$ vs $P \in H_1$.

"Single-split" statistic: $T_n(X_1, \ldots, X_n; \Omega)$, where Ω is 🎲 .

- Extra randomness $\Omega \sim P_\Omega$ independent of X.
- Ω is used to split data, perform resampling, etc.

Assumption. For $P \in H_0$, $T_n(X; \Omega) \to_d F_0$ as $n \to \infty$ unconditionally.

(1) $F_0 = \text{unif}(0,1)$ for p-value
(2) $F_0 = \mathcal{N}(0,1)$ for Z-statistic

- "unconditionally" = over randomness of both X and Ω
- "conditionally" = over randomness of $\Omega \mid X$

"Single-split" test: Reject H_0 whenever $T_n \gtrless (\alpha$ quantile of $F_0)$.

⚠️ High conditional variability. ⚠️ Low power.

Setup: Aggregation

“Multiple-split”, exchangeable statistics: Fix X. Draw $\Omega^{(1)}, \ldots, \Omega^{(L)}$ as L independent copies of Ω and let

$$T_n^{(1)} := T_n(X; \Omega^{(1)}), \quad \ldots \quad , T_n^{(L)} := T_n(X; \Omega^{(L)}) \,.$$

Setup: Aggregation

"Multiple-split", exchangeable statistics: Fix X. Draw $\Omega^{(1)}, \ldots, \Omega^{(L)}$ as L independent copies of Ω and let

$$T_n^{(1)} := T_n(X; \Omega^{(1)}), \quad \ldots \quad , T_n^{(L)} := T_n(X; \Omega^{(L)}) \, .$$

👉 By construction, $T_n^{(1)}, \ldots, T_n^{(L)}$ are unconditionally **exchangeable**.

Setup: Aggregation

“Multiple-split”, exchangeable statistics: Fix X. Draw $\Omega^{(1)}, \ldots, \Omega^{(L)}$ as L independent copies of Ω and let

$$T_n^{(1)} := T_n(X; \Omega^{(1)}), \quad \ldots \quad , T_n^{(L)} := T_n(X; \Omega^{(L)}) .$$

👉 By construction, $T_n^{(1)}, \ldots, T_n^{(L)}$ are unconditionally **exchangeable**.

💡 **Aggregated statistic:**

$$S_n := S(T_n^{(1)}, \ldots, T_n^{(L)}),$$

for a chosen aggregation function $S : \mathbb{R}^L \to \mathbb{R}$.

👉 S should be symmetric and Lipschitz in $\|\cdot\|_\infty$. 👉 Examples: $S = $ avg, $S = \min$.

Setup: Aggregation

“Multiple-split”, exchangeable statistics: Fix X. Draw $\Omega^{(1)}, \ldots, \Omega^{(L)}$ as L independent copies of Ω and let

$$T_n^{(1)} := T_n(X; \Omega^{(1)}), \quad \ldots \quad , T_n^{(L)} := T_n(X; \Omega^{(L)}) .$$

👉 By construction, $T_n^{(1)}, \ldots, T_n^{(L)}$ are unconditionally **exchangeable**.

💡 **Aggregated statistic:**

$$S_n := S(T_n^{(1)}, \ldots, T_n^{(L)}),$$

for a chosen aggregation function $S : \mathbb{R}^L \to \mathbb{R}$.

👉 S should be symmetric and Lipschitz in $\|\cdot\|_\infty$. 👉 Examples: $S = \text{avg}$, $S = \min$.

Aggregated test: Reject H_0 when $S_n = S(T_n^{(1)}, \ldots, T_n^{(L)}) \lessgtr$ **?**

Setup: Aggregation

“Multiple-split”, exchangeable statistics: Fix X. Draw $\Omega^{(1)}, \ldots, \Omega^{(L)}$ as L independent copies of Ω and let

$$T_n^{(1)} := T_n(X; \Omega^{(1)}), \quad \ldots \quad , T_n^{(L)} := T_n(X; \Omega^{(L)})\,.$$

👉 By construction, $T_n^{(1)}, \ldots, T_n^{(L)}$ are unconditionally **exchangeable**.

💡 **Aggregated statistic:**

$$S_n := S(T_n^{(1)}, \ldots, T_n^{(L)}),$$

for a chosen aggregation function $S : \mathbb{R}^L \to \mathbb{R}$.

👉 S should be symmetric and Lipschitz in $\|\cdot\|_\infty$. 👉 Examples: $S = \text{avg}$, $S = \min$.

Aggregated test: Reject H_0 when $S_n = S(T_n^{(1)}, \ldots, T_n^{(L)}) \lessgtr$ ❓

✅ Lower conditional variability and ✅ more power compared to the single-split test: $T_n^{(1)} \gtrless (\alpha \text{ quantile of } F_0)$.

Main challenge

🤔 **Aggregated test:** Reject H_0 when $S_n = S(T_n^{(1)}, \dots, T_n^{(L)}) \lesseqgtr$ **?**

Main challenge

🤔 **Aggregated test:** Reject H_0 when $S_n = S(T_n^{(1)}, \ldots, T_n^{(L)}) \lessgtr$ **?**

🤔 $(T_n^{(1)}, \ldots, T_n^{(L)})$
under H_0

Main challenge

🤔 **Aggregated test:** Reject H_0 when $S_n = S(T_n^{(1)}, \ldots, T_n^{(L)}) \lessgtr$ **?**

🤔 $(T_n^{(1)}, \ldots, T_n^{(L)})$ under H_0

(1) Marginal:

(2) Copula:

Main challenge

🤔 **Aggregated test:** Reject H_0 when $S_n = S(T_n^{(1)}, \ldots, T_n^{(L)}) \lessgtr$ ❓

🤔 $(T_n^{(1)}, \ldots, T_n^{(L)})$ under H_0

(1) Marginal: Every $T_n^{(l)} \to_d F_0$ under H_0 ✅

(2) Copula:

Main challenge

🤔 **Aggregated test:** Reject H_0 when $S_n = S(T_n^{(1)}, \ldots, T_n^{(L)}) \lessgtr$ ❓

🤔 $(T_n^{(1)}, \ldots, T_n^{(L)})$ under H_0

(1) Marginal: Every $T_n^{(l)} \to_d F_0$ under H_0 ✅

(2) Copula: Joint distribution of $F_{n,P}(T_n^{(1)}), \ldots, F_{n,P}(T_n^{(L)})$, where $F_{n,P}$ is the CDF of $T_n^{(1)}$

Main challenge

🤔 **Aggregated test:** Reject H_0 when $S_n = S(T_n^{(1)}, \ldots, T_n^{(L)}) \lesseqgtr$ ❓

🤔 $(T_n^{(1)}, \ldots, T_n^{(L)})$ under H_0

(1) Marginal: Every $T_n^{(l)} \to_d F_0$ under H_0 ✅

(2) Copula: Joint distribution of $F_{n,P}(T_n^{(1)}), \ldots, F_{n,P}(T_n^{(L)})$, where $F_{n,P}$ is the CDF of $T_n^{(1)}$

❓ Unknown, except for its symmetry.

Main challenge

🤔 **Aggregated test:** Reject H_0 when $S_n = S(T_n^{(1)}, \ldots, T_n^{(L)}) \lessgtr$ ❓

🤔 $(T_n^{(1)}, \ldots, T_n^{(L)})$ under H_0

(1) Marginal: Every $T_n^{(l)} \to_d F_0$ under H_0 ✅

Main challenge

(2) Copula: Joint distribution of $F_{n,P}(T_n^{(1)}), \ldots, F_{n,P}(T_n^{(L)})$, where $F_{n,P}$ is the CDF of $T_n^{(1)}$

❓ Unknown, except for its symmetry.

👉 Under H_0, S_n converges to some unknown distribution that depends on $P \in H_0$.

Typically, S_n will converge to some non-degenerate limit distribution under H_0.

Existing approaches: Two types

Existing approaches: Two types

(1) Assumes a **parametric copula** (e.g., Gaussian) and fits it.

Existing approaches: Two types

(1) Assumes a **parametric copula** (e.g., Gaussian) and fits it.

⚠️ Easily **misspecified** in real applications. **Cannot control type-I error.**

Kim & Ramdas (2020)

$X_1, \ldots, X_n \sim \mathcal{N}(\mu, \Sigma)$ in $\mathbb{R}^3$.

Single-split statistic for testing $H_0 : \mu = \mathbf{0}$

$$T_n := \frac{\sqrt{n_2}\hat{\mu}_1^\top \hat{\mu}_2}{\hat{\mu}_1^\top \hat{\Sigma}_2 \hat{\mu}_1} \to_d \mathcal{N}(0,1) \text{ under } H_0.$$

Existing approaches: Two types

(1) Assumes a **parametric copula** (e.g., Gaussian) and fits it.

⚠️ Easily **misspecified** in real applications. **Cannot control type-I error.**

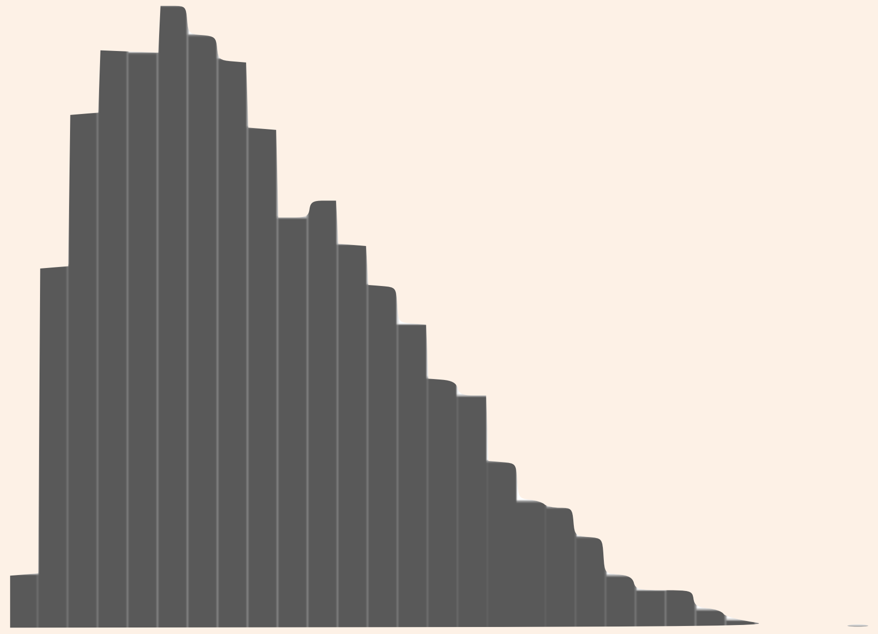
Kim & Ramdas (2020)

$X_1, \dots, X_n \sim \mathcal{N}(\mu, \Sigma)$ in $\mathbb{R}^3$.

Single-split statistic for testing $H_0 : \mu = \mathbf{0}$

$$T_n := \frac{\sqrt{n_2}\hat{\mu}_1^\top \hat{\mu}_2}{\hat{\mu}_1^\top \hat{\Sigma}_2 \hat{\mu}_1} \to_d \mathcal{N}(0,1) \text{ under } H_0.$$

⚠️ Copula can be complex. No generic approximation.



Null distribution of $(T_n^{(1)} + \ldots + T_n^{(200)})/200$

Existing approaches: Two types

(1) Assumes a **parametric copula** (e.g., Gaussian) and fits it.

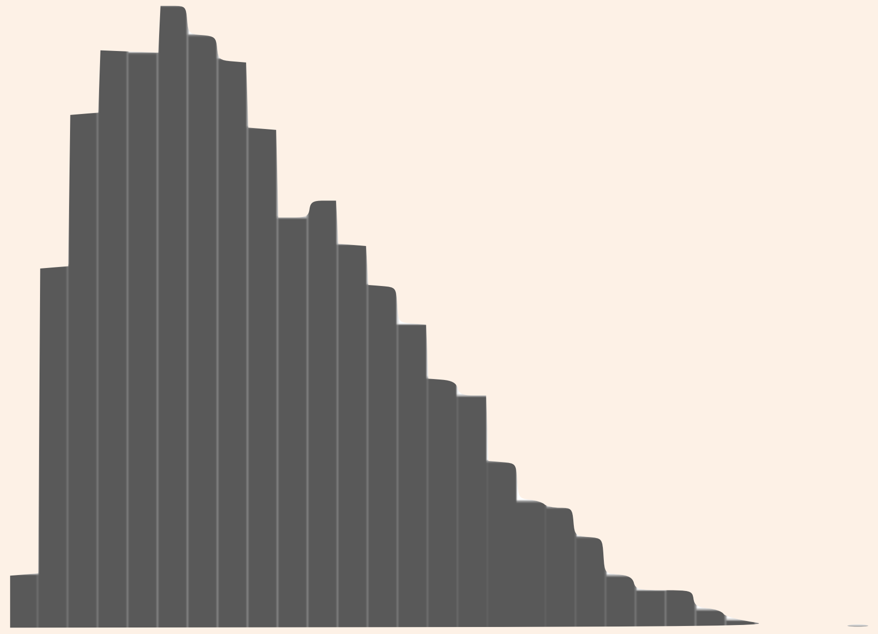
⚠️ Easily **misspecified** in real applications. **Cannot control type-I error.**

Kim & Ramdas (2020)

$X_1, \ldots, X_n \sim \mathcal{N}(\mu, \Sigma)$ in $\mathbb{R}^3$.

Single-split statistic for testing $H_0 : \mu = \mathbf{0}$

$$T_n := \frac{\sqrt{n_2}\hat{\mu}_1^\top \hat{\mu}_2}{\hat{\mu}_1^\top \hat{\Sigma}_2 \hat{\mu}_1} \to_d \mathcal{N}(0,1) \text{ under } H_0.$$



Null distribution of $(T_n^{(1)} + \ldots + T_n^{(200)})/200$

⚠️ Copula can be complex. No generic approximation.

(2) Guards against the **worst-case** copula.

Existing approaches: Two types

(1) Assumes a **parametric copula** (e.g., Gaussian) and fits it.

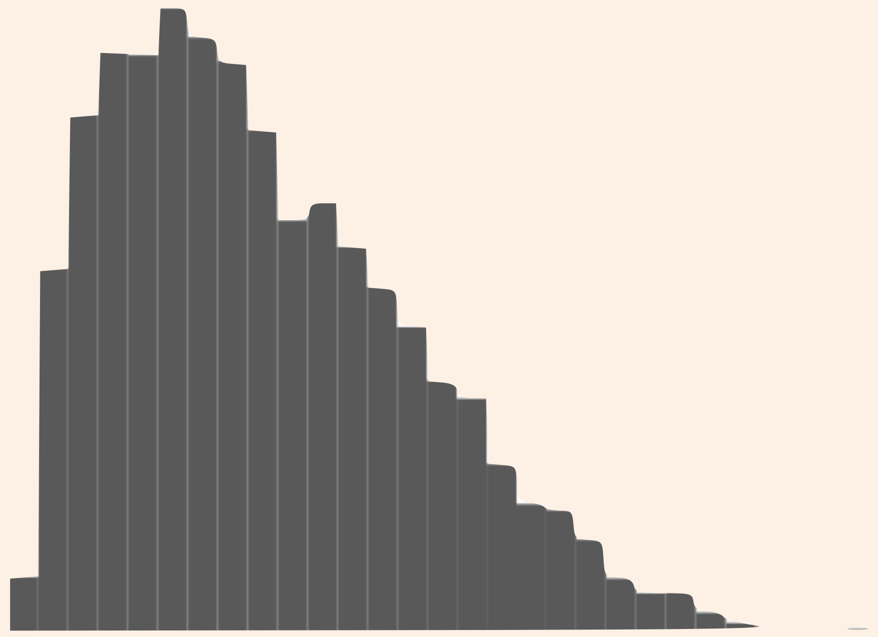
⚠️ Easily **misspecified** in real applications. **Cannot control type-I error.**

Kim & Ramdas (2020)

$X_1, \ldots, X_n \sim \mathcal{N}(\mu, \Sigma)$ in $\mathbb{R}^3$.

Single-split statistic for testing $H_0 : \mu = \mathbf{0}$

$$T_n := \frac{\sqrt{n_2}\hat{\mu}_1^\top \hat{\mu}_2}{\hat{\mu}_1^\top \hat{\Sigma}_2 \hat{\mu}_1} \to_d \mathcal{N}(0,1) \text{ under } H_0.$$



Null distribution of $(T_n^{(1)} + \ldots + T_n^{(200)})/200$

⚠️ Copula can be complex. No generic approximation.

(2) Guards against the **worst-case** copula.

👉 A large body of literature on combining p-values under **arbitrary dependence**.

- Averaging p-values multiplied by two (Rüschendorf, 1982; Meng, 1994)
- Generalized means (Vovk & Wang, 2020)
- Quantiles (Meinshausen et al., 2009; DiCiccio et al., 2020)
- Concentration inequalities (DiCiccio et al., 2020)
- Cauchy transformations (Liu & Xie, 2020)
- e-values (Vovk & Wang, 2021)

…

Existing approaches: Two types

(1) Assumes a **parametric copula** (e.g., Gaussian) and fits it.

⚠️ Easily **misspecified** in real applications. **Cannot control type-I error.**

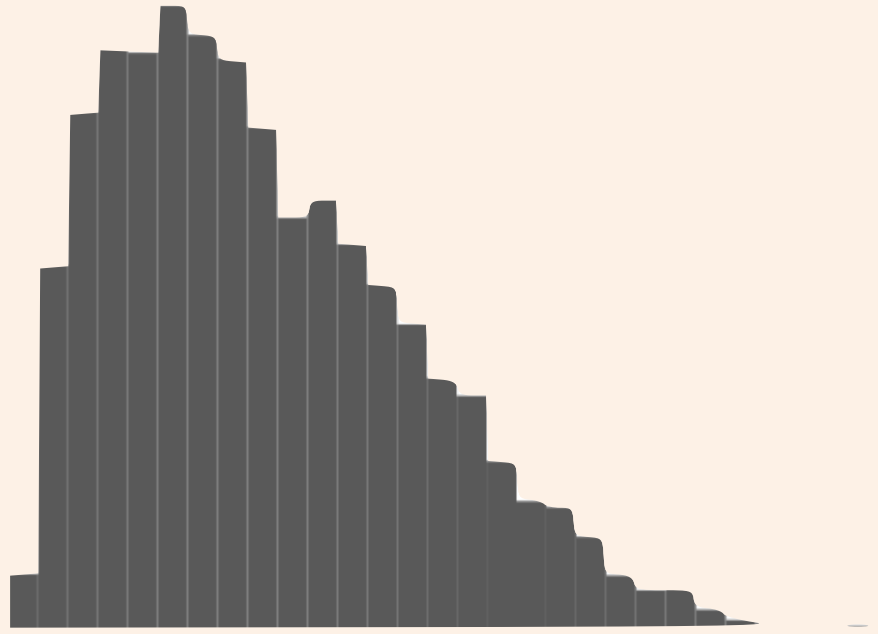
Kim & Ramdas (2020)

$X_1, \ldots, X_n \sim \mathcal{N}(\mu, \Sigma)$ in $\mathbb{R}^3$.

Single-split statistic for testing $H_0 : \mu = \mathbf{0}$

$$T_n := \frac{\sqrt{n_2}\hat{\mu}_1^\top \hat{\mu}_2}{\hat{\mu}_1^\top \hat{\Sigma}_2 \hat{\mu}_1} \to_d \mathcal{N}(0,1) \text{ under } H_0.$$

⚠️ Copula can be complex. No generic approximation.



Null distribution of $(T_n^{(1)} + \ldots + T_n^{(200)})/200$

(2) Guards against the **worst-case** copula.

👉 A large body of literature on combining p-values under **arbitrary dependence**.

- Averaging p-values multiplied by two (Rüschendorf, 1982; Meng, 1994)
- Generalized means (Vovk & Wang, 2020)
- Quantiles (Meinshausen et al., 2009; DiCiccio et al., 2020)
- Concentration inequalities (DiCiccio et al., 2020)
- Cauchy transformations (Liu & Xie, 2020)
- e-values (Vovk & Wang, 2021)

…

⚠️ **Very conservative**

actual type-I error $\ll \alpha$, typically

Existing approaches: Two types

(1) Assumes a **parametric copula** (e.g., Gaussian) and fits it.

⚠ Easily **misspecified** in real applications. **Cannot control type-I error.**

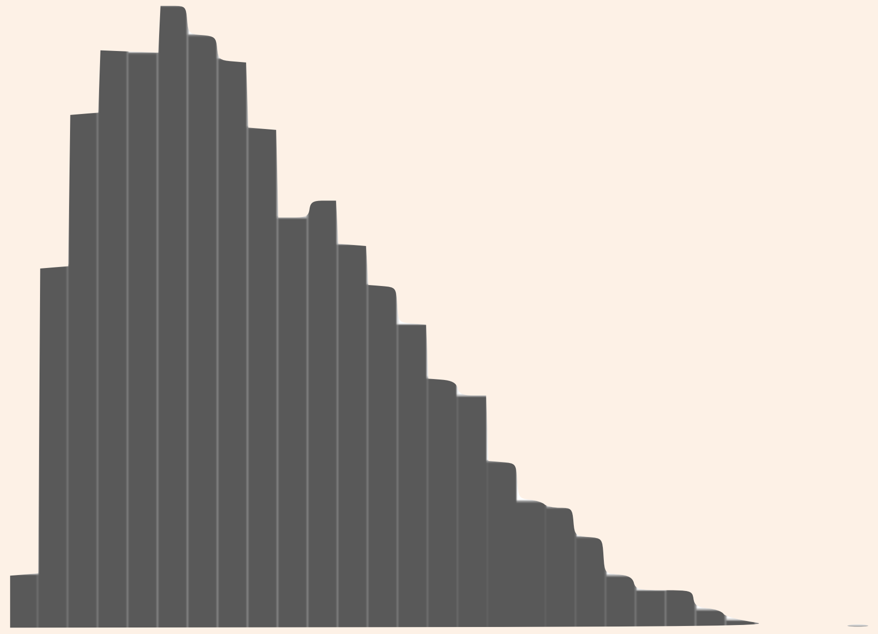
Kim & Ramdas (2020)

$X_1, \ldots, X_n \sim \mathcal{N}(\mu, \Sigma)$ in $\mathbb{R}^3$.

Single-split statistic for testing $H_0 : \mu = \mathbf{0}$

$$T_n := \frac{\sqrt{n_2}\hat{\mu}_1^\top \hat{\mu}_2}{\hat{\mu}_1^\top \hat{\Sigma}_2 \hat{\mu}_1} \to_d \mathcal{N}(0,1) \text{ under } H_0.$$

⚠ Copula can be complex. No generic approximation.



Null distribution of $(T_n^{(1)} + \ldots + T_n^{(200)})/200$

(2) Guards against the **worst-case** copula.

👉 A large body of literature on combining p-values under **arbitrary dependence**.

- Averaging p-values multiplied by two (Rüschendorf, 1982; Meng, 1994)
- Generalized means (Vovk & Wang, 2020)
- Quantiles (Meinshausen et al., 2009; DiCiccio et al., 2020)
- Concentration inequalities (DiCiccio et al., 2020)
- Cauchy transformations (Liu & Xie, 2020)
- e-values (Vovk & Wang, 2021)

...

⚠ **Very conservative**

actual type-I error $\ll \alpha$, typically

👉 Symmetry does **not** help. (Choi & Kim, 2022)

Outline

- Setup and main challenge
- **Method: Rank-transformed subsampling**
- Applications
    - Hunt and test
    - Improving inference for double machine learning
    - Testing no direct effect in a sequentially randomized trial
- Future directions

Approach

$(T_n^{(1)}, \ldots, T_n^{(L)})$
under H_0

(1) Marginal: F_0 ✓

Main Challenge

(2) Copula: $C_{n,P}(u_1, \ldots, u_L) = \mathbb{P}\left(F_{n,P}(T_n^{(1)}) \leq u_1, \ldots, F_{n,P}(T_n^{(L)}) \leq u_L\right)$?

Approach

$(T_n^{(1)}, \ldots, T_n^{(L)})$ under H_0

- (1) Marginal: F_0 ✅

Main Challenge

- (2) Copula: $C_{n,P}(u_1, \ldots, u_L) = \mathbb{P}\left(F_{n,P}(T_n^{(1)}) \leq u_1, \ldots, F_{n,P}(T_n^{(L)}) \leq u_L\right)$ **?**

🤔 **Aggregated test:** Reject H_0 when $S_n = S(T_n^{(1)}, \ldots, T_n^{(L)}) \lesseqgtr$ **?**

Approach

$(T_n^{(1)}, \ldots, T_n^{(L)})$ under H_0

(1) Marginal: F_0 ✅

Main Challenge

(2) Copula: $C_{n,P}(u_1, \ldots, u_L) = \mathbb{P}\left(F_{n,P}(T_n^{(1)}) \leq u_1, \ldots, F_{n,P}(T_n^{(L)}) \leq u_L\right)$ **?**

👆 Estimate it **nonparametrically** with **subsampling**!

🤔 **Aggregated test:** Reject H_0 when $S_n = S(T_n^{(1)}, \ldots, T_n^{(L)}) \lessgtr$ **?**

Approach

$(T_n^{(1)}, \dots, T_n^{(L)})$ under H_0

(1) Marginal: F_0 ✅

Main Challenge

(2) Copula: $C_{n,P}(u_1, \dots, u_L) = \mathbb{P}\left(F_{n,P}(T_n^{(1)}) \leq u_1, \dots, F_{n,P}(T_n^{(L)}) \leq u_L\right)$ **?**

👆Estimate it **nonparametrically** with **subsampling**!

🤔 **Aggregated test:** Reject H_0 when $S_n = S(T_n^{(1)}, \dots, T_n^{(L)}) \lesseqgtr$ **?**

(1) Marginal F_0

(2) Estimated Copula

Approach

$(T_n^{(1)}, \ldots, T_n^{(L)})$
under H_0

(1) Marginal: F_0 ✓

Main Challenge

(2) Copula: $C_{n,P}(u_1, \ldots, u_L) = \mathbb{P}\left(F_{n,P}(T_n^{(1)}) \le u_1, \ldots, F_{n,P}(T_n^{(L)}) \le u_L\right)$ **?**

👆 Estimate it **nonparametrically** with **subsampling**!

🤔 **Aggregated test:** Reject H_0 when $S_n = S(T_n^{(1)}, \ldots, T_n^{(L)}) \lesseqgtr$ **?**

(1) Marginal F_0

(2) Estimated Copula

$(\widetilde{T}_n^{(1)}, \ldots, \widetilde{T}_n^{(L)})$

Approach

$(T_n^{(1)}, \dots, T_n^{(L)})$ under H_0 $\begin{cases} \text{(1) Marginal: } F_0 \text{ ✅} \\ \text{(2) Copula: } C_{n,P}(u_1,\dots,u_L) = \mathbb{P}\left(F_{n,P}(T_n^{(1)}) \le u_1, \dots, F_{n,P}(T_n^{(L)}) \le u_L\right) \text{ ?} \end{cases}$

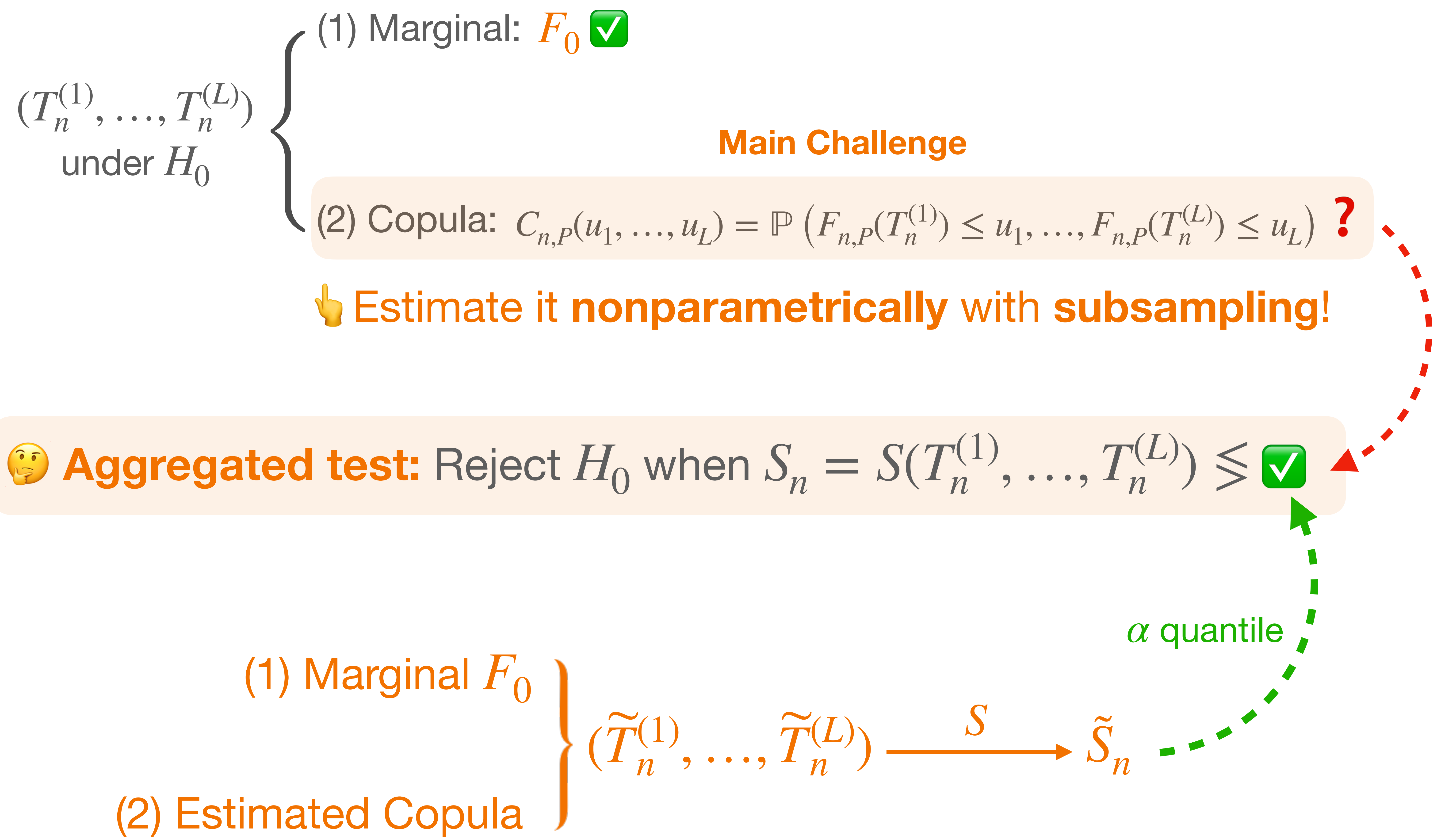
Main Challenge

👆Estimate it **nonparametrically** with **subsampling**!

🤔 **Aggregated test:** Reject H_0 when $S_n = S(T_n^{(1)}, \dots, T_n^{(L)}) \lessgtr$ **?**

$$\left.\begin{array}{r} \text{(1) Marginal } F_0 \\ \\ \text{(2) Estimated Copula} \end{array}\right\} (\widetilde{T}_n^{(1)}, \dots, \widetilde{T}_n^{(L)}) \xrightarrow{S} \tilde{S}_n$$

Approach



$(T_n^{(1)}, \dots, T_n^{(L)})$ under H_0

(1) Marginal: F_0 ✅

Main Challenge

(2) Copula: $C_{n,P}(u_1, \dots, u_L) = \mathbb{P}\left(F_{n,P}(T_n^{(1)}) \le u_1, \dots, F_{n,P}(T_n^{(L)}) \le u_L\right)$?

👆Estimate it **nonparametrically** with **subsampling**!

🤔 **Aggregated test:** Reject H_0 when $S_n = S(T_n^{(1)}, \dots, T_n^{(L)}) \lesseqgtr$ ✅

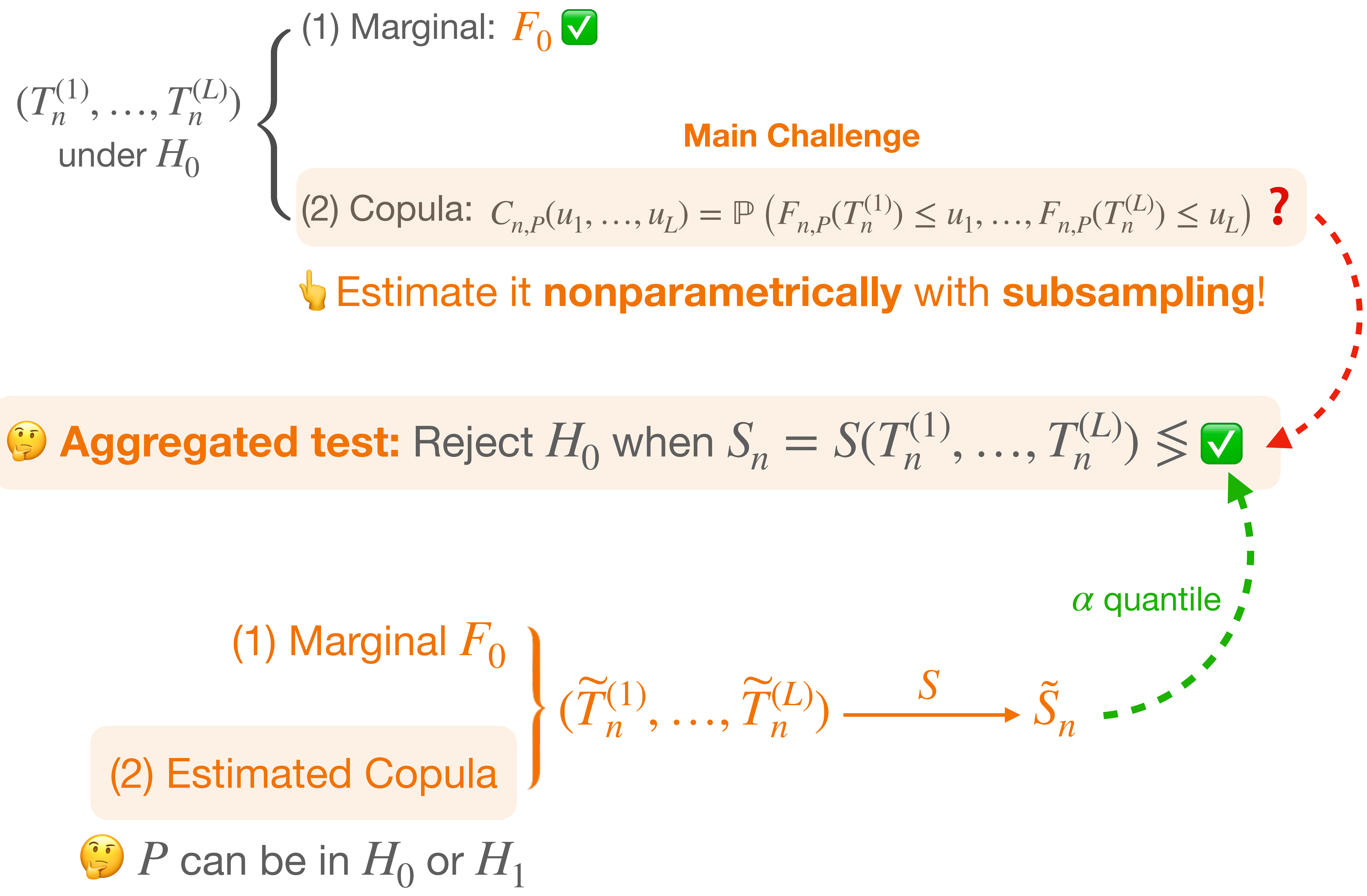
α quantile

(1) Marginal F_0

(2) Estimated Copula

$$(\widetilde{T}_n^{(1)}, \dots, \widetilde{T}_n^{(L)}) \xrightarrow{S} \tilde{S}_n$$

Approach



$(T_n^{(1)}, \dots, T_n^{(L)})$ under H_0

(1) Marginal: F_0 ✅

Main Challenge

(2) Copula: $C_{n,P}(u_1, \dots, u_L) = \mathbb{P}\left(F_{n,P}(T_n^{(1)}) \le u_1, \dots, F_{n,P}(T_n^{(L)}) \le u_L\right)$?

👆 Estimate it **nonparametrically** with **subsampling**!

🤔 **Aggregated test:** Reject H_0 when $S_n = S(T_n^{(1)}, \dots, T_n^{(L)}) \lessgtr$ ✅

α quantile

(1) Marginal F_0

(2) Estimated Copula

$(\widetilde{T}_n^{(1)}, \dots, \widetilde{T}_n^{(L)}) \xrightarrow{S} \tilde{S}_n$

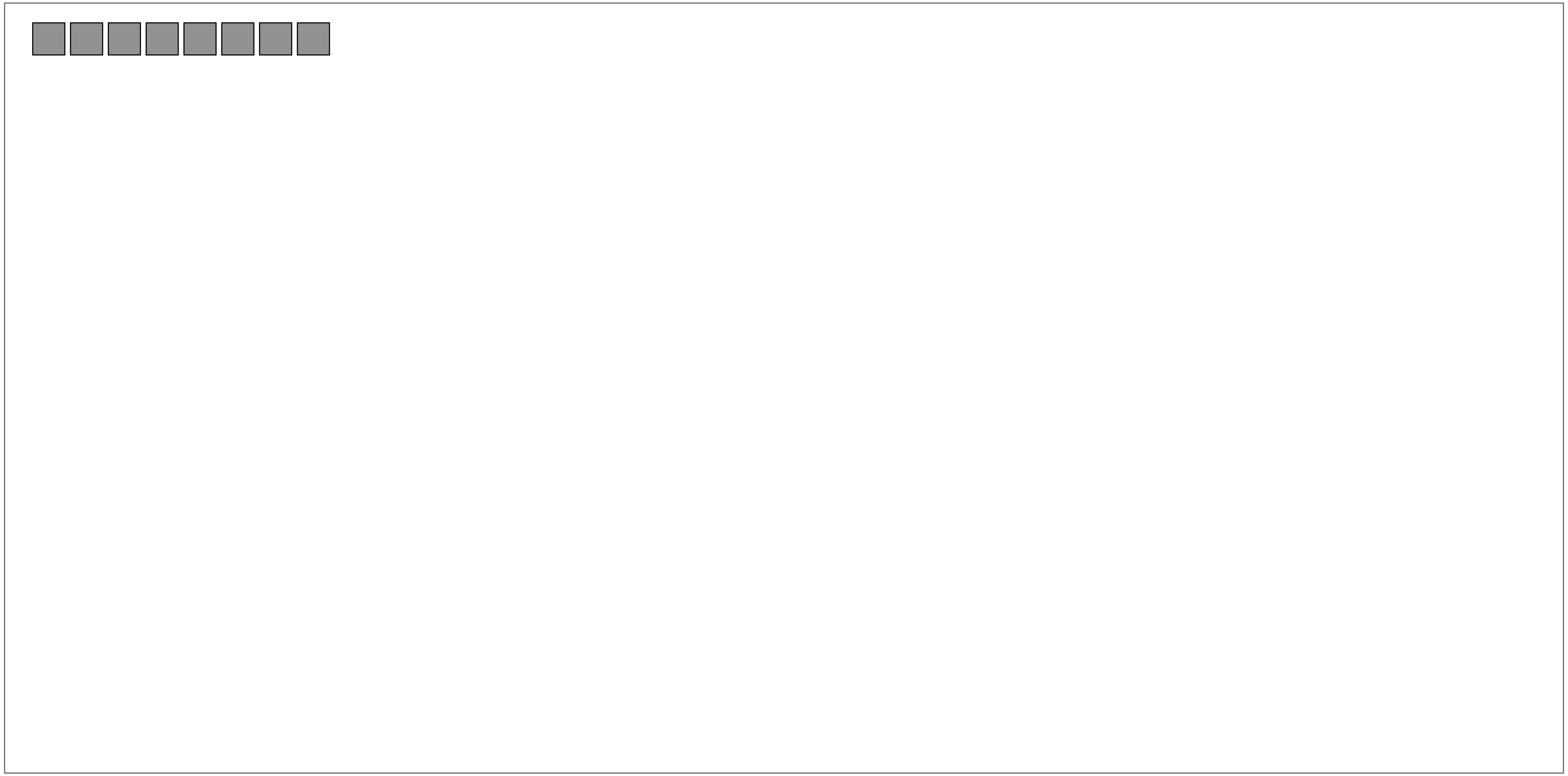
🤔 P can be in H_0 or H_1

Rank-transformed subsampling

Rank-transformed subsampling

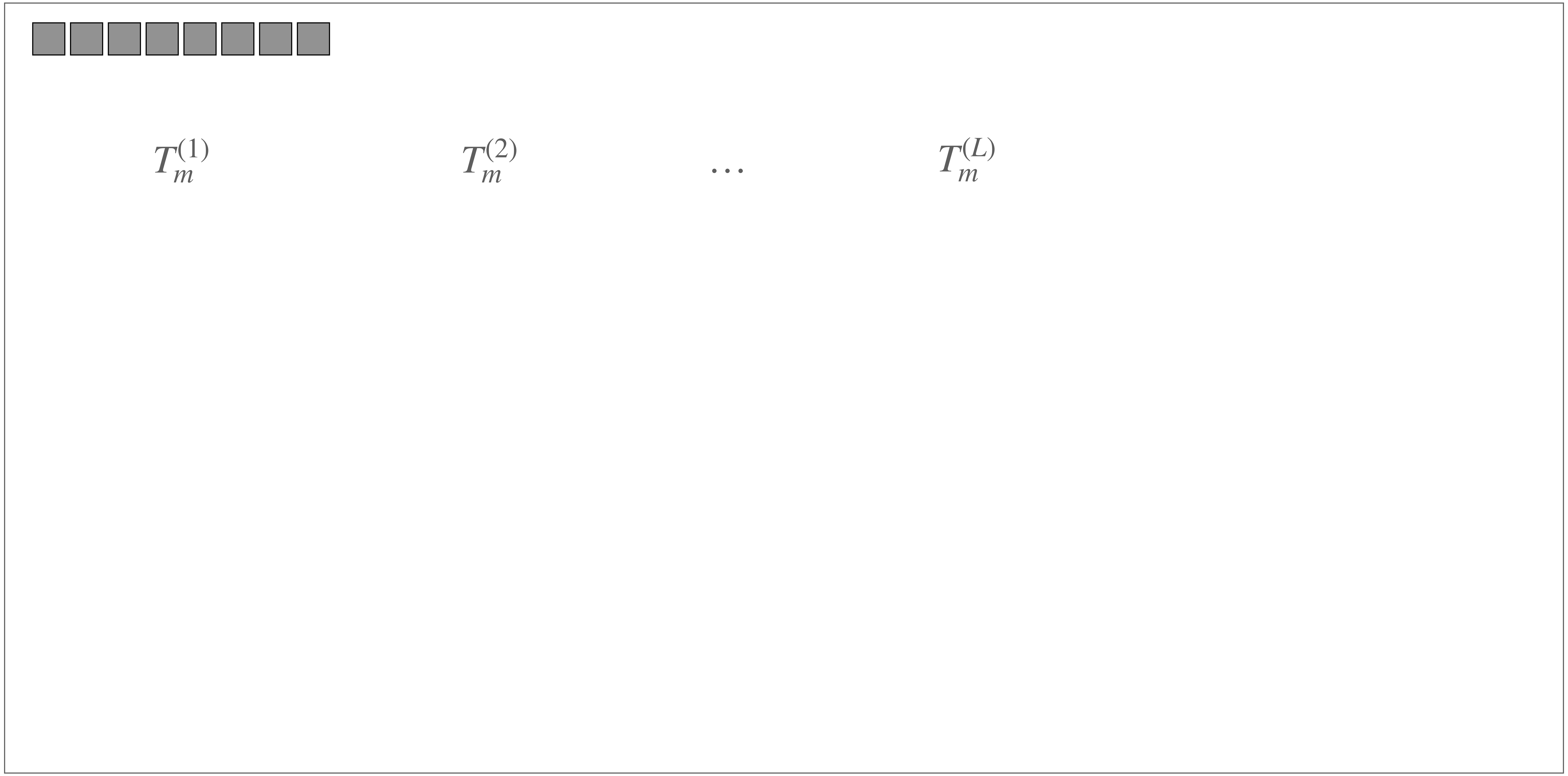
1. Randomly pick B subsamples of size $m = [n/\log n]$

