A Framework for Efficient Model Evaluation via Stratification, Sampling, and Estimation

Riccardo Fogliato (AWS Themis Responsible AI)

arXiv: https://arxiv.org/abs/2406.07320 (to appear also at ECCV ’24)
GitHub: https://github.com/amazon-science/ssepy

Joint work with

Pratik Patil

UC Berkeley

Mathew Monfort

AWS

Pietro Perona

Caltech & AWS

A Framework for **Quick and Cheap** Model Evaluation via **Tailored Inference Strategies**

Why quick and cheap evaluations?

TPM

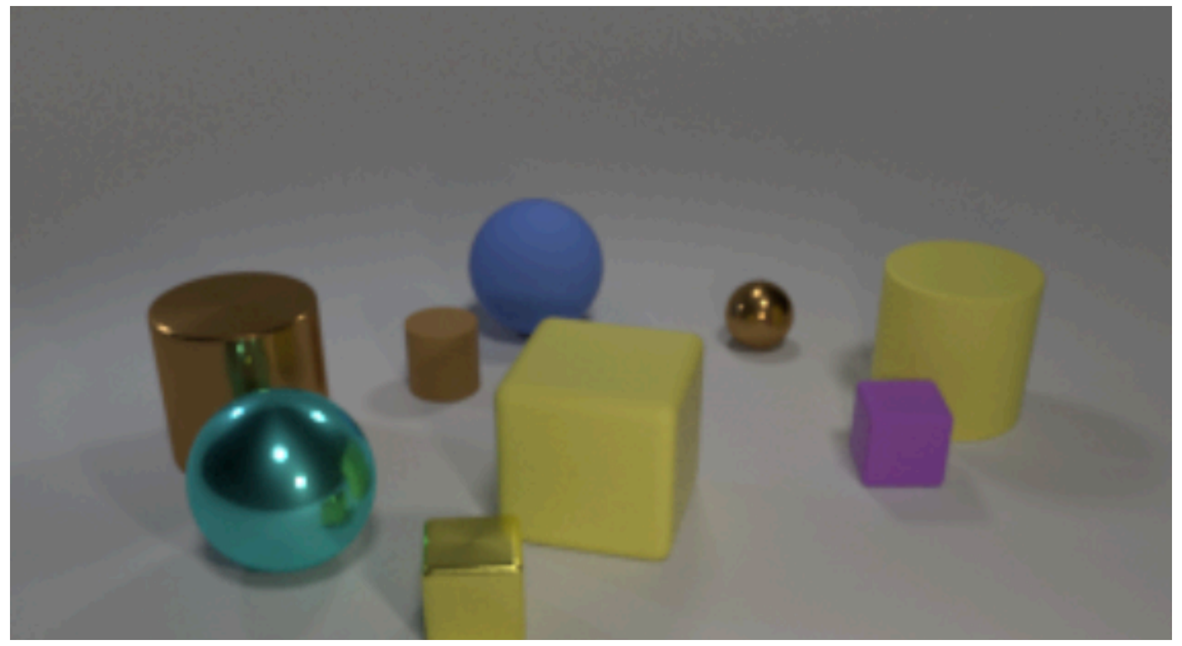
Can you quickly check the accuracy of our CV service on the customer's data?

FYI deadline is tomorrow

Scientist

No way… It will take 10 days and cost **$5k** to obtain ground-truth labels! :(

Customer data



">5 objects?"

ML API: yes

ML confidence: 90%

Why quick and cheap evaluations?

TPM

Can you quickly check the accuracy of our CV service on the customer's data?