Rank-transformed subsampling



1. Randomly pick B subsamples of size $m = [n/\log n]$

Rank-transformed subsampling

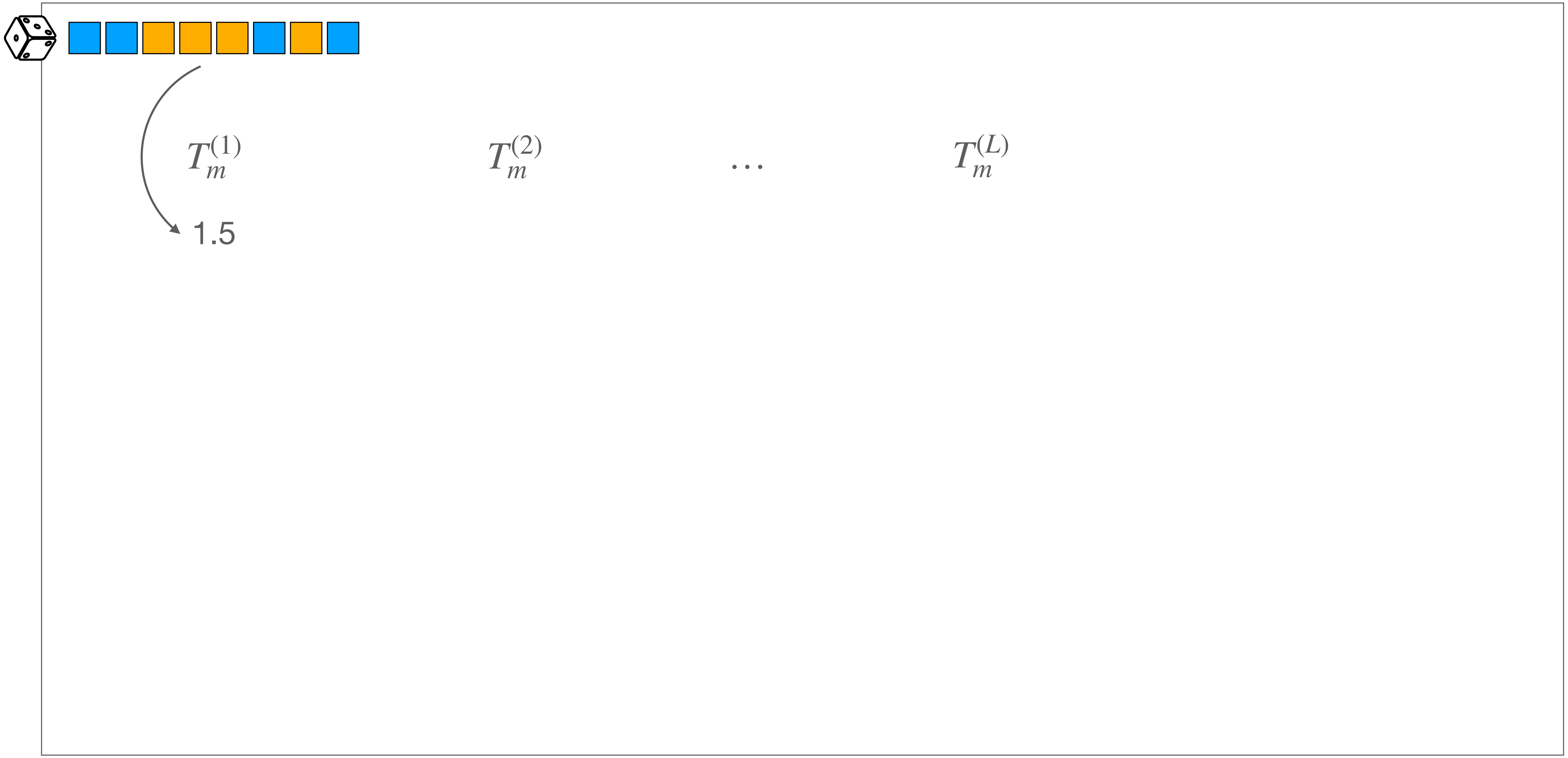


$T_m^{(1)}$ $T_m^{(2)}$ $\dots$ $T_m^{(L)}$

1. Randomly pick B subsamples of size $m = [n/\log n]$

2. Compute $(T_m^{(1)}, \dots, T_m^{(L)})$ for each subsample

Rank-transformed subsampling

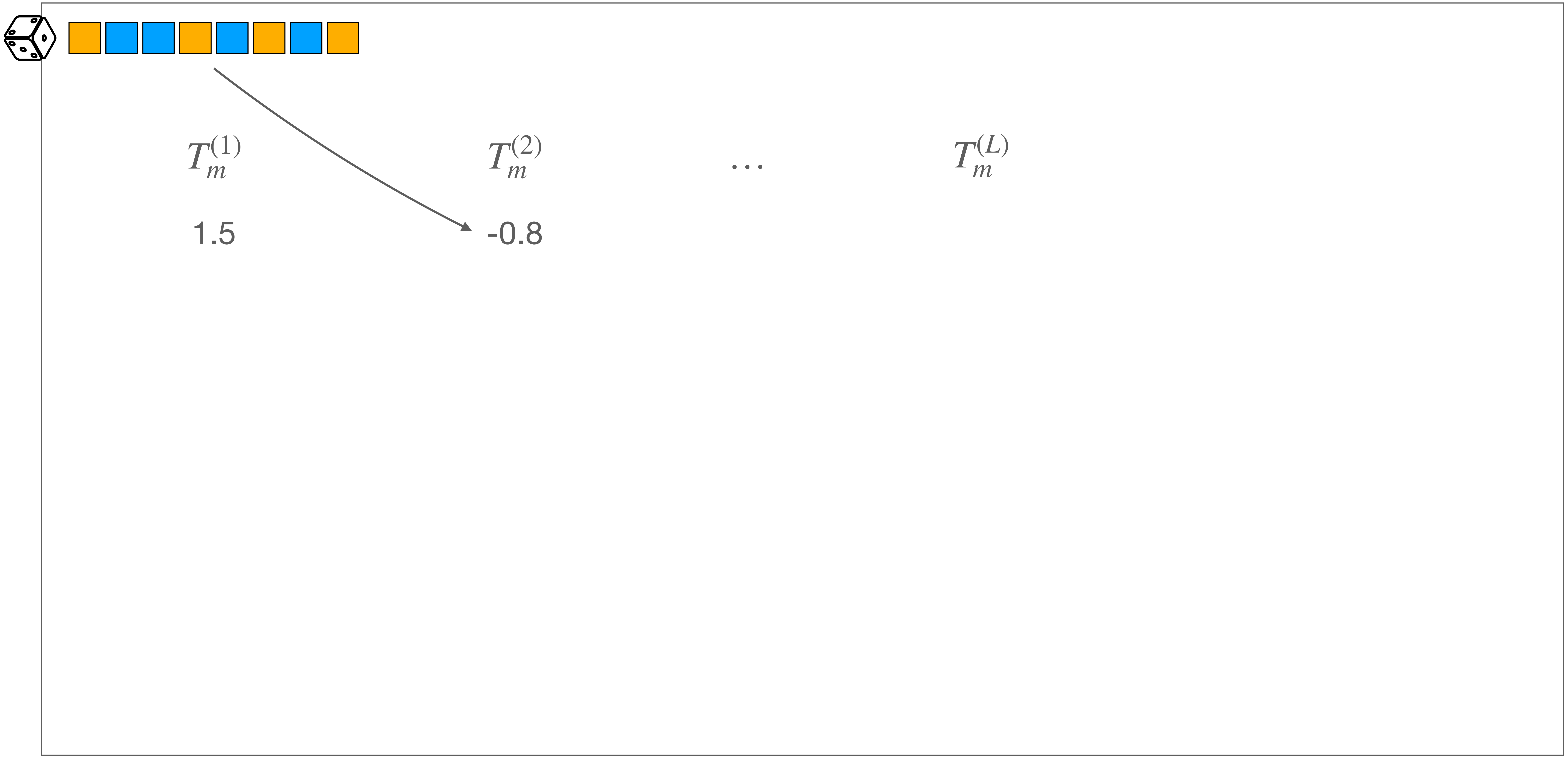


$T_m^{(1)}$ $T_m^{(2)}$ $\dots$ $T_m^{(L)}$

1.5

1. Randomly pick B subsamples of size $m = [n/\log n]$
2. Compute $(T_m^{(1)}, \dots, T_m^{(L)})$ for each subsample

Rank-transformed subsampling

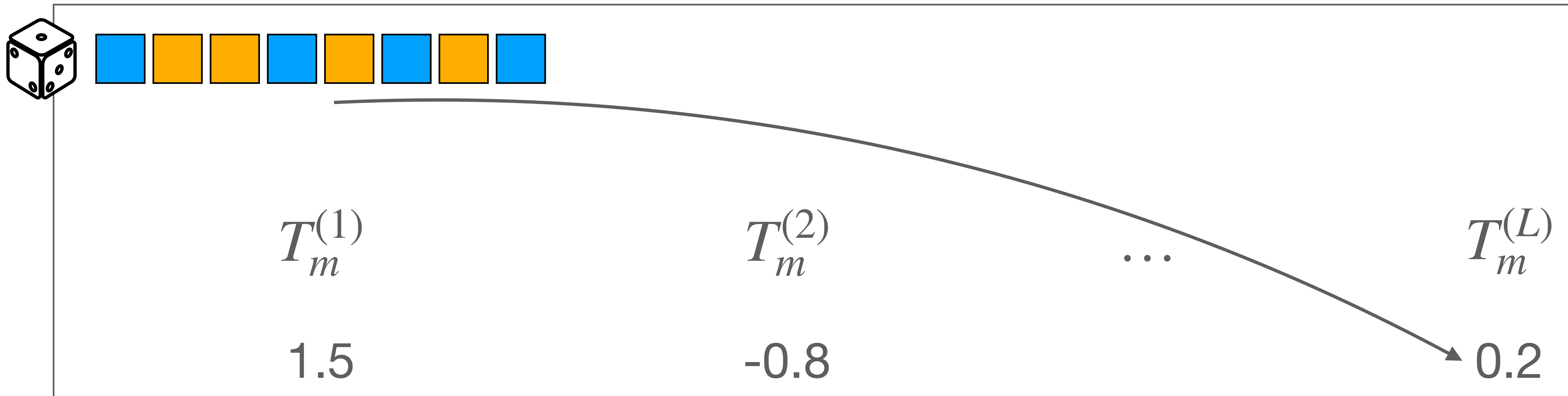


$T_m^{(1)}$ $T_m^{(2)}$ $\dots$ $T_m^{(L)}$

1.5 -0.8

1. Randomly pick B subsamples of size $m = [n/\log n]$
2. Compute $(T_m^{(1)}, \dots, T_m^{(L)})$ for each subsample

Rank-transformed subsampling



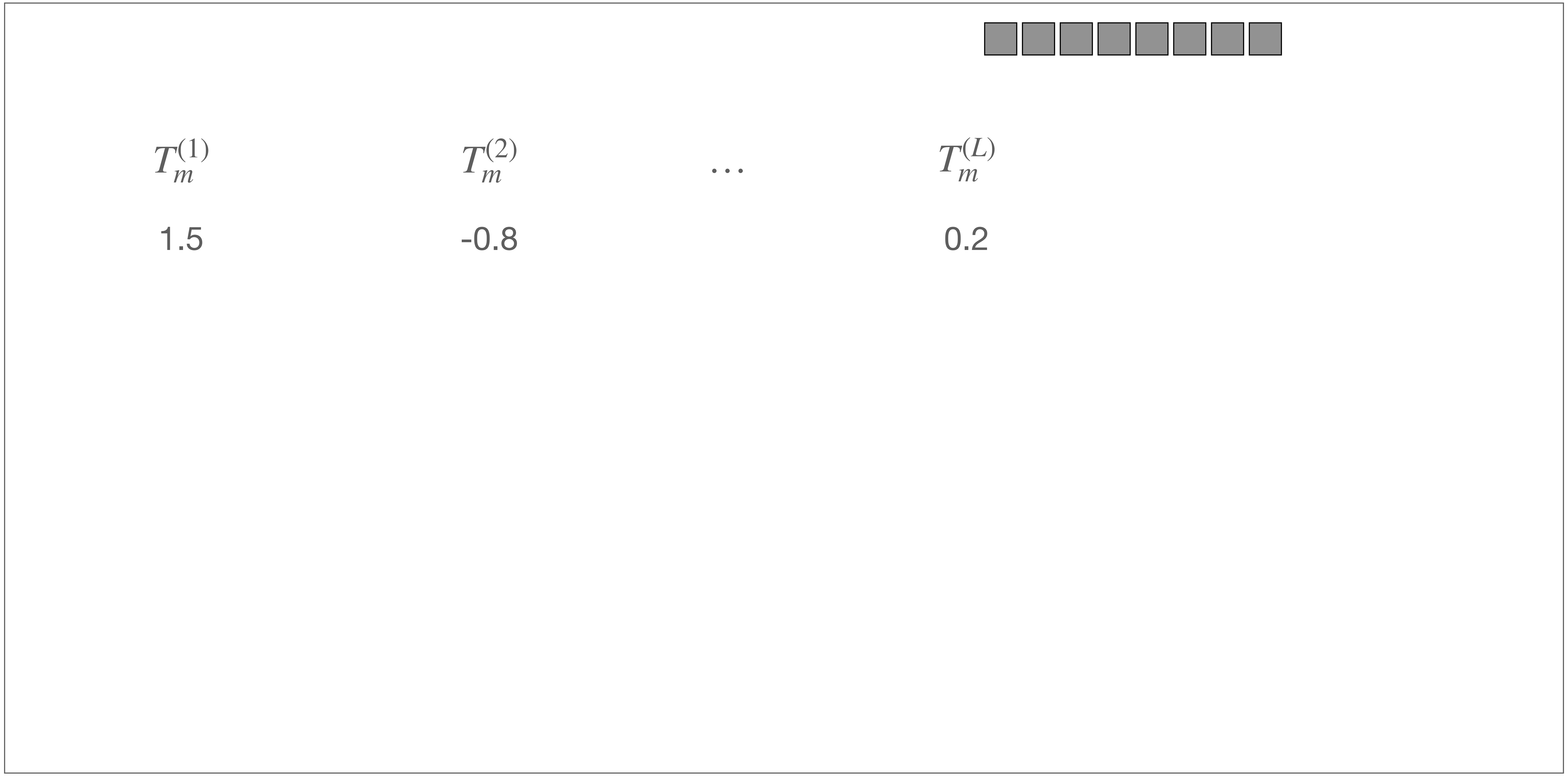
$T_m^{(1)}$ $T_m^{(2)}$ $\dots$ $T_m^{(L)}$

1.5 -0.8 0.2

1. Randomly pick B subsamples of size $m = [n/\log n]$

2. Compute $(T_m^{(1)}, \dots, T_m^{(L)})$ for each subsample

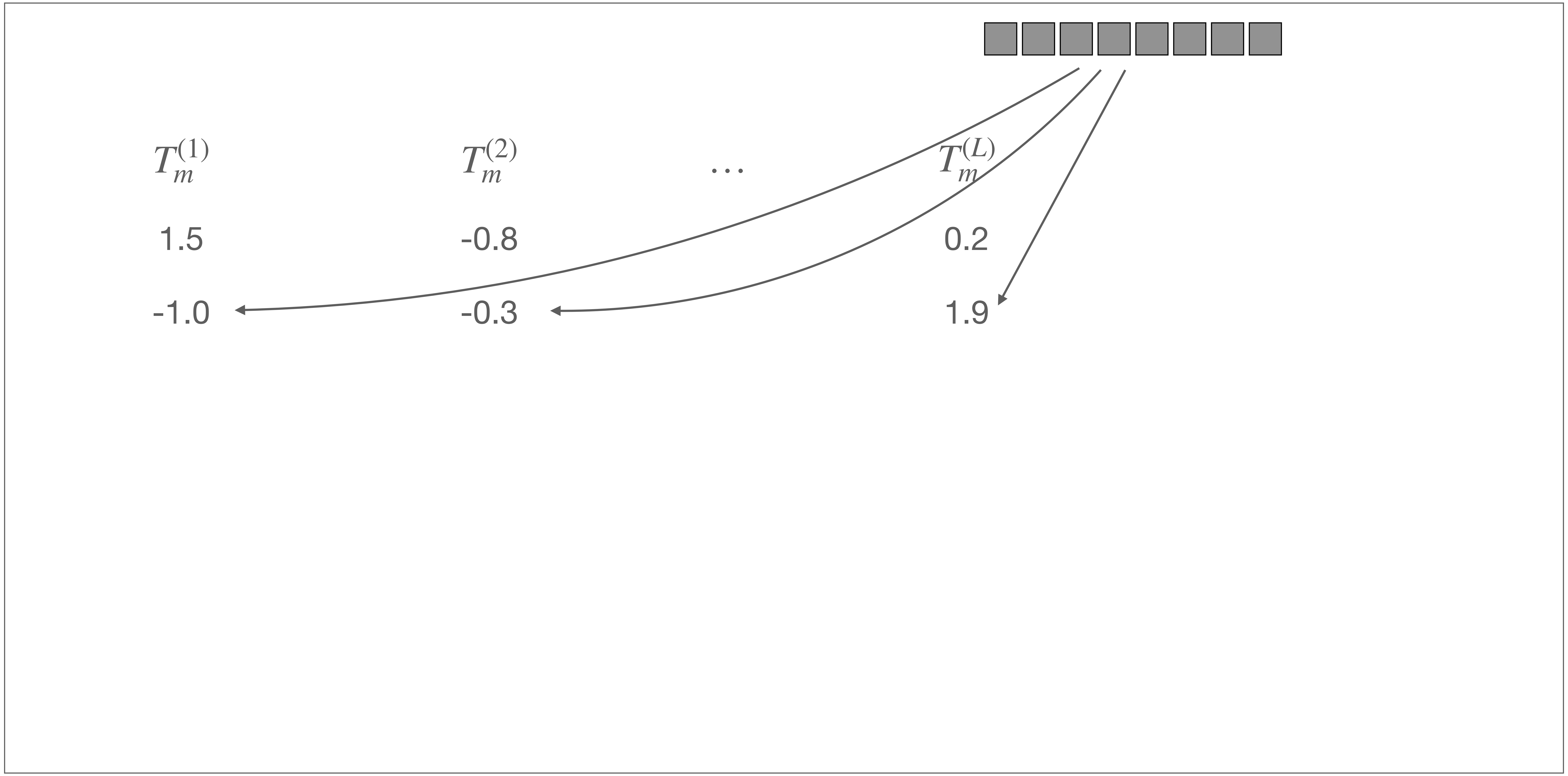
Rank-transformed subsampling



$T_m^{(1)}$	$T_m^{(2)}$	$\dots$	$T_m^{(L)}$
1.5	-0.8		0.2

1. Randomly pick B subsamples of size $m = [n/\log n]$
2. Compute $(T_m^{(1)}, \dots, T_m^{(L)})$ for each subsample

Rank-transformed subsampling



1. Randomly pick B subsamples of size $m = [n/\log n]$

2. Compute $(T_m^{(1)}, \dots, T_m^{(L)})$ for each subsample

Rank-transformed subsampling

$T_m^{(1)}$	$T_m^{(2)}$	$\dots$	$T_m^{(L)}$
1.5	-0.8		0.2
-1.0	-0.3		1.9

1. Randomly pick B subsamples of size $m = [n/\log n]$
2. Compute $(T_m^{(1)}, \dots, T_m^{(L)})$ for each subsample

Rank-transformed subsampling

$T_m^{(1)}$	$T_m^{(2)}$	$\dots$	$T_m^{(L)}$
1.5	-0.8		0.2
-1.0	-0.3		1.9
$\vdots$	$\vdots$		$\vdots$
2.7	0.1		3.0

B rows

L columns

1. Randomly pick B subsamples of size $m = [n/\log n]$
2. Compute $(T_m^{(1)}, \dots, T_m^{(L)})$ for each subsample

Rank-transformed subsampling

$$\begin{array}{c|cccc} & T_m^{(1)} & T_m^{(2)} & \dots & T_m^{(L)} \\ \hline & 1.5 & -0.8 & & 0.2 \\ & -1.0 & -0.3 & & 1.9 \\ \begin{array}{c} B \\ \text{rows} \end{array} & & & & \\ & \vdots & \vdots & & \vdots \\ & 2.7 & 0.1 & & 3.0 \end{array}$$

L columns

1. Randomly pick B subsamples of size $m = [n/\log n]$
2. Compute $(T_m^{(1)}, \dots, T_m^{(L)})$ for each subsample
3. In this $B \times L$ matrix, replace each entry by its rank

Rank-transformed subsampling

$$\begin{array}{c|cccc} & T_m^{(1)} & T_m^{(2)} & \cdots & T_m^{(L)} \\ \hline & 933 & 212 & & 580 \\ & 158 & 380 & & 971 \\ B \text{ rows} & \vdots & \vdots & & \vdots \\ & 990 & 539 & & 998 \end{array}$$

L columns

1. Randomly pick B subsamples of size $m = [n/\log n]$
2. Compute $(T_m^{(1)}, \ldots, T_m^{(L)})$ for each subsample
3. In this $B \times L$ matrix, replace each entry by its rank

Rank-transformed subsampling

$$
\begin{array}{c|cccc}
 & T_m^{(1)} & T_m^{(2)} & \cdots & T_m^{(L)} \\
\hline
 & 0.993 & 0.212 & & 0.580 \\
 & 0.158 & 0.380 & & 0.971 \\
B \text{ rows} & & & & \\
 & \vdots & \vdots & & \vdots \\
 & 0.990 & 0.539 & & 0.998
\end{array}
$$

L columns

1. Randomly pick B subsamples of size $m = [n/\log n]$
2. Compute $(T_m^{(1)}, \ldots, T_m^{(L)})$ for each subsample
3. In this $B \times L$ matrix, replace each entry by its rank
4. Normalize the ranks

(Copula estimate)

Rank-transformed subsampling

$$\begin{array}{c|cccc} & \widetilde{T}_m^{(1)} & \widetilde{T}_m^{(2)} & \cdots & \widetilde{T}_m^{(L)} \\ \hline & 1.6 & -0.8 & & 0.1 \\ & -1.1 & -0.2 & & 1.8 \\ B \text{ rows} & \vdots & \vdots & & \vdots \\ & 2.7 & 0.2 & & 2.8 \end{array}$$

L columns

1. Randomly pick B subsamples of size $m = [n/\log n]$
2. Compute $(T_m^{(1)}, \ldots, T_m^{(L)})$ for each subsample
3. In this $B \times L$ matrix, replace each entry by its rank
4. Normalize the ranks

   (Copula estimate)
5. Apply F_0^{-1} entry-wise

   (Enforce the margin)

Rank-transformed subsampling

$$\begin{array}{c|cccc|cc}
 & \widetilde{T}_m^{(1)} & \widetilde{T}_m^{(2)} & \dots & \widetilde{T}_m^{(L)} & \xrightarrow{S} & \tilde{S}_n \\
 & 1.6 & -0.8 & & 0.1 & & 0.5 \\
 & -1.1 & -0.2 & & 1.8 & & -0.1 \\
B \text{ rows} & \vdots & \vdots & & \vdots & & \vdots \\
 & 2.7 & 0.2 & & 2.8 & & 1.2
\end{array}$$

L columns

1. Randomly pick B subsamples of size $m = [n/\log n]$
2. Compute $(T_m^{(1)}, \dots, T_m^{(L)})$ for each subsample
3. In this $B \times L$ matrix, replace each entry by its rank
4. Normalize the ranks **(Copula estimate)**
5. Apply F_0^{-1} entry-wise **(Enforce the margin)**
6. Aggregate

Rank-transformed subsampling

$$\begin{array}{c|cccc|c}
 & & & & & ✅ \\
 & \widetilde{T}_m^{(1)} & \widetilde{T}_m^{(2)} & \cdots & \widetilde{T}_m^{(L)} & \tilde{S}_n \\
 & 1.6 & -0.8 & & 0.1 & 0.5 \\
 & -1.1 & -0.2 & & 1.8 & -0.1 \\
B \text{ rows} & & & & & \\
 & \vdots & \vdots & & \vdots & \vdots \\
 & 2.7 & 0.2 & & 2.8 & 1.2
\end{array}$$

L columns

1. Randomly pick B subsamples of size $m = [n/\log n]$
2. Compute $(T_m^{(1)}, \ldots, T_m^{(L)})$ for each subsample
3. In this $B \times L$ matrix, replace each entry by its rank
4. Normalize the ranks **(Copula estimate)**
5. Apply F_0^{-1} entry-wise **(Enforce the margin)**
6. Aggregate
7. Use upper α quantile of $\tilde{S}_n$ as critical value

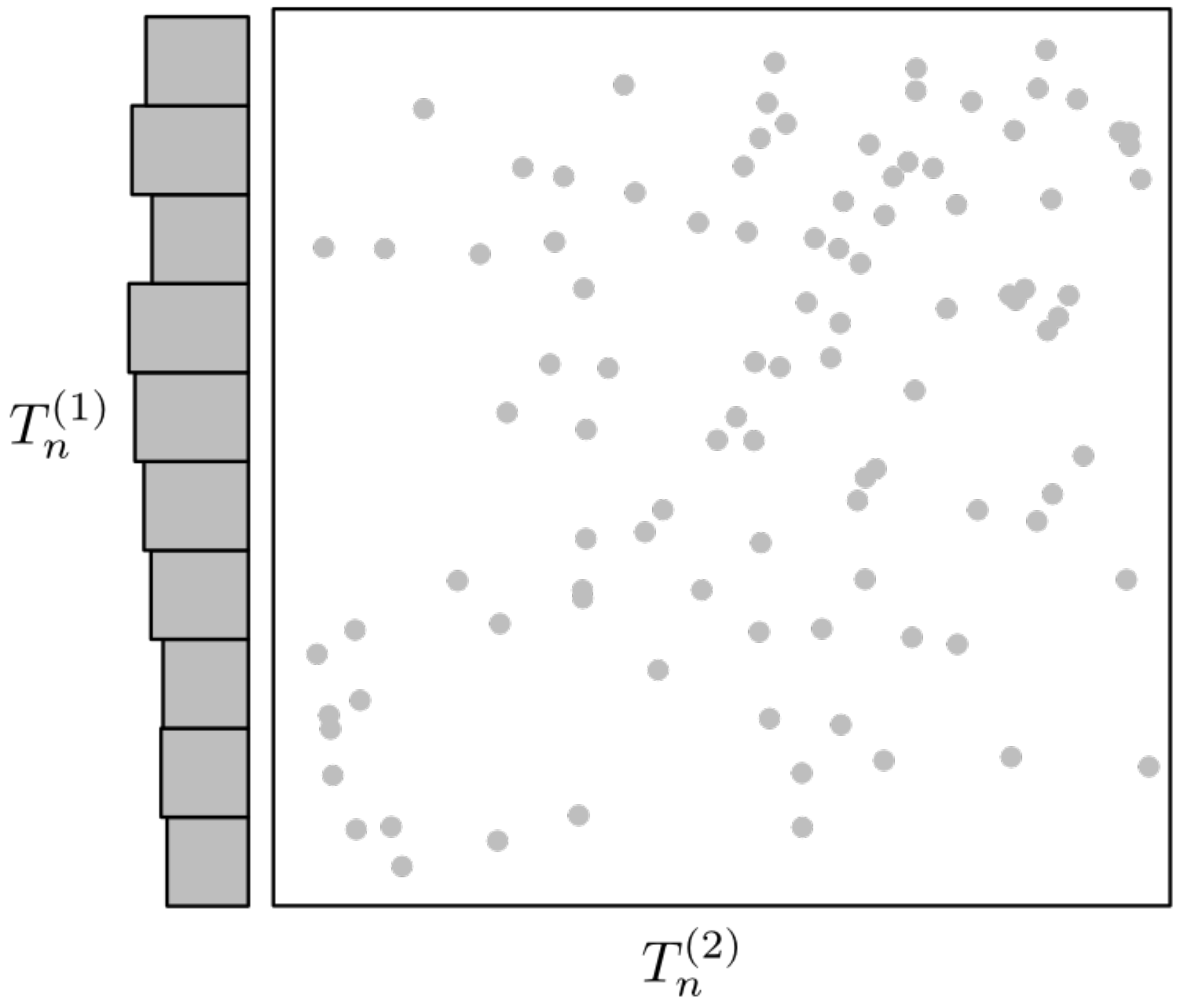
Rank-transformed subsampling: under H_0

Rank-transformed subsampling: under H_0

$L = 2$, $F_0 =$ unif$(0,1)$

Rank-transformed subsampling: under H_0

$L = 2$, $F_0 =$ unif$(0,1)$

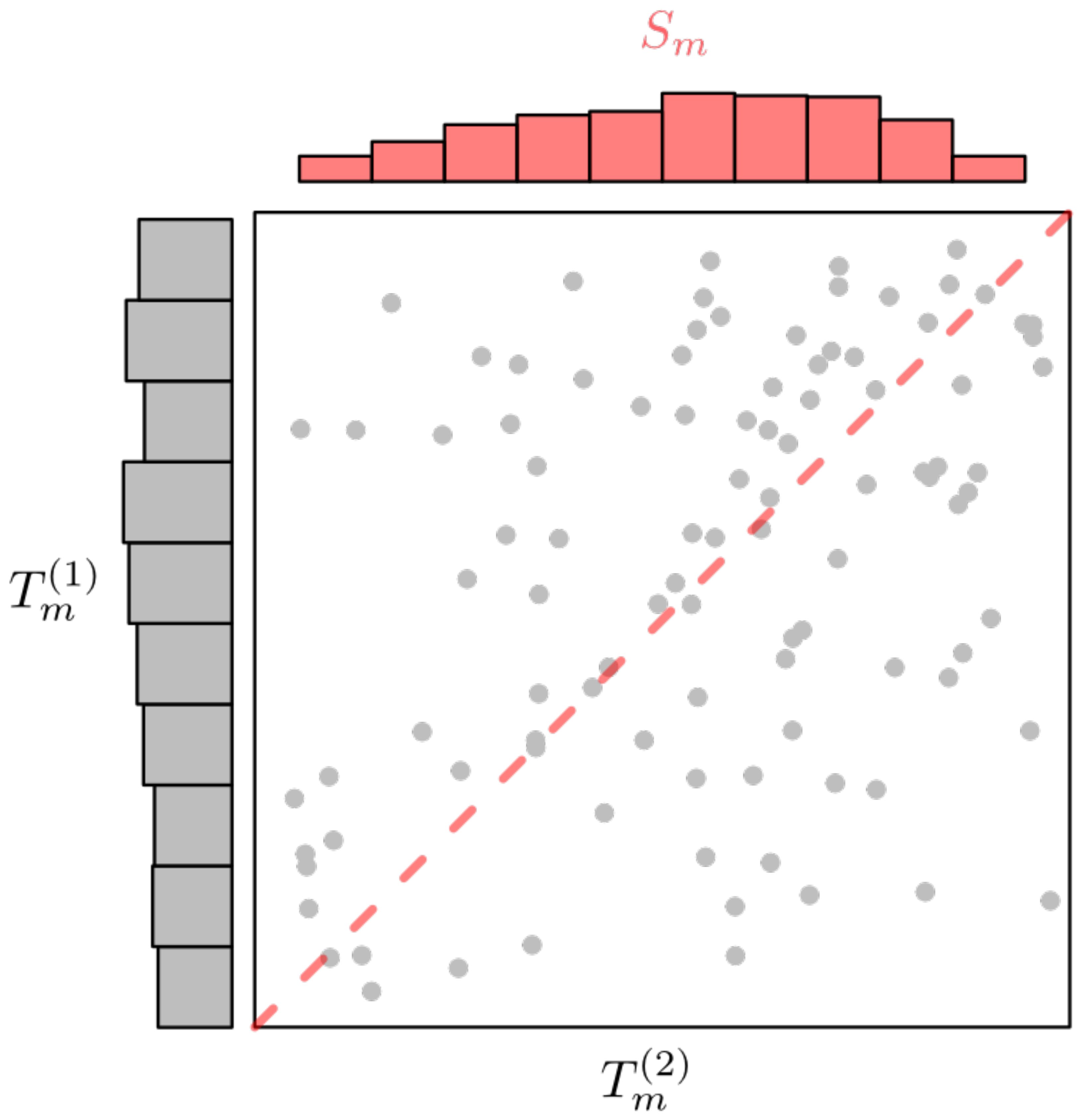


before rank transform

Rank-transformed subsampling: under H_0

$L = 2$, $F_0 =$ unif$(0,1)$

$S =$ avg



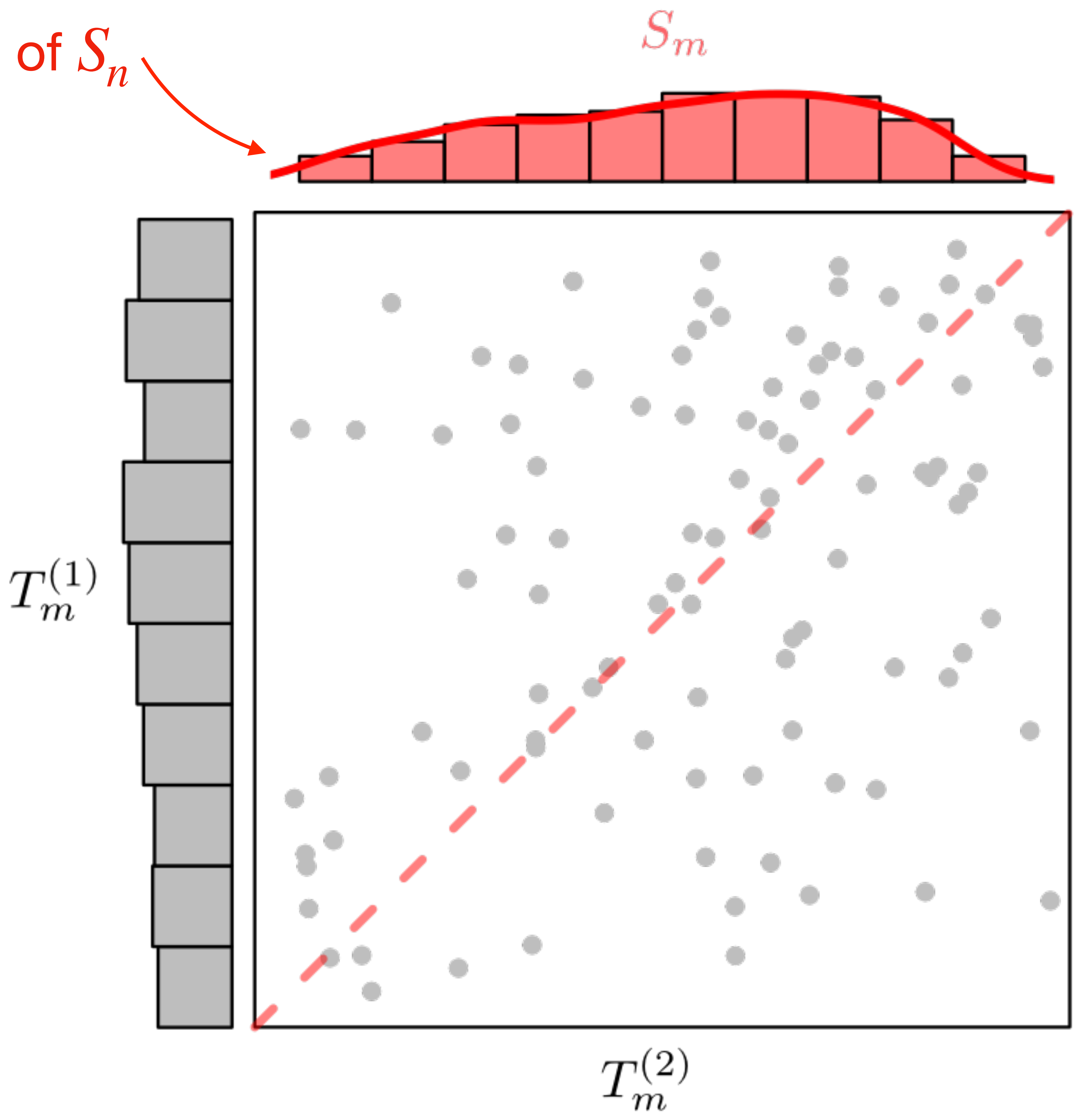
before rank transform

Rank-transformed subsampling: under H_0

$L = 2$, $F_0 =$ unif$(0,1)$

$S =$ avg

Null distribution of S_n



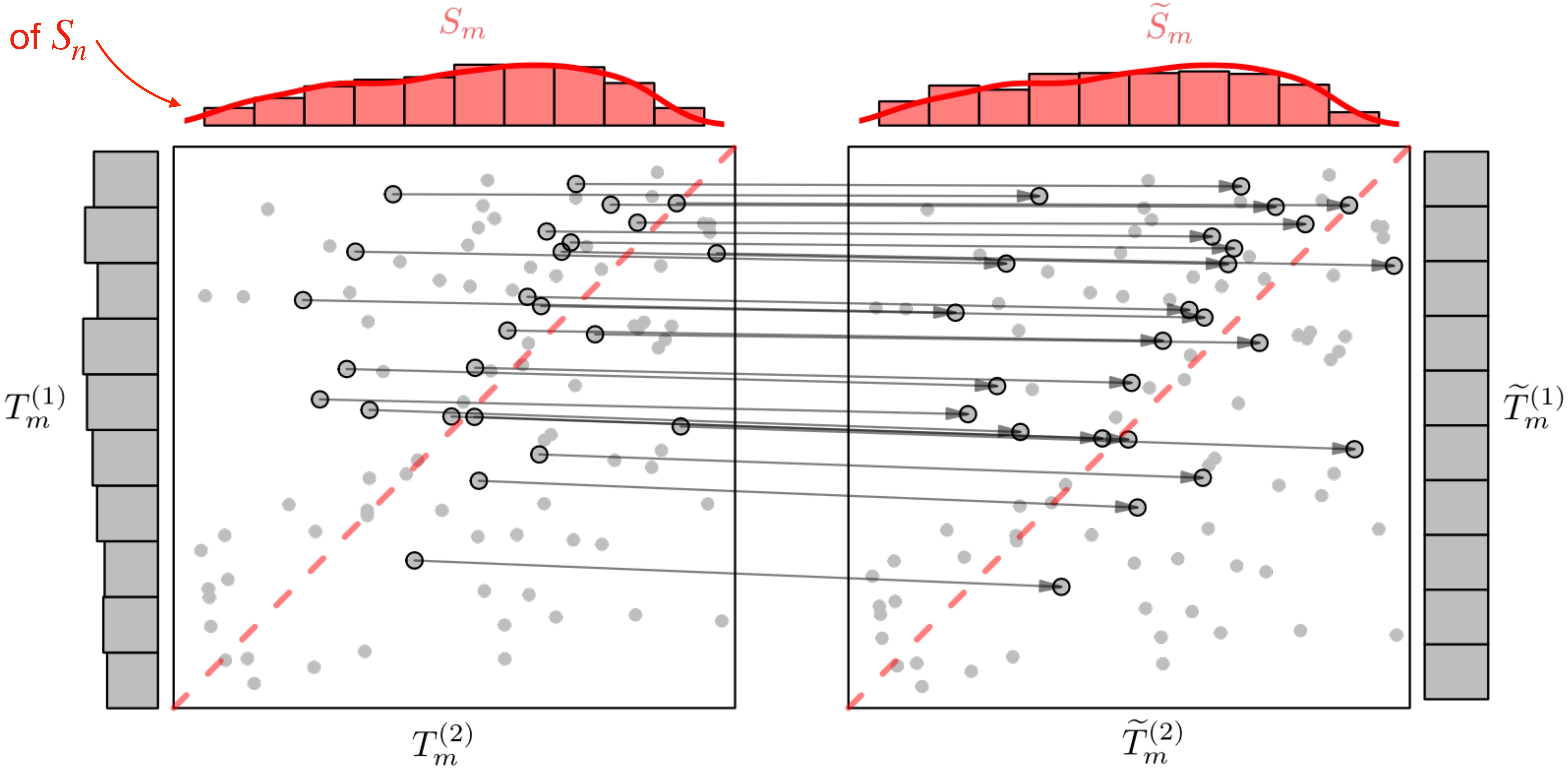
before rank transform

Rank-transformed subsampling: under H_0

$L = 2$, $F_0 =$ unif$(0,1)$

$S =$ avg

Null distribution of S_n



before rank transform

after rank transform

Theory: under H_0

Theory: under H_0

A1. For $P \in H_0$, $T_n(X;\Omega) \to_d F_0 \in \{\text{unif}(0,1), \mathcal{N}(0,1)\}$ as $n \to \infty$.

Theory: under H_0

A1. For $P \in H_0$, $T_n(X;\Omega) \to_d F_0 \in \{\text{unif}(0,1), \mathcal{N}(0,1)\}$ as $n \to \infty$.

Theorem Suppose $S(\,\cdot\,)$ is symmetric and Lipschitz. Suppose the aggregated S_n has a continuous asymptotic law under H_0.
Then, under A1, our test is pointwise asymptotically level α.

✅ **Not conservative**

Theory: under H_0

A1. For $P \in H_0$, $T_n(X;\Omega) \to_d F_0 \in \{\text{unif}(0,1), \mathcal{N}(0,1)\}$ as $n \to \infty$.

Theorem Suppose $S(\cdot)$ is symmetric and Lipschitz. Suppose the aggregated S_n has a continuous asymptotic law under H_0.
Then, under A1, our test is pointwise asymptotically level α.

Further, if T_n and S_n converge to their respective limit distributions uniformly over H_0, then our test is uniformly asymptotic level α.

✅ **Not conservative**

Rank-transformed subsampling: Local alternative

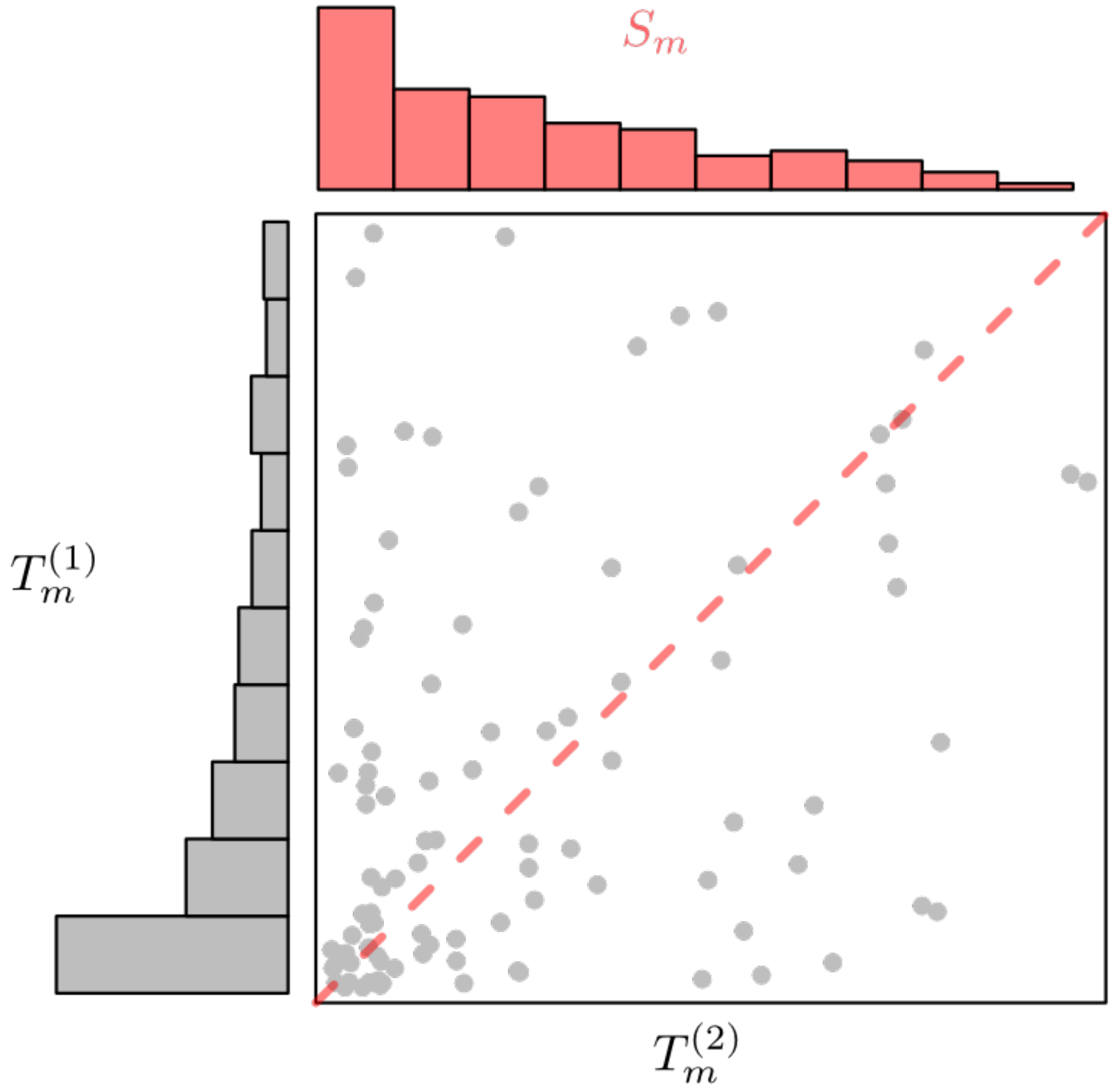
$L = 2$, $F_0 = \text{unif}(0,1)$

$S = \text{avg}$

Rank-transformed subsampling: Local alternative

$L = 2$, $F_0 =$ unif$(0,1)$

$S =$ avg



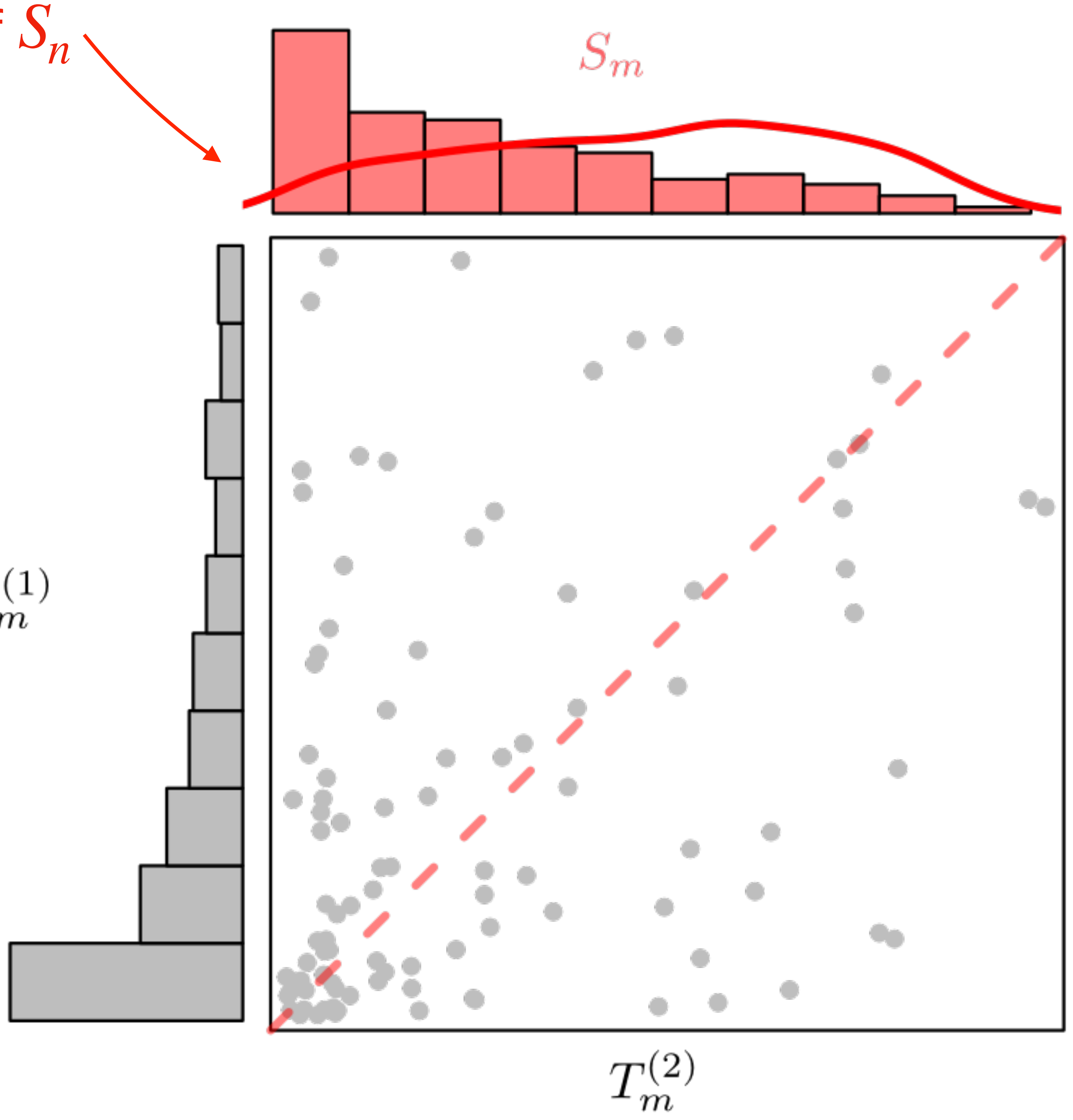
before rank transform

Rank-transformed subsampling: Local alternative

$L = 2, F_0 =$ unif$(0,1)$

$S =$ avg

Null distribution of S_n



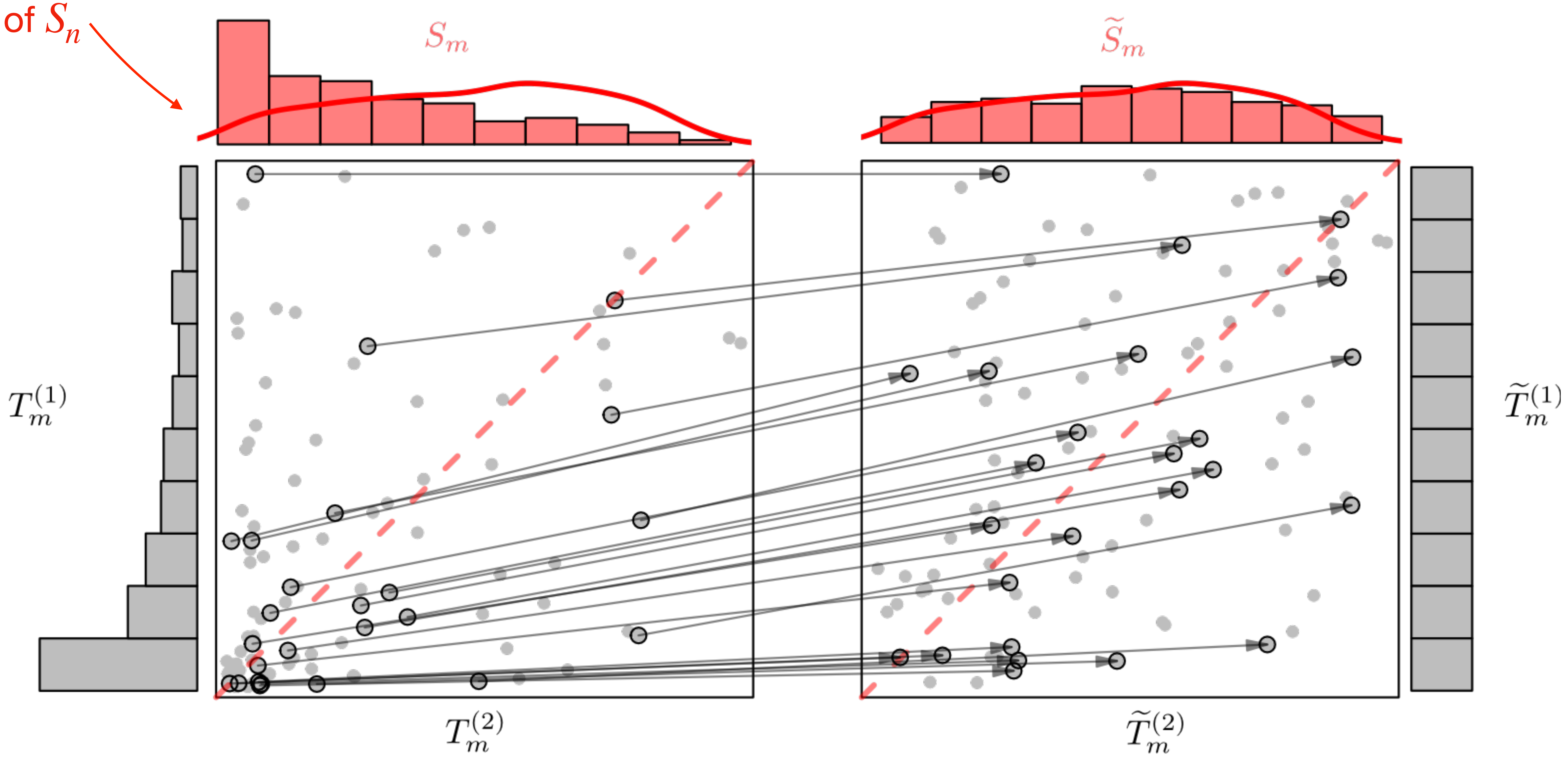
before rank transform

Rank-transformed subsampling: Local alternative

$L = 2, F_0 =$ unif$(0,1)$

$S =$ avg

Null distribution of S_n



before rank transform

after rank transform

Rank-transformed subsampling: Local alternative

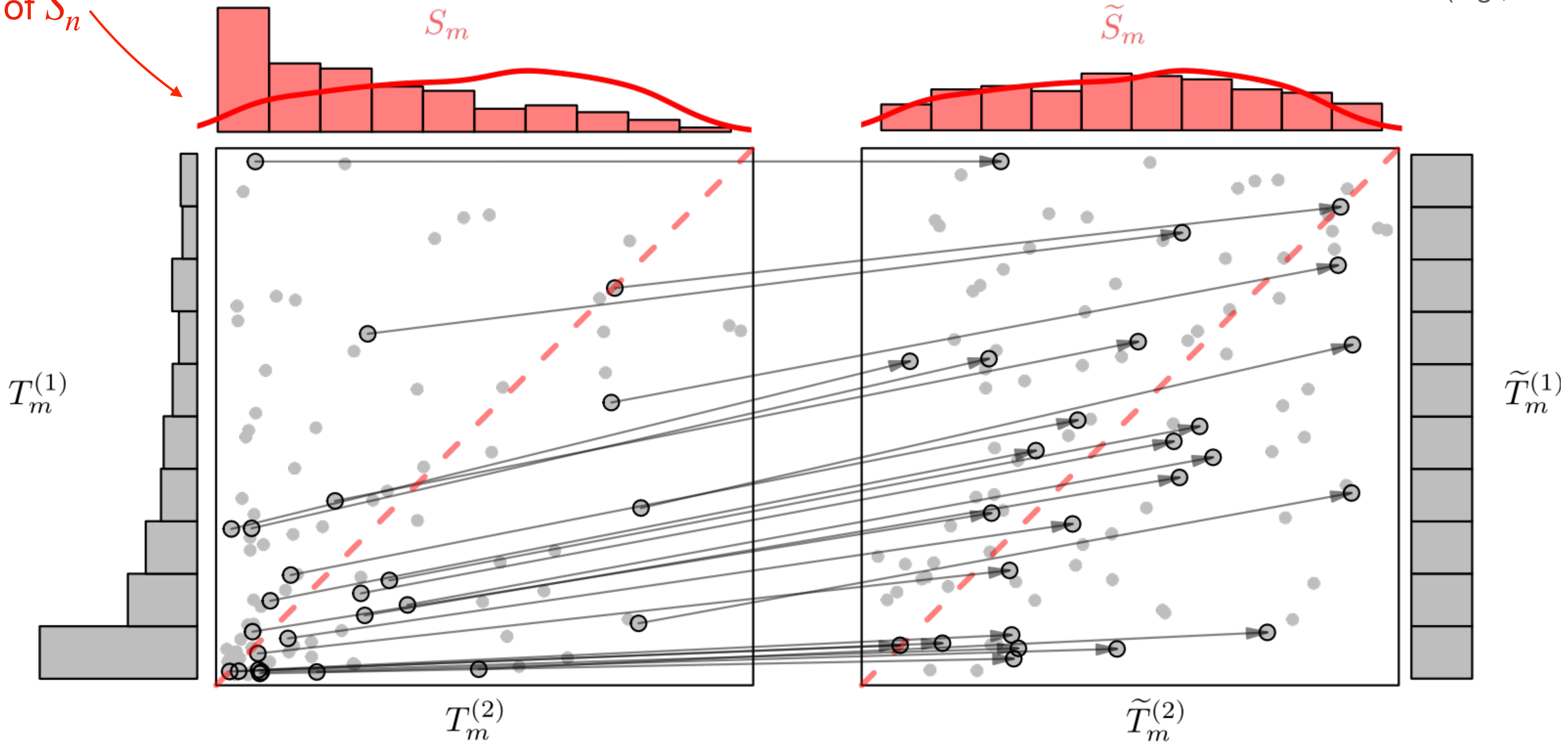
$L = 2$, $F_0 =$ unif$(0,1)$

$S =$ avg

💡 **Intuition:** copula under the null ≈ copula under local alternatives

(e.g., Le Cam's 3rd Lemma)

Null distribution of S_n



before rank transform

after rank transform

Theory: Local power

Theory: Local power

Theorem (informal) Fix $P_0 \in H_0$.

If the copula of $(T_n^{(1)}, \ldots, T_n^{(L)})$ converges in a locally uniform fashion at P_0, then for P_0's local alternatives,

| Power(our test) - Power(oracle test) | $\to 0,$

where the oracle test has access to S_n's null distribution under P_0.

💡 For example, when Le Cam's 3rd lemma is applicable to $(T_n^{(1)}, \ldots, T_n^{(L)})$.

Outline

- Setup and main challenge
- Method: Rank-transformed subsampling
- Applications
  - **Hunt and test**
  - Improving inference for double machine learning
  - Testing no direct effect in a sequentially randomized trial
- Future directions

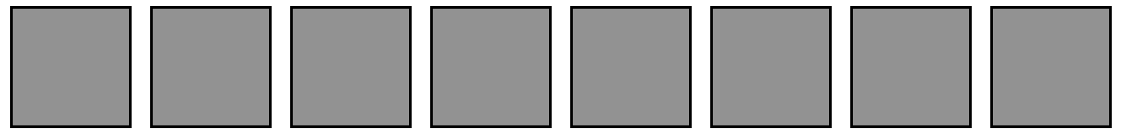
Hunt and test

Hunt and test

👉 Test hypothesis of the form $H_0 = \cap_d H_0(d)$, where each $H_0(d)$ is relatively easy to test.

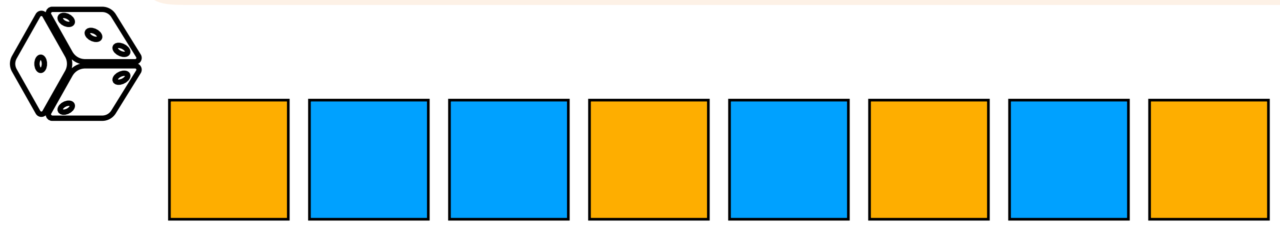
Hunt and test

👉 Test hypothesis of the form $H_0 = \cap_d H_0(d)$, where each $H_0(d)$ is relatively easy to test.



Hunt and test

👉 Test hypothesis of the form $H_0 = \cap_d H_0(d)$, where each $H_0(d)$ is relatively easy to test.



Hunt and test

👉 Test hypothesis of the form $H_0 = \cap_d H_0(d)$, where each $H_0(d)$ is relatively easy to test.

🎲 🟧🟦🟦🟧🟦🟧🟦🟧 (1) Use 🟦🟦🟦🟦 to find $\hat{d}$ such that $H_0(\hat{d})$ is most likely to be rejected.

Hunt and test

👉 Test hypothesis of the form $H_0 = \cap_d H_0(d)$, where each $H_0(d)$ is relatively easy to test.

🎲 🟧🟦🟦🟧🟦🟧🟦🟧

(1) Use 🟦🟦🟦🟦 to find $\hat{d}$ such that $H_0(\hat{d})$ is most likely to be rejected.

(2) Use 🟧🟧🟧🟧 to compute a test statistic for $H_0(\hat{d})$ and call it T_n.

Hunt and test

👉 Test hypothesis of the form $H_0 = \cap_d H_0(d)$, where each $H_0(d)$ is relatively easy to test.

🎲 🟧🟦🟦🟧🟦🟧🟦🟧

(1) Use 🟦🟦🟦🟦 to find $\hat{d}$ such that $H_0(\hat{d})$ is most likely to be rejected.

⚠️ **NOT selective inference!**

(2) Use 🟧🟧🟧🟧 to compute a test statistic for $H_0(\hat{d})$ and call it T_n.

Hunt and test

👉 Test hypothesis of the form $H_0 = \cap_d H_0(d)$, where each $H_0(d)$ is relatively easy to test.

🎲 🟧🟦🟦🟧🟦🟧🟦🟧

⚠️ NOT selective inference!

(1) Use 🟦🟦🟦🟦 to find $\hat{d}$ such that $H_0(\hat{d})$ is most likely to be rejected.

(2) Use 🟧🟧🟧🟧 to compute a test statistic for $H_0(\hat{d})$ and call it T_n.

Laura

$X \in \mathbb{R}^p$: gene expression of a random cell in the sample.

Hunt and test

👉 Test hypothesis of the form $H_0 = \cap_d H_0(d)$, where each $H_0(d)$ is relatively easy to test.

🟧🟦🟦🟧🟦🟧🟦🟧

⚠️ NOT selective inference!

(1) Use 🟦🟦🟦🟦 to find $\hat{d}$ such that $H_0(\hat{d})$ is most likely to be rejected.

(2) Use 🟧🟧🟧🟧 to compute a test statistic for $H_0(\hat{d})$ and call it T_n.

Laura

$X \in \mathbb{R}^p$: gene expression of a random cell in the sample.

$H_0 = \{X \sim \text{only one subtype}\}$

$= \{X \sim \text{unimodal}\}$

👉 very hard

Hunt and test

👉 Test hypothesis of the form $H_0 = \cap_d H_0(d)$, where each $H_0(d)$ is relatively easy to test.

🎲 🟧🟦🟦🟧🟦🟧🟦🟧

⚠️ NOT selective inference!

(1) Use 🟦🟦🟦🟦 to find $\hat{d}$ such that $H_0(\hat{d})$ is most likely to be rejected.

(2) Use 🟧🟧🟧🟧 to compute a test statistic for $H_0(\hat{d})$ and call it T_n.

Laura

$X \in \mathbb{R}^p$: gene expression of a random cell in the sample.

$H_0 = \{X \sim \text{only one subtype}\}$

$= \{X \sim \text{unimodal}\}$ 👉 very hard

$= \cap_{d \in \mathbb{R}^p} \{d^\top X \sim \text{unimodal}\}$ 👉 linear unimodality

Hunt and test

👉 Test hypothesis of the form $H_0 = \cap_d H_0(d)$, where each $H_0(d)$ is relatively easy to test.

🟧🟦🟦🟧🟦🟧🟦🟧

⚠️ NOT selective inference!

(1) Use 🟦🟦🟦🟦 to find $\hat{d}$ such that $H_0(\hat{d})$ is most likely to be rejected.

(2) Use 🟧🟧🟧🟧 to compute a test statistic for $H_0(\hat{d})$ and call it T_n.

Laura

$X \in \mathbb{R}^p$: gene expression of a random cell in the sample.

$H_0 = \{X \sim \text{only one subtype}\}$

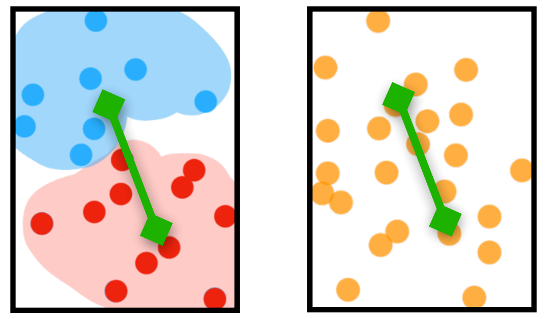
$= \{X \sim \text{unimodal}\}$ 👉 very hard

$= \cap_{d \in \mathbb{R}^p} \{d^\top X \sim \text{unimodal}\}$ 👉 linear unimodality

(1) Find $\hat{d}$ by running 2-means on 🟦🟦🟦🟦 .

(2) Compute T_n := dip test p-value on 🟧🟧🟧🟧.

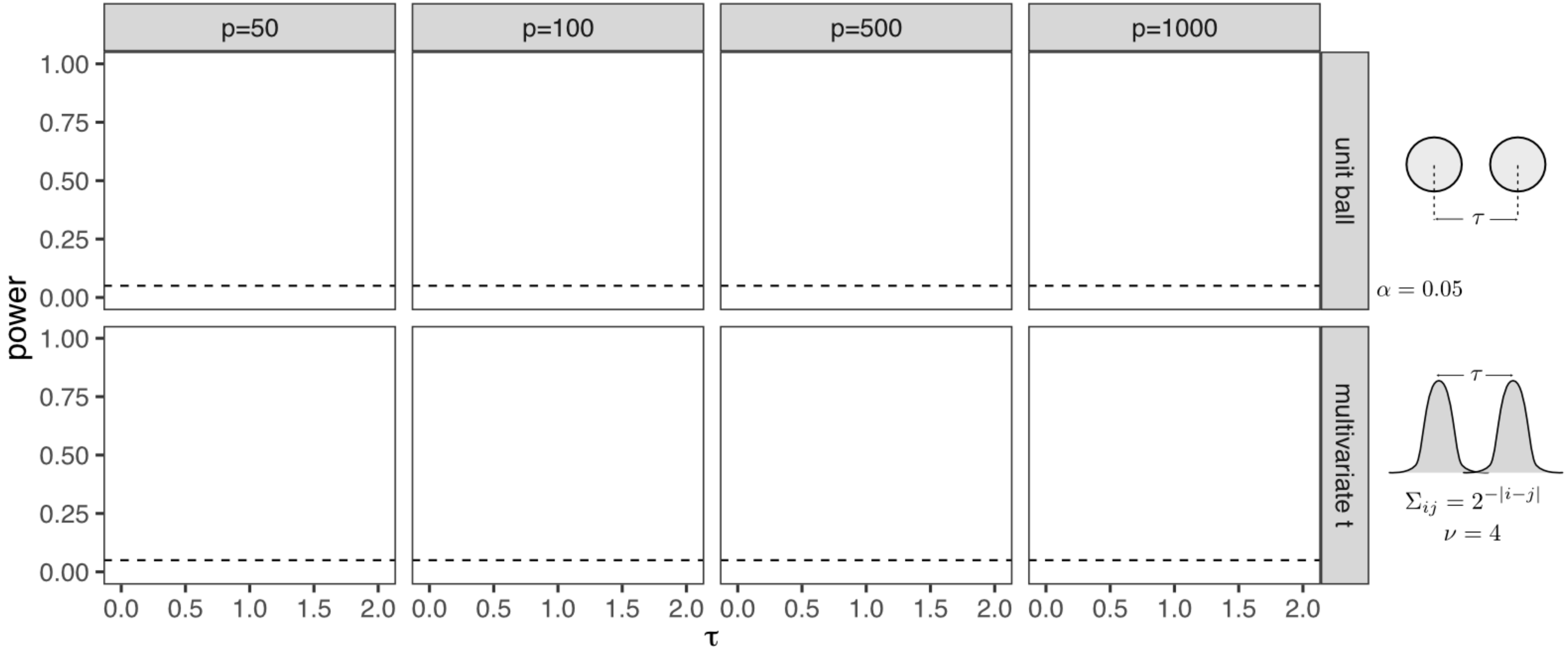
Cheng, M-Y., and Peter Hall. "Calibrating the excess mass and dip tests of modality." JRSS-B (1998)



⚠️ Low power

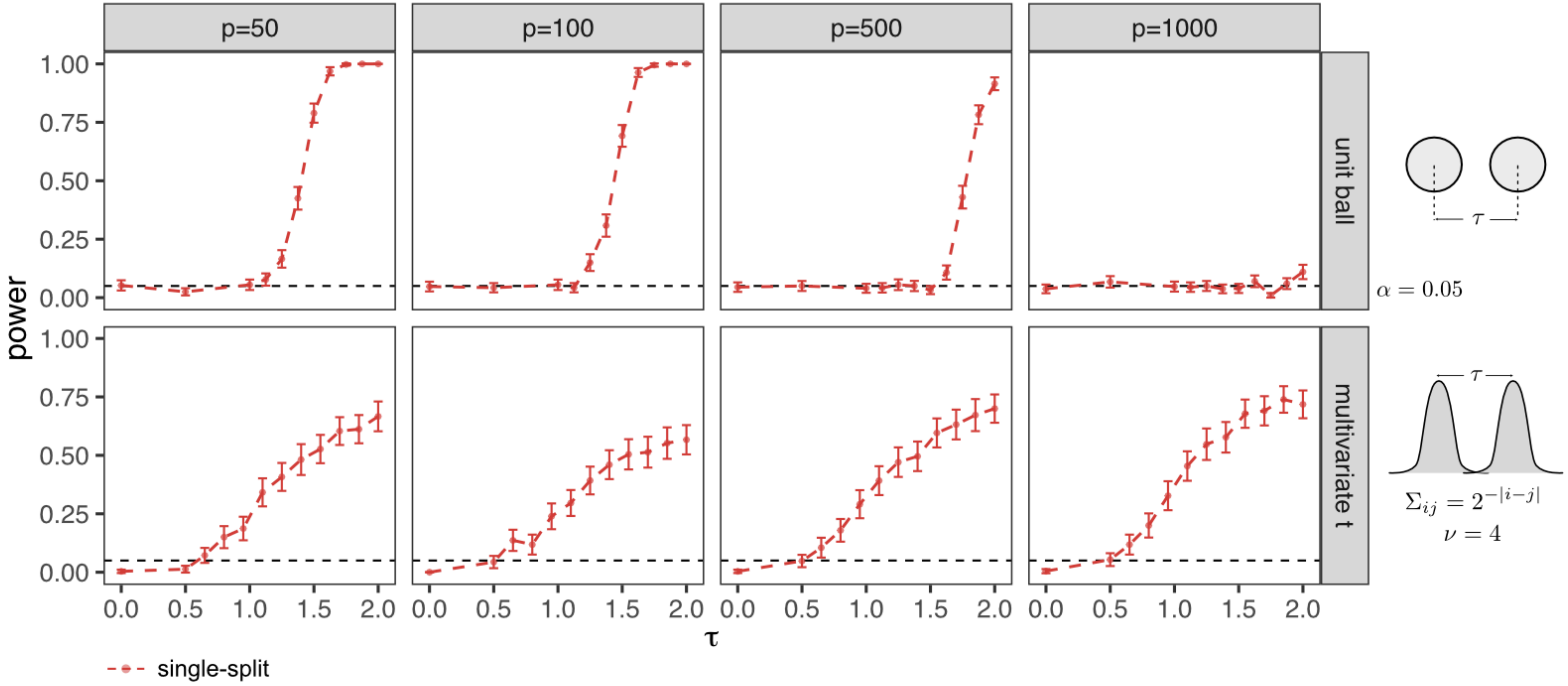
Hunt and test: Detecting cancer subtypes

Simulation in $\mathbb{R}^p$



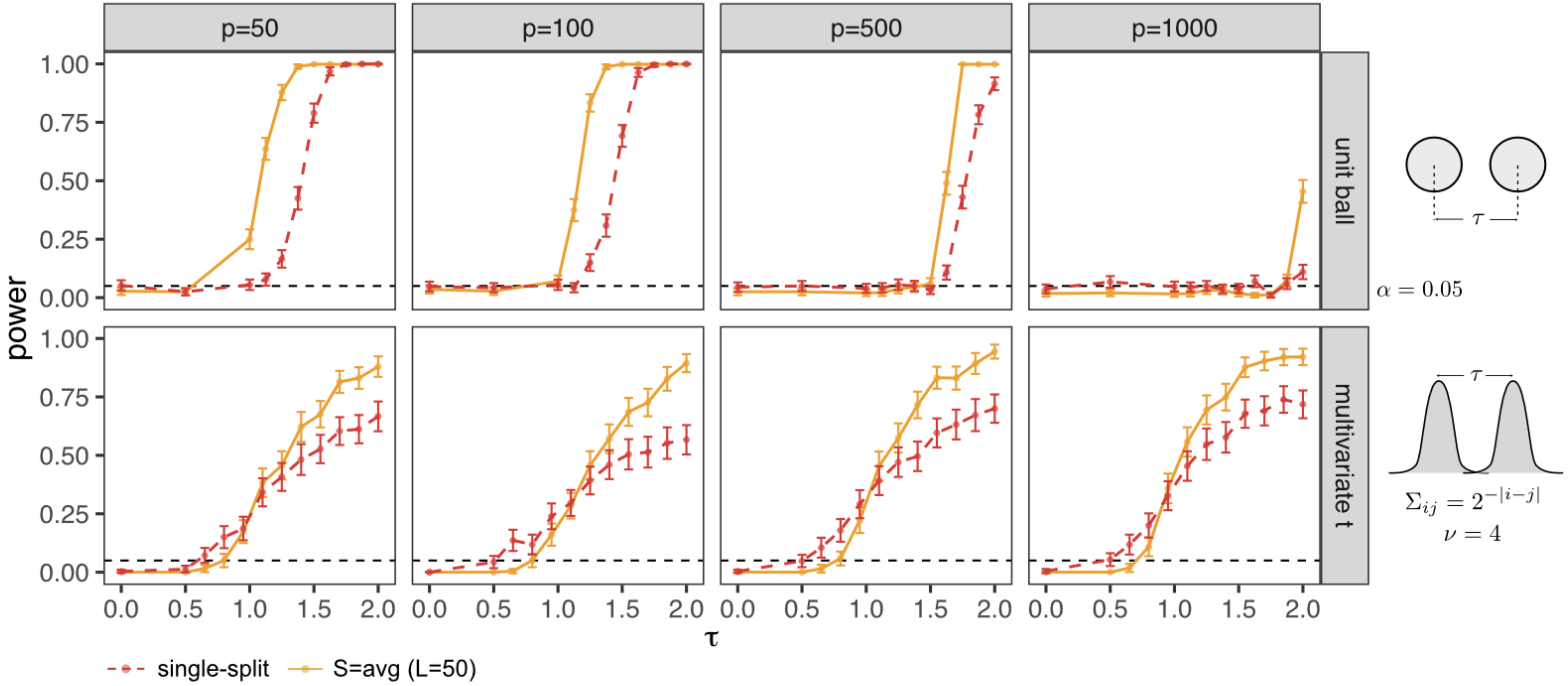
Hunt and test: Detecting cancer subtypes

Simulation in $\mathbb{R}^p$



Hunt and test: Detecting cancer subtypes

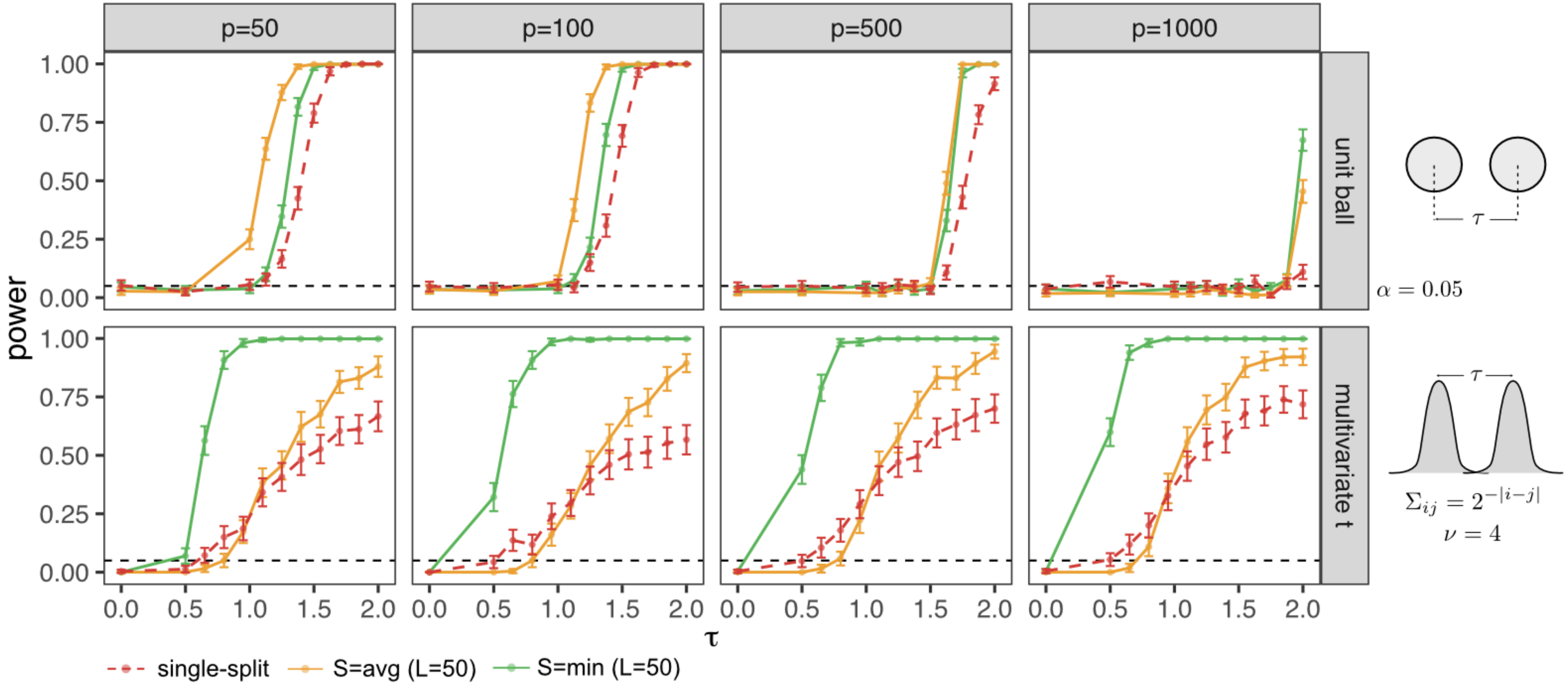
Simulation in $\mathbb{R}^p$



- Rank-transform subsampling maintains the correct level and significantly improves power.

Hunt and test: Detecting cancer subtypes

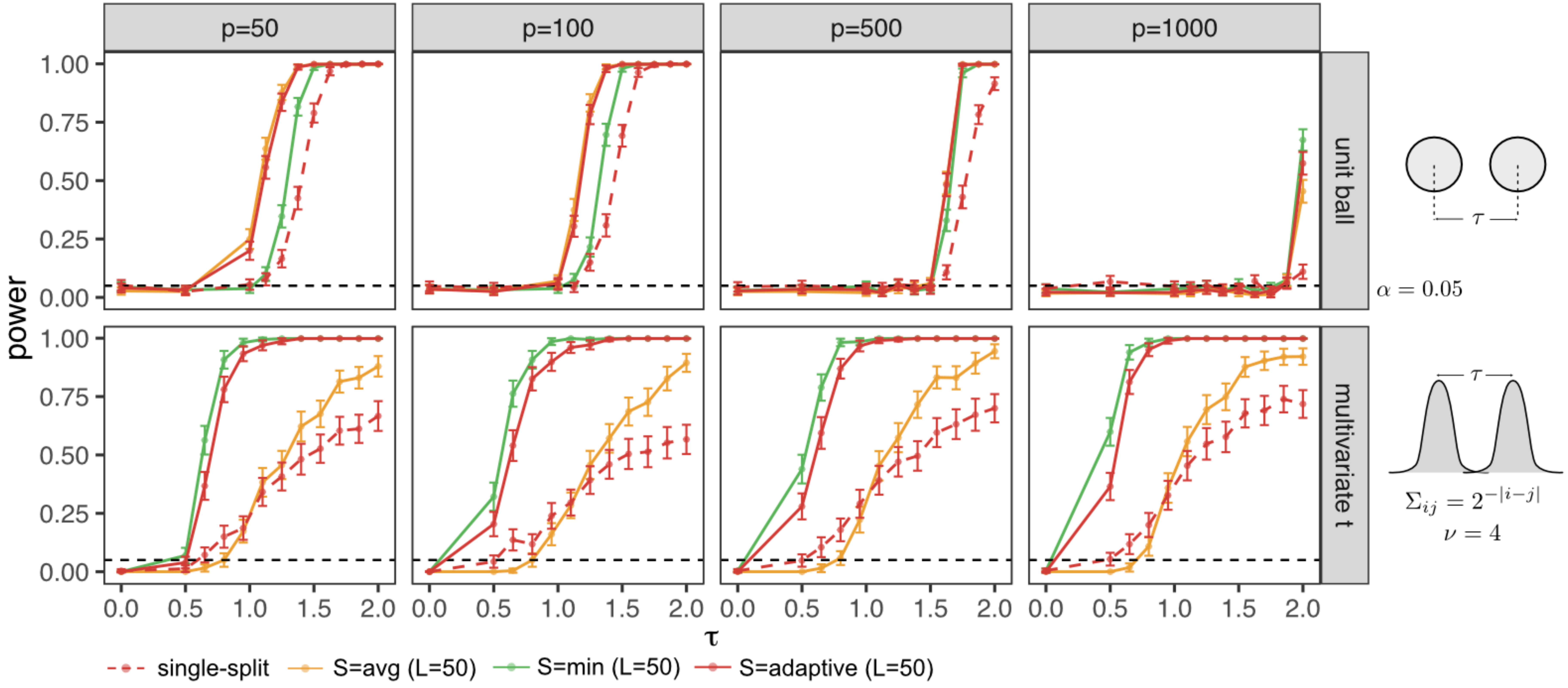
Simulation in $\mathbb{R}^p$



- Rank-transform subsampling maintains the correct level and significantly improves power.

Hunt and test: Detecting cancer subtypes

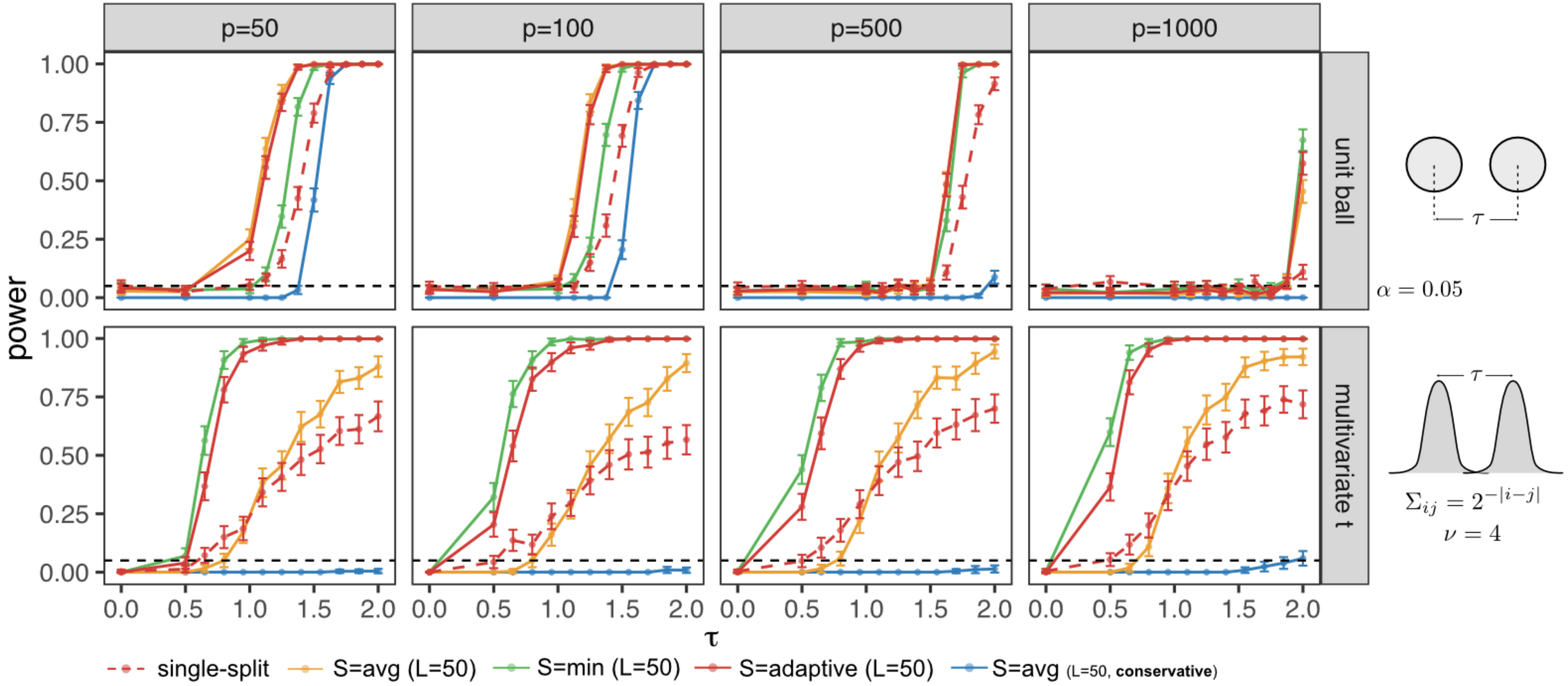
Simulation in $\mathbb{R}^p$



- Rank-transform subsampling maintains the correct level and significantly improves power.
- Adaptive version of the algorithm achieves the better performance between the two choices of S.

Hunt and test: Detecting cancer subtypes

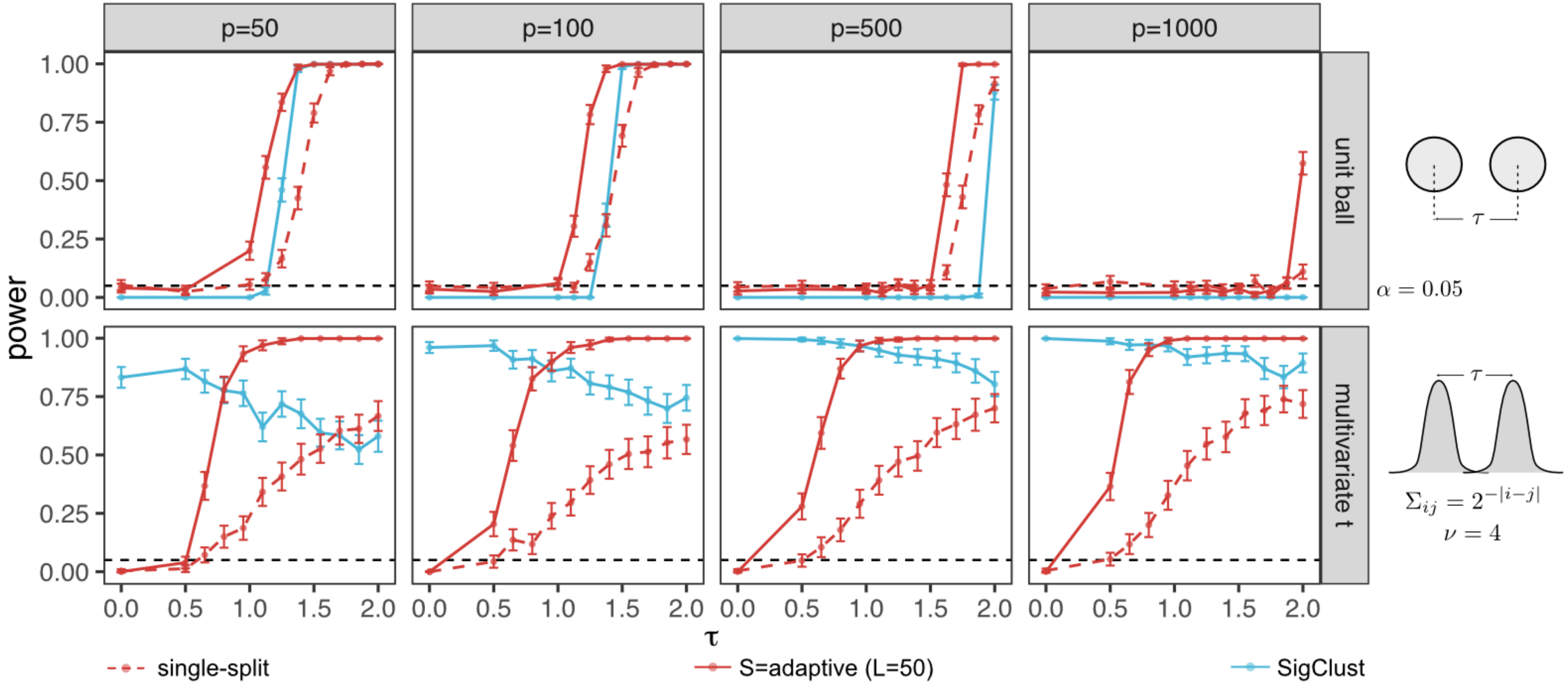
Simulation in $\mathbb{R}^p$



- Rank-transform subsampling maintains the correct level and significantly improves power.
- Adaptive version of the algorithm achieves the better performance between the two choices of S.
- Conservatively averaged p-value is not competitive.

Hunt and test: Detecting cancer subtypes

Simulation in $\mathbb{R}^p$

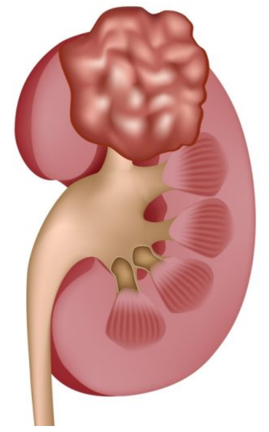


- Rank-transform subsampling maintains the correct level and significantly improves power.
- Adaptive version of the algorithm achieves the better performance between the two choices of S.
- Conservatively averaged p-value is not competitive.
- SigClust: for unit balls, it loses power as p increases; for multivariate t, it does not control type-I error.

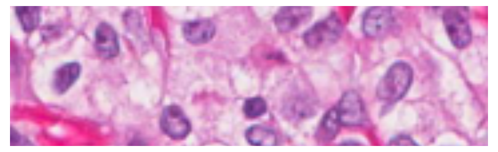
Yufeng Liu, David Neil Hayes, Andrew Nobel, and J. S Marron.
Statistical significance of clustering for high-dimension, low–sample size data.
Journal of the American Statistical Association (2008).
https://CRAN.R-project.org/package=sigclust

Hunt and test: Detecting cancer subtypes

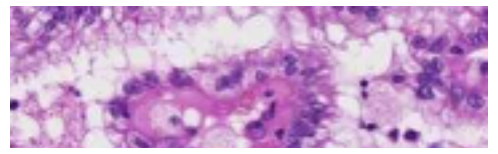
ICGC/TCGA Pan-Cancer dataset



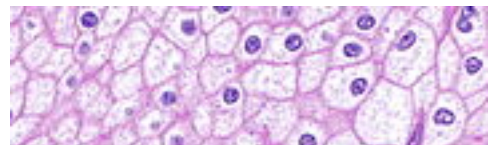
111 Kidney Cancer
(Renal Cell Carcinoma)
Cases



37 Clear Cell (cc)



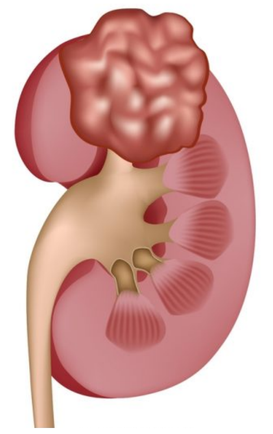
31 Papillary (P)



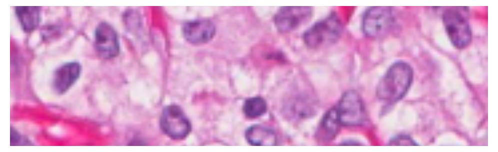
43 Chromophobe (Ch)

Hunt and test: Detecting cancer subtypes

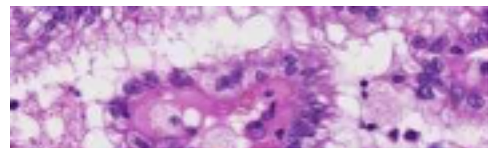
ICGC/TCGA Pan-Cancer dataset



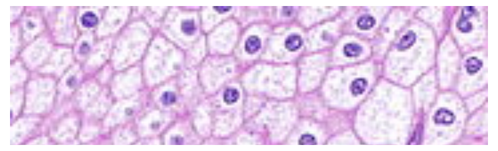
111 Kidney Cancer
(Renal Cell Carcinoma)
Cases



37 Clear Cell (cc)



31 Papillary (P)



43 Chromophobe (Ch)

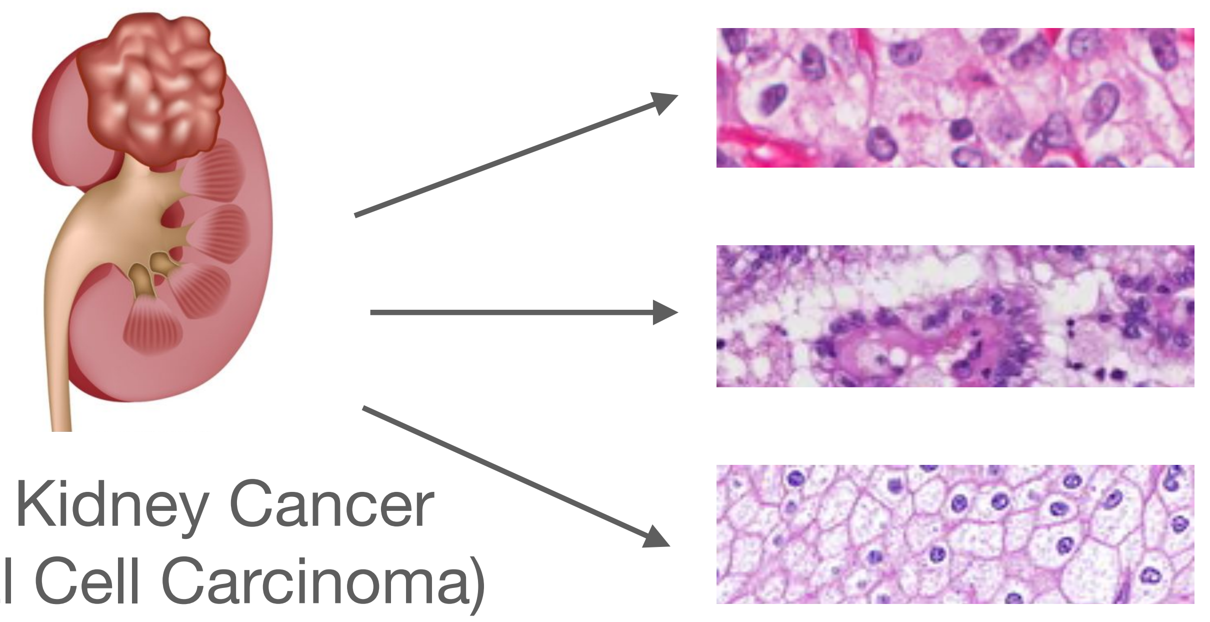
Selected 1000 genes by comparing to control

	Gene 1	**Gene 2**	**Gene 3**	**...**
Cell 1	-1.2	0.5	6.2	
Cell 2	0.1	12	1.1	
Cell 3	-2.2	0	-2	
⋮				

Normalized mRNA expression

Hunt and test: Detecting cancer subtypes

ICGC/TCGA Pan-Cancer dataset



111 Kidney Cancer (Renal Cell Carcinoma) Cases

37 Clear Cell (cc)

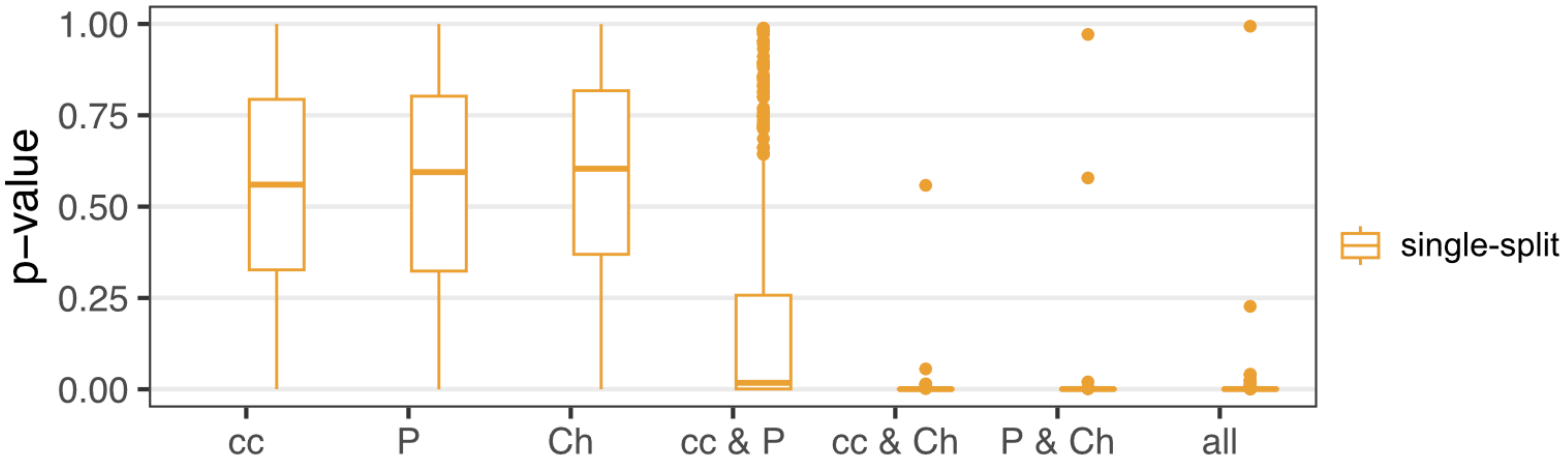
31 Papillary (P)

43 Chromophobe (Ch)

Selected 1000 genes by comparing to control

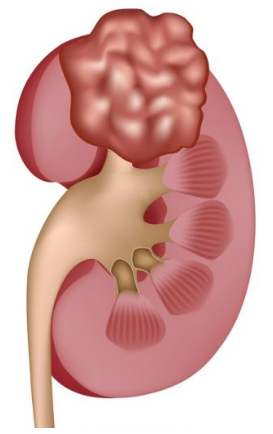
	Gene 1	Gene 2	Gene 3	...
Cell 1	-1.2	0.5	6.2	
Cell 2	0.1	12	1.1	
Cell 3	-2.2	0	-2	
⋮				

Normalized mRNA expression

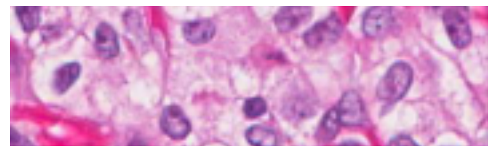


Hunt and test: Detecting cancer subtypes

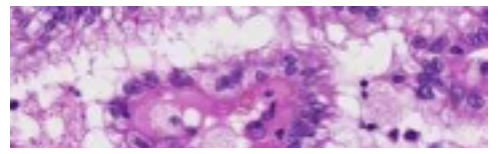
ICGC/TCGA Pan-Cancer dataset



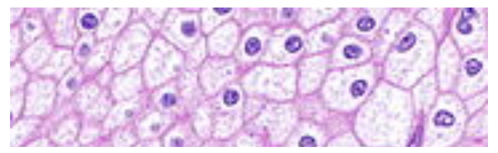
111 Kidney Cancer
(Renal Cell Carcinoma)
Cases



37 Clear Cell (cc)



31 Papillary (P)

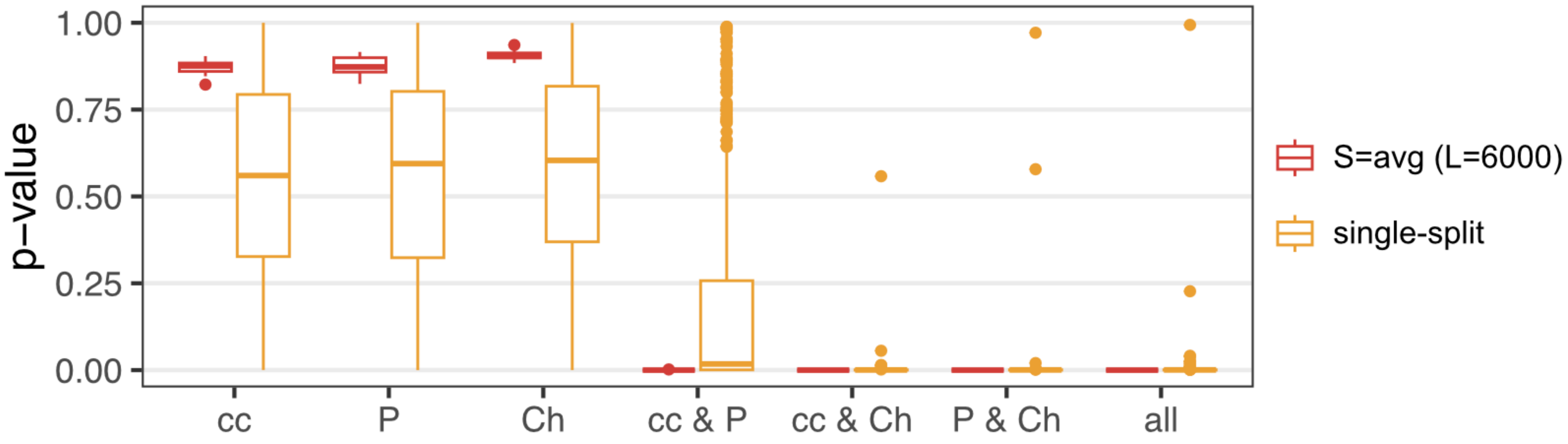


43 Chromophobe (Ch)

Selected 1000 genes by comparing to control

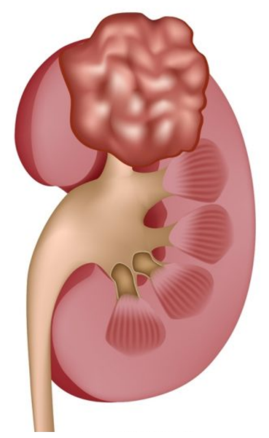
	Gene 1	**Gene 2**	**Gene 3**	**…**
Cell 1	-1.2	0.5	6.2	
Cell 2	0.1	12	1.1	
Cell 3	-2.2	0	-2	
⋮				