FYI deadline is tomorrow

Scientist

No way… It will take 10 days and cost **$5k** to obtain ground-truth labels! :(

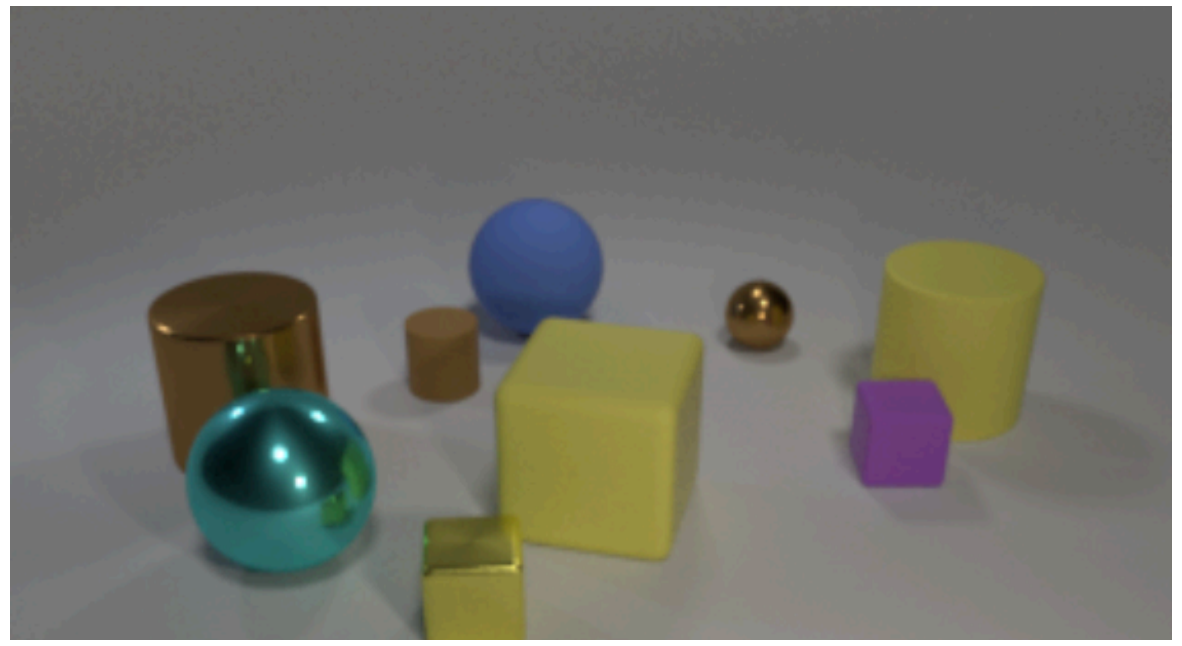
Senior Scientist

Don't worry TPM…It'll take 1 day and cost **$<1k** with sampling techniques and prediction-powered inference!

Scientist

What's this??