Normalized mRNA expression

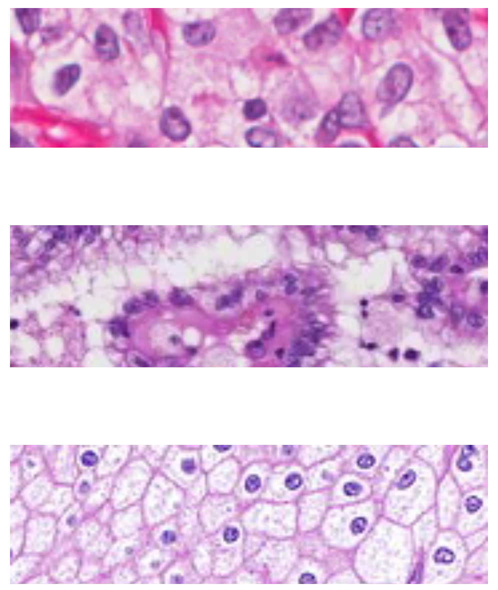


Hunt and test: Detecting cancer subtypes

ICGC/TCGA Pan-Cancer dataset



111 Kidney Cancer
(Renal Cell Carcinoma)
Cases



37 Clear Cell (cc)

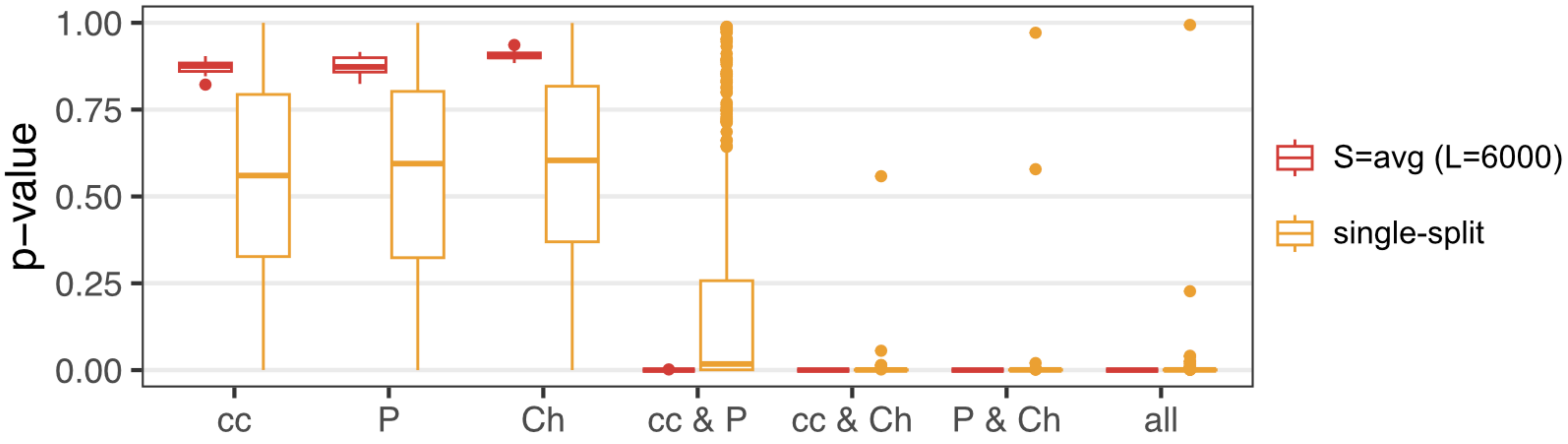
31 Papillary (P)

43 Chromophobe (Ch)

Selected 1000 genes by comparing to control

	Gene 1	**Gene 2**	**Gene 3**	**...**
Cell 1	-1.2	0.5	6.2	
Cell 2	0.1	12	1.1	
Cell 3	-2.2	0	-2	
⋮				

Normalized mRNA expression



Happy Laura

Other hunt-and-test / data-split procedures

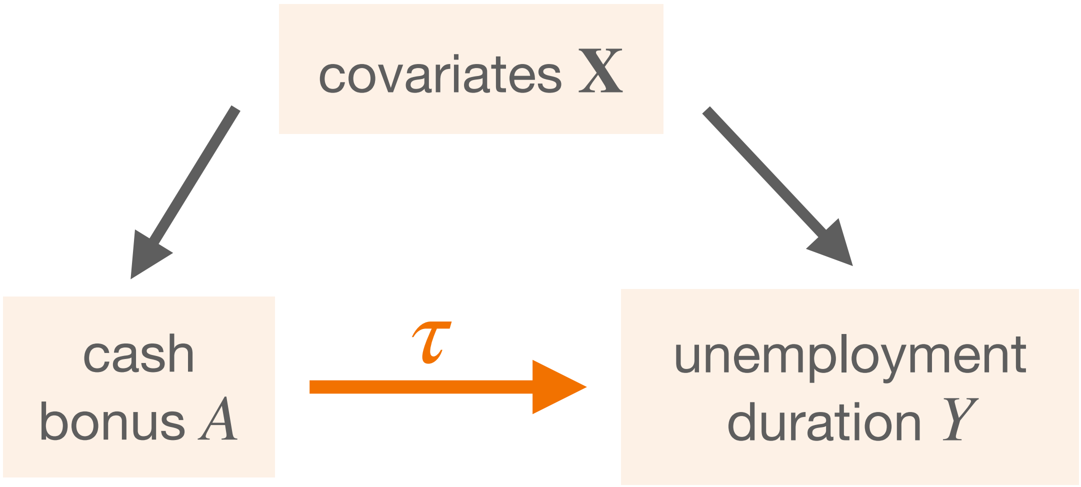
- Testing multiple sample (Cox, 1975)
- Split conformal prediction (Lei et al., 2018; Solari & Djordjilović, 2022)
- Goodness-of-fit testing (Janková et al., 2020)
- Conditional (mean) independence testing (Scheidegger et al., 2021; Lundborg et al., 2022)
- Dimension-agnostic inference (Kim & Ramdas, 2020)

  …

Outline

- Setup and main challenge
- Method: Rank-transformed subsampling
- Applications
  - Hunt and test
  - **Improving inference for double machine learning**
  - Testing no direct effect of a sequentially randomized trial
- Future directions

Bill

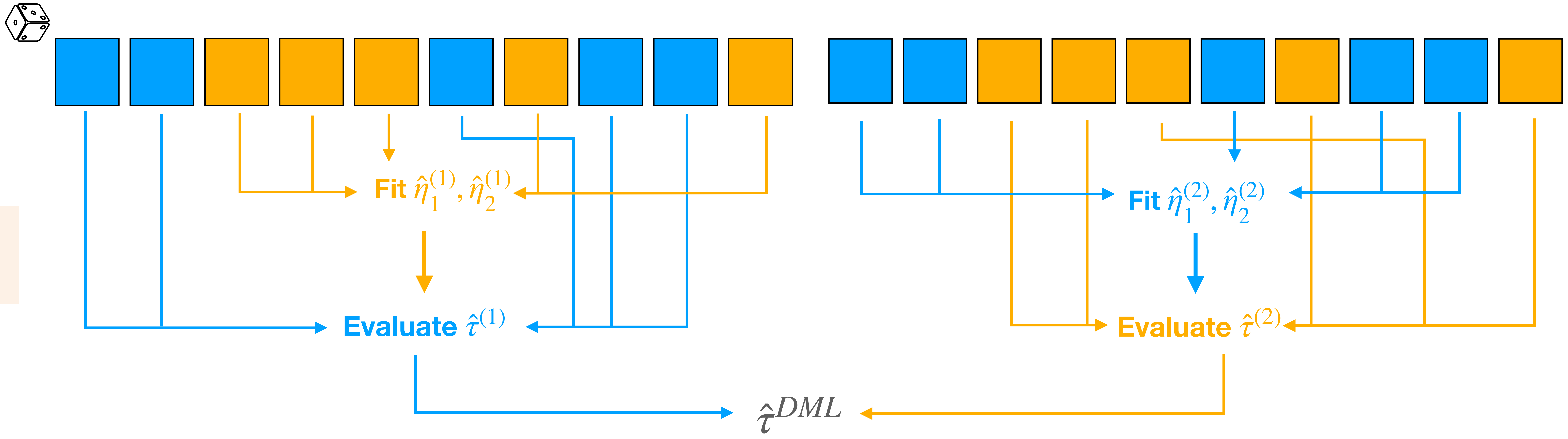


covariates $\mathbf{X}$

cash bonus A

τ

unemployment duration Y



Fit $\hat{\eta}_1^{(1)}, \hat{\eta}_2^{(1)}$

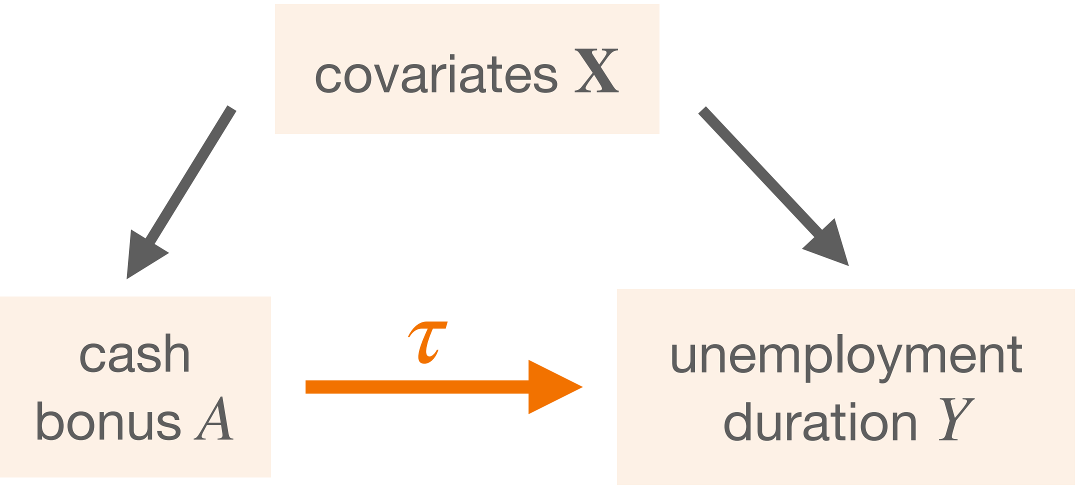
Evaluate $\hat{\tau}^{(1)}$

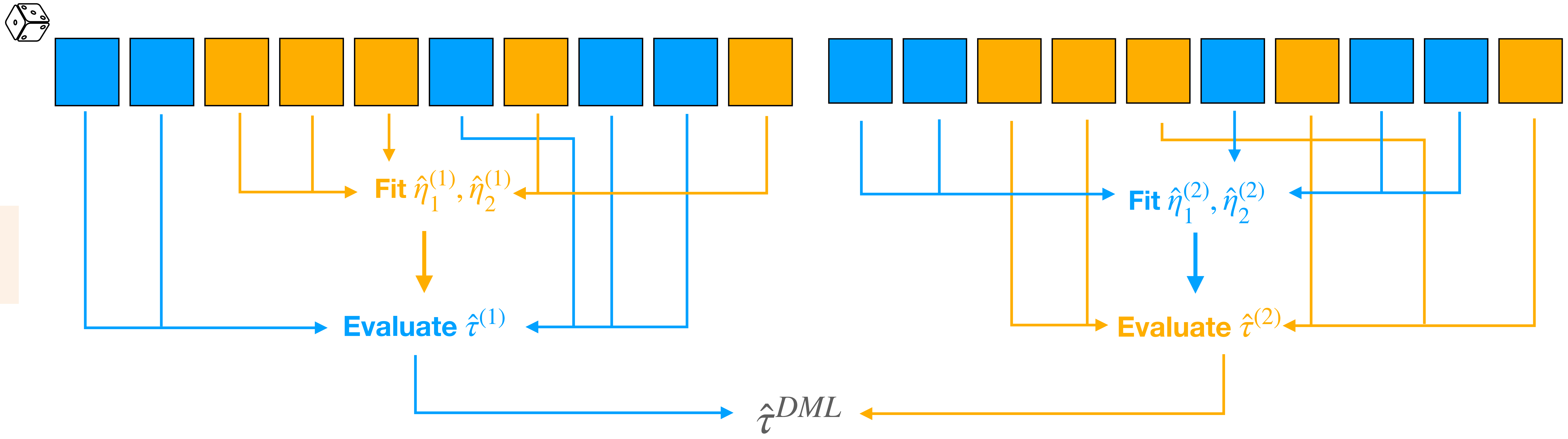
Fit $\hat{\eta}_1^{(2)}, \hat{\eta}_2^{(2)}$

Evaluate $\hat{\tau}^{(2)}$

$\hat{\tau}^{DML}$

Bill



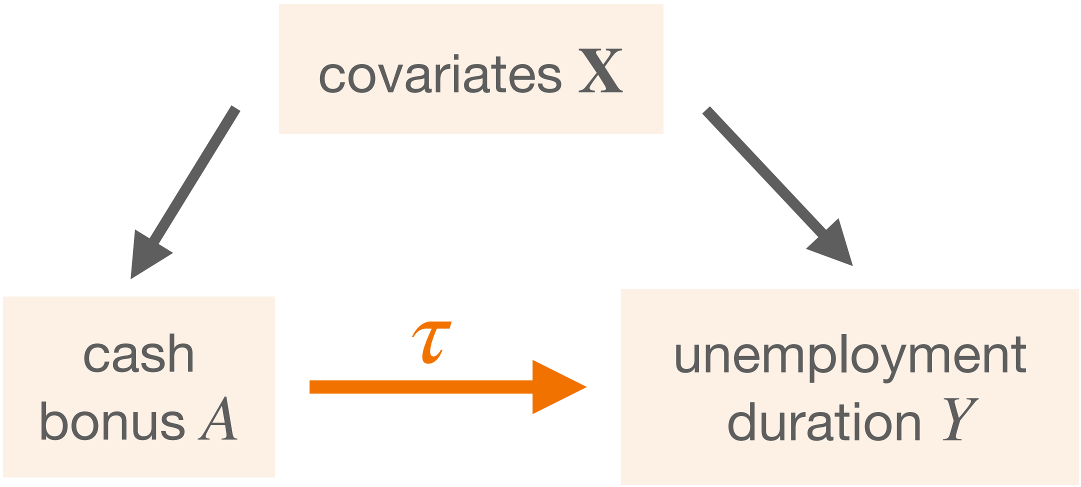

Fit $\hat{\eta}_1^{(1)}, \hat{\eta}_2^{(1)}$
Fit $\hat{\eta}_1^{(2)}, \hat{\eta}_2^{(2)}$
Evaluate $\hat{\tau}^{(1)}$
Evaluate $\hat{\tau}^{(2)}$
$\hat{\tau}^{DML}$

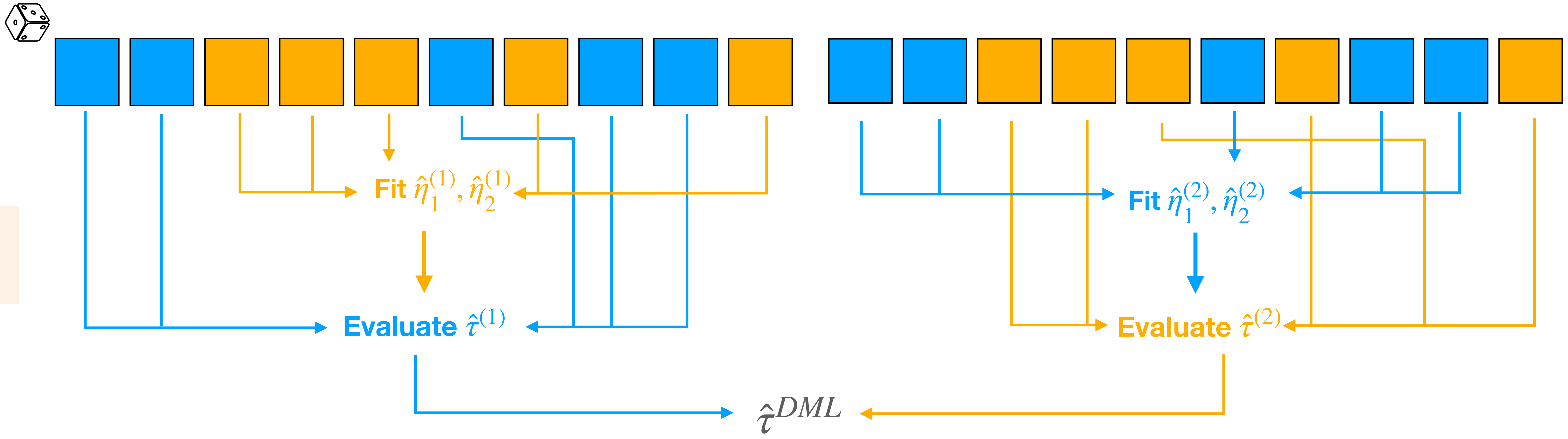
```
> set.seed(42)
> dml$fit()
      Estimate. Std. Error t value Pr(>|t|)
tau   -0.1      0.035       -2.86  0.004 **
```

```
> set.seed(43)
> dml$fit()
      Estimate. Std. Error t value Pr(>|t|)
tau   -0.06     0.035       -1.71  0.08 .
```

Problem 1. Conditional variability due to data splitting

Bill



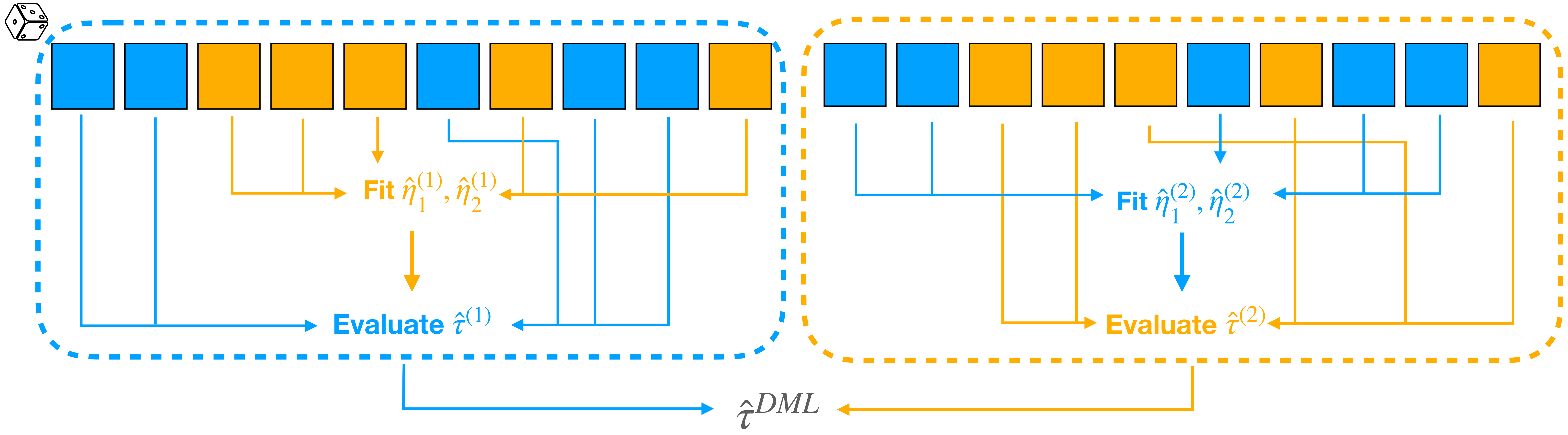

Fit $\hat{\eta}_1^{(1)}, \hat{\eta}_2^{(1)}$
Evaluate $\hat{\tau}^{(1)}$
Fit $\hat{\eta}_1^{(2)}, \hat{\eta}_2^{(2)}$
Evaluate $\hat{\tau}^{(2)}$
$\hat{\tau}^{DML}$

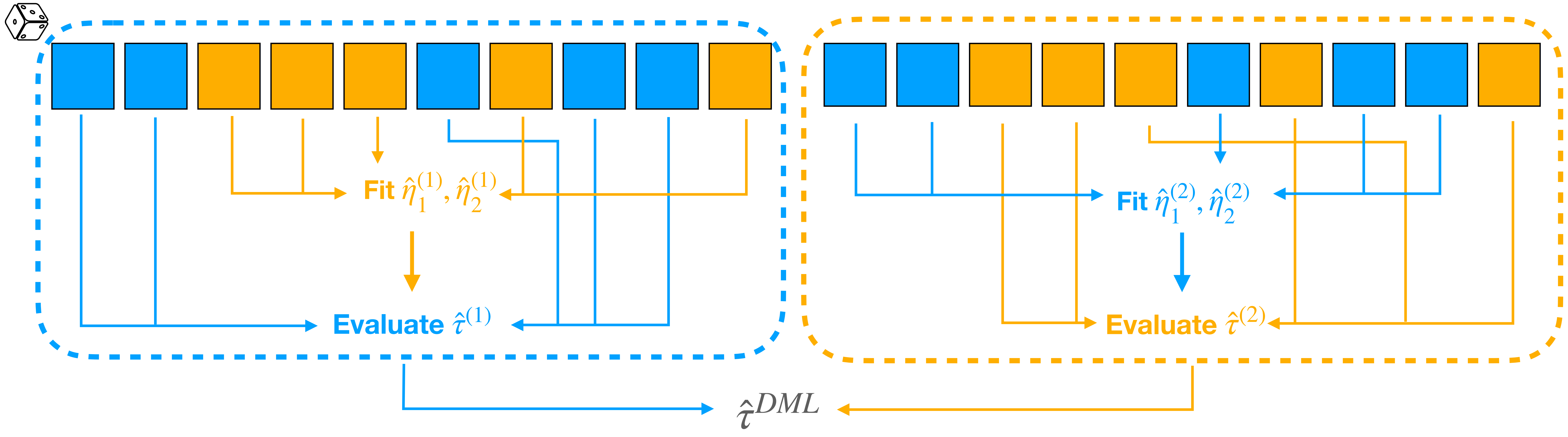
```
> set.seed(42)
> dml$fit()
     Estimate. Std. Error t value Pr(>|t|)
tau  -0.1      0.035      -2.86   0.004 **

> set.seed(43)
> dml$fit()
     Estimate. Std. Error t value Pr(>|t|)
tau  -0.06     0.035      -1.71   0.08 .
```

Problem 1. Conditional variability due to data splitting

Problem 2. DML Std. Error tends to be too small

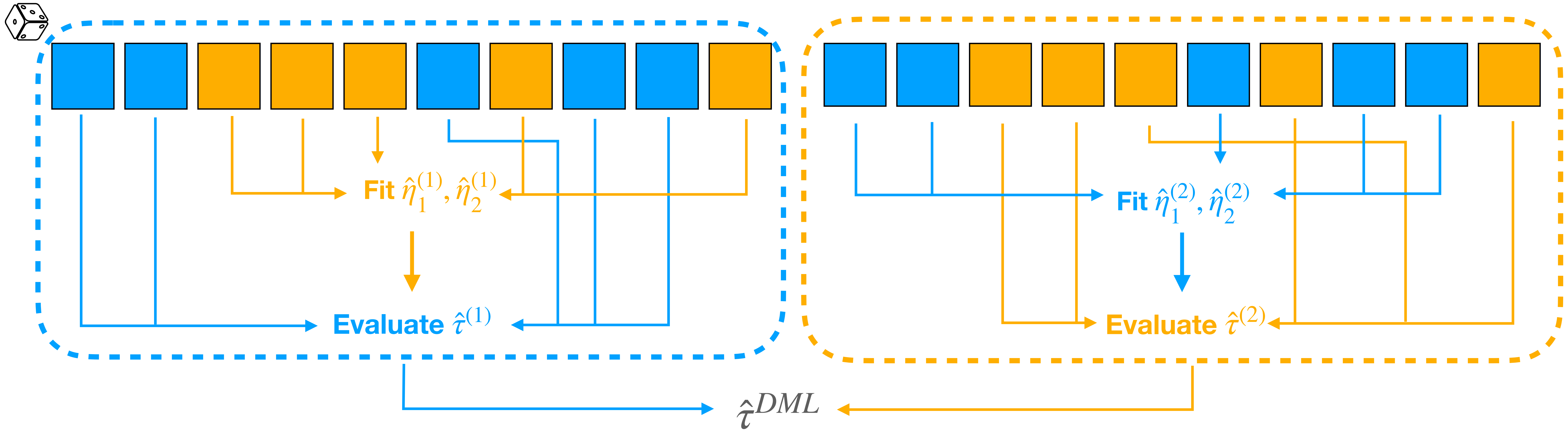
👉 It ignores cross-fold correlation


Fit $\hat{\eta}_1^{(1)}, \hat{\eta}_2^{(1)}$
Evaluate $\hat{\tau}^{(1)}$
Fit $\hat{\eta}_1^{(2)}, \hat{\eta}_2^{(2)}$
Evaluate $\hat{\tau}^{(2)}$
$\hat{\tau}^{DML}$


Fit $\hat{\eta}_1^{(1)}, \hat{\eta}_2^{(1)}$
Evaluate $\hat{\tau}^{(1)}$
Fit $\hat{\eta}_1^{(2)}, \hat{\eta}_2^{(2)}$
Evaluate $\hat{\tau}^{(2)}$
$\hat{\tau}^{DML}$

💡 Each fold defines a "single-split" statistic $T_n^{(1)} := \dfrac{\sqrt{n/2}(\hat{\tau}_1^{(1)} - \tau)}{\sigma} \to_d \mathcal{N}(0,1)$

$$T_n^{(2)} := \frac{\sqrt{n/2}(\hat{\tau}^{(2)} - \tau)}{\sigma} \to_d \mathcal{N}(0,1)$$



Fit $\hat{\eta}_1^{(1)}, \hat{\eta}_2^{(1)}$

Evaluate $\hat{\tau}^{(1)}$

Fit $\hat{\eta}_1^{(2)}, \hat{\eta}_2^{(2)}$

Evaluate $\hat{\tau}^{(2)}$

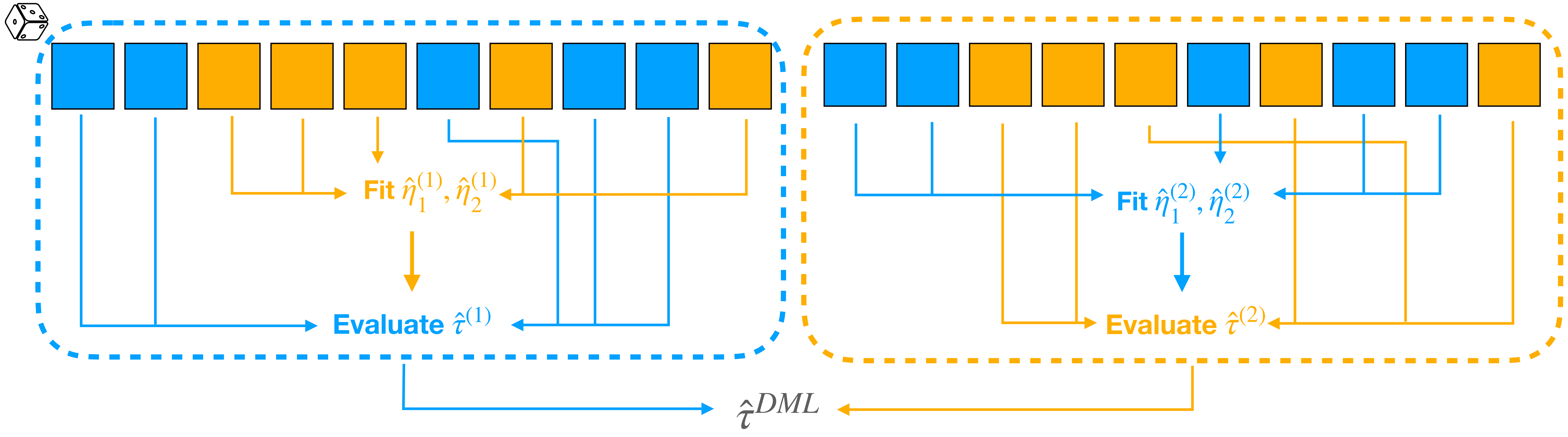
$\hat{\tau}^{DML}$

💡 Each fold defines a “single-split” statistic $T_n^{(1)} := \dfrac{\sqrt{n/2}(\hat{\tau}_1^{(1)} - \tau)}{\sigma} \to_d \mathcal{N}(0,1)$ $\quad T_n^{(2)} := \dfrac{\sqrt{n/2}(\hat{\tau}^{(2)} - \tau)}{\sigma} \to_d \mathcal{N}(0,1)$

For $\hat{\tau}_{\text{DML}} := (\hat{\tau}^{(1)} + \hat{\tau}^{(2)})/2$,

DML CLT: $$\frac{\sqrt{n}(\hat{\tau}_{\text{DML}} - \tau)}{\sigma} = \frac{1}{\sqrt{2}}(T_n^{(1)} + T_n^{(2)}) \to \mathcal{N}(0,1)\,.$$

🤔 Under conditions required by DML, between-fold correlation $\rho \to 0$.


Fit $\hat\eta_1^{(1)}, \hat\eta_2^{(1)}$
Evaluate $\hat\tau^{(1)}$
Fit $\hat\eta_1^{(2)}, \hat\eta_2^{(2)}$
Evaluate $\hat\tau^{(2)}$
$\hat\tau^{DML}$

💡 Each fold defines a “single-split” statistic $T_n^{(1)} := \dfrac{\sqrt{n/2}(\hat\tau_1^{(1)} - \tau)}{\sigma} \to_d \mathcal{N}(0,1)$ $\quad T_n^{(2)} := \dfrac{\sqrt{n/2}(\hat\tau^{(2)} - \tau)}{\sigma} \to_d \mathcal{N}(0,1)$

For $\hat\tau_{\text{DML}} := (\hat\tau^{(1)} + \hat\tau^{(2)})/2$,

DML CLT: $$\frac{\sqrt{n}(\hat\tau_{\text{DML}} - \tau)}{\sigma} = \frac{1}{\sqrt{2}}(T_n^{(1)} + T_n^{(2)}) \to \mathcal{N}(0,1)\,.$$

🤔 Under conditions required by DML, between-fold correlation $\rho \to 0$.

⚠️ For finite sample, $\rho > 0$.

	Std. Error
DML	$\sigma/\sqrt{n}$
Actual	$\sigma\sqrt{1+\rho(L-1)}\;/\sqrt{n}$

Improved DML inference

Improved DML inference

💡 Rank-transformed subsampling **automatically accounts for ρ.**

Improved DML inference

💡 Rank-transformed subsampling **automatically accounts for ρ.**

👉 Can be performed without knowing τ or σ:

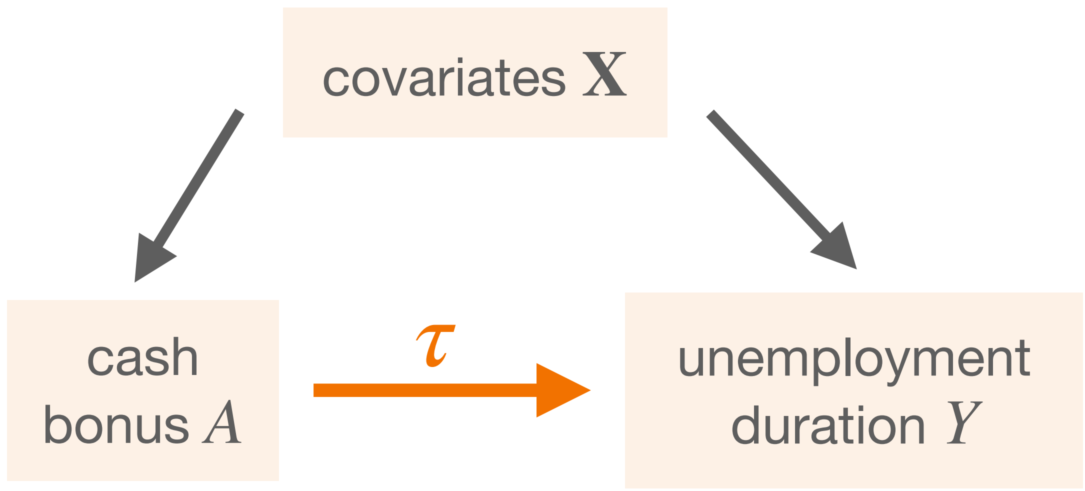
$$\text{rank}\left(T_m^{(l)}\right) = \text{rank}\left\{\frac{\sqrt{m/2}(\hat{\tau}_m^{(l)} - \tau)}{\sigma}\right\} = \text{rank}\left(\hat{\tau}_m^{(l)}\right)$$

Improved DML inference

💡 Rank-transformed subsampling **automatically accounts for ρ.**

👉 Can be performed without knowing τ or σ:

$$\text{rank}\left(T_m^{(l)}\right) = \text{rank}\left\{\frac{\sqrt{m/2}(\hat{\tau}_m^{(l)} - \tau)}{\sigma}\right\} = \text{rank}\left(\hat{\tau}_m^{(l)}\right)$$



covariates $\mathbf{X}$

cash bonus A

τ

unemployment duration Y

Table 1: Coverage of nominal 95% confidence intervals

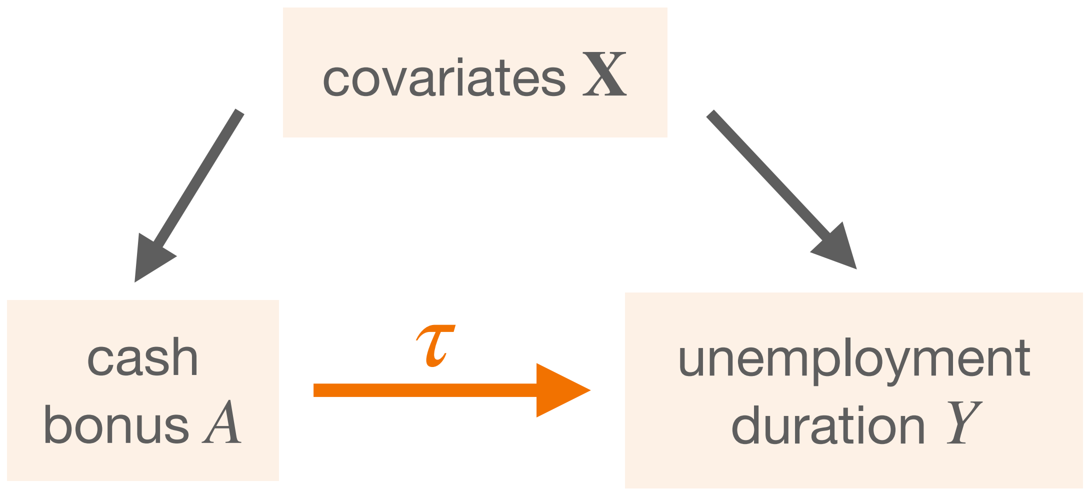
method	$n = 500$		$n = 1000$		$n = 2000$	
	$L = 2$	$L = 5$	$L = 2$	$L = 5$	$L = 2$	$L = 5$
$\rho(L-1)$	0.46	0.31	0.36	0.18	0.25	0.14
Corrected	0.94	0.93	0.95	0.95	0.96	0.95
DML	0.86	0.88	0.88	0.92	0.91	0.92

Improved DML inference

💡 Rank-transformed subsampling **automatically accounts for ρ.**

👉 Can be performed without knowing τ or σ:

$$\text{rank}\left(T_m^{(l)}\right) = \text{rank}\left\{\frac{\sqrt{m/2}(\hat{\tau}_m^{(l)} - \tau)}{\sigma}\right\} = \text{rank}\left(\hat{\tau}_m^{(l)}\right)$$



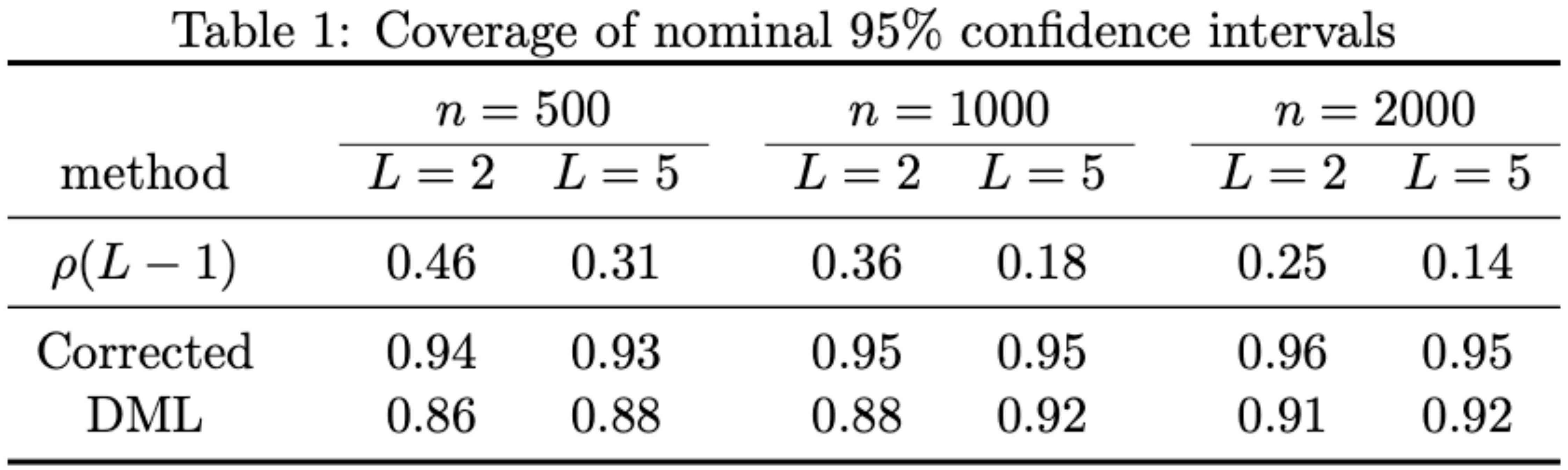


Table 1: Coverage of nominal 95% confidence intervals

method	$n = 500$		$n = 1000$		$n = 2000$	
	$L = 2$	$L = 5$	$L = 2$	$L = 5$	$L = 2$	$L = 5$
$\rho(L-1)$	0.46	0.31	0.36	0.18	0.25	0.14
Corrected	0.94	0.93	0.95	0.95	0.96	0.95
DML	0.86	0.88	0.88	0.92	0.91	0.92

✅ Calibrated CI’s by accounting for correlation.

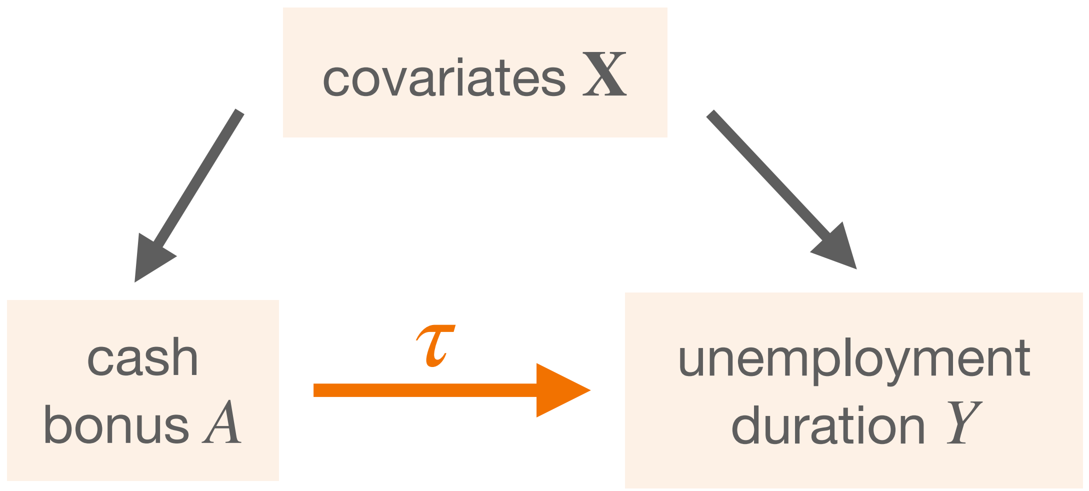
✅ Improved replicability by averaging over data splits.

Improved DML inference

💡 Rank-transformed subsampling **automatically accounts for ρ.**

👉 Can be performed without knowing τ or σ:

$$\text{rank}\left(T_m^{(l)}\right) = \text{rank}\left\{\frac{\sqrt{m/2}(\hat{\tau}_m^{(l)} - \tau)}{\sigma}\right\} = \text{rank}\left(\hat{\tau}_m^{(l)}\right)$$



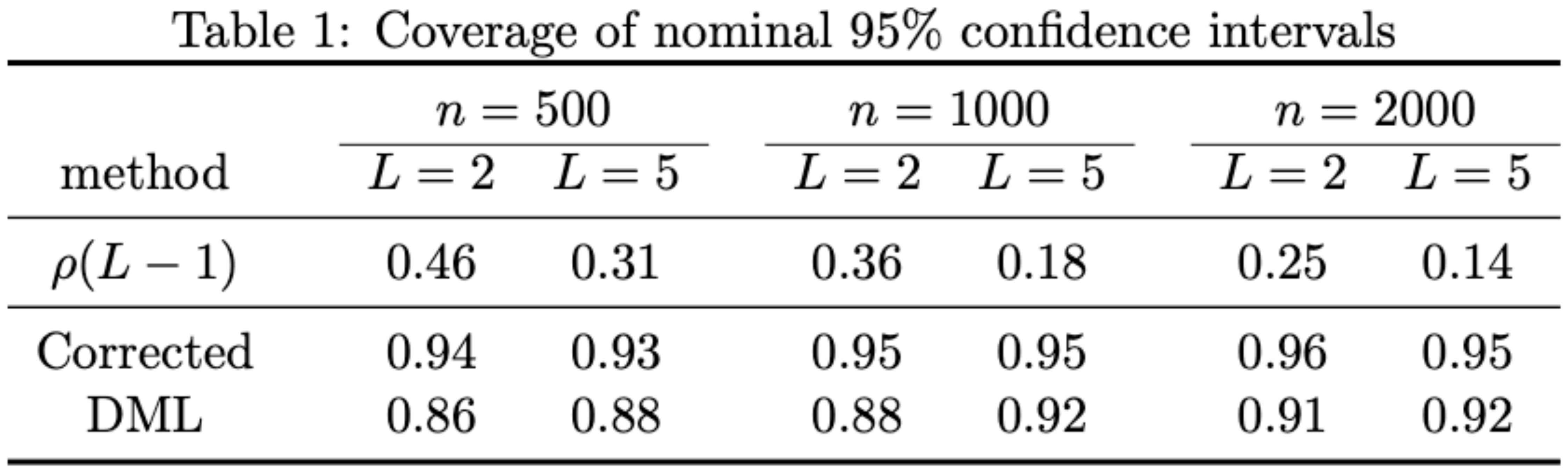


Table 1: Coverage of nominal 95% confidence intervals

	$n = 500$		$n = 1000$		$n = 2000$	
method	$L = 2$	$L = 5$	$L = 2$	$L = 5$	$L = 2$	$L = 5$
$\rho(L-1)$	0.46	0.31	0.36	0.18	0.25	0.14
Corrected	0.94	0.93	0.95	0.95	0.96	0.95
DML	0.86	0.88	0.88	0.92	0.91	0.92

Relieved Bill

✅ Calibrated CI's by accounting for correlation.

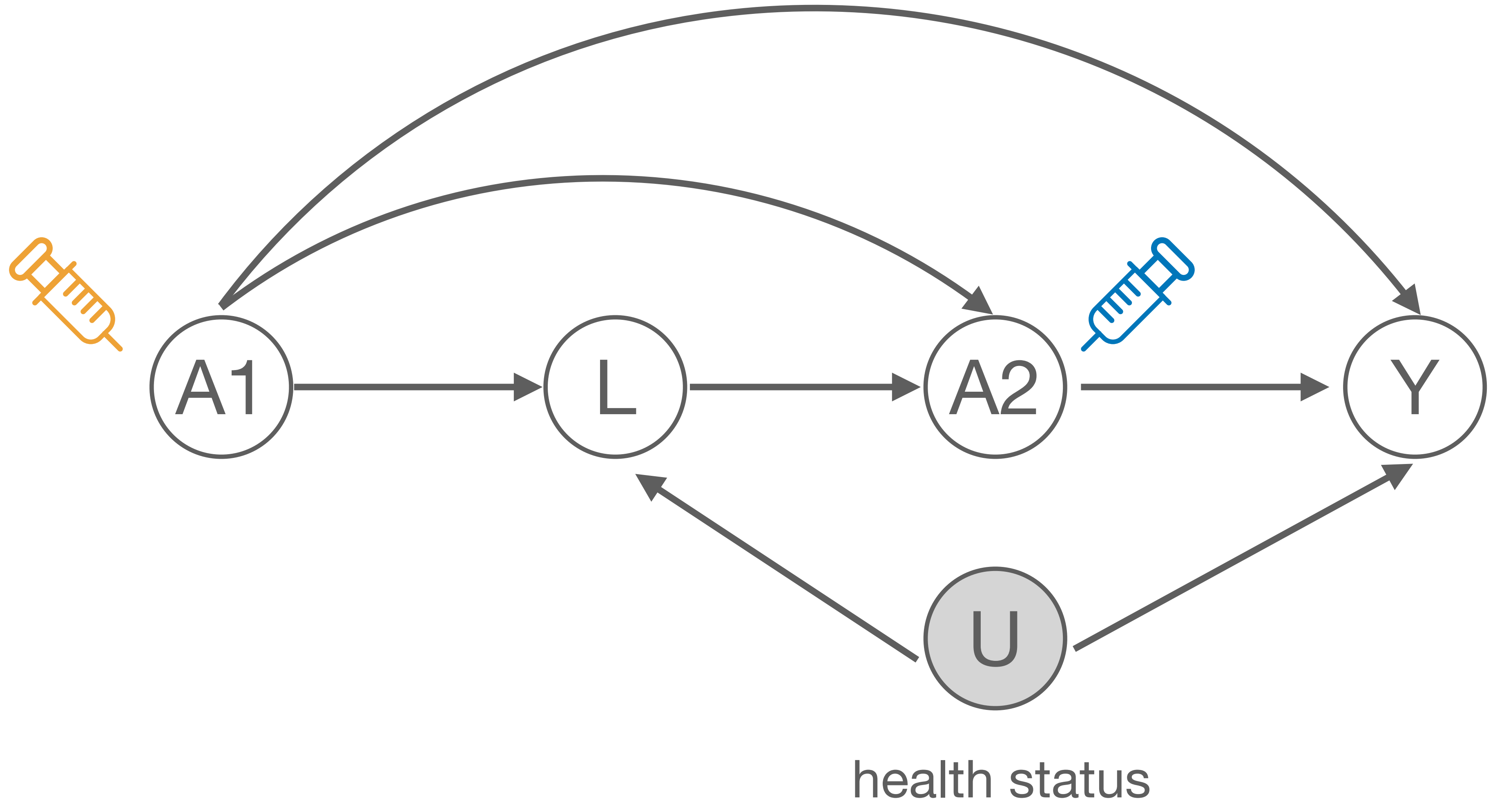
✅ Improved replicability by averaging over data splits.

Outline

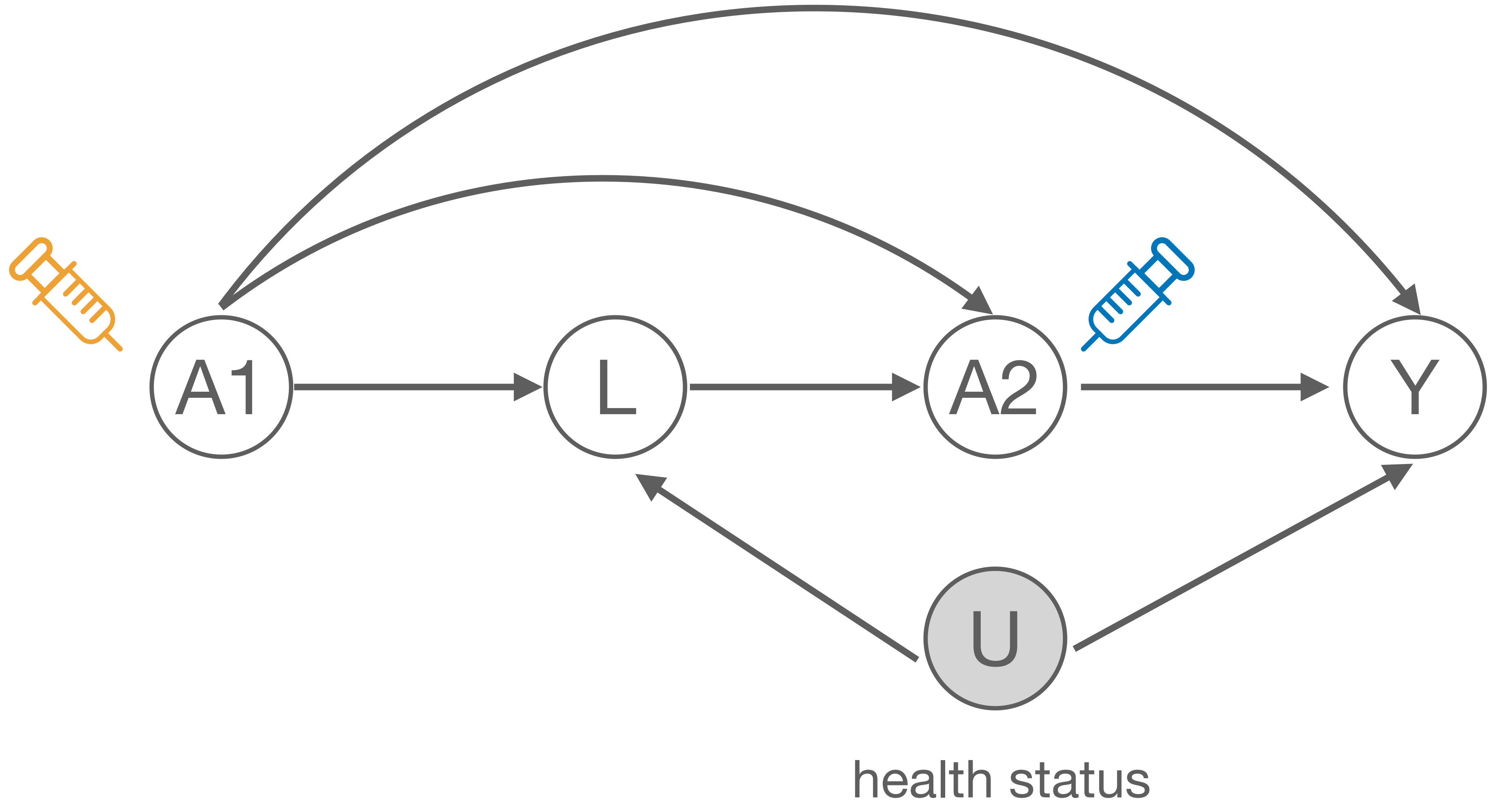
- Setup and main challenge
- Method: Rank-transformed subsampling
- Applications
    - Hunt and test
    - Improving double machine learning
    - **Testing no direct effect in a sequentially randomized trial**
- Future directions

Sequentially randomized trial

Sequentially randomized trial

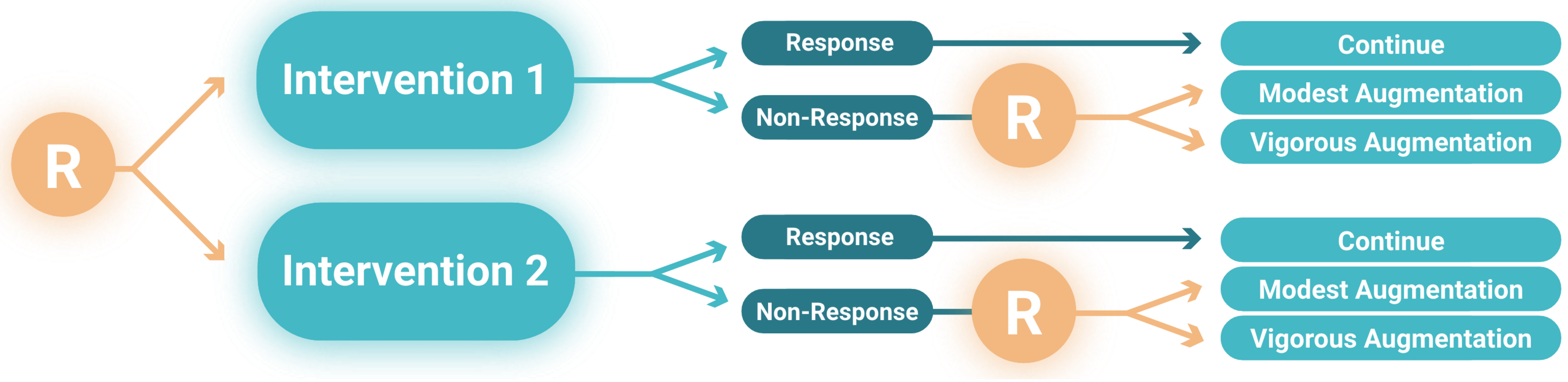


Sequentially randomized trial



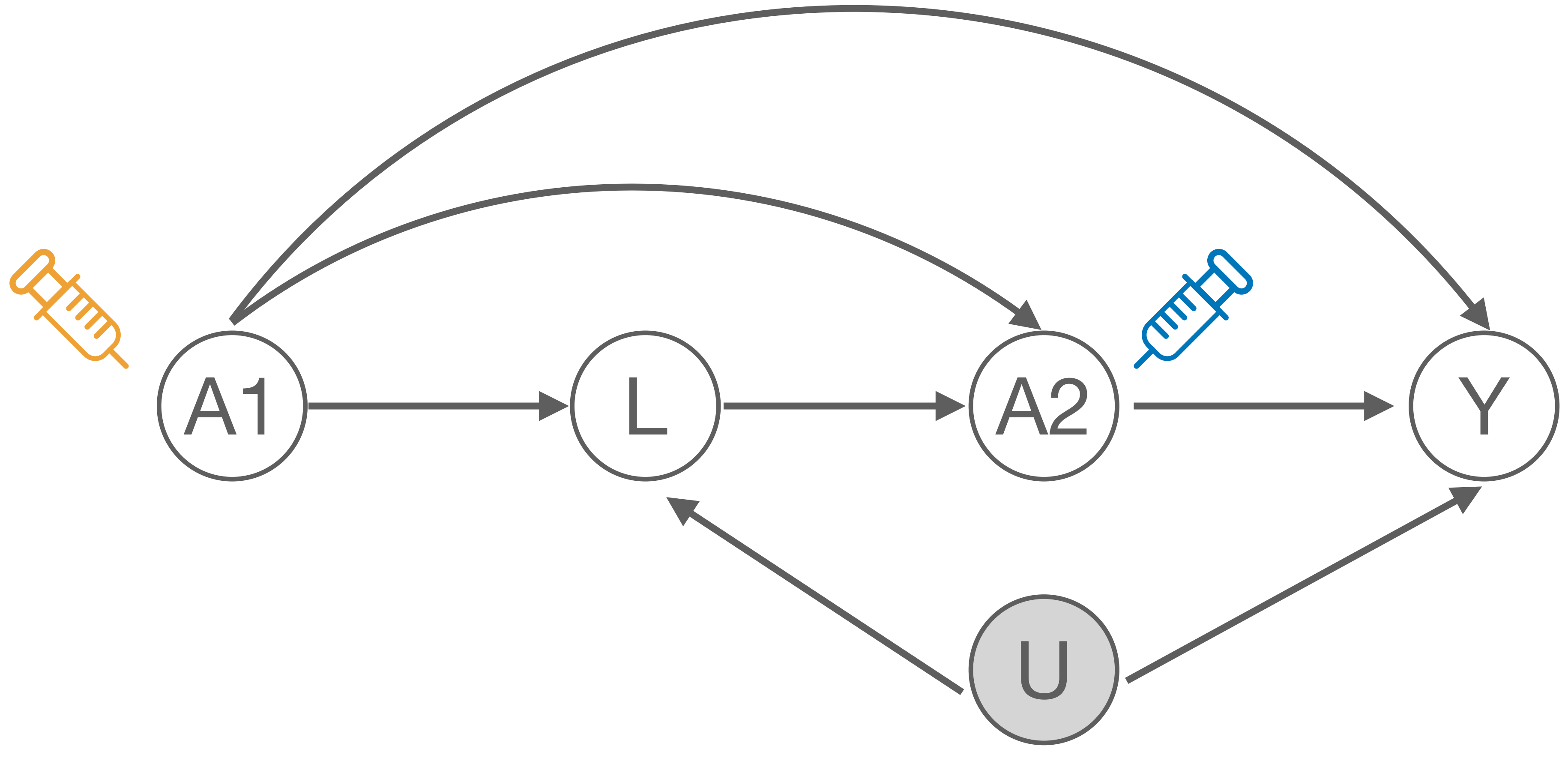
- SMART trials

(Murphy, 2005; Murphy et al., 2006)



© d3c.isr.umich.edu

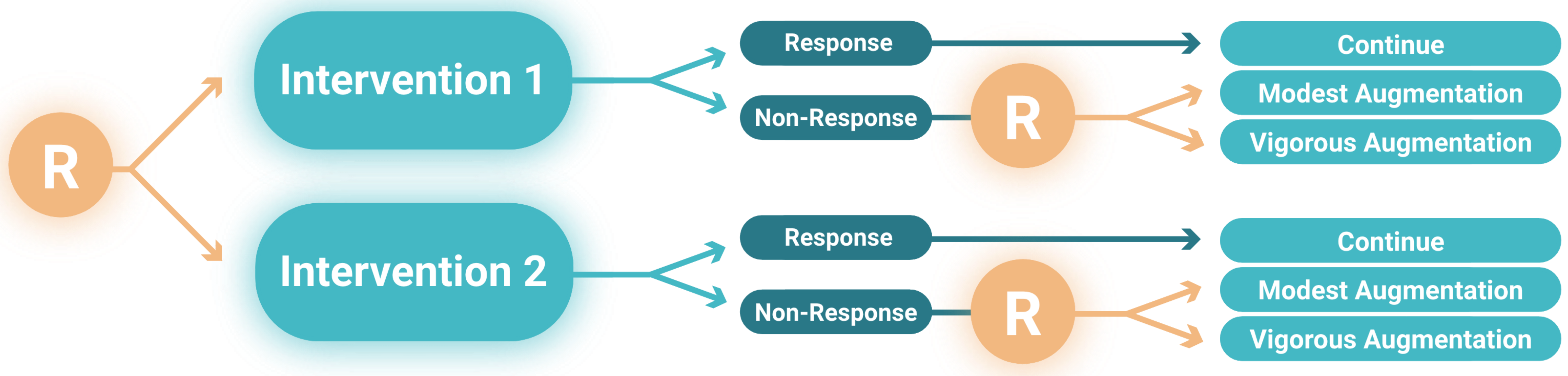
Sequentially randomized trial



health status

- SMART trials

(Murphy, 2005; Murphy et al., 2006)



R

Intervention 1

Response

Non-Response

R

Continue

Modest Augmentation

Vigorous Augmentation

Intervention 2

Response

Non-Response

R

Continue

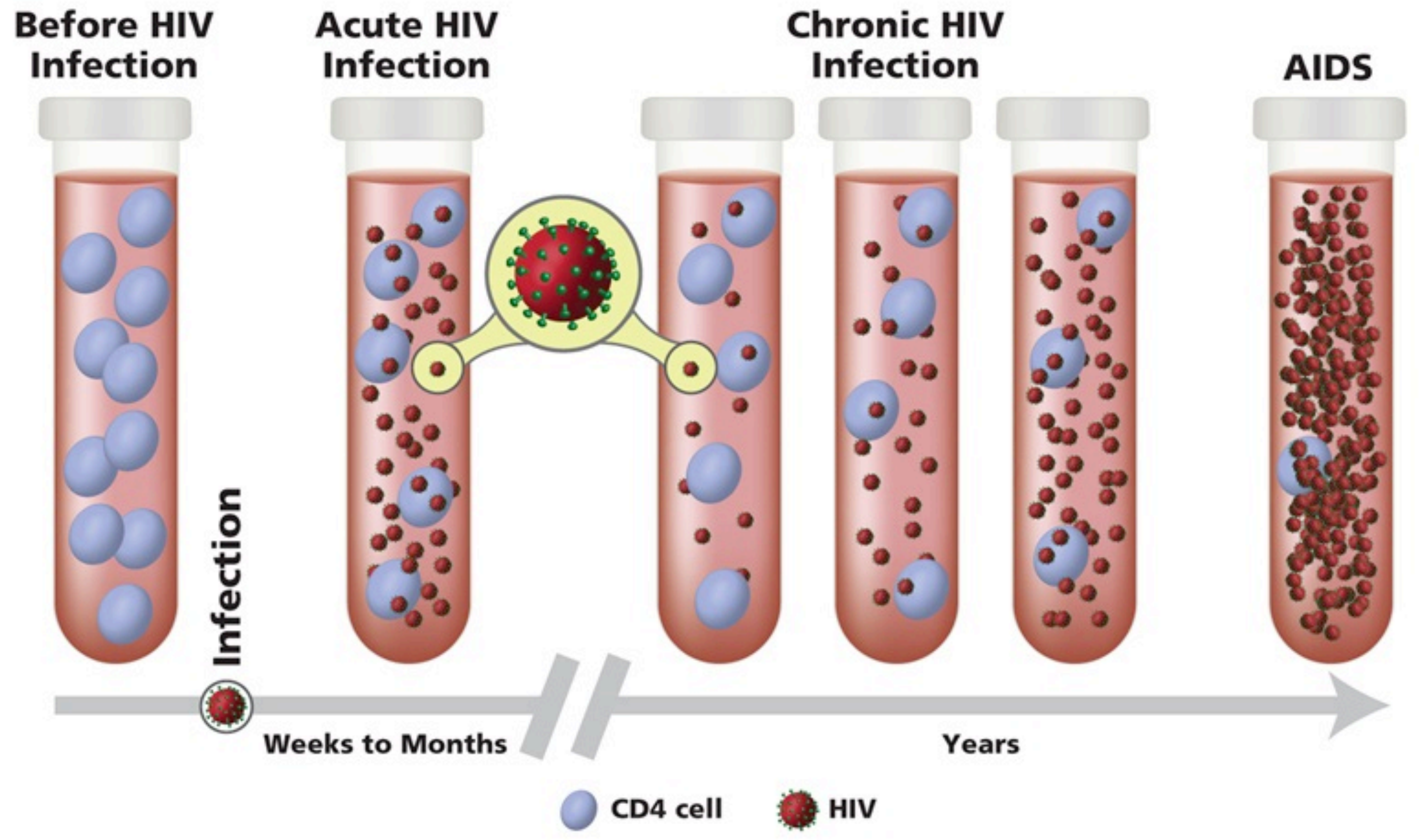
Modest Augmentation

Vigorous Augmentation

© d3c.isr.umich.edu

- Observational / follow-up studies

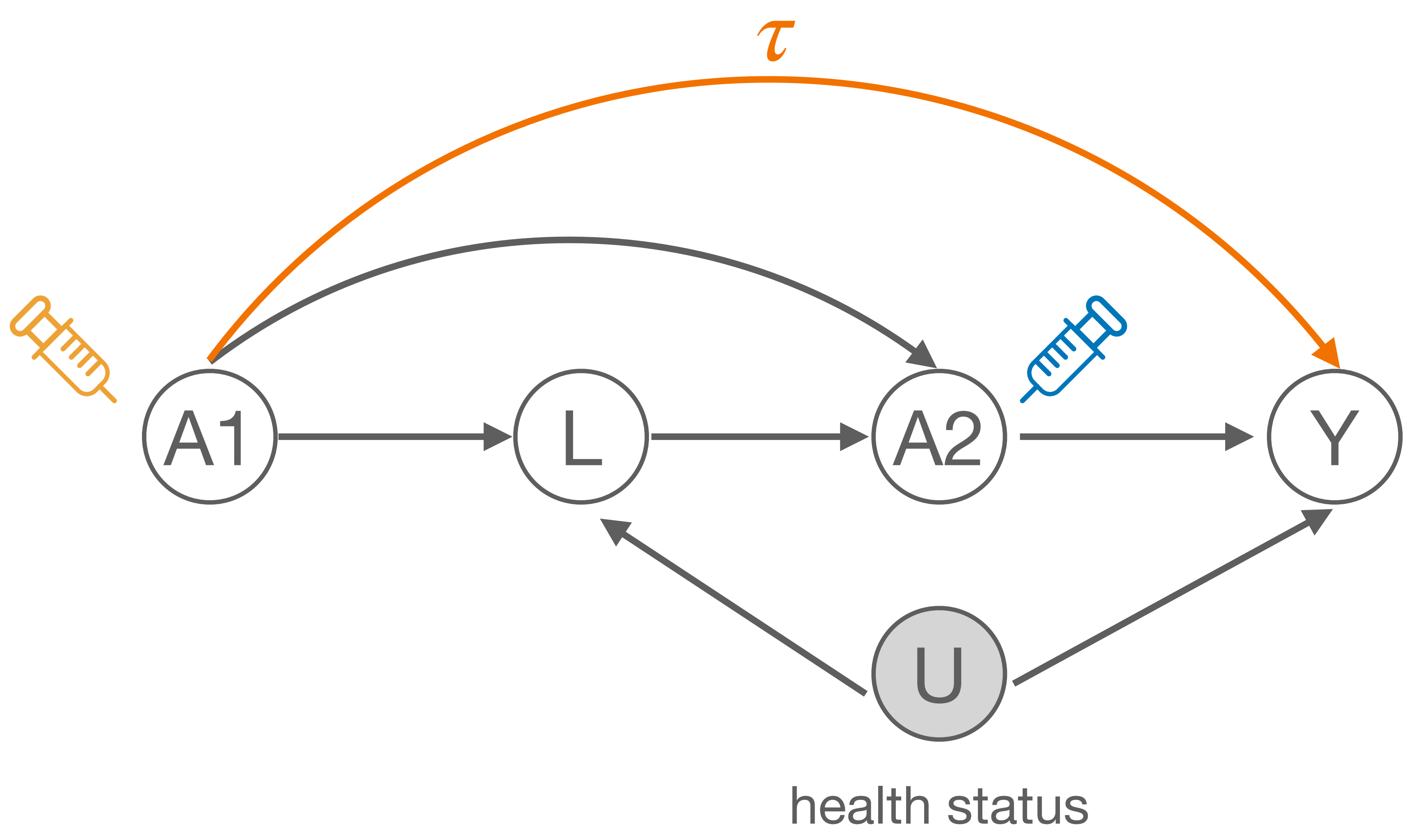
HIV studies: A_1, A_2: antiretroviral therapy; L, Y: CD4 cell counts



© NIH

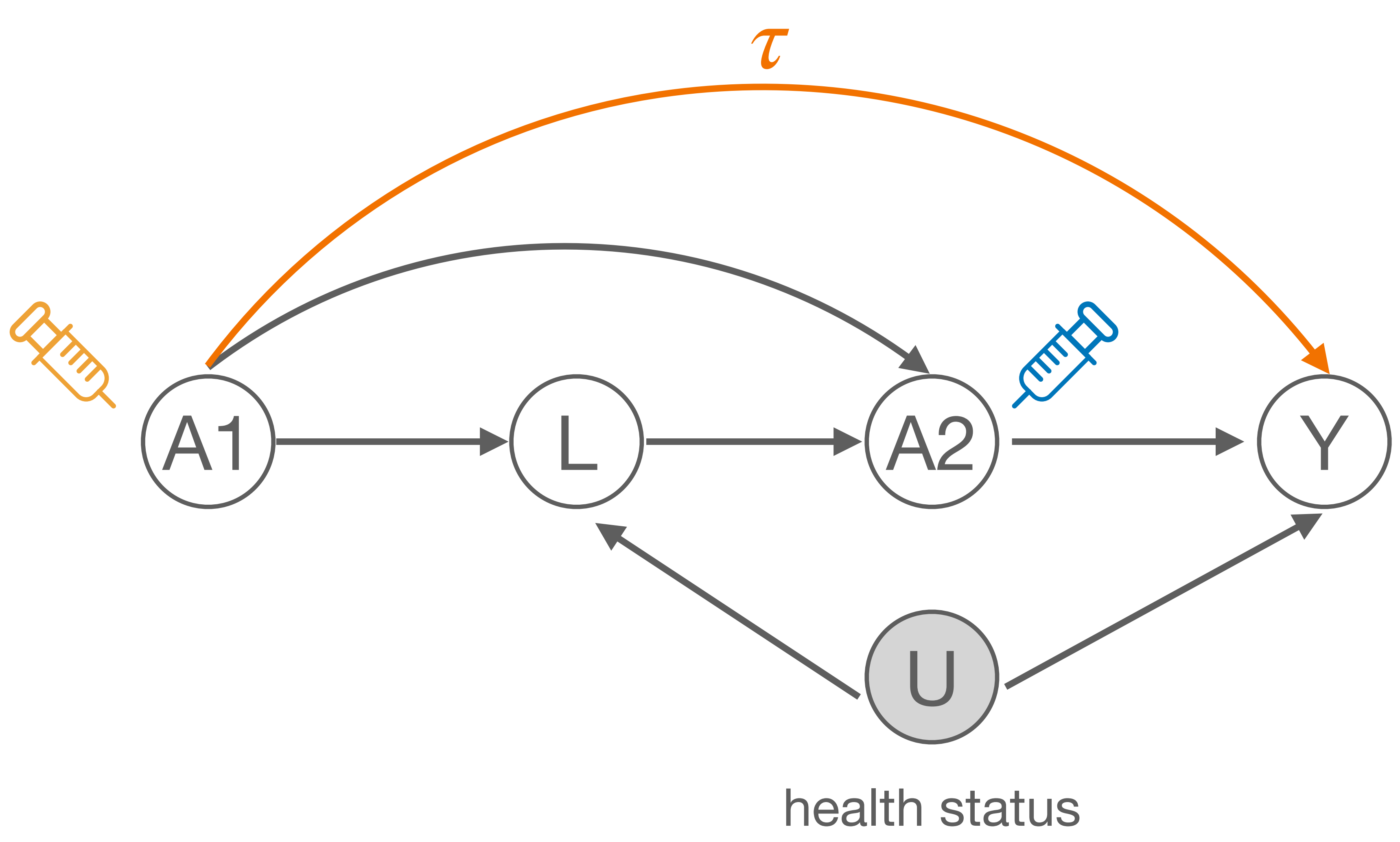
Sharp null of no direct effect

Sharp null of no direct effect



τ: the direct effect of A_1 on Y (i.e., not through A_2).

Sharp null of no direct effect

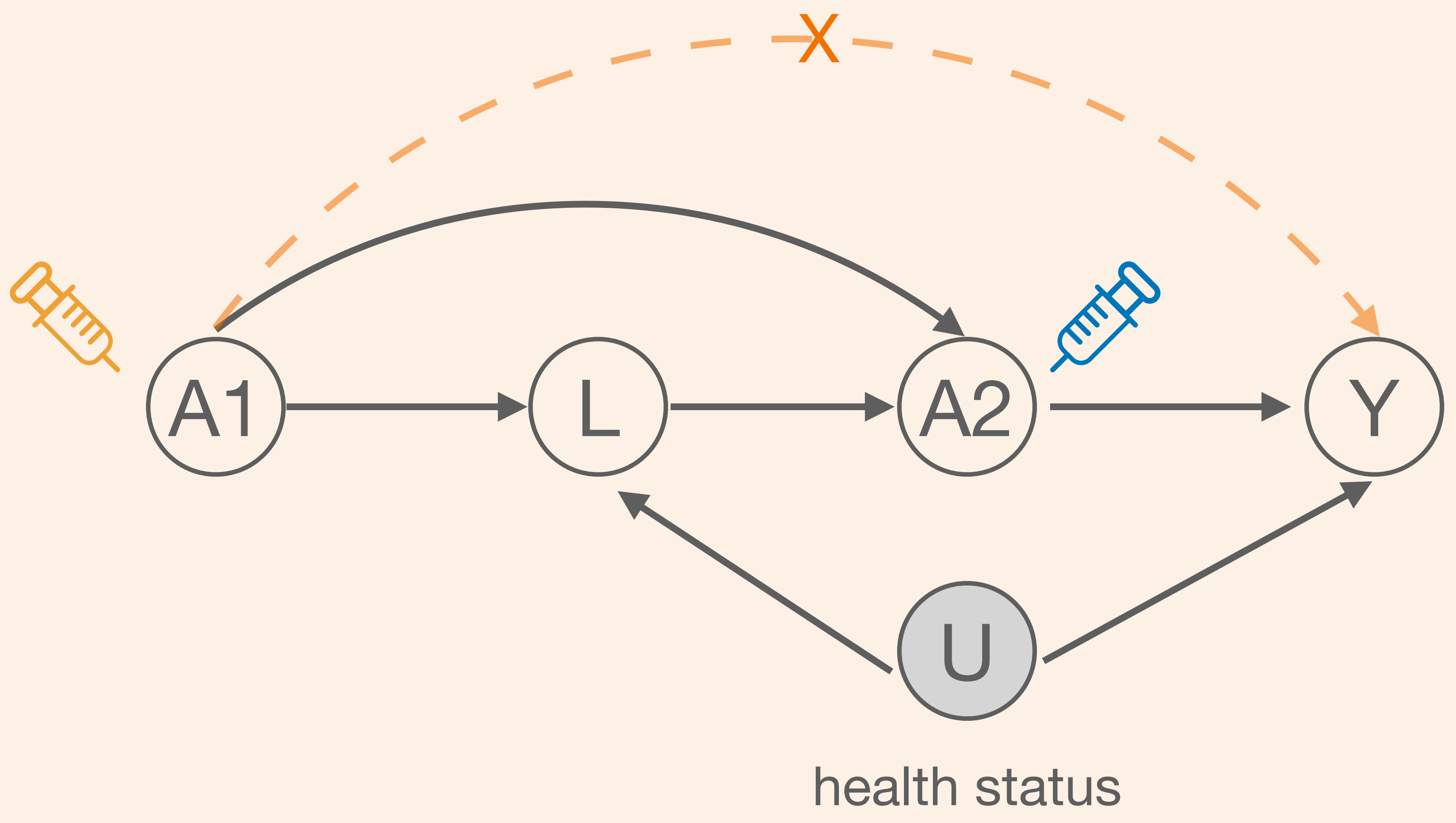


τ: the direct effect of A_1 on Y (i.e., not through A_2).

Sharp null hypothesis H_0: $\tau_i \equiv 0$ for every individual i.

* More precisely, $Y_i(1,0) - Y_i(0,0) \equiv 0$ and $Y_i(1,1) - Y_i(0,1) = 0$ for every i. $Y_i(a_1, a_2)$ is the potential outcome had subject i taken treatments (a_1, a_2).

Sharp null of no direct effect



Graph under the sharp null H_0

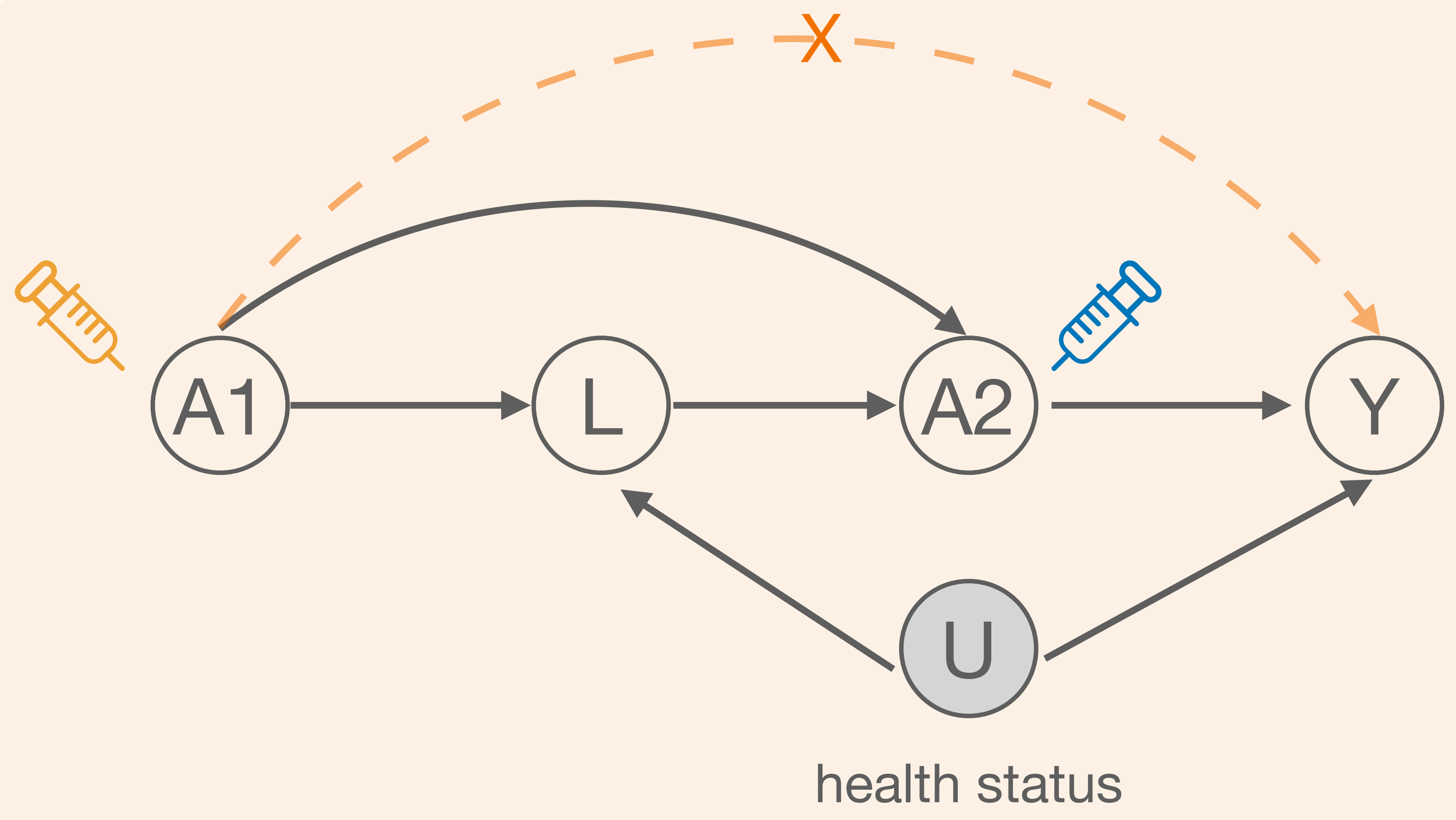
τ: the direct effect of A_1 on Y (i.e., not through A_2).

Sharp null hypothesis H_0: $\tau_i \equiv 0$ for every individual i.

* More precisely, $Y_i(1,0) - Y_i(0,0) \equiv 0$ and $Y_i(1,1) - Y_i(0,1) = 0$ for every i.

$Y_i(a_1, a_2)$ is the potential outcome had subject i taken treatments (a_1, a_2).

Sharp null of no direct effect



Graph under the sharp null H_0

τ: the direct effect of A_1 on Y (i.e., not through A_2).

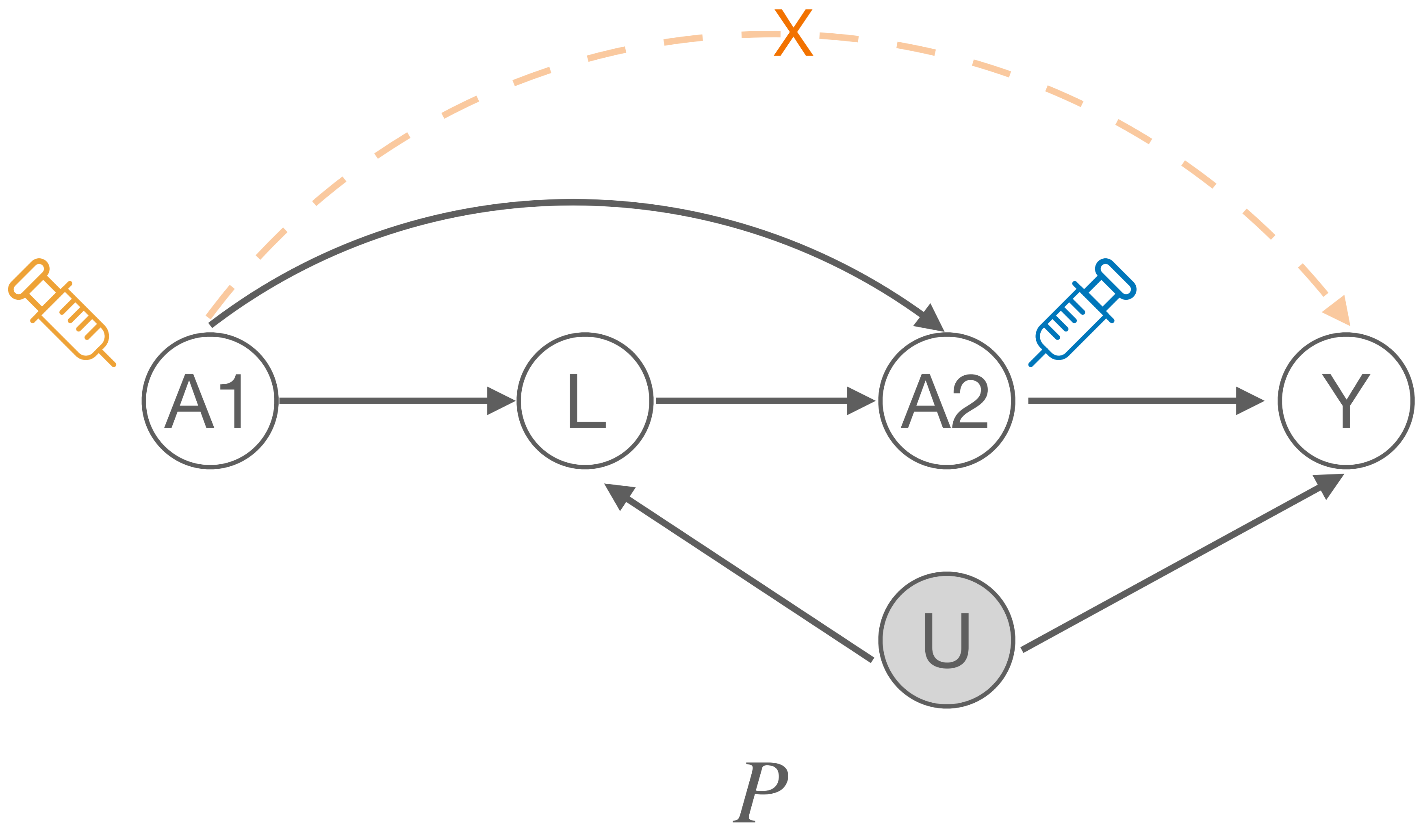
Sharp null hypothesis H_0: $\tau_i \equiv 0$ for every individual i.

* More precisely, $Y_i(1,0) - Y_i(0,0) \equiv 0$ and $Y_i(1,1) - Y_i(0,1) = 0$ for every i.
$Y_i(a_1, a_2)$ is the potential outcome had subject i taken treatments (a_1, a_2).

🤔 H_0 cannot be formulated as an independence or conditional independence.

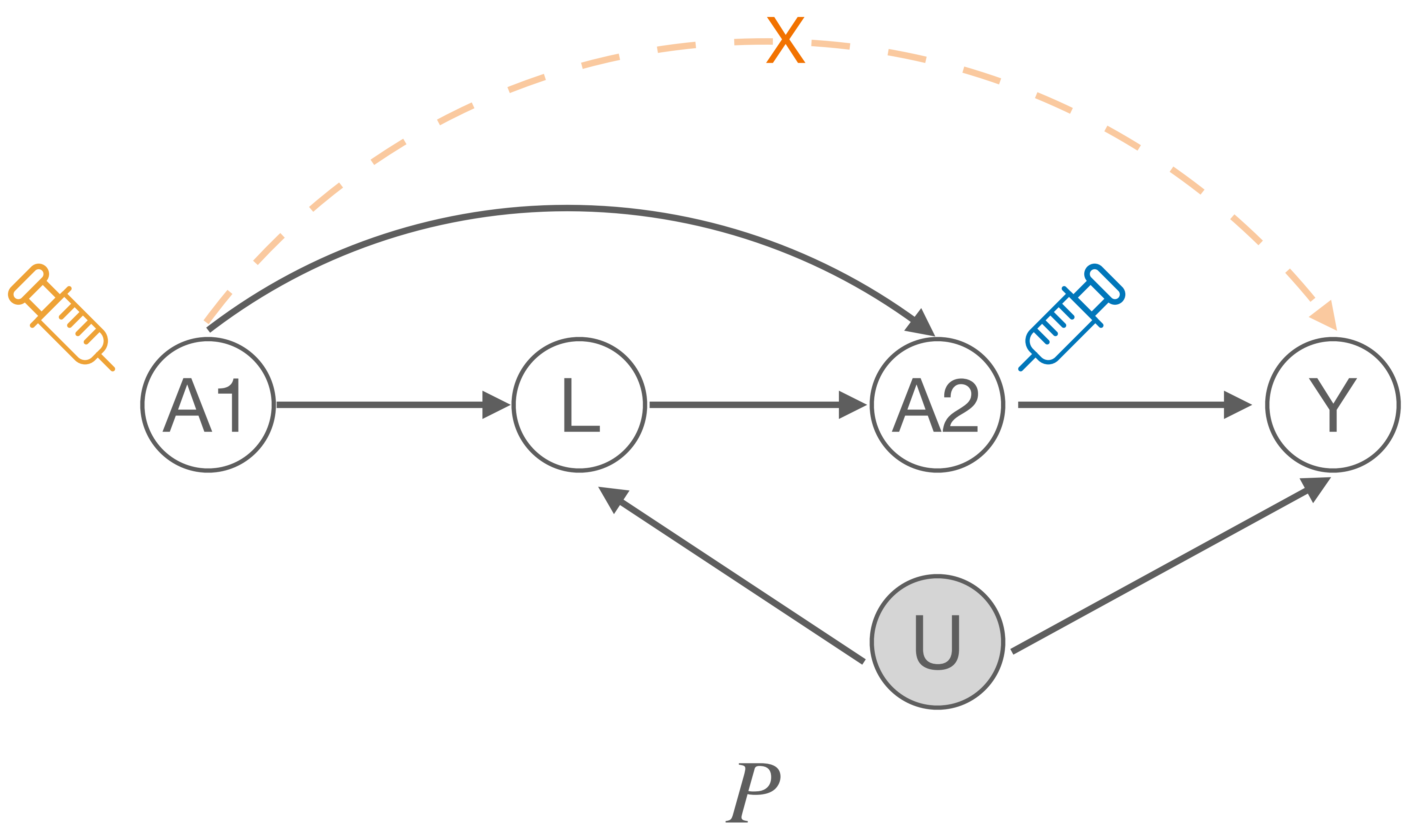
Testing the sharp null

Sequentially randomized trial under the sharp null

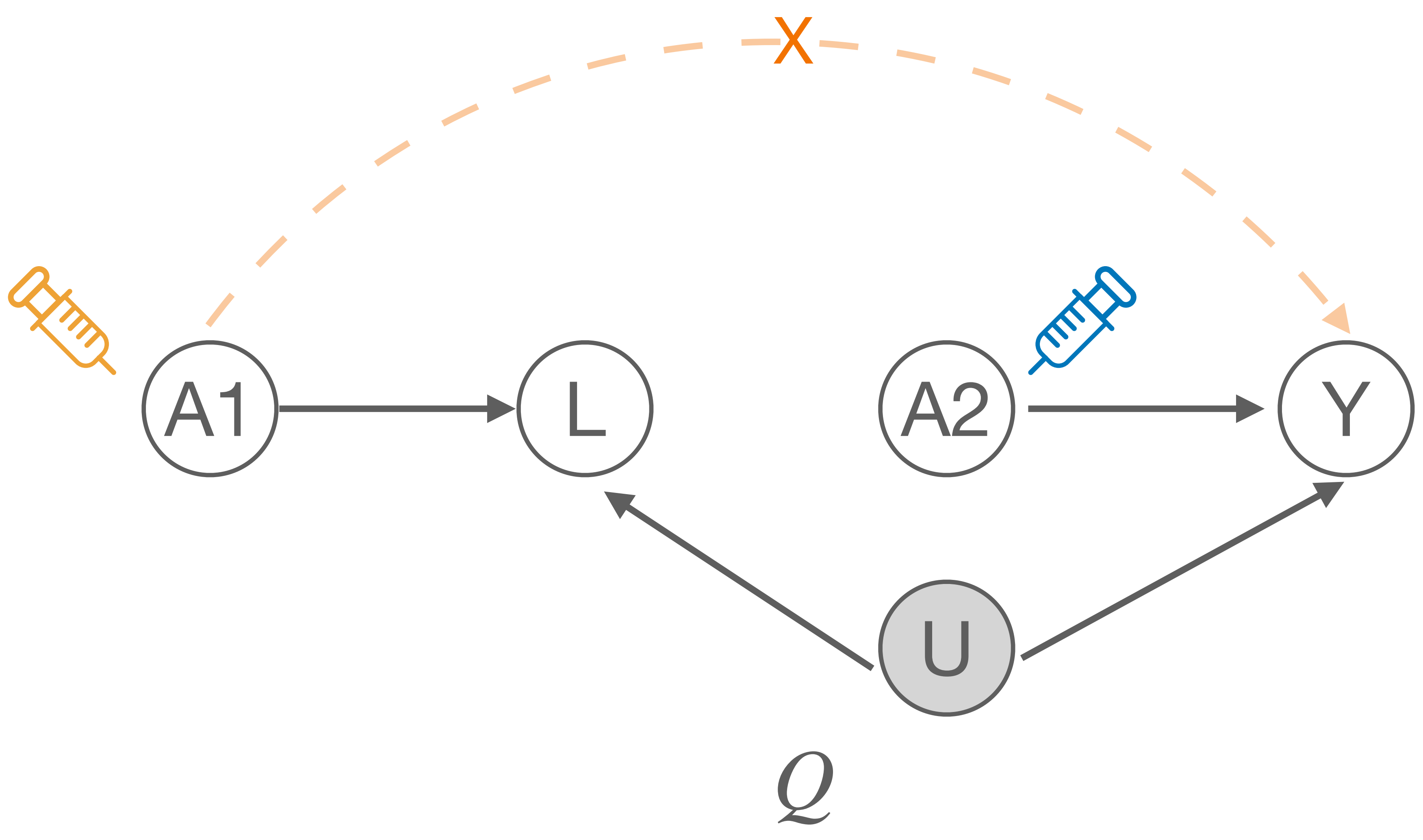


Testing the sharp null

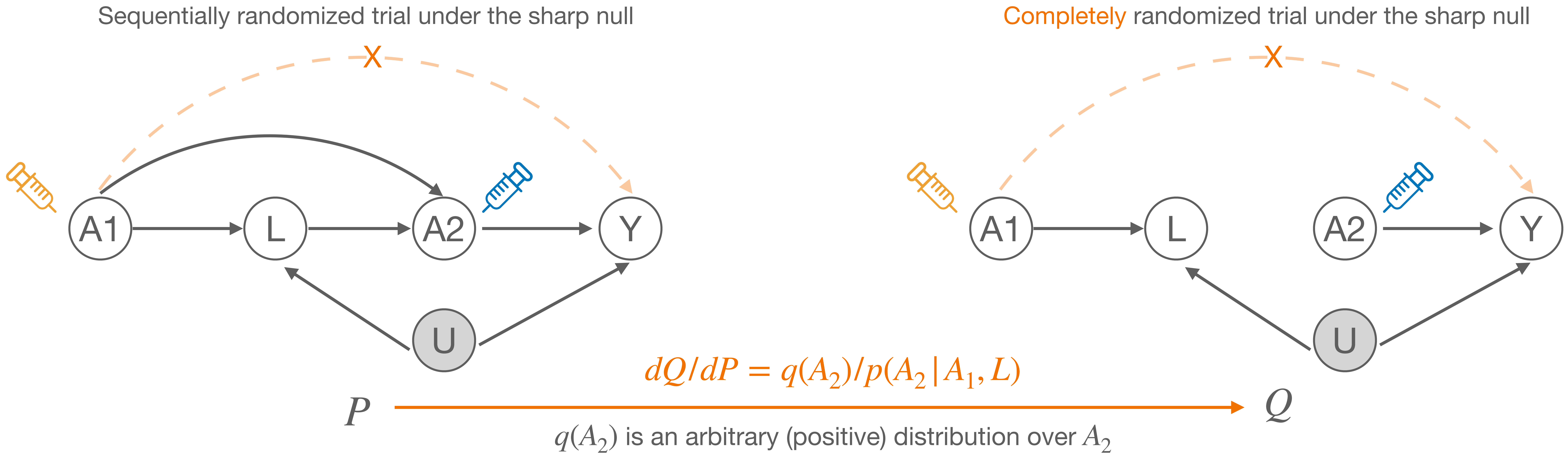
Sequentially randomized trial under the sharp null