Customer data

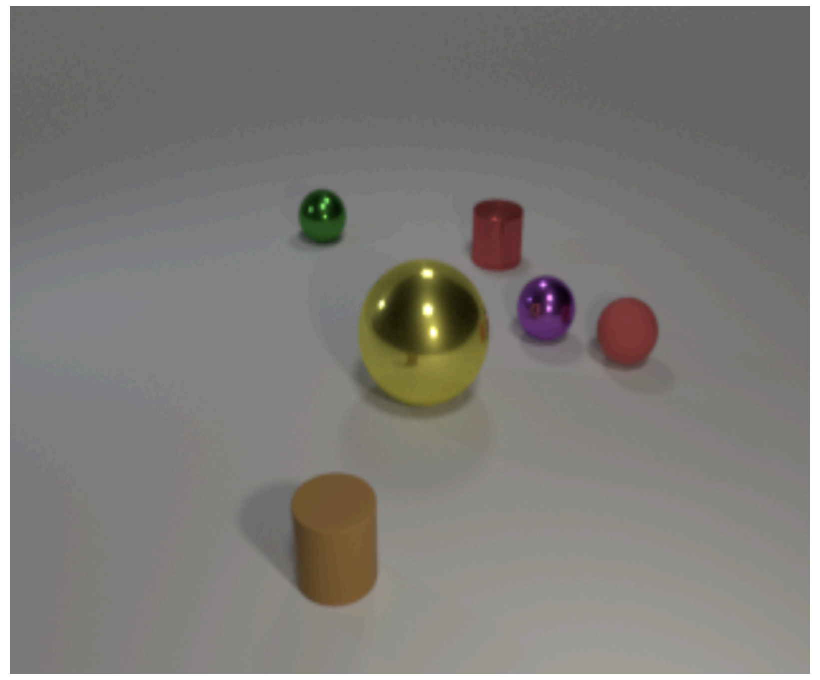


">5 objects?"
ML API: yes
ML confidence: 90%

Ideal estimation setup

X $\hat{Y}$ $f(X)$ Z

$\mathscr{D} = \{($

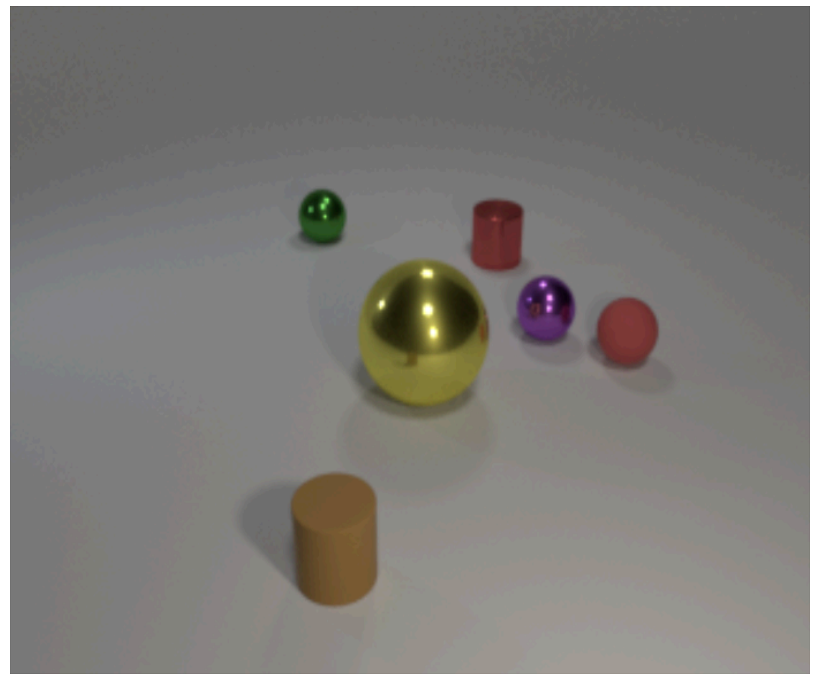


, ML prediction: >5, confidence: 50%, ML correct: yes $)\}_{i=1}^{N}$

$$\Rightarrow \theta = \frac{1}{N} \sum_{i \in \mathscr{D}} Z_i$$

Ground truth labels often are not available

$$\mathscr{D} = \{(X, \hat{Y}, f(X), Z)\}_{i=1}^{N}$$



ML prediction: >5

confidence: 50%

~~ML c : yes~~

$$\Rightarrow \cancel{\theta = \sum Z_i}$$

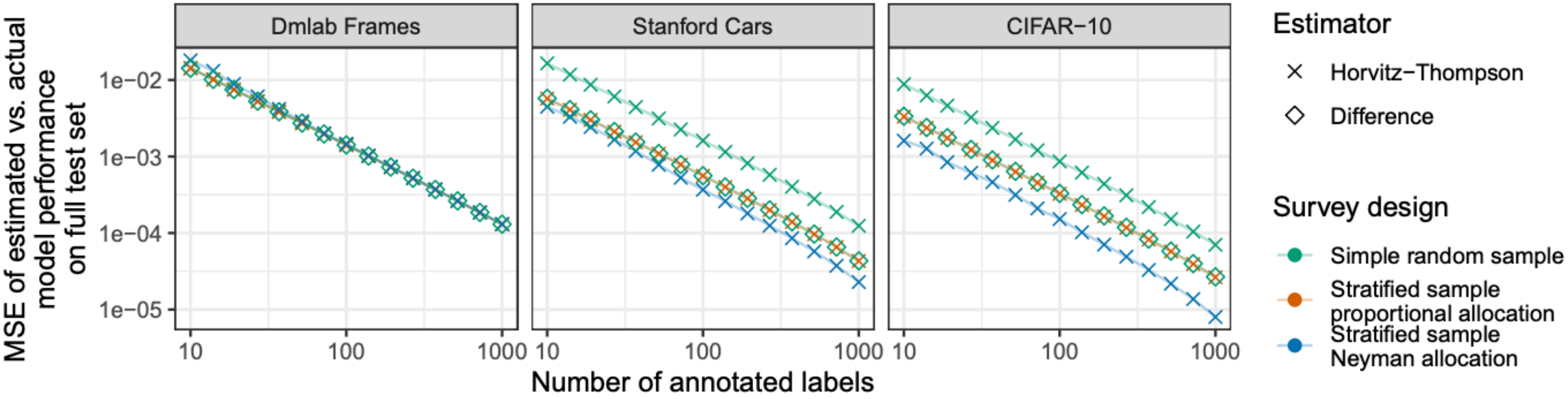
Workflows

Inputs: test data $\mathscr{D} = \{(X_i, \hat{Y}_i, f(X_i)\}_{i=1}^{N}$ **and annotation budget** n

Scientist workflow

Senior scientist workflow

Which method you use does matter



Large-scale benchmarking