Completely randomized trial under the sharp null



Testing the sharp null



Sequentially randomized trial under the sharp null

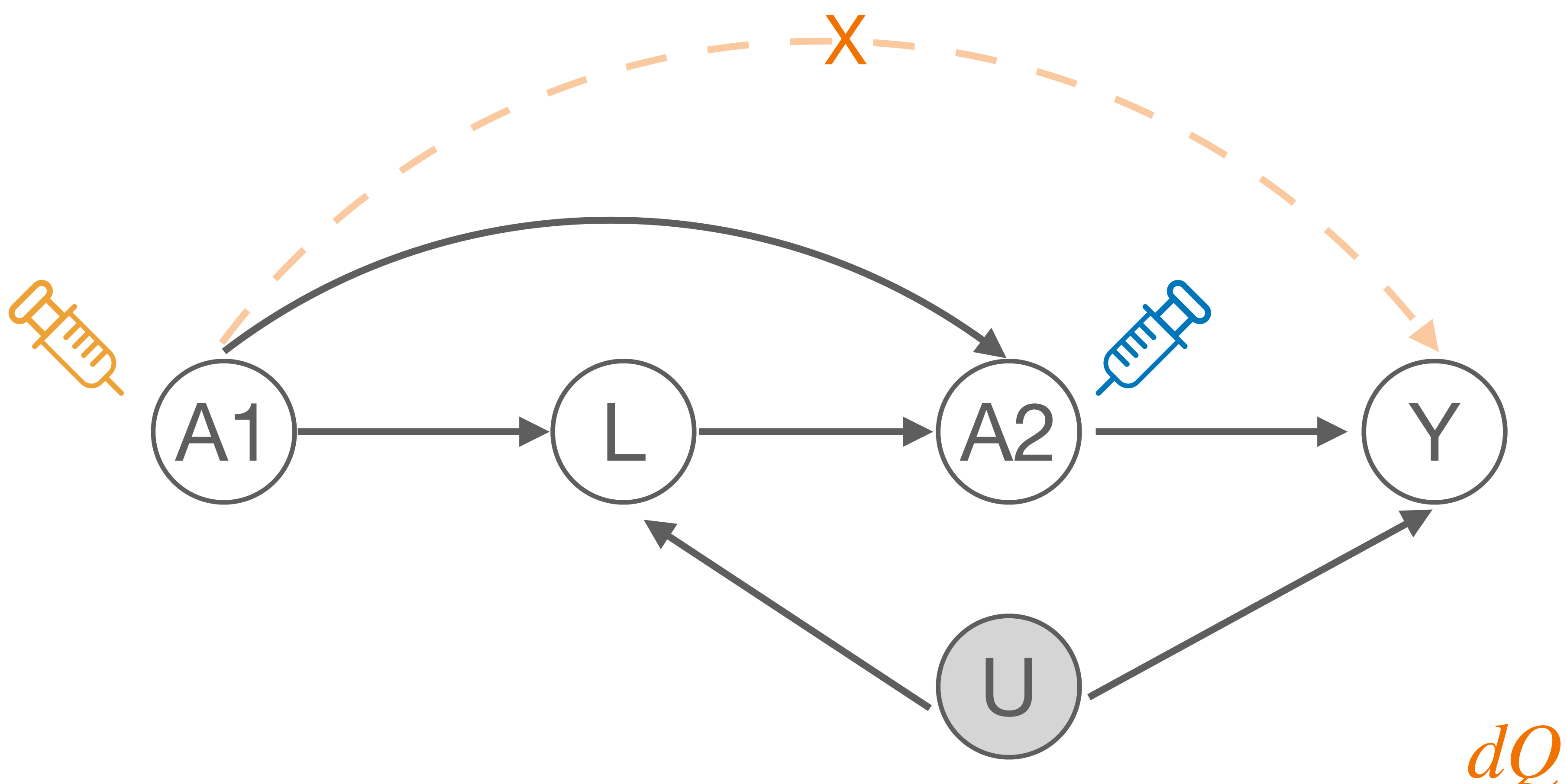
Completely randomized trial under the sharp null

$$dQ/dP = q(A_2)/p(A_2\,|\,A_1, L)$$

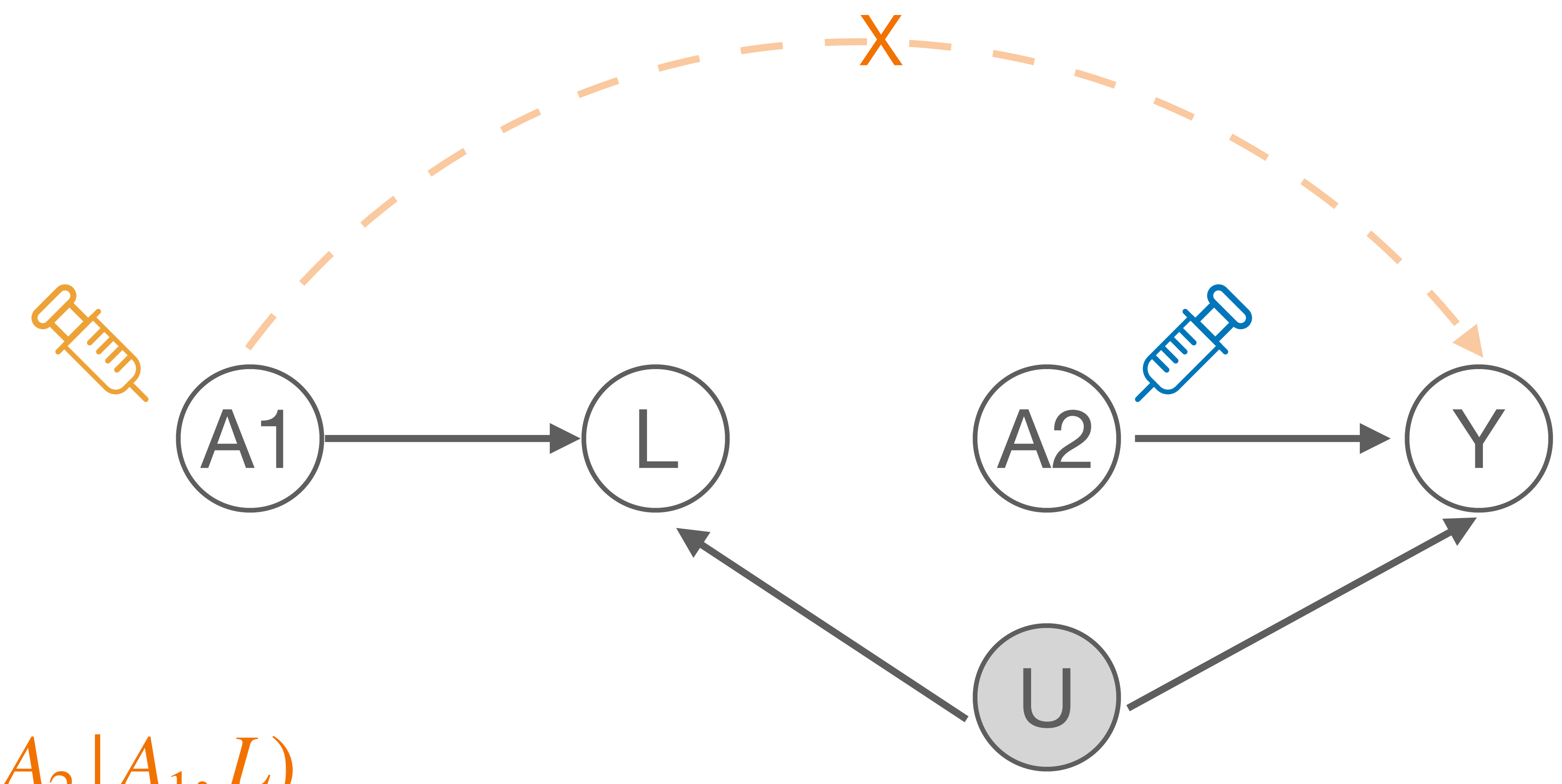
$q(A_2)$ is an arbitrary (positive) distribution over A_2

Testing the sharp null

Sequentially randomized trial under the sharp null



Completely randomized trial under the sharp null



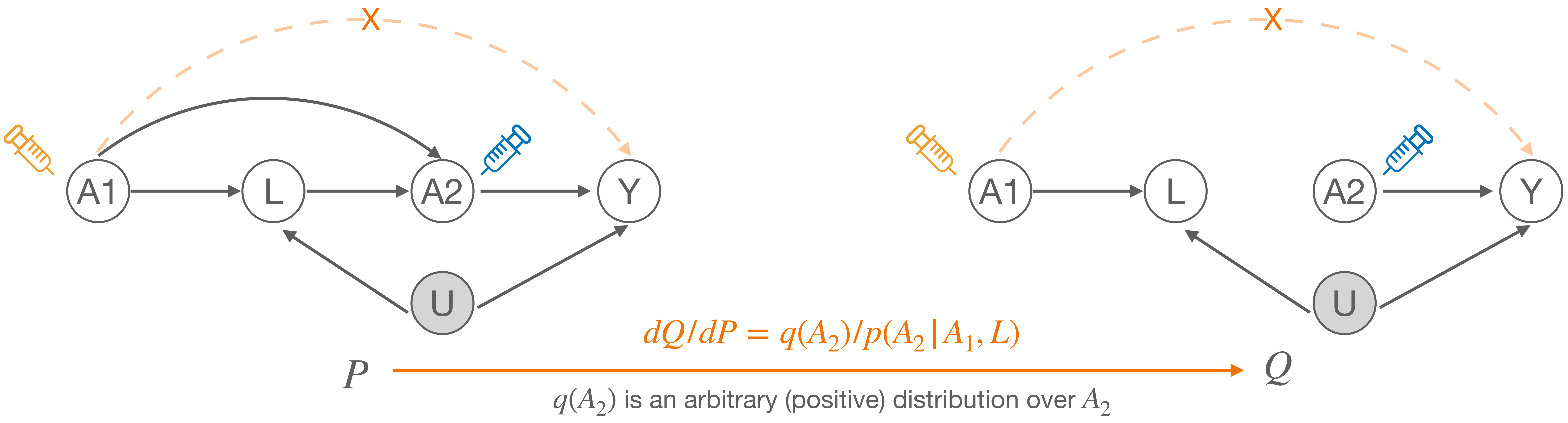
$$P \xrightarrow[q(A_2)\text{ is an arbitrary (positive) distribution over }A_2]{dQ/dP = q(A_2)/p(A_2\,|\,A_1,L)} Q$$

💡 Sharp null H_0: $A_1 \perp\!\!\!\perp Y\ (Q),\ dQ/dP = q(A_2)/p(A_2\,|\,A_1,L)$.

Testing the sharp null

Sequentially randomized trial under the sharp null

Completely randomized trial under the sharp null



$dQ/dP = q(A_2)/p(A_2 \mid A_1, L)$

$P \xrightarrow{\qquad} Q$

$q(A_2)$ is an arbitrary (positive) distribution over A_2

💡 Sharp null H_0: $A_1 \perp\!\!\!\perp Y \ (Q),\ dQ/dP = q(A_2)/p(A_2 \mid A_1, L)$.

👉 This is a generalized / “dormant” independence, aka. Verma constraint on P.

Robins (1986, 1999), Verma & Pearl (1990), Wermuth & Cox (2008), Richardson et al. (2017)

An instance of “distribution shift”.

Testing generalized (conditional) independence

Testing generalized (conditional) independence

A lot of recent progress in independence / conditional independence testing.

Independence: kernel embedding (Gretton et al., 2005, 2007), rank correlation coefficients (Bergsma & Dassios, 2014; Drton et al., 2020; Shi et al., 2021), optimal rates via U-statistics (Berrett et al., 2021), optimal transport (Liu et al., 2022), etc.

Conditional Independence: kernel method (Zhang et al., 2011), generalized covariance measure (Shah & Peters, 2020; Scheidegger et al., 2022), copula (Petersen & Hansen, 2021), projected covariance (Lundborg et al., 2022), model-X (Candès et al., 2018; Berrett et al., 2020), etc.

Testing generalized (conditional) independence

A lot of recent progress in independence / conditional independence testing.

Independence: kernel embedding (Gretton et al., 2005, 2007), rank correlation coefficients (Bergsma & Dassios, 2014; Drton et al., 2020; Shi et al., 2021), optimal rates via U-statistics (Berrett et al., 2021), optimal transport (Liu et al., 2022), etc.

Conditional Independence: kernel method (Zhang et al., 2011), generalized covariance measure (Shah & Peters, 2020; Scheidegger et al., 2022), copula (Petersen & Hansen, 2021), projected covariance (Lundborg et al., 2022), model-X (Candès et al., 2018; Berrett et al., 2020), etc.

💡 We can simulate data from Q.

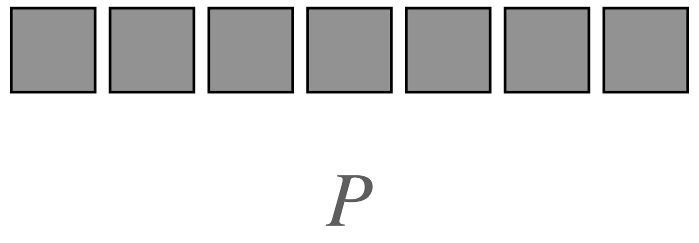
Testing generalized (conditional) independence

A lot of recent progress in independence / conditional independence testing.

Independence: kernel embedding (Gretton et al., 2005, 2007), rank correlation coefficients (Bergsma & Dassios, 2014; Drton et al., 2020; Shi et al., 2021), optimal rates via U-statistics (Berrett et al., 2021), optimal transport (Liu et al., 2022), etc.

Conditional Independence: kernel method (Zhang et al., 2011), generalized covariance measure (Shah & Peters, 2020; Scheidegger et al., 2022), copula (Petersen & Hansen, 2021), projected covariance (Lundborg et al., 2022), model-X (Candès et al., 2018; Berrett et al., 2020), etc.

💡 We can simulate data from Q.



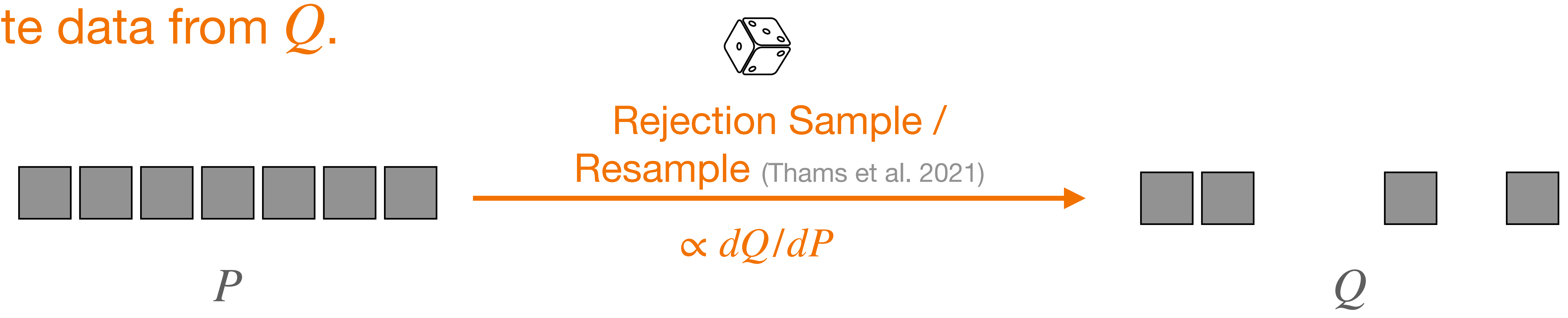
Testing generalized (conditional) independence

A lot of recent progress in independence / conditional independence testing.

Independence: kernel embedding (Gretton et al., 2005, 2007), rank correlation coefficients (Bergsma & Dassios, 2014; Drton et al., 2020; Shi et al., 2021), optimal rates via U-statistics (Berrett et al., 2021), optimal transport (Liu et al., 2022), etc.

Conditional Independence: kernel method (Zhang et al., 2011), generalized covariance measure (Shah & Peters, 2020; Scheidegger et al., 2022), copula (Petersen & Hansen, 2021), projected covariance (Lundborg et al., 2022), model-X (Candès et al., 2018; Berrett et al., 2020), etc.

💡 We can simulate data from Q.



Rejection Sample / Resample (Thams et al. 2021)

$\propto dQ/dP$

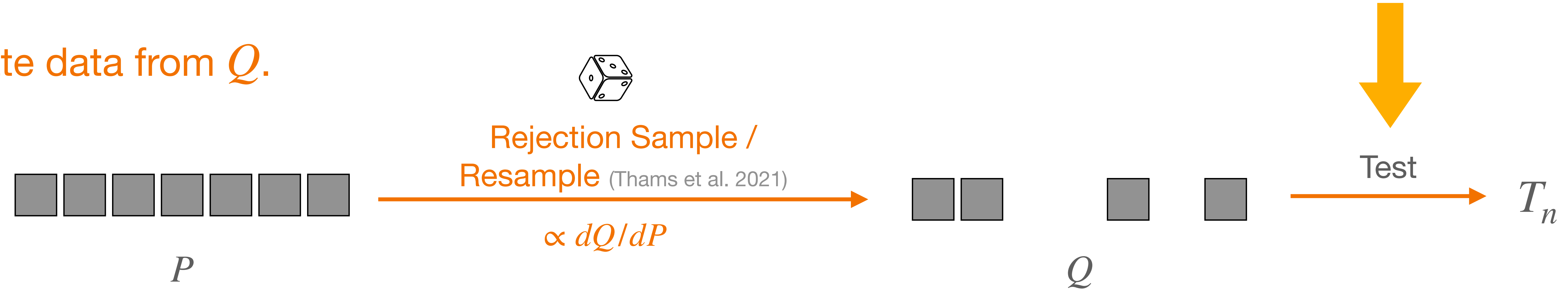
Testing generalized (conditional) independence

A lot of recent progress in independence / conditional independence testing.

Independence: kernel embedding (Gretton et al., 2005, 2007), rank correlation coefficients (Bergsma & Dassios, 2014; Drton et al., 2020; Shi et al., 2021), optimal rates via U-statistics (Berrett et al., 2021), optimal transport (Liu et al., 2022), etc.

Conditional Independence: kernel method (Zhang et al., 2011), generalized covariance measure (Shah & Peters, 2020; Scheidegger et al., 2022), copula (Petersen & Hansen, 2021), projected covariance (Lundborg et al., 2022), model-X (Candès et al., 2018; Berrett et al., 2020), etc.

💡 We can simulate data from Q.



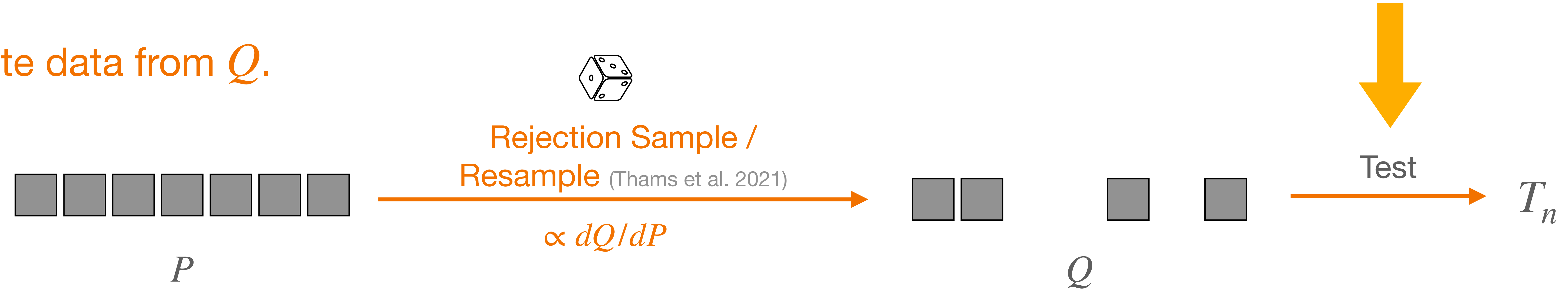
Testing generalized (conditional) independence

A lot of recent progress in independence / conditional independence testing.

Independence: kernel embedding (Gretton et al., 2005, 2007), rank correlation coefficients (Bergsma & Dassios, 2014; Drton et al., 2020; Shi et al., 2021), optimal rates via U-statistics (Berrett et al., 2021), optimal transport (Liu et al., 2022), etc.

Conditional Independence: kernel method (Zhang et al., 2011), generalized covariance measure (Shah & Peters, 2020; Scheidegger et al., 2022), copula (Petersen & Hansen, 2021), projected covariance (Lundborg et al., 2022), model-X (Candès et al., 2018; Berrett et al., 2020), etc.

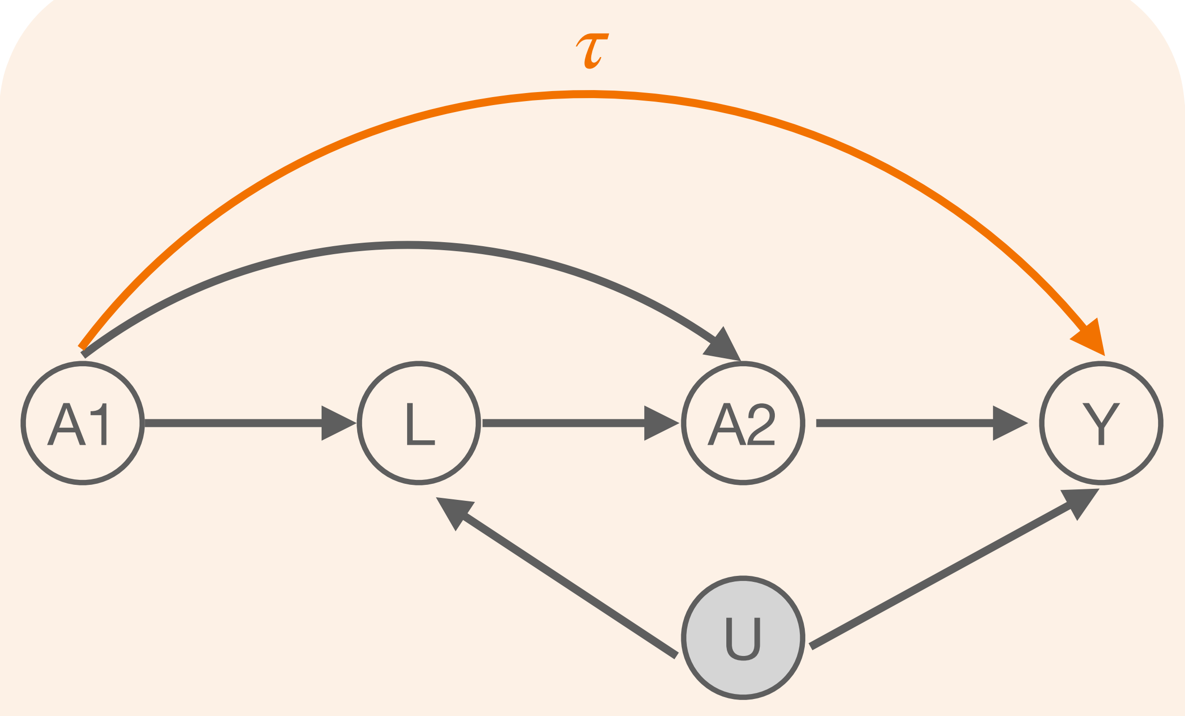
💡 We can simulate data from Q.



	Need re-calibration	Reduced sample size	Randomized
Sampling	No	Yes	Yes
Inverse probability weighting (IPW)	Yes	No	No

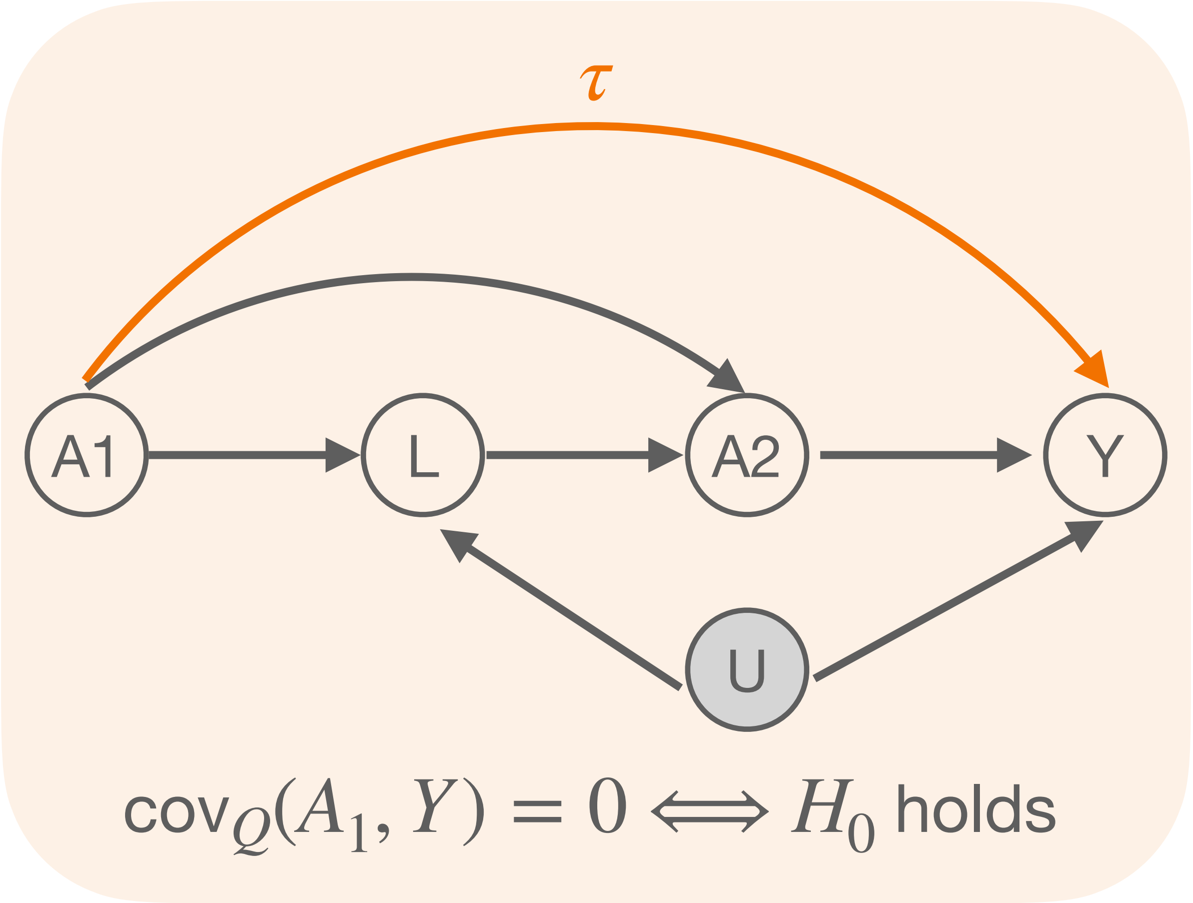
Simulation: Linear SEM

Simulation: Linear SEM



$\mathrm{cov}_Q(A_1, Y) = 0 \iff H_0$ holds

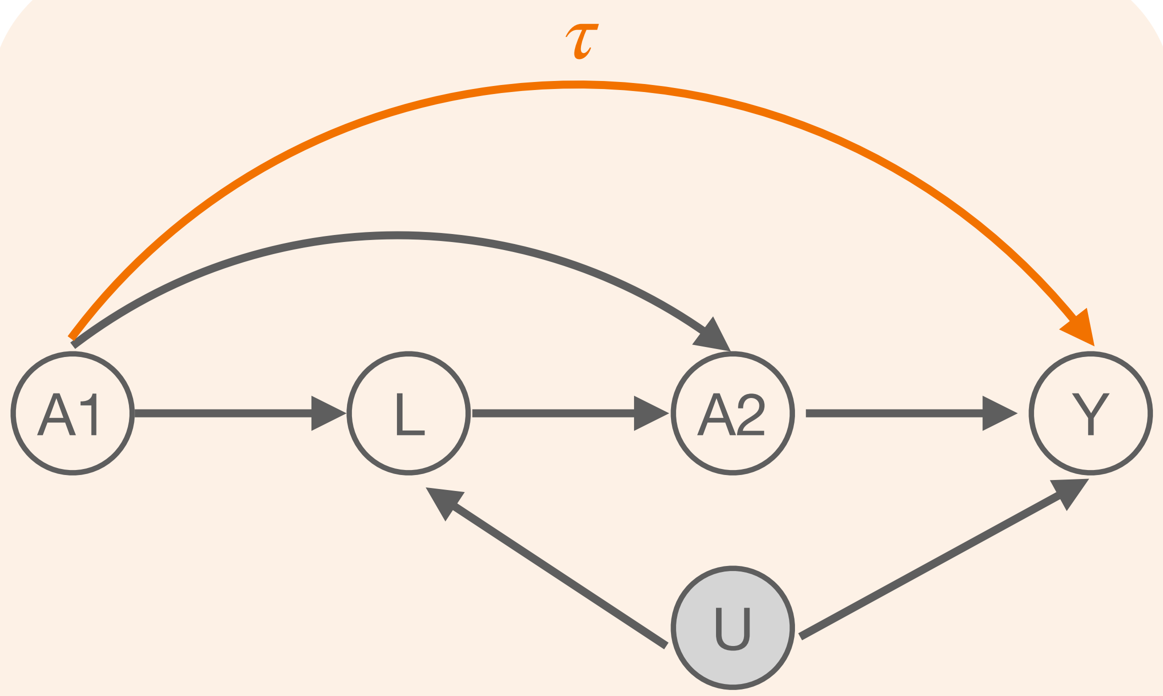
Simulation: Linear SEM



$\text{cov}_Q(A_1, Y) = 0 \iff H_0$ holds

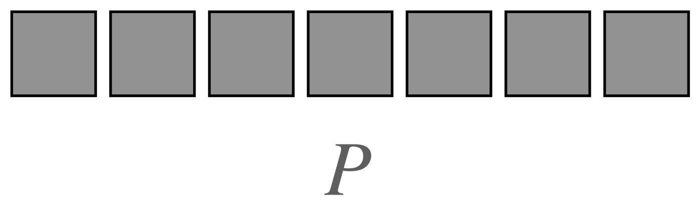
(1) Rej. Sample + Permutation

Simulation: Linear SEM

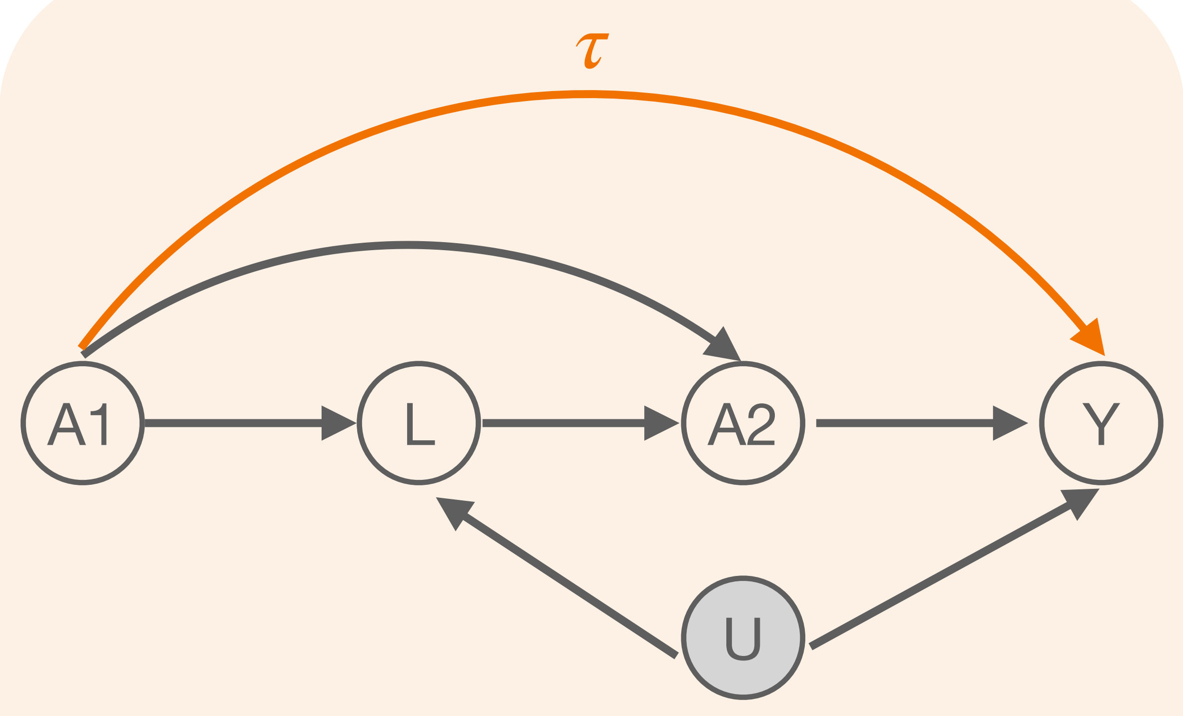


cov$_Q(A_1, Y) = 0 \iff H_0$ holds

(1) Rej. Sample + Permutation

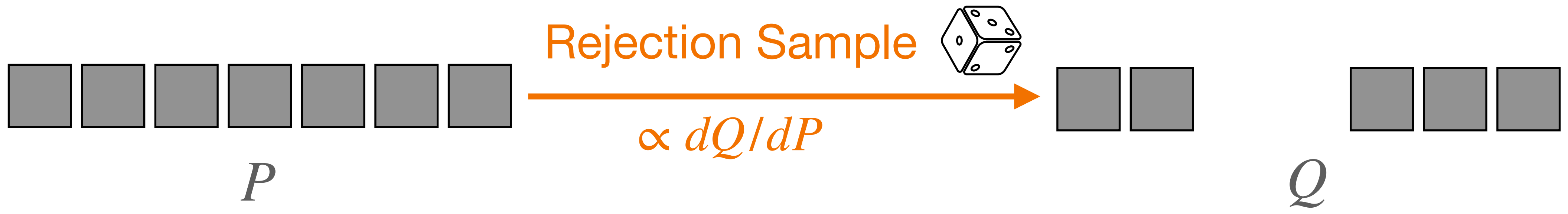


Simulation: Linear SEM



$\text{cov}_Q(A_1, Y) = 0 \iff H_0$ holds

(1) Rej. Sample + Permutation

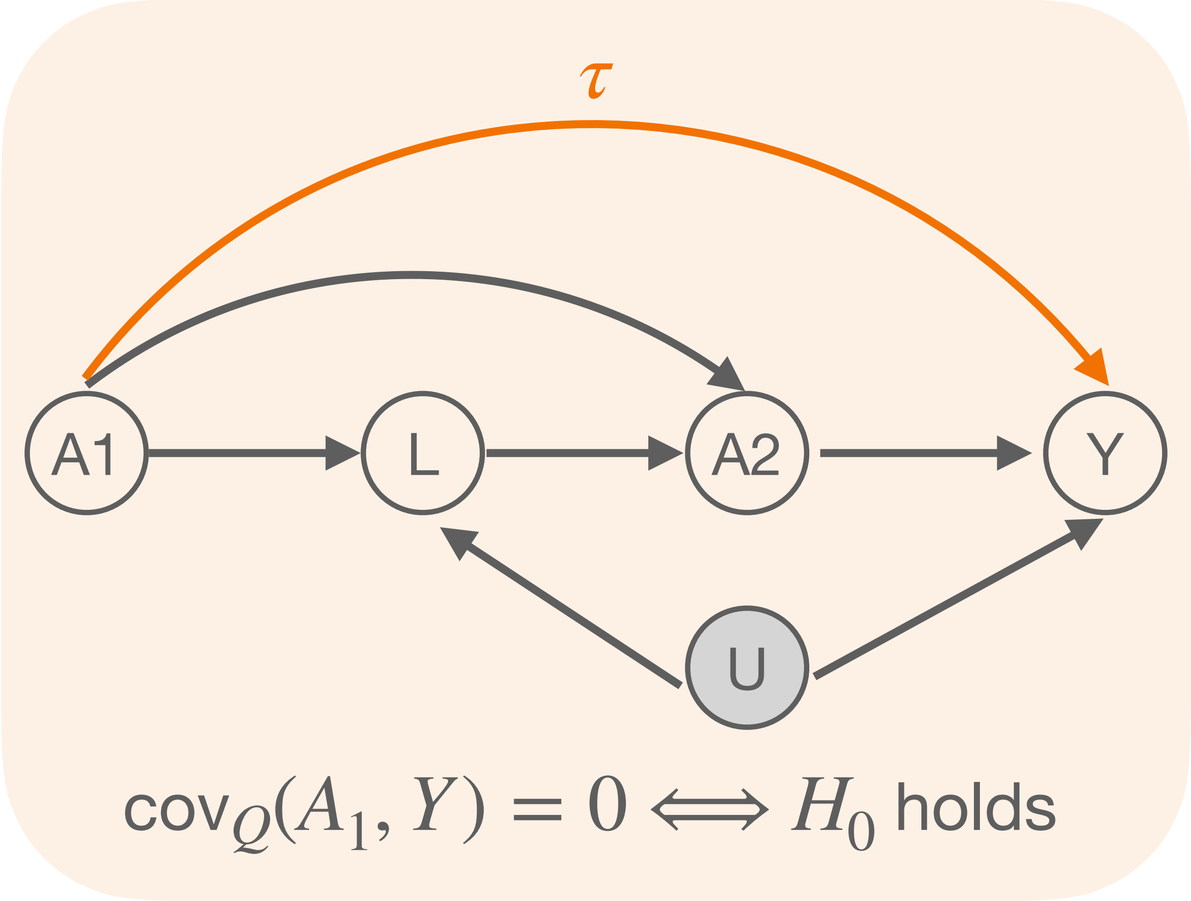

Rejection Sample

$\propto dQ/dP$

P

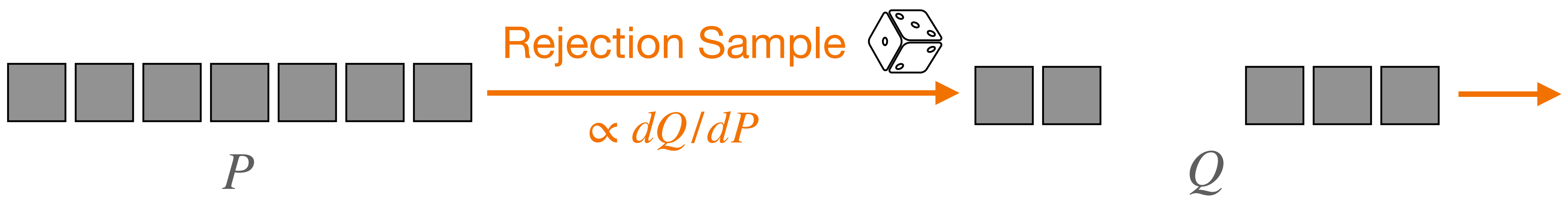
Q

Simulation: Linear SEM



$\mathrm{cov}_Q(A_1, Y) = 0 \iff H_0$ holds

(1) Rej. Sample + Permutation



Rejection Sample

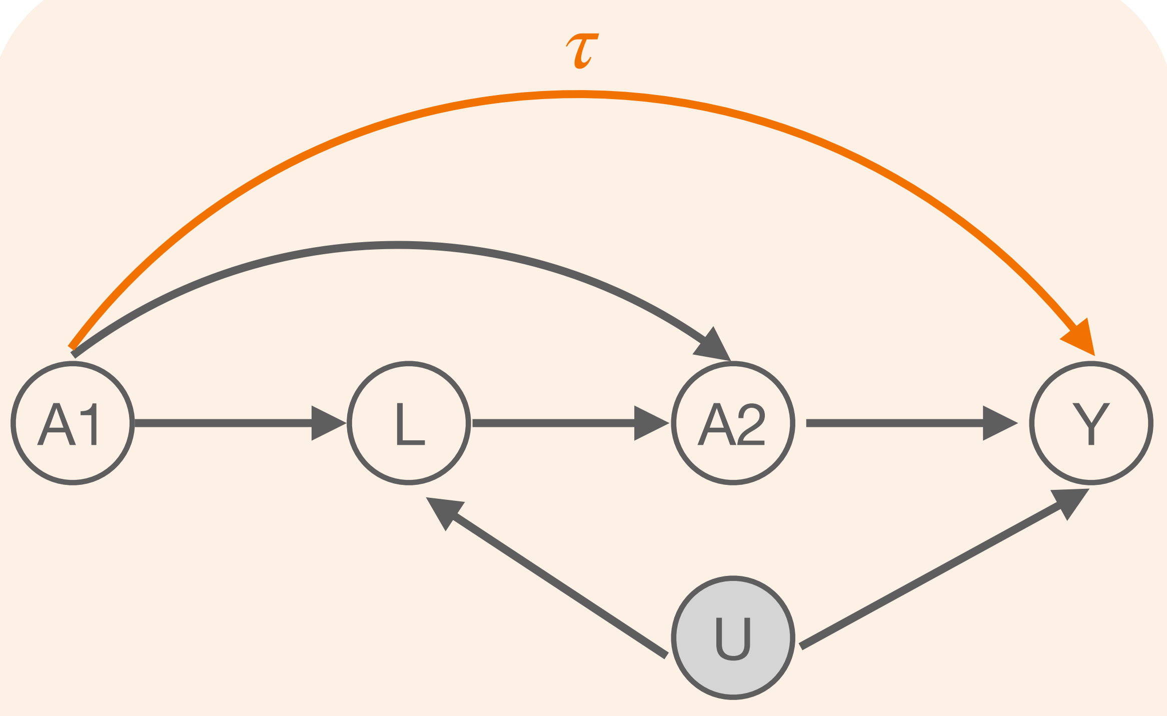
$\propto dQ/dP$

P

Q

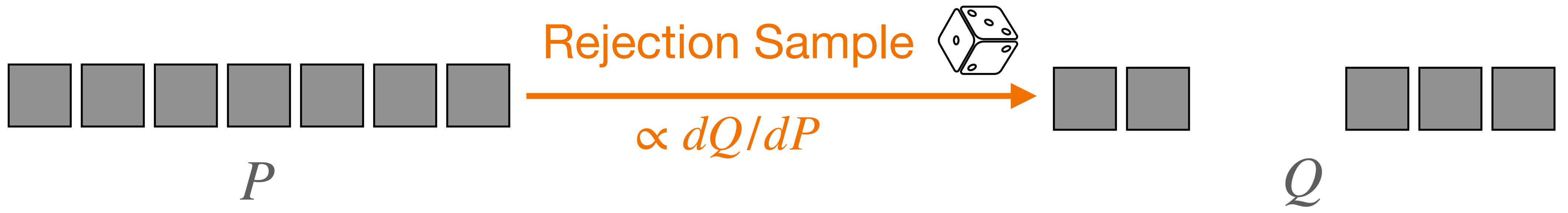
T_n := Perm. p-value of $\mathrm{cov}_Q(A_1, Y)$