$$\mathscr{D} = \left\{ \left(\overset{X}{\boxed{\begin{array}{c}\text{~40 datasets from}\\ \text{CLIP benchmark}\end{array}}}, \overset{\hat{Y}}{\boxed{\begin{array}{c}\text{Zero-shot CLIP w/}\\ \text{ViT-B-32}\end{array}}}, \overset{f(X)}{\boxed{\begin{array}{c}\text{Zero-shot CLIP w/}\\ \text{ViT-B-32 or CLIP}\\ \text{w/ ViT-L-14}\end{array}}}, \overset{Z}{\boxed{\text{Ground-truth labels}}} \right) \right\}_{\mathbf{i=1}}^{\mathbf{N}}$$

Sampling workflows

Inputs: test data $\mathscr{D} = \{(X_i, \hat{Y}_i, f(X_i)\}_{i=1}^{N}$ and annotation budget n

Simple random sampling (SRS)

➡ $$\hat{\theta} = \frac{1}{n} \sum_{i \in \mathcal{S}} Z_i$$

Stratified simple random sampling (SSRS)

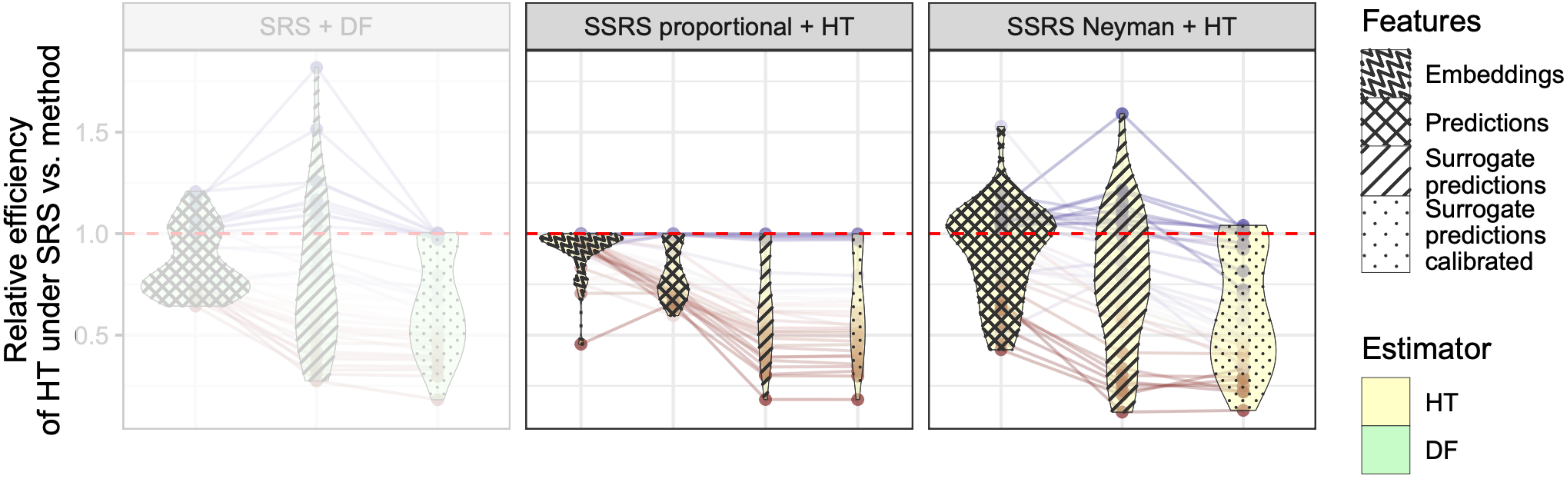
- proportional allocation $n_h \propto N_h / N$
- Neyman or optimal allocation $n_h \propto \sqrt{\text{Var}_h(Z)}$

➡ $$\hat{\theta} = \sum_{h=1}^{H} \frac{N_h}{N} \frac{1}{n_h} \sum_{i \in \mathcal{S}_h} Z_i$$

Result:

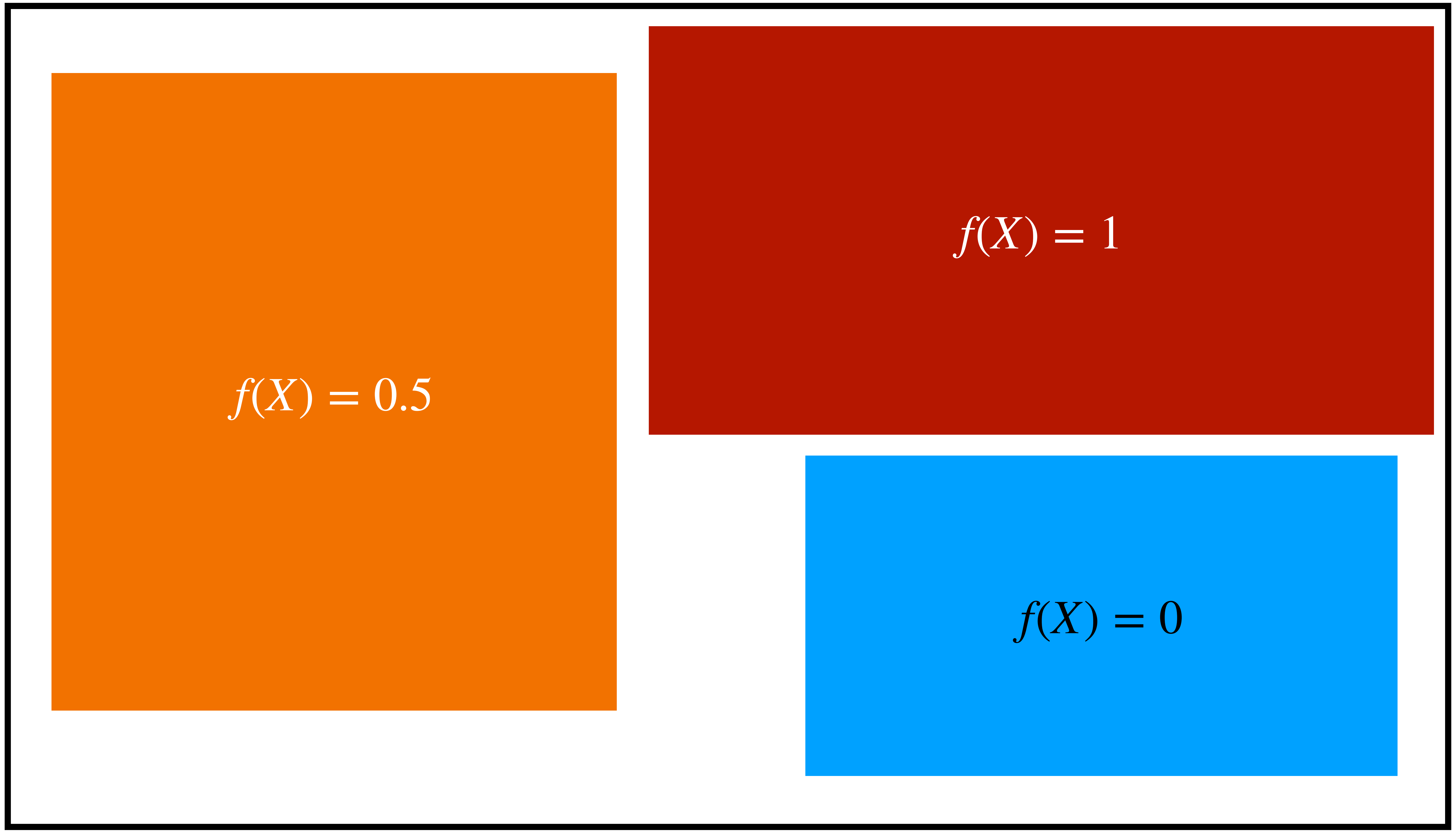
$$\text{Var}_{\text{SRS}}(\hat{\theta}) \geq \text{Var}_{\text{prop}}(\hat{\theta}) \geq \text{Var}_{\text{opt}}(\hat{\theta})$$