Simulation: Linear SEM

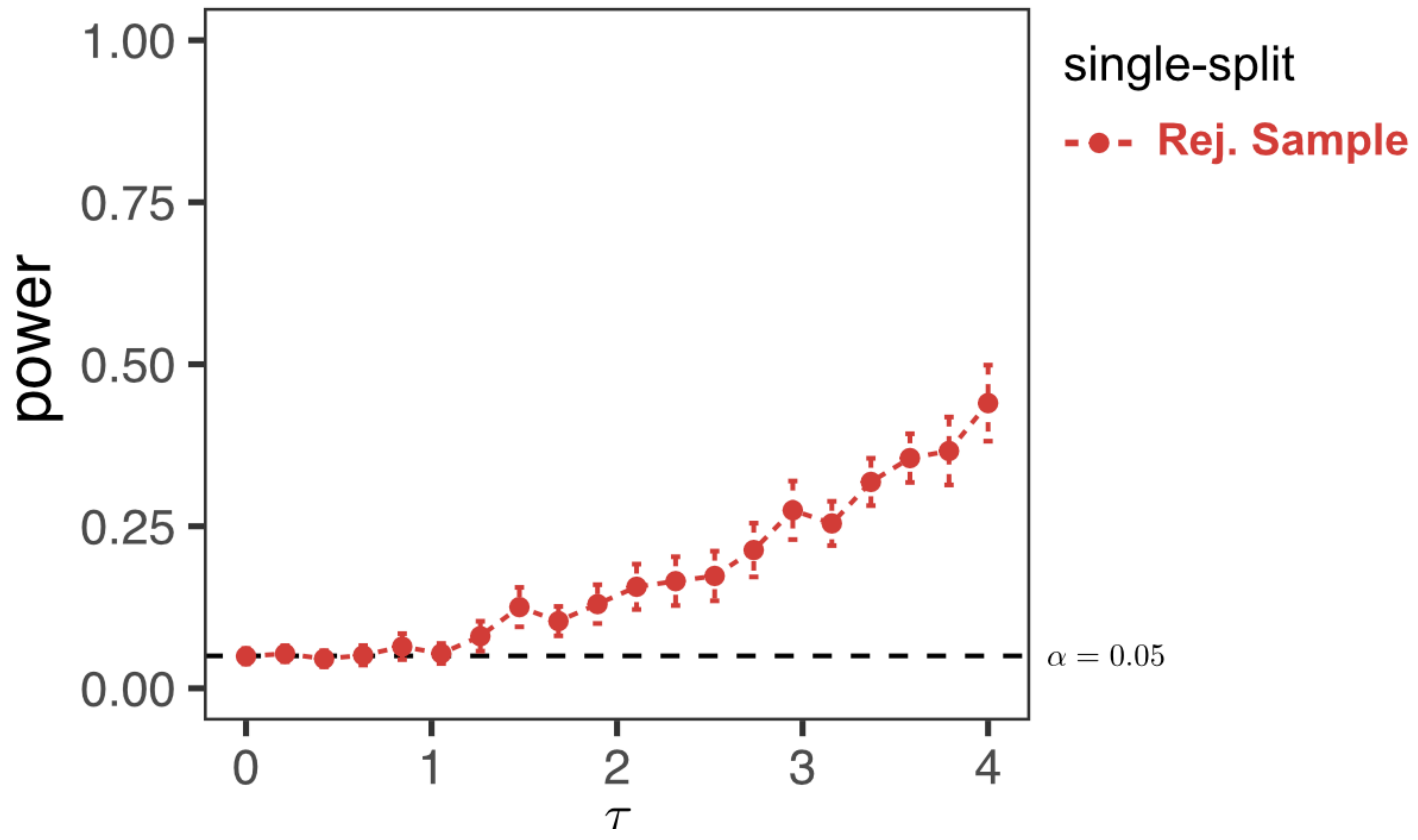


$\text{cov}_Q(A_1, Y) = 0 \iff H_0$ holds

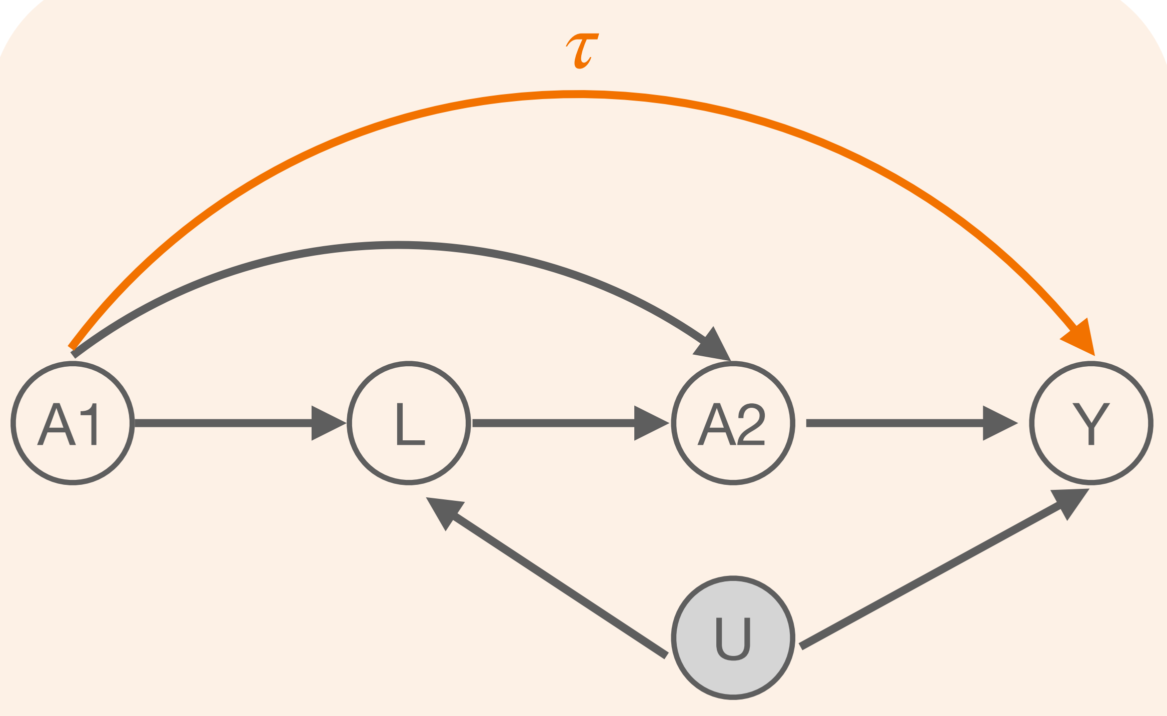
(1) Rej. Sample + Permutation



T_n := Perm. p-value of $\text{cov}_Q(A_1, Y)$

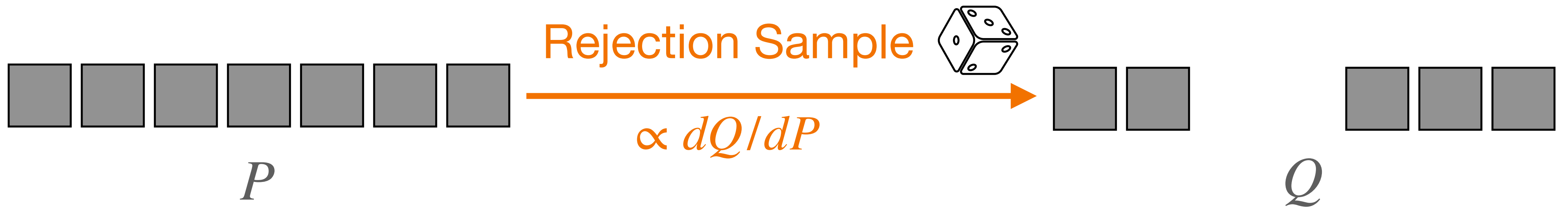


Simulation: Linear SEM

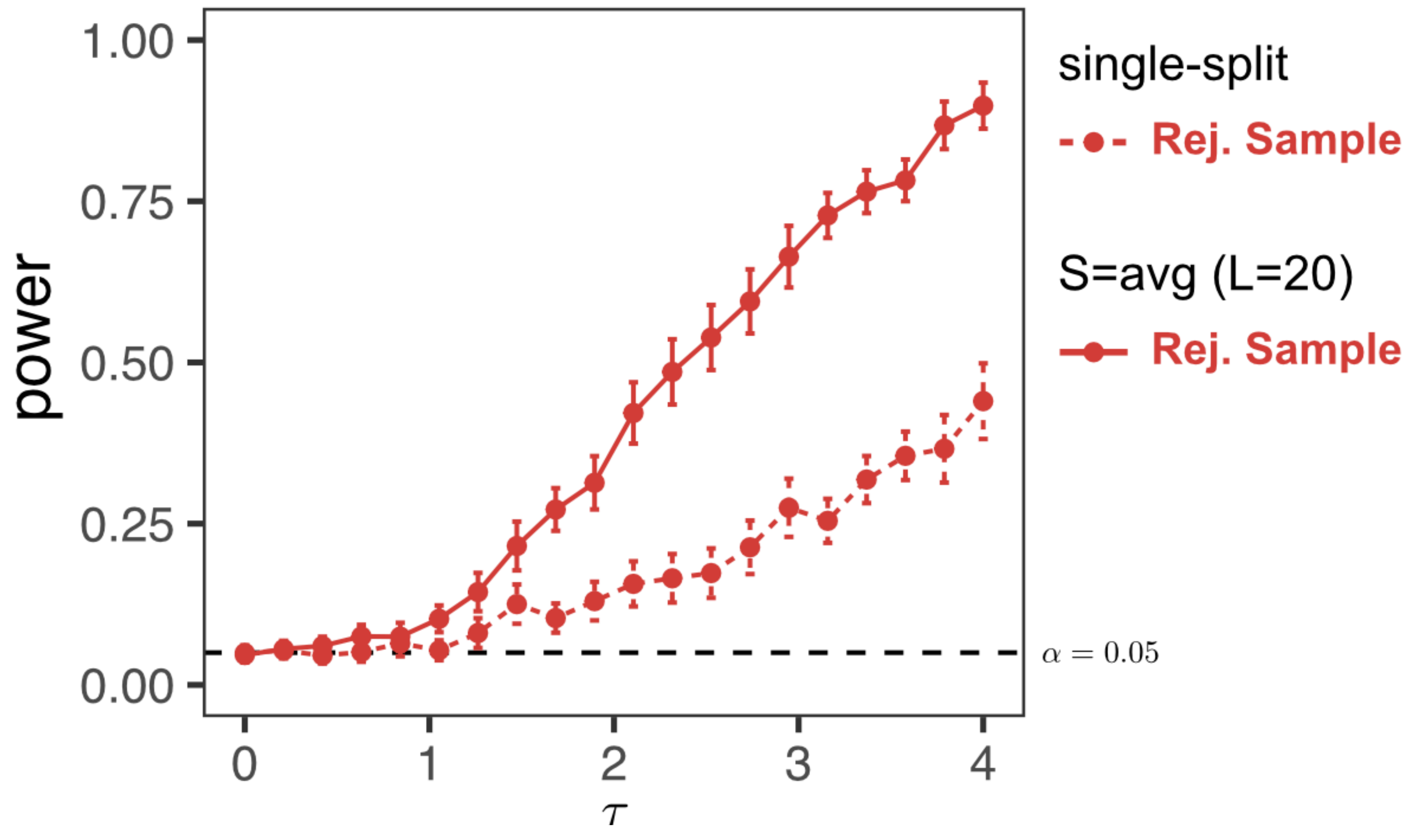


$\mathrm{cov}_Q(A_1, Y) = 0 \iff H_0$ holds

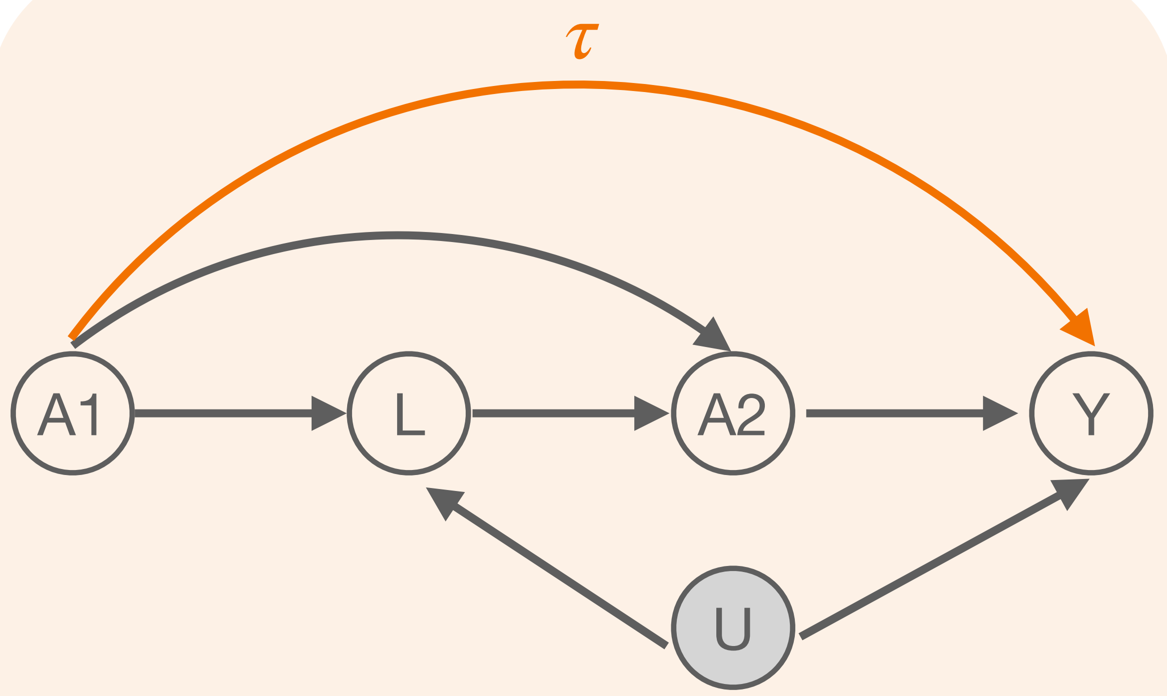
(1) Rej. Sample + Permutation



→ T_n := Perm. p-value of $\mathrm{cov}_Q(A_1, Y)$

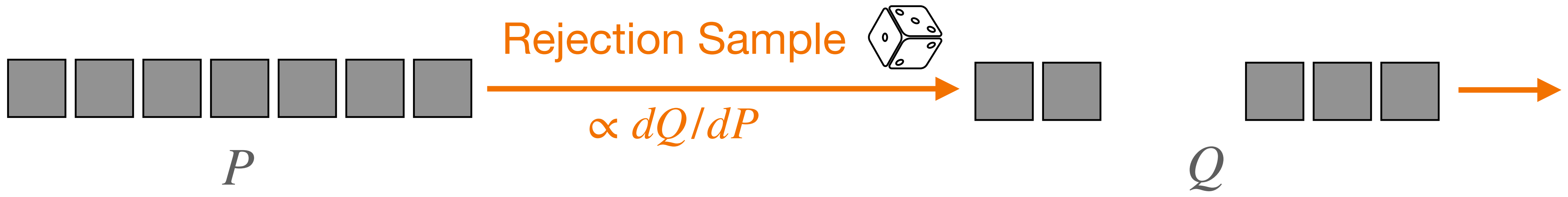


Simulation: Linear SEM



$\mathrm{cov}_Q(A_1, Y) = 0 \iff H_0$ holds

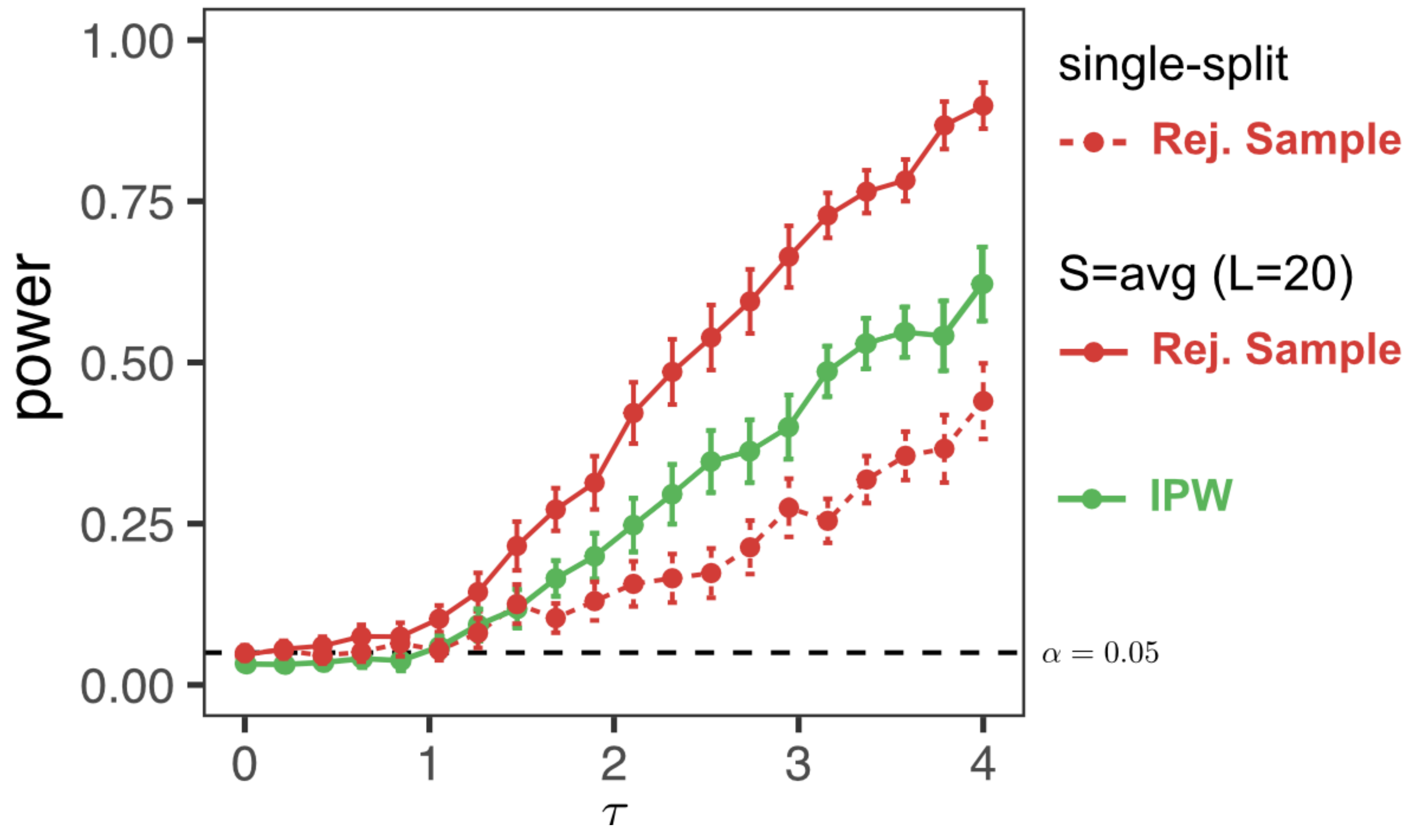
(1) Rej. Sample + Permutation



$T_n :=$ Perm. p-value of $\mathrm{cov}_Q(A_1, Y)$

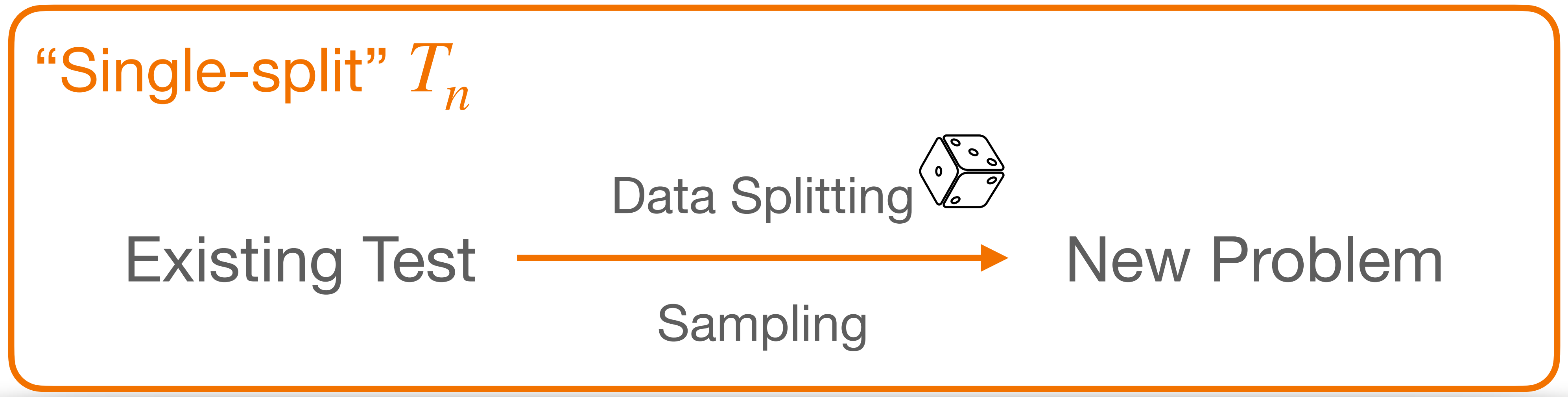
(2) IPW for $\mathrm{cov}_Q(A_1, Y)$ (Robins, 1999)

$$Z_i := \frac{Y_i(A_{1,i} - \mathbb{E}A_1)}{P(A_{2,i} \mid L_i, A_{1,i})}, \qquad \chi_n := \frac{\sum_i Z_i}{\sqrt{\sum_i Z_i^2}} \to_d \mathcal{N}(0,1)\,.$$



Theme so far

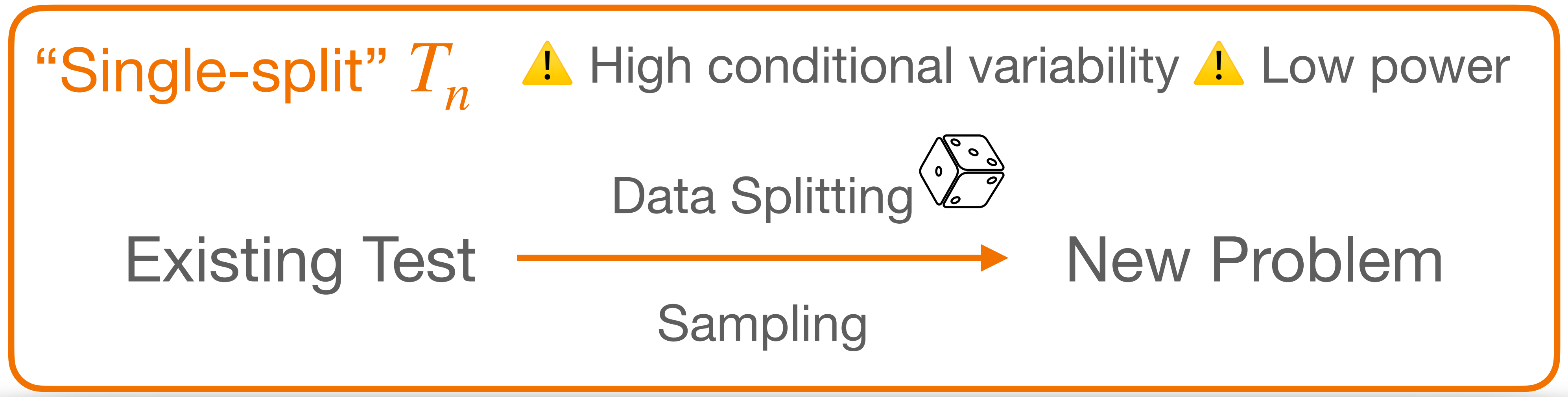
Theme so far



“Single-split” T_n

Existing Test $\xrightarrow[\text{Sampling}]{\text{Data Splitting}}$ New Problem

Theme so far

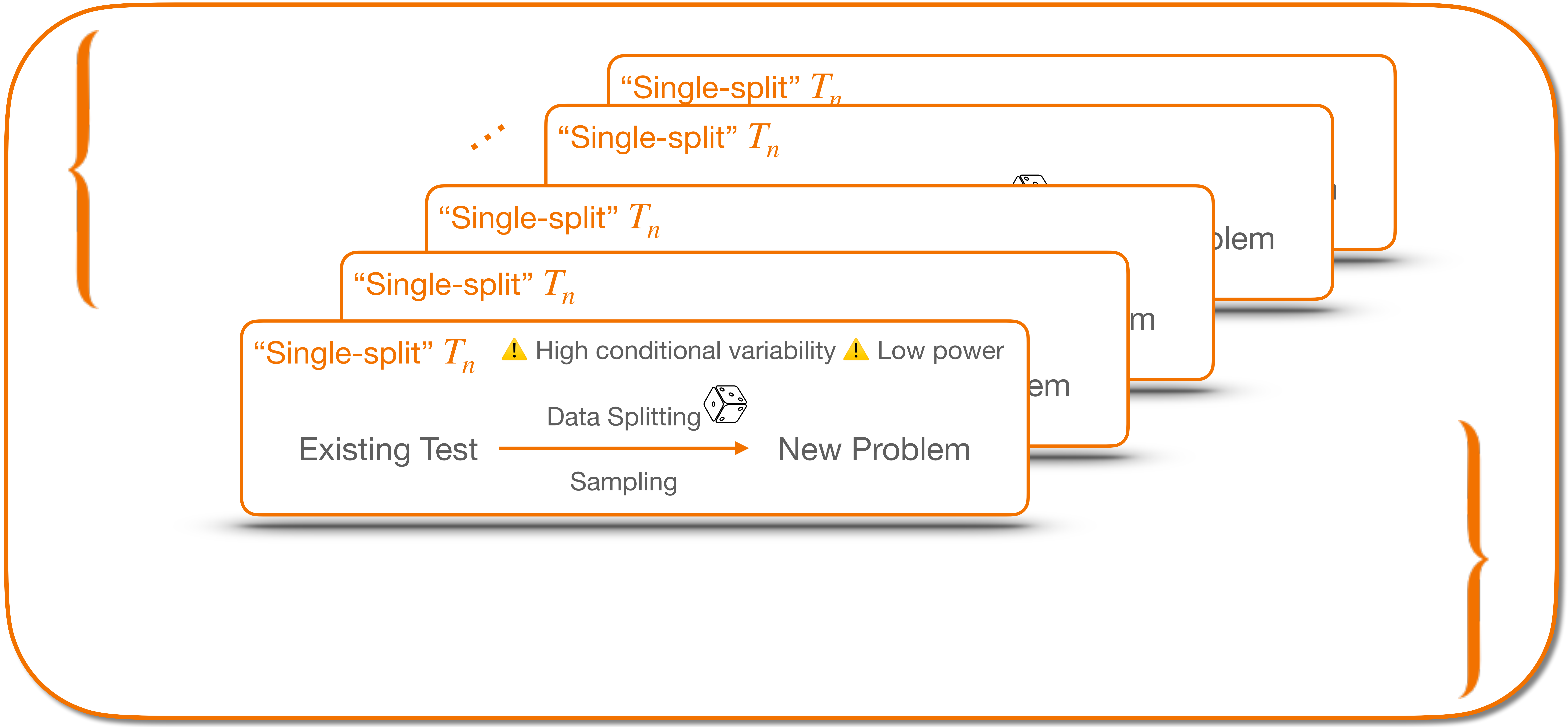


“Single-split” T_n ⚠ High conditional variability ⚠ Low power

$$\text{Existing Test} \xrightarrow[\text{Sampling}]{\text{Data Splitting}} \text{New Problem}$$

Theme so far

Meta-algorithm: **Rank-transformed Subsampling**



“Single-split” T_n

“Single-split” T_n

“Single-split” T_n

“Single-split” T_n

“Single-split” T_n ⚠ High conditional variability ⚠ Low power

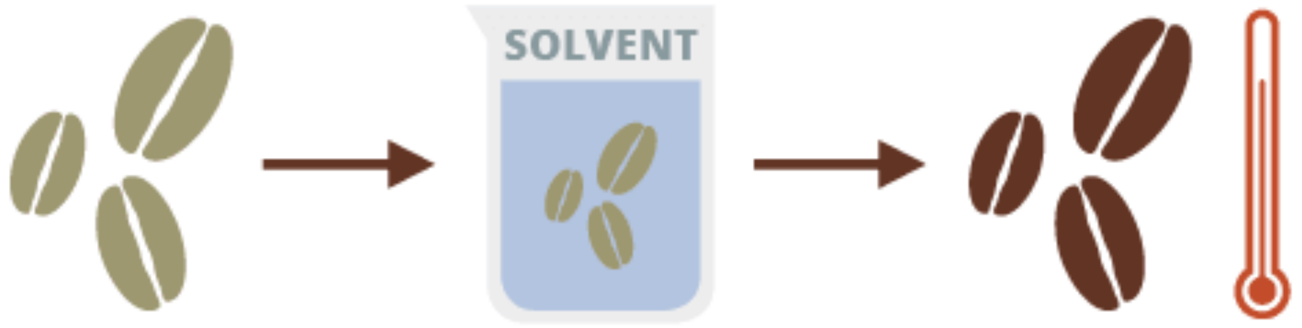
$$\text{Existing Test} \xrightarrow[\text{Sampling}]{\text{Data Splitting}} \text{New Problem}$$

Reduces (conditional) variability & Boosts power!

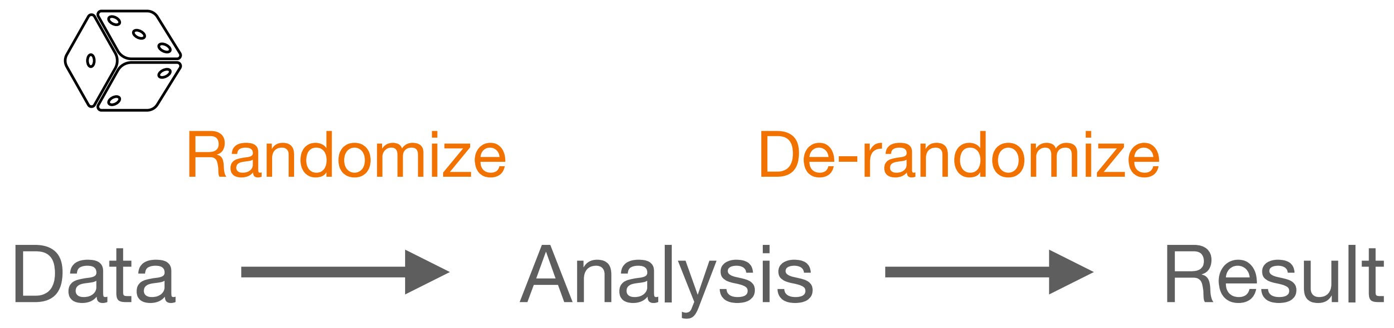
Outline

- Setup and main challenge
- Method: Rank-transformed subsampling
- Applications
    - Hunt and test
    - Improving double machine learning
    - Testing no direct effect of a sequentially randomized trial
- **Future directions**

Harness extra randomness



© www.compoundchem.com

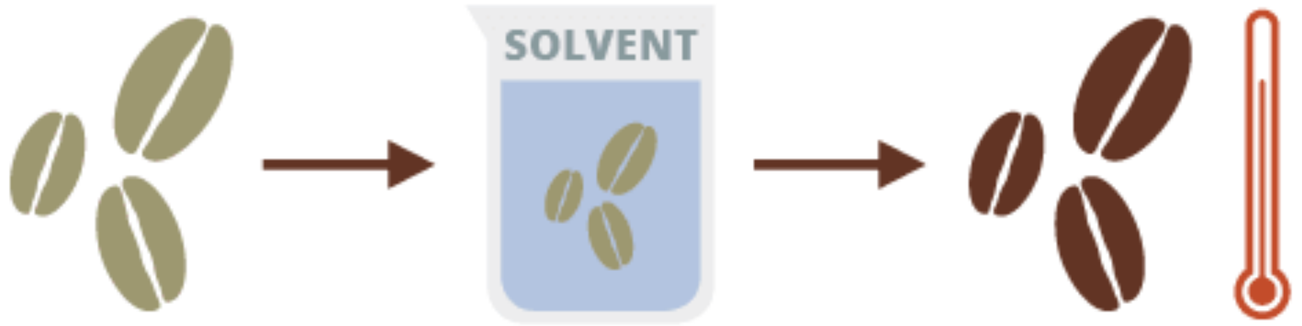


Randomize

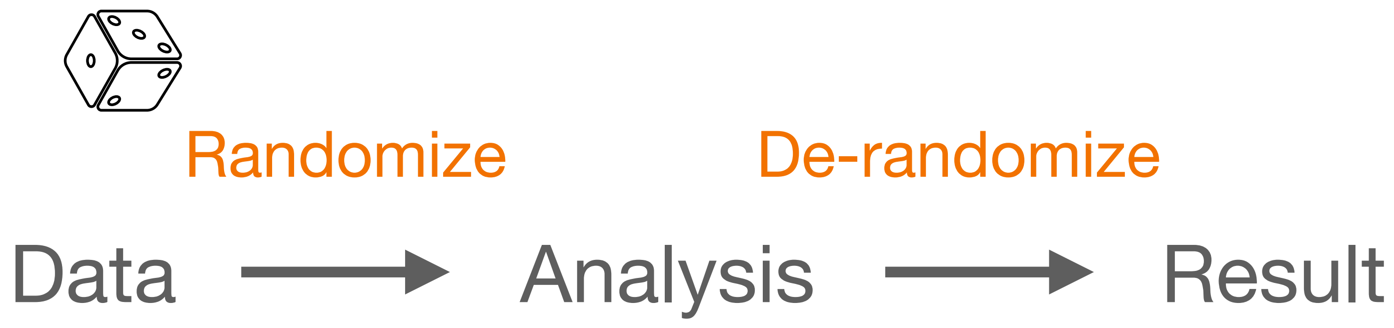
De-randomize

Data → Analysis → Result

Harness extra randomness



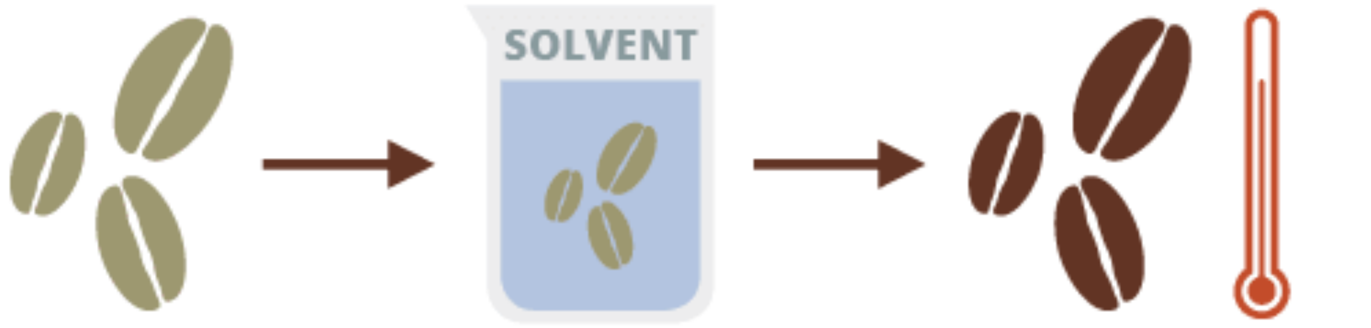
© www.compoundchem.com



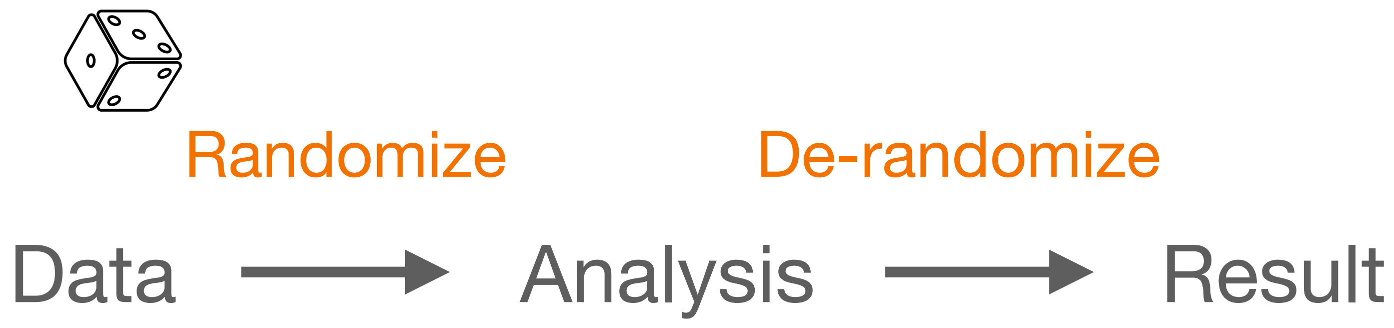
👉 Flexible goodness-of-fit

e.g., quantile regression

Harness extra randomness



© www.compoundchem.com

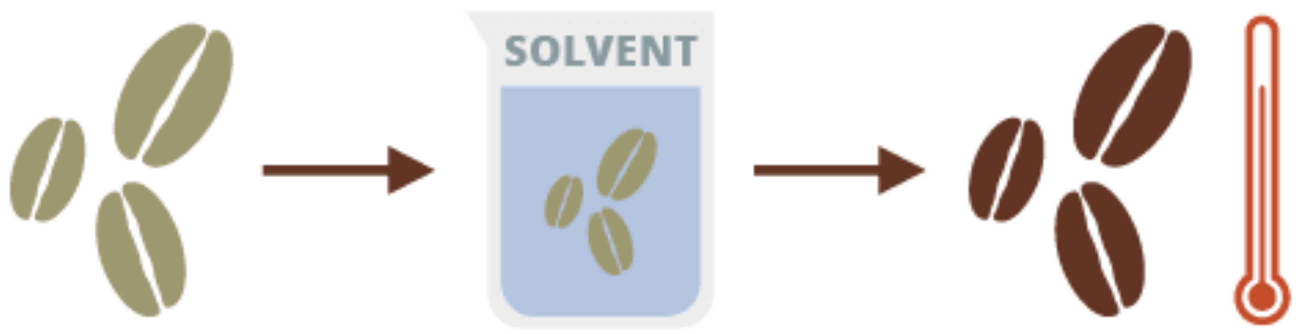


👉 Flexible goodness-of-fit

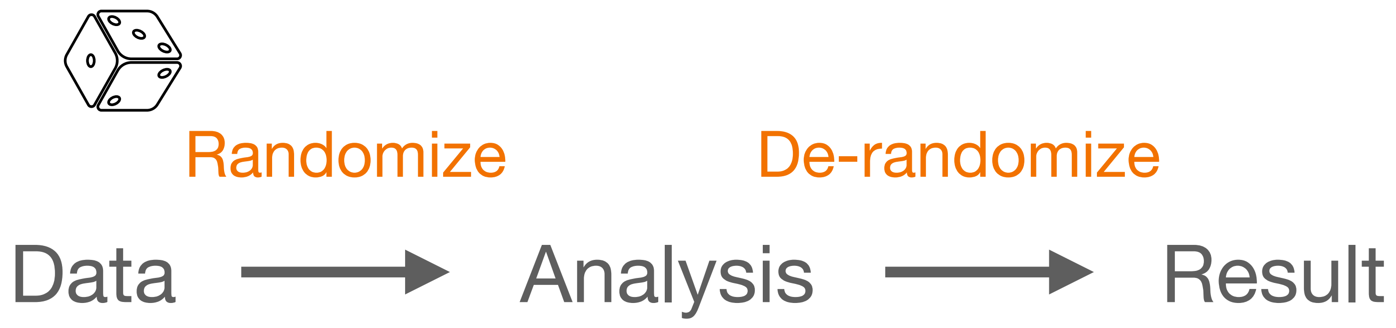
e.g., quantile regression

👉 Missing data / imputation

Harness extra randomness



© www.compoundchem.com


Randomize

De-randomize

Data → Analysis → Result

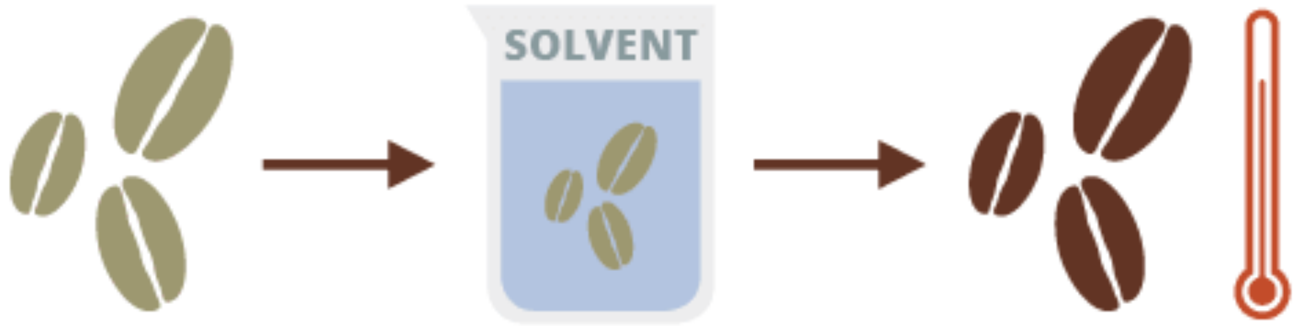
👉 Flexible goodness-of-fit

e.g., quantile regression

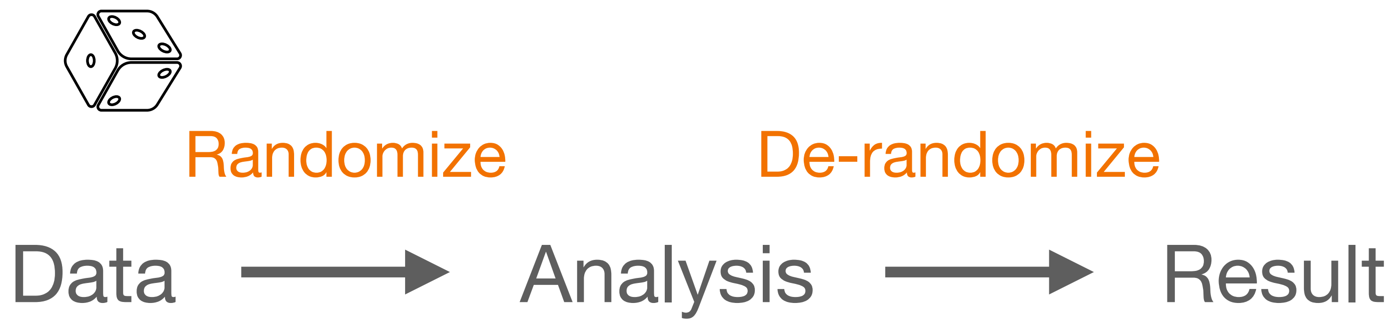
👉 Missing data / imputation

👉 Random projection

Harness extra randomness



© www.compoundchem.com



👉 Flexible goodness-of-fit

e.g., quantile regression

👉 Missing data / imputation

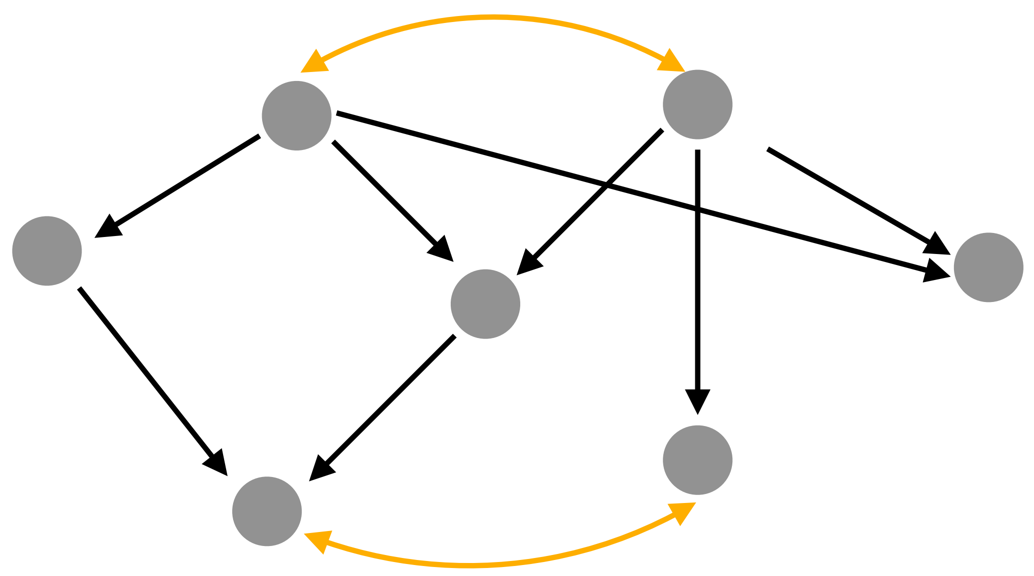
👉 Random projection

👉 Causal inference & causal discovery

- Observed distribution → Intervened distribution

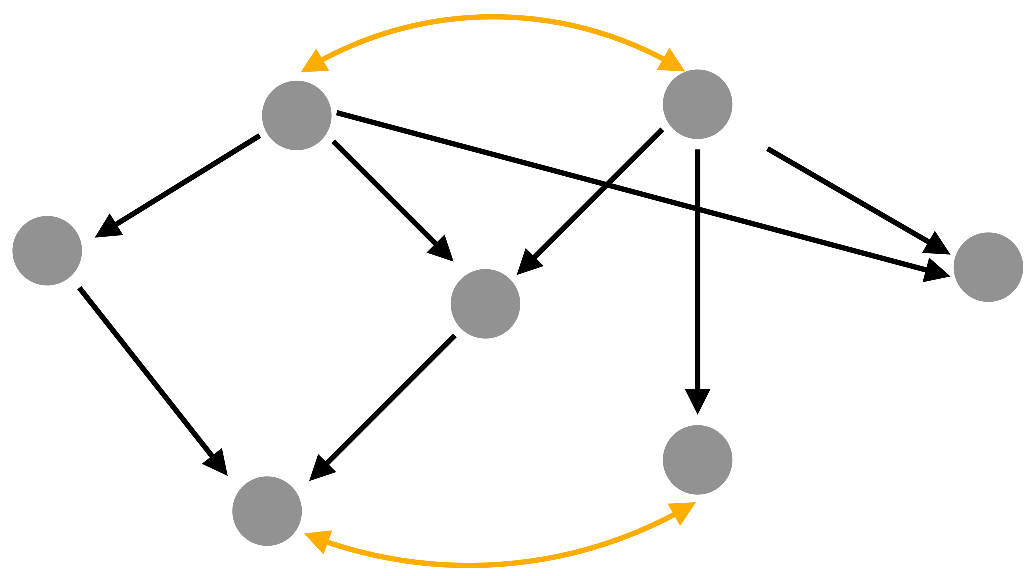
Empowering causal discovery

	Gene 1	Gene 2	Gene 3	...
Cell 1	10	10	0	
Cell 2	0	15	4	
Cell 3	600	0	20	
⋮				



Empowering causal discovery

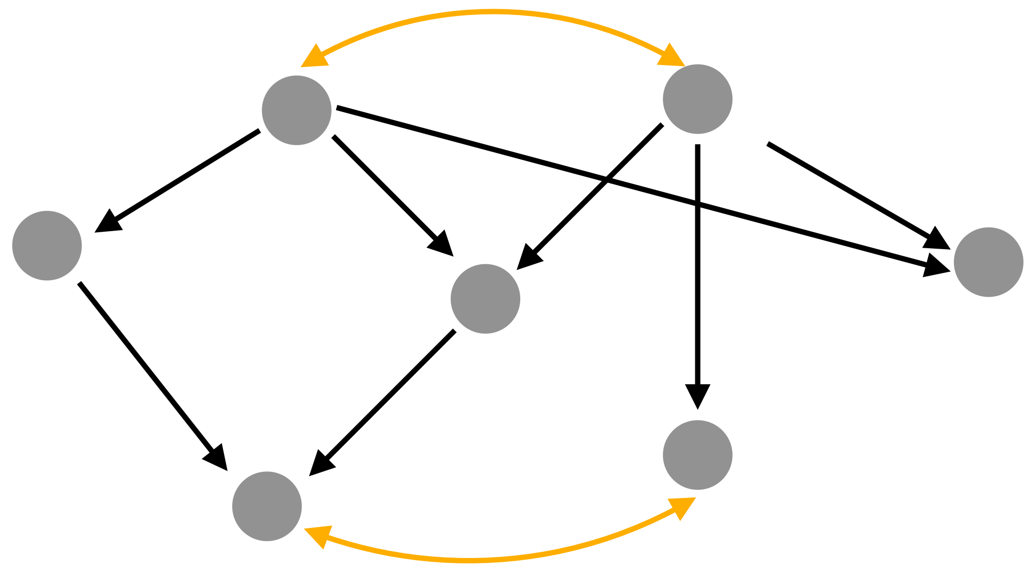
	Gene 1	Gene 2	Gene 3	...
Cell 1	10	10	0	
Cell 2	0	15	4	
Cell 3	600	0	20	
⋮				



🙁 **State of the art:** cannot utilize generalized conditional independence.

Empowering causal discovery

	Gene 1	Gene 2	Gene 3	...
Cell 1	10	10	0	
Cell 2	0	15	4	
Cell 3	600	0	20	
⋮				

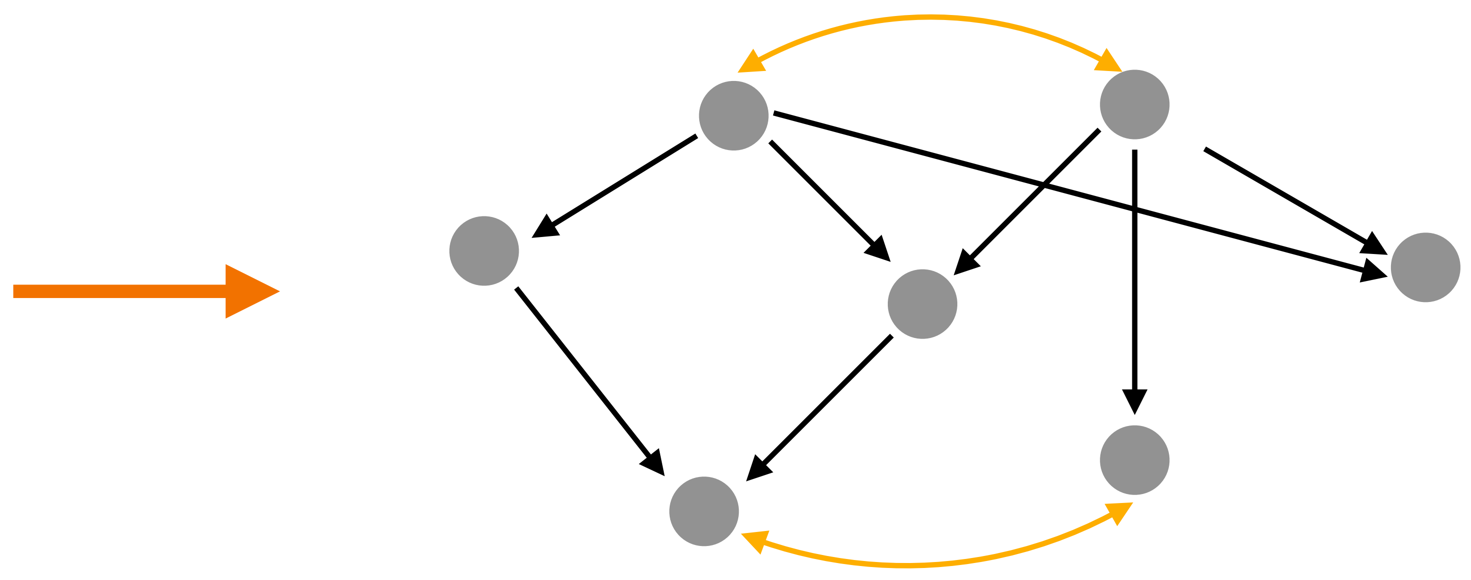


🙁 **State of the art:** cannot utilize generalized conditional independence.

💡 Generalized conditional independence can be very informative about the graph!

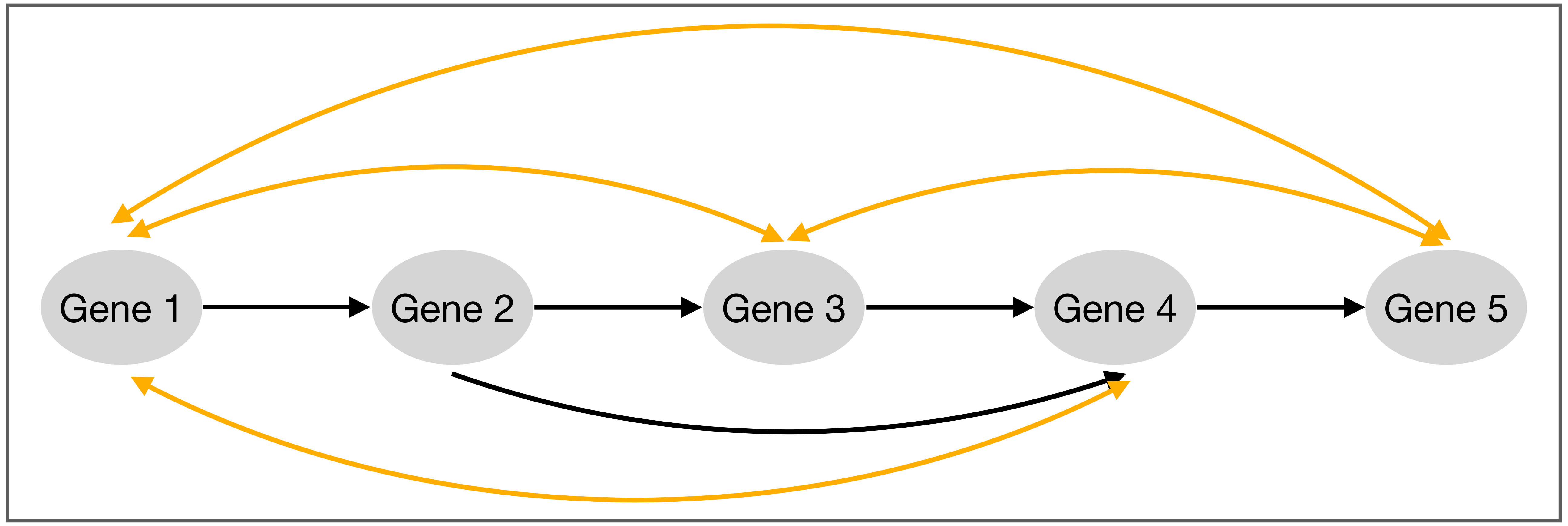
Empowering causal discovery

	Gene 1	Gene 2	Gene 3	...
Cell 1	10	10	0	
Cell 2	0	15	4	
Cell 3	600	0	20	
⋮				



🙁 **State of the art:** cannot utilize generalized conditional independence.

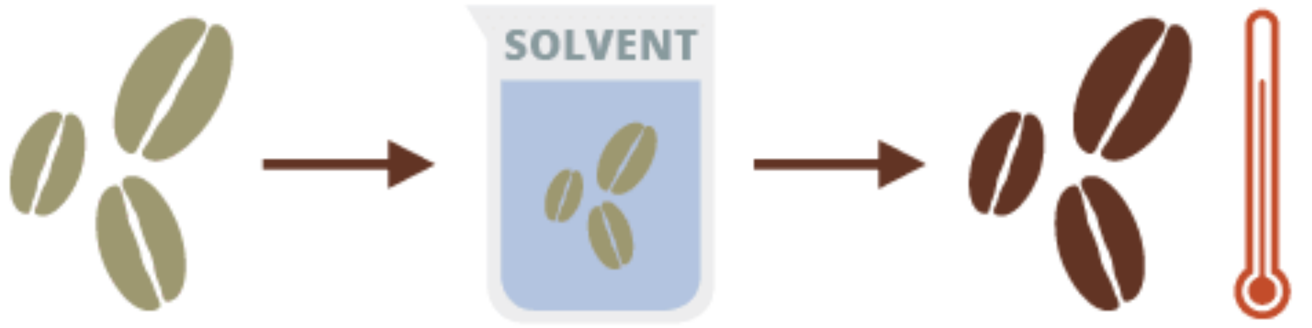
💡 Generalized conditional independence can be very informative about the graph!



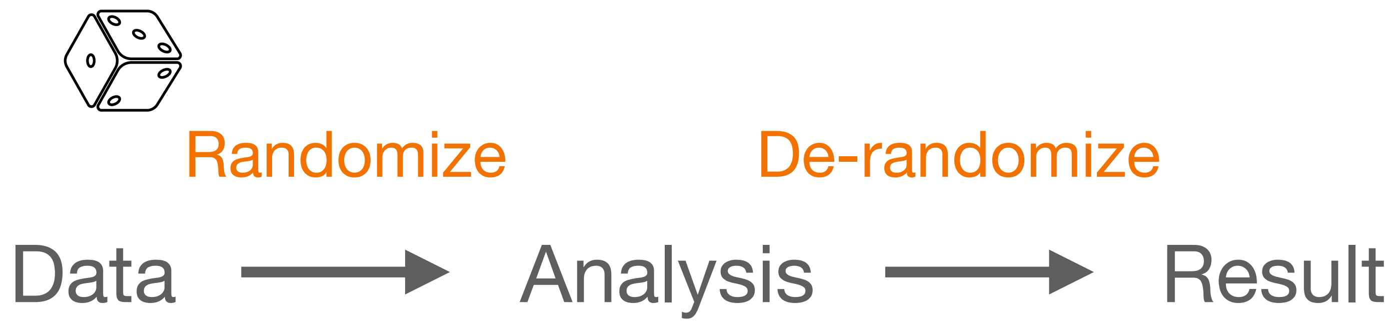
😮 Uniquely identified (from ~30,000 possibilities) from one single generalized conditional independence constraint!

Robins. Interview with Jamie Robins. *Observational Studies* (2022).

Harness extra randomness



© www.compoundchem.com



Randomize De-randomize

Data → Analysis → Result

👉 Flexible goodness-of-fit

e.g., quantile regression

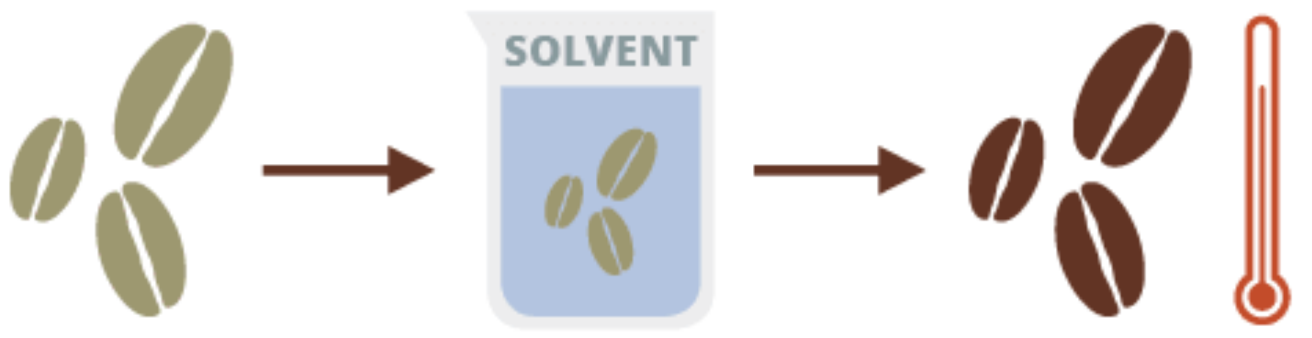
👉 Missing data / imputation

👉 Random projection

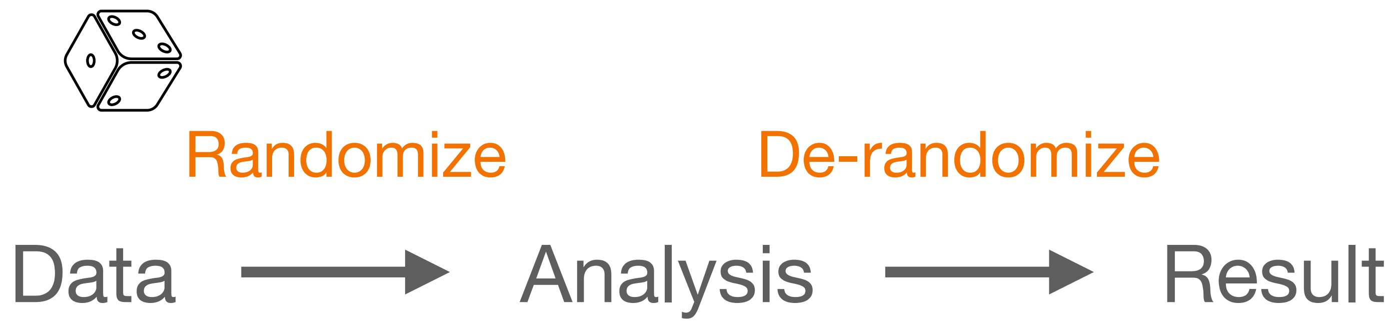
👉 Causal inference & causal discovery

- Observed distribution → Intervened distribution

Harness extra randomness



© www.compoundchem.com



👉 Flexible goodness-of-fit

e.g., quantile regression

👉 Missing data / imputation

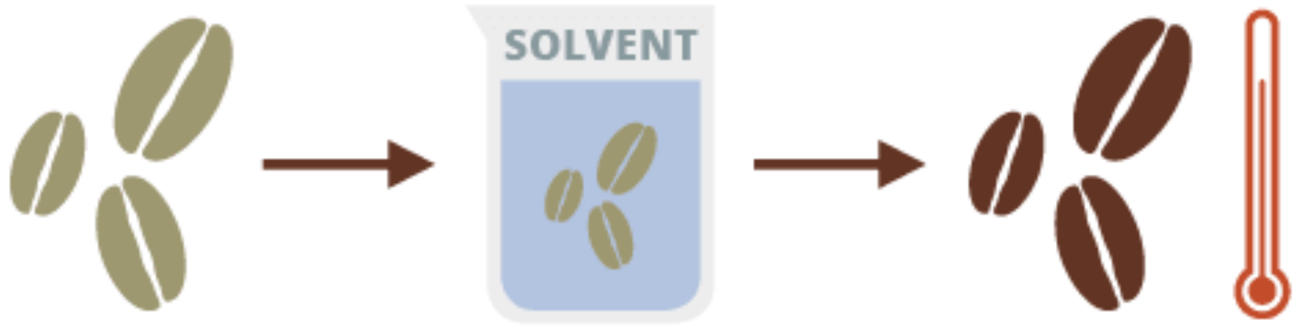
👉 Random projection

👉 Causal inference & causal discovery

- Observed distribution → Intervened distribution

🤔 How much power can we hope to extract?

Harness extra randomness



© www.compoundchem.com

Randomize De-randomize

Data ⟶ Analysis ⟶ Result

👉 Flexible goodness-of-fit

e.g., quantile regression

👉 Missing data / imputation

👉 Random projection

👉 Causal inference & causal discovery

- Observed distribution → Intervened distribution

🤔 How much power can we hope to extract?

🤔 Replicability: **computational** ➠ **statistical**

Multiple aggregations: Adaptive algorithm

Multiple aggregations: Adaptive algorithm

$T_n^{(1)}$	$T_n^{(2)}$	$T_n^{(3)}$	$T_n^{(4)}$	$\cdots$	$T_n^{(L)}$	🤔
*	**	.	*		*	$S = \text{avg}$ ✅
.			***			$S = \min$ ✅

Multiple aggregations: Adaptive algorithm

$T_n^{(1)}$	$T_n^{(2)}$	$T_n^{(3)}$	$T_n^{(4)}$	$\cdots$	$T_n^{(L)}$	🤔
*	**	.	*		*	$S =$ avg ✅
.			***			$S = \min$ ✅

💡 Allow the user to specify $S^1, \ldots, S^W$

Multiple aggregations: Adaptive algorithm

$T_n^{(1)}$	$T_n^{(2)}$	$T_n^{(3)}$	$T_n^{(4)}$	$\cdots$	$T_n^{(L)}$	🤔
*	**	.	*		*	$S =$ avg ✅
.			***			$S = \min$ ✅

$\widetilde{T}_m^{(1)}$	$\widetilde{T}_m^{(2)}$	$\ldots$	$\widetilde{T}_m^{(L)}$
1.6	-0.8		0.1
-1.1	-0.2		1.8
$\vdots$	$\vdots$		$\vdots$
2.7	0.2		2.8

Observed

$T_n^{(1)}$	$T_n^{(2)}$	$\ldots$	$T_n^{(L)}$
2.1	-1.2		0.3

💡 Allow the user to specify $S^1, \ldots, S^W$

Multiple aggregations: Adaptive algorithm

$T_n^{(1)}$	$T_n^{(2)}$	$T_n^{(3)}$	$T_n^{(4)}$	$\cdots$	$T_n^{(L)}$	🤔
*	**	.	*		*	$S = \text{avg}$ ✅
.			***			$S = \min$ ✅

💡 Allow the user to specify $S^1, \ldots, S^W$

S^1

$\widetilde{T}_m^{(1)}$	$\widetilde{T}_m^{(2)}$	$\ldots$	$\widetilde{T}_m^{(L)}$	$\tilde{S}_n^1$
1.6	-0.8		0.1	0.5
-1.1	-0.2		1.8	-0.1
$\vdots$	$\vdots$		$\vdots$	$\vdots$
2.7	0.2		2.8	1.2

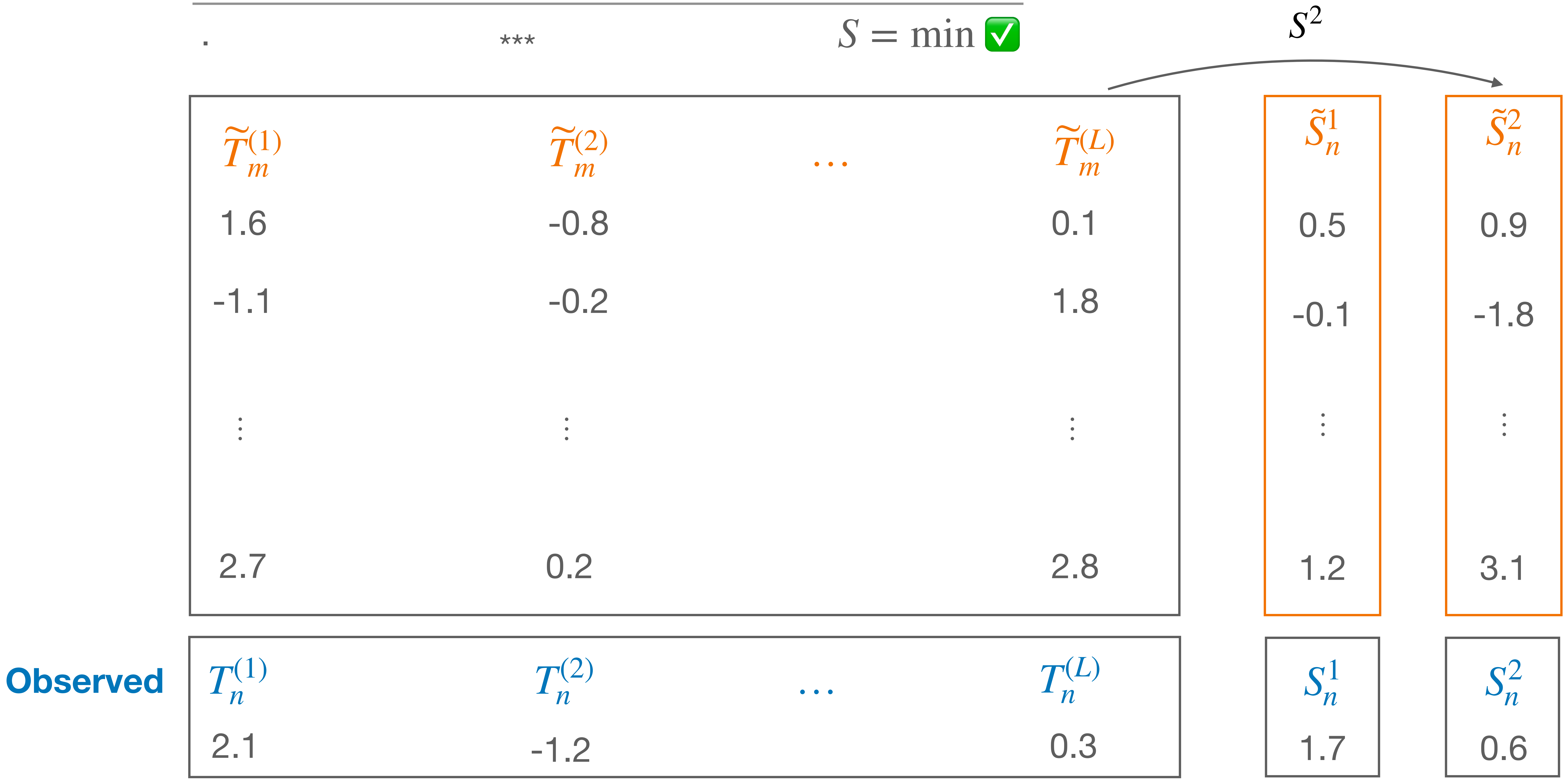
Observed

$T_n^{(1)}$	$T_n^{(2)}$	$\ldots$	$T_n^{(L)}$	S_n^1
2.1	-1.2		0.3	1.7

Multiple aggregations: Adaptive algorithm

$T_n^{(1)}$	$T_n^{(2)}$	$T_n^{(3)}$	$T_n^{(4)}$	$\cdots$	$T_n^{(L)}$	🤔
*	**	.	*		*	$S =$ avg ✅
.			***			$S = \min$ ✅

💡 Allow the user to specify $S^1, \ldots, S^W$



S^2

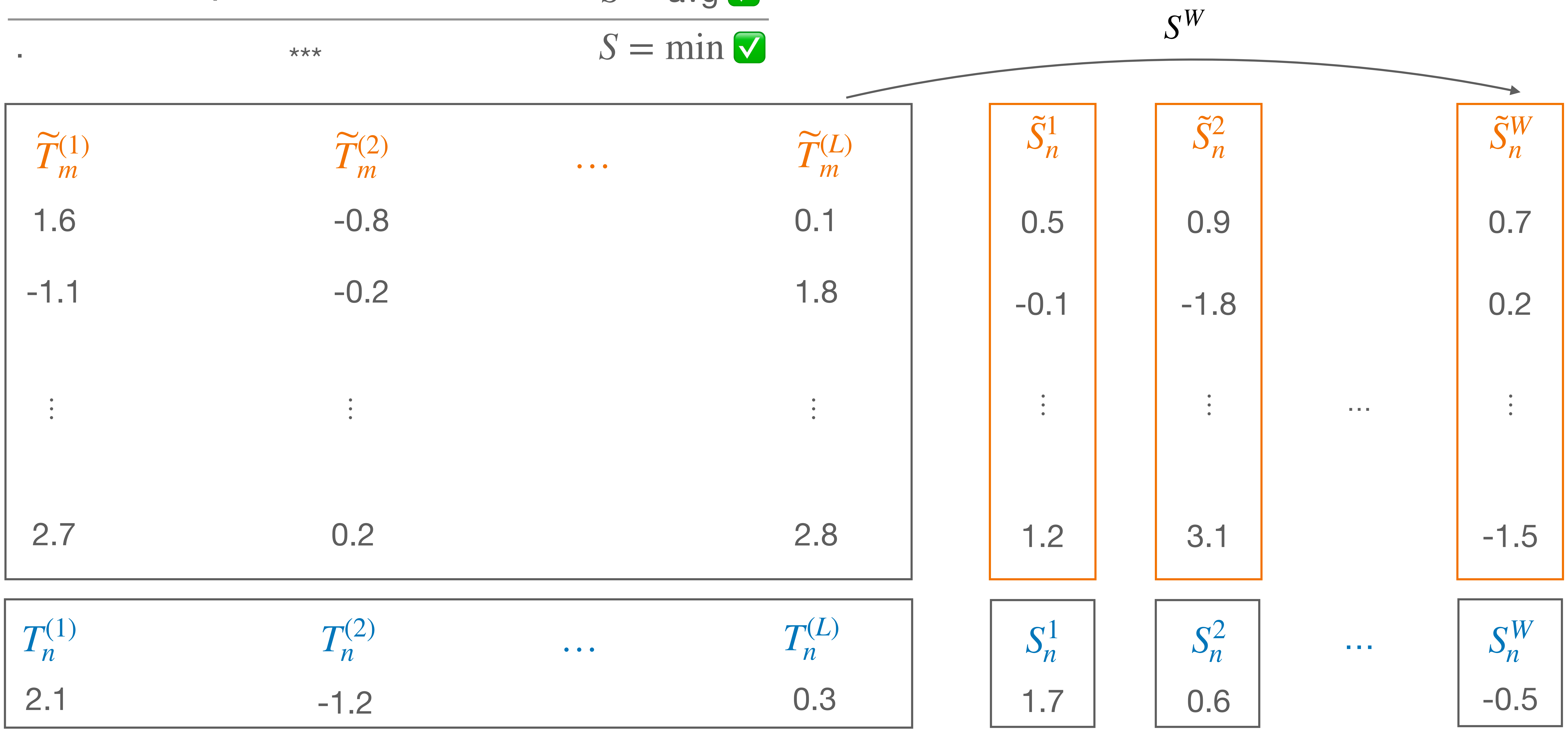
$\widetilde{T}_m^{(1)}$	$\widetilde{T}_m^{(2)}$	$\ldots$	$\widetilde{T}_m^{(L)}$	$\tilde{S}_n^1$	$\tilde{S}_n^2$
1.6	-0.8		0.1	0.5	0.9
-1.1	-0.2		1.8	-0.1	-1.8
$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$
2.7	0.2		2.8	1.2	3.1

Observed	$T_n^{(1)}$	$T_n^{(2)}$	$\ldots$	$T_n^{(L)}$	S_n^1	S_n^2
	2.1	-1.2		0.3	1.7	0.6

Multiple aggregations: Adaptive algorithm

$T_n^{(1)}$	$T_n^{(2)}$	$T_n^{(3)}$	$T_n^{(4)}$	$\cdots$	$T_n^{(L)}$	🤔
*	**	.	*		*	$S = \text{avg}$ ✅
.			***			$S = \min$ ✅

💡 Allow the user to specify $S^1, \ldots, S^W$



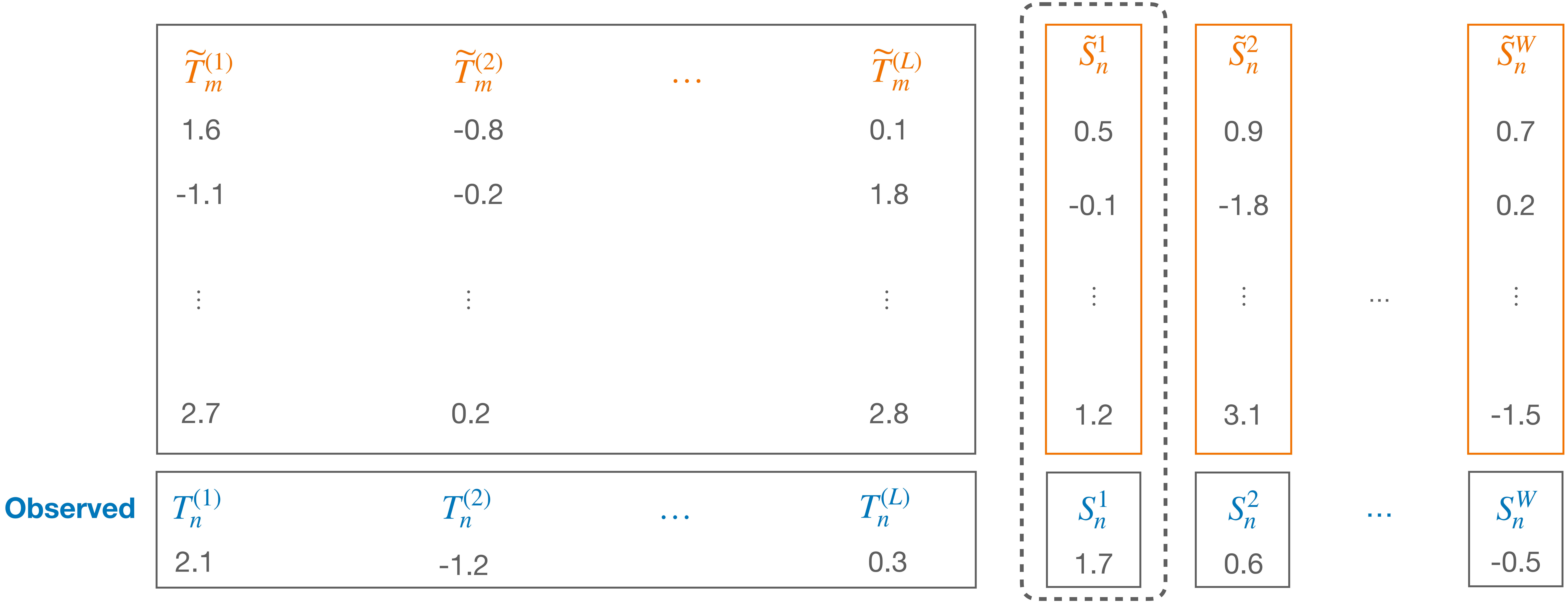
$\widetilde{T}_m^{(1)}$	$\widetilde{T}_m^{(2)}$	$\ldots$	$\widetilde{T}_m^{(L)}$	$\tilde{S}_n^1$	$\tilde{S}_n^2$		$\tilde{S}_n^W$
1.6	-0.8		0.1	0.5	0.9		0.7
-1.1	-0.2		1.8	-0.1	-1.8		0.2
⋮	⋮		⋮	⋮	⋮	…	⋮
2.7	0.2		2.8	1.2	3.1		-1.5

	$T_n^{(1)}$	$T_n^{(2)}$	$\ldots$	$T_n^{(L)}$	S_n^1	S_n^2	…	S_n^W
Observed	2.1	-1.2		0.3	1.7	0.6		-0.5

Multiple aggregations: Adaptive algorithm

$T_n^{(1)}$	$T_n^{(2)}$	$T_n^{(3)}$	$T_n^{(4)}$	$\cdots$	$T_n^{(L)}$	🤔
*	**	.	*		*	$S =$ avg ✅
.			***			$S = \min$ ✅

💡 Allow the user to specify $S^1, \ldots, S^W$



	$\widetilde{T}_m^{(1)}$	$\widetilde{T}_m^{(2)}$	$\ldots$	$\widetilde{T}_m^{(L)}$	$\tilde{S}_n^1$	$\tilde{S}_n^2$		$\tilde{S}_n^W$
	1.6	-0.8		0.1	0.5	0.9		0.7
	-1.1	-0.2		1.8	-0.1	-1.8		0.2
	⋮	⋮		⋮	⋮	⋮	…	⋮
	2.7	0.2		2.8	1.2	3.1		-1.5
Observed	$T_n^{(1)}$	$T_n^{(2)}$	$\ldots$	$T_n^{(L)}$	S_n^1	S_n^2	…	S_n^W
	2.1	-1.2		0.3	1.7	0.6		-0.5

Multiple aggregations: Adaptive algorithm

$T_n^{(1)}$	$T_n^{(2)}$	$T_n^{(3)}$	$T_n^{(4)}$	$\cdots$	$T_n^{(L)}$	🤔
*	**	.	*		*	$S =$ avg ✅
.			***			$S = \min$ ✅

💡 Allow the user to specify $S^1, \ldots, S^W$

	$\widetilde{T}_m^{(1)}$	$\widetilde{T}_m^{(2)}$	$\ldots$	$\widetilde{T}_m^{(L)}$	$\tilde{S}_n^1$	$\tilde{S}_n^2$	$\cdots$	$\tilde{S}_n^W$
	1.6	-0.8		0.1	200	0.9		0.7
	-1.1	-0.2		1.8	77	-1.8		0.2
	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$
	2.7	0.2		2.8	431	3.1		-1.5
Observed	$T_n^{(1)}$	$T_n^{(2)}$	$\ldots$	$T_n^{(L)}$	S_n^1	S_n^2	$\cdots$	S_n^W
	2.1	-1.2		0.3	489	0.6		-0.5

Multiple aggregations: Adaptive algorithm

$T_n^{(1)}$	$T_n^{(2)}$	$T_n^{(3)}$	$T_n^{(4)}$	$\cdots$	$T_n^{(L)}$	🤔
*	**	.	*		*	$S =$ avg ✅
.			***			$S = \min$ ✅

💡 Allow the user to specify $S^1, \ldots, S^W$

$\widetilde{T}_m^{(1)}$	$\widetilde{T}_m^{(2)}$	$\ldots$	$\widetilde{T}_m^{(L)}$	$\tilde{S}_n^1$	$\tilde{S}_n^2$	$\cdots$	$\tilde{S}_n^W$
1.6	-0.8		0.1	200	142		289
-1.1	-0.2		1.8	77	33		260
$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$		$\vdots$
2.7	0.2		2.8	431	460		12

Observed

$T_n^{(1)}$	$T_n^{(2)}$	$\ldots$	$T_n^{(L)}$	S_n^1	S_n^2	$\cdots$	S_n^W
2.1	-1.2		0.3	489	281		32

Multiple aggregations: Adaptive algorithm

$T_n^{(1)}$	$T_n^{(2)}$	$T_n^{(3)}$	$T_n^{(4)}$	$\cdots$	$T_n^{(L)}$	🤔
*	**	.	*		*	$S =$ avg ✅
.			***			$S = \min$ ✅

💡 Allow the user to specify $S^1, \ldots, S^W$

$\min$ ✅

	$\widetilde{T}_m^{(1)}$	$\widetilde{T}_m^{(2)}$	$\ldots$	$\widetilde{T}_m^{(L)}$	$\tilde{S}_n^1$	$\tilde{S}_n^2$	$\cdots$	$\tilde{S}_n^W$	$\tilde{R}_n$
	1.6	-0.8		0.1	200	142		289	**110**
	-1.1	-0.2		1.8	77	33		260	**25**
	⋮	⋮		⋮	⋮	⋮	$\cdots$	⋮	⋮
	2.7	0.2		2.8	431	460		12	**7**
Observed	$T_n^{(1)}$	$T_n^{(2)}$	$\ldots$	$T_n^{(L)}$	S_n^1	S_n^2	$\cdots$	S_n^W	R_n
	2.1	-1.2		0.3	489	281		32	**16**

Hunt and test: Flexible goodness-of-fit

Hunt and test: Flexible goodness-of-fit

Linear model $Y \sim \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$

Hunt and test: Flexible goodness-of-fit

Linear model $Y \sim \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$

Consider introducing a new covariate $X_{p+1} := \xi(X)$ for as a non-linear $\xi(\,\cdot\,)$.

Hunt and test: Flexible goodness-of-fit

Linear model $Y \sim \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$

Consider introducing a new covariate $X_{p+1} := \xi(X)$ for as a non-linear $\xi(\,\cdot\,)$.

💡 If linear model is well-specified, then the should have $\beta_{p+1} = 0$ in

$$Y \sim \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p + \beta_{p+1} X_{p+1}.$$

Hunt and test: Flexible goodness-of-fit

Linear model $Y \sim \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$

Consider introducing a new covariate $X_{p+1} := \xi(X)$ for as a non-linear $\xi(\cdot)$.

💡 If linear model is well-specified, then the should have $\beta_{p+1} = 0$ in

$$Y \sim \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p + \beta_{p+1} X_{p+1}.$$

👉 Test $\cap_\xi \{H_0(\xi) : \beta_{p+1} = 0\}$.

Hunt and test: Flexible goodness-of-fit

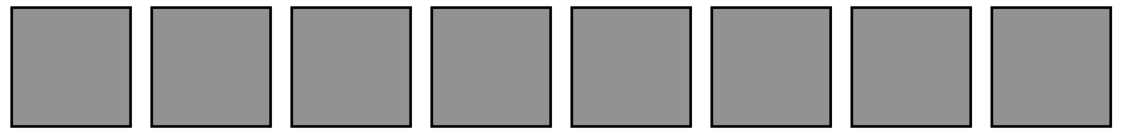
Linear model $Y \sim \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$

Consider introducing a new covariate $X_{p+1} := \xi(X)$ for as a non-linear $\xi(\cdot)$.

💡 If linear model is well-specified, then the should have $\beta_{p+1} = 0$ in

$$Y \sim \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p + \beta_{p+1} X_{p+1}.$$

👉 Test $\cap_\xi \{H_0(\xi) : \beta_{p+1} = 0\}$.



Hunt and test: Flexible goodness-of-fit

Linear model $Y \sim \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$

Consider introducing a new covariate $X_{p+1} := \xi(X)$ for as a non-linear $\xi(\,\cdot\,)$.

💡 If linear model is well-specified, then the should have $\beta_{p+1} = 0$ in

$$Y \sim \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p + \beta_{p+1} X_{p+1}.$$

👉 Test $\cap_\xi \{H_0(\xi) : \beta_{p+1} = 0\}$.

🟧🟦🟧🟦🟦🟧🟦🟧

🎲

Hunt and test: Flexible goodness-of-fit

Linear model $Y \sim \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$

Consider introducing a new covariate $X_{p+1} := \xi(X)$ for as a non-linear $\xi(\cdot)$.

💡 If linear model is well-specified, then the should have $\beta_{p+1} = 0$ in

$$Y \sim \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p + \beta_{p+1} X_{p+1}.$$

👉 Test $\cap_\xi \{H_0(\xi) : \beta_{p+1} = 0\}$.

🟧🟦🟧🟦🟦🟧🟦🟧 (1) Use 🟦🟦🟦🟦 to find $\hat{\xi}$ such that $X_{p+1} = \xi(X)$ is likely to be “significant”.

Hunt and test: Flexible goodness-of-fit

Linear model $Y \sim \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$

Consider introducing a new covariate $X_{p+1} := \xi(X)$ for as a non-linear $\xi(\cdot)$.

💡 If linear model is well-specified, then the should have $\beta_{p+1} = 0$ in

$$Y \sim \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p + \beta_{p+1} X_{p+1}.$$

👉 Test $\cap_\xi \{H_0(\xi) : \beta_{p+1} = 0\}$.

🟧🟦🟧🟦🟦🟧🟦🟧

(1) Use 🟦🟦🟦🟦 to find $\hat{\xi}$ such that $X_{p+1} = \xi(X)$ is likely to be “significant”.

(2) Use 🟧🟧🟧🟧 to compute a test statistic for $\beta_{p+1} = 0$.

👆 Use any existing device for parameter inference.

Hunt and test: Flexible goodness-of-fit

Linear model $Y \sim \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$

Consider introducing a new covariate $X_{p+1} := \xi(X)$ for as a non-linear $\xi(\cdot)$.

💡 If linear model is well-specified, then the should have $\beta_{p+1} = 0$ in

$$Y \sim \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p + \beta_{p+1} X_{p+1}.$$

👉 Test $\cap_\xi \{H_0(\xi) : \beta_{p+1} = 0\}$.

🟧🟦🟧🟦🟦🟧🟦🟧

(1) Use 🟦🟦🟦🟦 to find $\hat{\xi}$ such that $X_{p+1} = \xi(X)$ is likely to be “significant”.

(2) Use 🟧🟧🟧🟧 to compute a test statistic for $\beta_{p+1} = 0$.

👆 Use any existing device for parameter inference.

🤔 How to find $\hat{\xi}$?

Hunt and test: Flexible goodness-of-fit

Linear model $Y \sim \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$

Consider introducing a new covariate $X_{p+1} := \xi(X)$ for as a non-linear $\xi(\cdot)$.

💡 If linear model is well-specified, then the should have $\beta_{p+1} = 0$ in

$$Y \sim \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p + \beta_{p+1} X_{p+1}.$$

👉 Test $\cap_\xi \{H_0(\xi) : \beta_{p+1} = 0\}$.

🟧🟦🟧🟦🟦🟧🟦🟧

(1) Use 🟦🟦🟦🟦 to find $\hat{\xi}$ such that $X_{p+1} = \xi(X)$ is likely to be "significant".

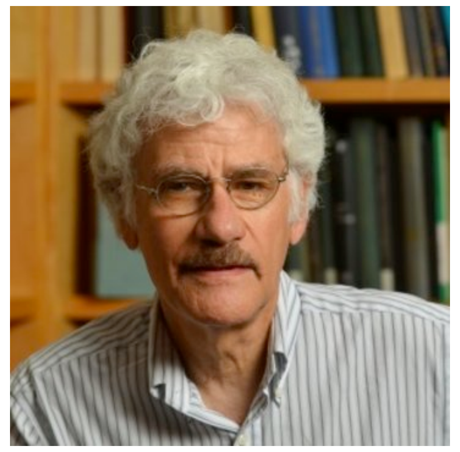
(2) Use 🟧🟧🟧🟧 to compute a test statistic for $\beta_{p+1} = 0$.

👆 Use any existing device for parameter inference.

🤔 How to find $\hat{\xi}$?

💡 **Gradient boosting!**

Jerome H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of Statistics* (2001).



Hunt and test: Flexible goodness-of-fit

Regression: $\min \mathbb{E} l(Y - \beta^\top X)$ for an **arbitrary** loss function $l(\,\cdot\,)$.

Hunt and test: Flexible goodness-of-fit

Regression: $\min \mathbb{E} l(Y - \beta^\top X)$ for an **arbitrary** loss function $l(\,\cdot\,)$.

Fitted $Y \sim \hat{\beta}^\top X$. With new covariate X_{p+1},

$$\sum_i l(Y_i - \hat{\beta}^\top X_i - \beta_{p+1} X_{i,p+1}) \approx \sum_i l(Y_i - \hat{\beta}^\top X_i) - \beta_i \sum_i l'(Y_i - \hat{\beta}^\top X_i)\, X_{i,p+1}$$

Hunt and test: Flexible goodness-of-fit

Regression: $\min \mathbb{E} l(Y - \beta^\top X)$ for an **arbitrary** loss function $l(\cdot)$.

Fitted $Y \sim \hat{\beta}^\top X$. With new covariate X_{p+1},

$$\sum_i l(Y_i - \hat{\beta}^\top X_i - \beta_{p+1} X_{i,p+1}) \approx \sum_i l(Y_i - \hat{\beta}^\top X_i) - \beta_i \sum_i l'(Y_i - \hat{\beta}^\top X_i)\, X_{i,p+1}$$

(1) On 🟦🟦🟦🟦 : Train any ML algorithm $\hat{\xi}$ to predict l'(resid) from X.

Hunt and test: Flexible goodness-of-fit

Regression: $\min \mathbb{E} l(Y - \beta^\top X)$ for an **arbitrary** loss function $l(\cdot)$.

Fitted $Y \sim \hat{\beta}^\top X$. With new covariate X_{p+1},

$$\sum_i l(Y_i - \hat{\beta}^\top X_i - \beta_{p+1} X_{i,p+1}) \approx \sum_i l(Y_i - \hat{\beta}^\top X_i) - \beta_i \sum_i l'(Y_i - \hat{\beta}^\top X_i)\, X_{i,p+1}$$

(1) On 🟦🟦🟦🟦 : Train any ML algorithm $\hat{\xi}$ to predict l'(resid) from X.

(2) On 🟧🟧🟧🟧 : Compute statistic for testing $\beta_{p+1} = 0$ in $Y \sim \beta^\top X + \beta_{p+1}\, \hat{\xi}(X)$.

Hunt and test: Flexible goodness-of-fit

Hunt and test: Flexible goodness-of-fit

Quantile regression

Hunt and test: Flexible goodness-of-fit

Quantile regression

1. Linear model is widely used.

Hunt and test: Flexible goodness-of-fit

Quantile regression

1. Linear model is widely used.

2. Developing goodness/lack-of-fit test is difficult.

   e.g., Zheng (1998), Horowitz & Spokoiny (2002), He & Zhu (2003),
   Escanciano and Velasco (2010), Escanciano & Goh (2014).

   (1) Asymptotic theory of certain residual statistics/processes.

   (2) Performance deteriorates when p is moderate or large.

Hunt and test: Flexible goodness-of-fit

Quantile regression

1. Linear model is widely used.

2. Developing goodness/lack-of-fit test is difficult.

   e.g., Zheng (1998), Horowitz & Spokoiny (2002), He & Zhu (2003), Escanciano and Velasco (2010), Escanciano & Goh (2014).

   (1) Asymptotic theory of certain residual statistics/processes.

   (2) Performance deteriorates when p is moderate or large.

3. Moderate/large p: active research.

   e.g., Conde-Amboage et al. (2015), Dong et al. (2019).

Hunt and test: Flexible goodness-of-fit

Quantile regression

1. Linear model is widely used.

2. Developing goodness/lack-of-fit test is difficult.

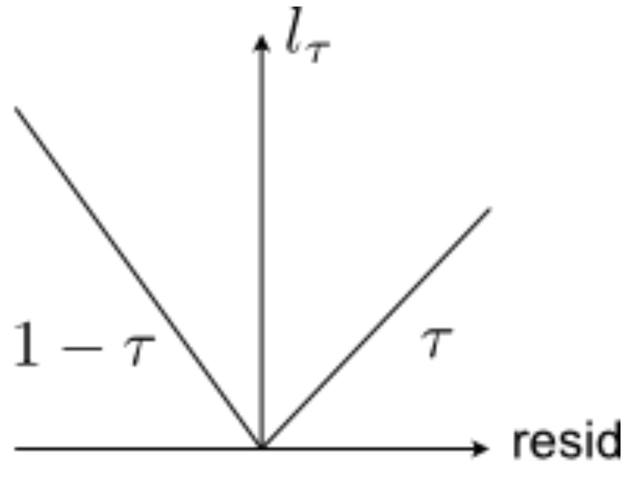
   e.g., Zheng (1998), Horowitz & Spokoiny (2002), He & Zhu (2003), Escanciano and Velasco (2010), Escanciano & Goh (2014).

   (1) Asymptotic theory of certain residual statistics/processes.

   (2) Performance deteriorates when p is moderate or large.

3. Moderate/large p: active research.

   e.g., Conde-Amboage et al. (2015), Dong et al. (2019).



$\hat{\xi}$: random forest classifier sign(resid) $\sim X$.

T_n: standard "t-value" from `quantreg`.

Hunt and test: Flexible goodness-of-fit

Quantile regression

$\tau = 0.5$ (median)

$$Y = 1 + \beta_0^\top X + 4\, v\, n^{-1/2}\sqrt{X_1^2 + X_2^2} + (1 + X_2 + X_3)\,\varepsilon$$

1. Linear model is widely used.

2. Developing goodness/lack-of-fit test is difficult.

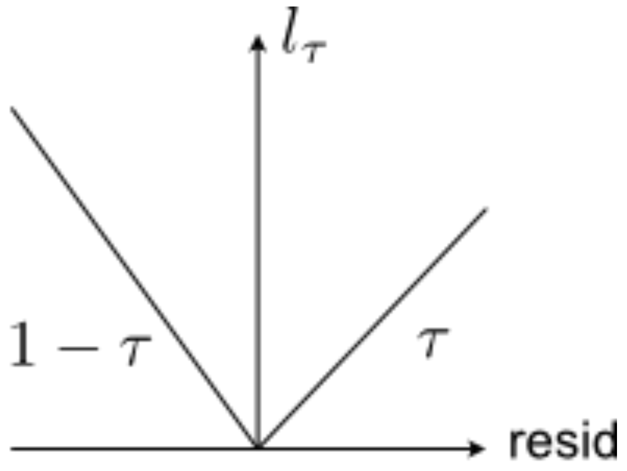
   e.g., Zheng (1998), Horowitz & Spokoiny (2002), He & Zhu (2003), Escanciano and Velasco (2010), Escanciano & Goh (2014).

   (1) Asymptotic theory of certain residual statistics/processes.

   (2) Performance deteriorates when p is moderate or large.

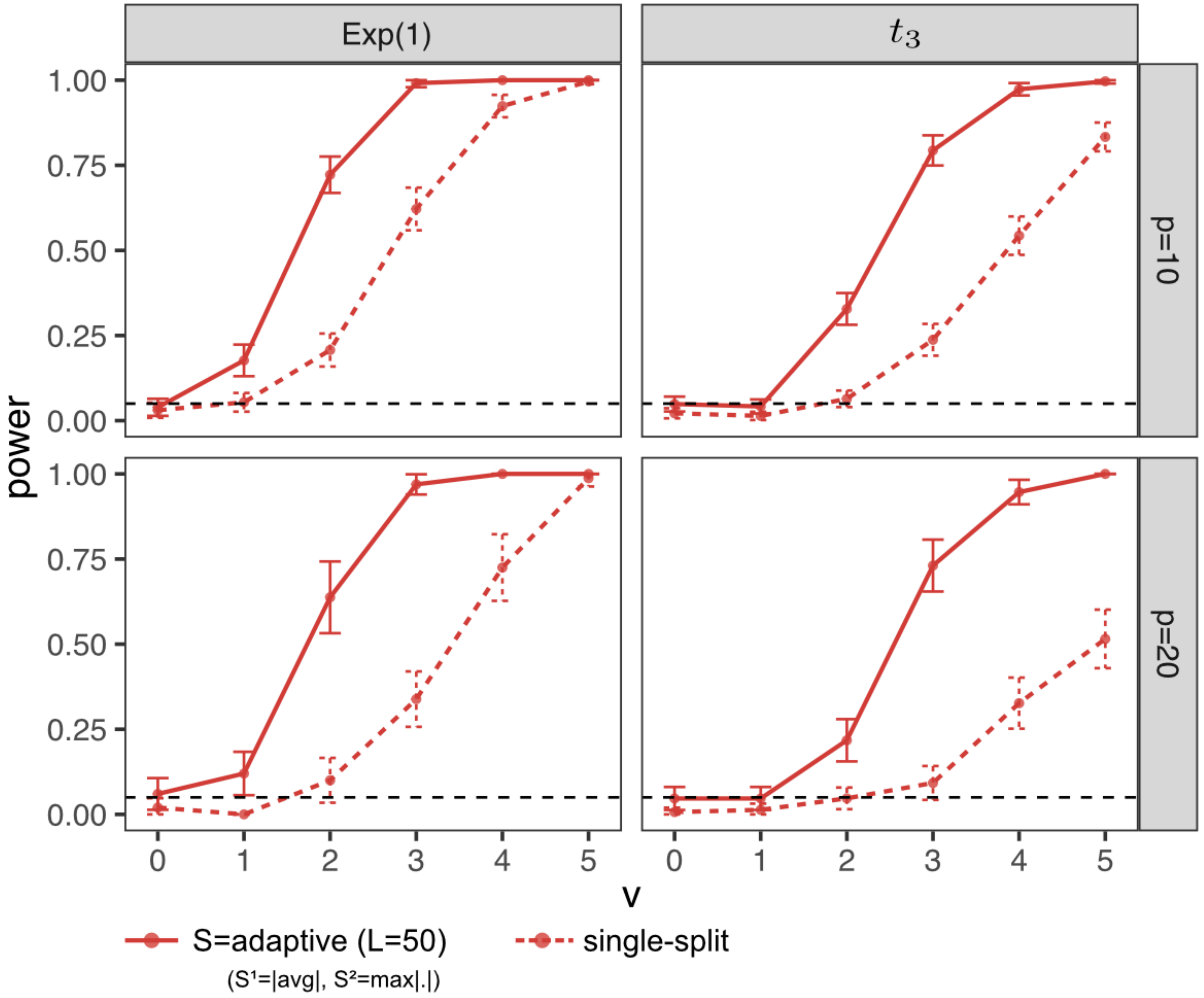
3. Moderate/large p: active research.

   e.g., Conde-Amboage et al. (2015), Dong et al. (2019).



$\hat{\xi}$: random forest classifier sign(resid) $\sim X$.

T_n: standard “t-value” from `quantreg`.



Hunt and test: Flexible goodness-of-fit

Quantile regression

1. Linear model is widely used.

2. Developing goodness/lack-of-fit test is difficult.

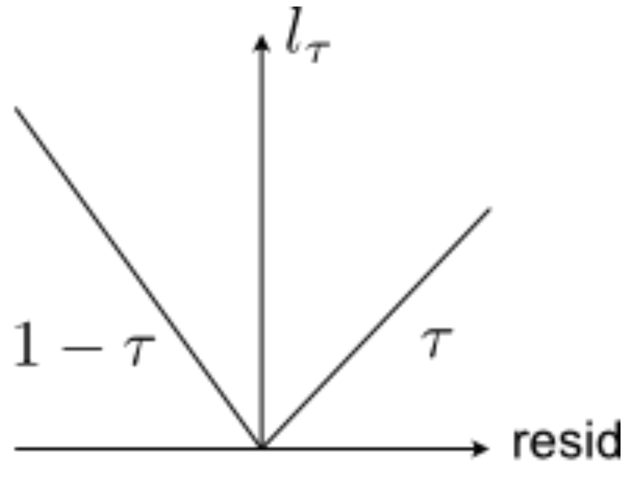
   e.g., Zheng (1998), Horowitz & Spokoiny (2002), He & Zhu (2003), Escanciano and Velasco (2010), Escanciano & Goh (2014).

   (1) Asymptotic theory of certain residual statistics/processes.

   (2) Performance deteriorates when p is moderate or large.

3. Moderate/large p: active research.

   e.g., Conde-Amboage et al. (2015), Dong et al. (2019).



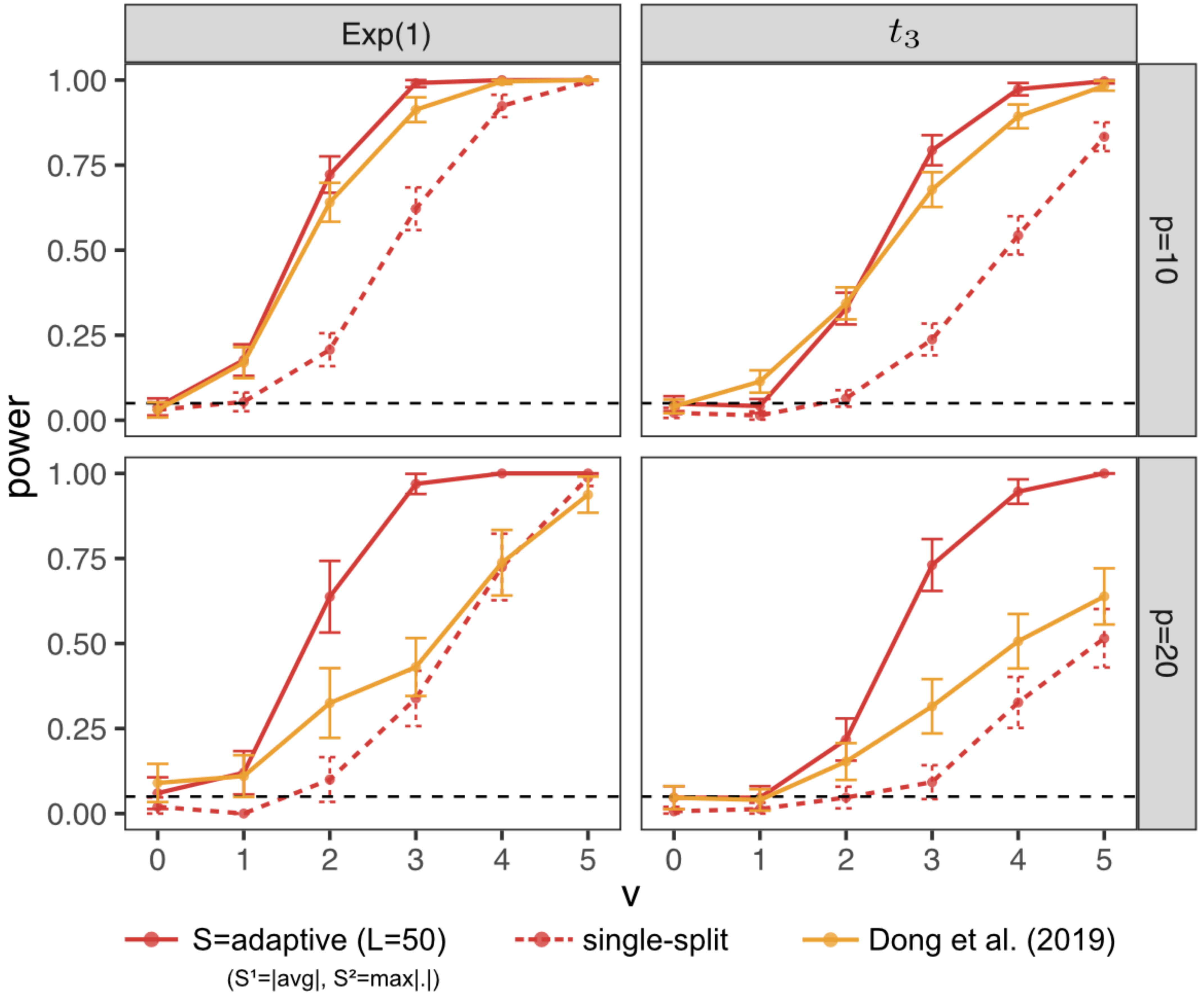
$\hat{\xi}$: random forest classifier sign(resid) $\sim X$.

T_n: standard “t-value” from `quantreg`.

Chen Dong, Guodong Li, and Xingdong Feng.
Lack-of-fit tests for quantile regression models.
Journal of the Royal Statistical Society: Series B (2019).

$\tau = 0.5$ (median)

$$Y = 1 + \beta_0^\top X + 4\, v\, n^{-1/2}\sqrt{X_1^2 + X_2^2} + (1 + X_2 + X_3)\,\varepsilon$$


Exp(1)
t_3
p=10
p=20
power
1.00
0.75
0.50
0.25
0.00
0
1
2
3
4
5
v
S=adaptive (L=50)
(S^1=|avg|, S^2=max|.|)
single-split
Dong et al. (2019)