Stratified sampling with proportional allocation consistently yields good results. Neyman can help

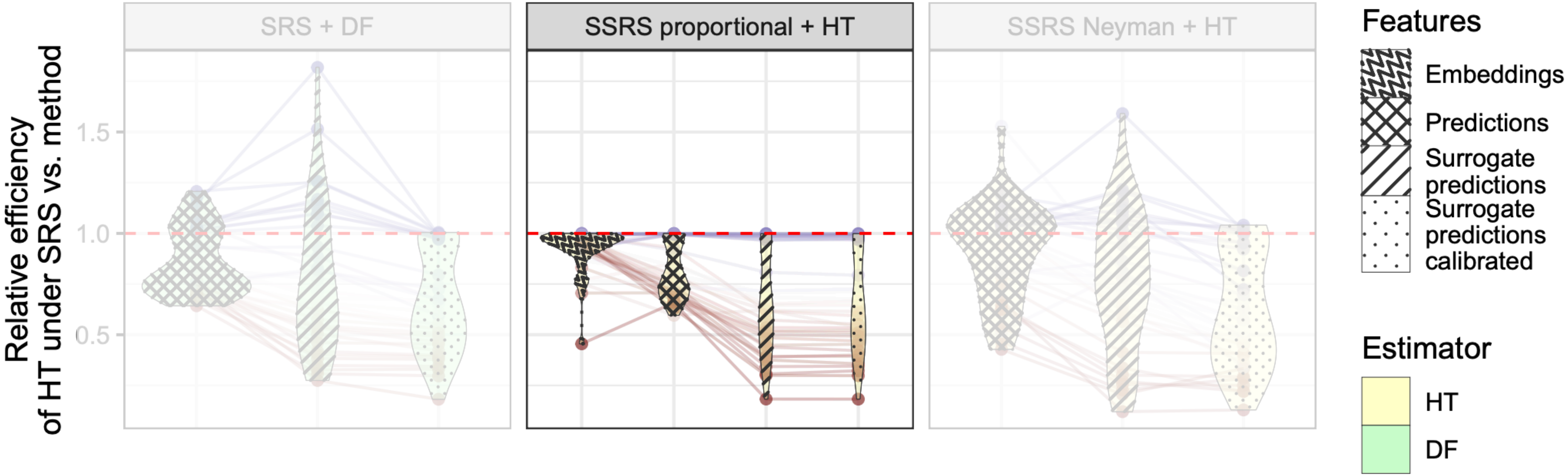


How to best stratify?

(hint: you should minimize Var$_{\text{prop}}(\hat{\theta})$ so stratify by $f(X) \approx Z$)



Stratifying on a more accurate $f(X)$ means lower variance



Estimation workflows

Inputs: $\mathscr{D} = \{(X_i, \hat{Y}_i, f(X_i)\}_{i=1}^{N}$ **and annotation budget** n

Horvitz-Thompson (HT) estimator

$$\hat{\theta}_{\text{HT}} = \frac{1}{n}\sum_{i\in\mathcal{S}} Z_i$$

Difference (DF) estimator (aka prediction-powered)

- Uses both labeled and unlabeled data

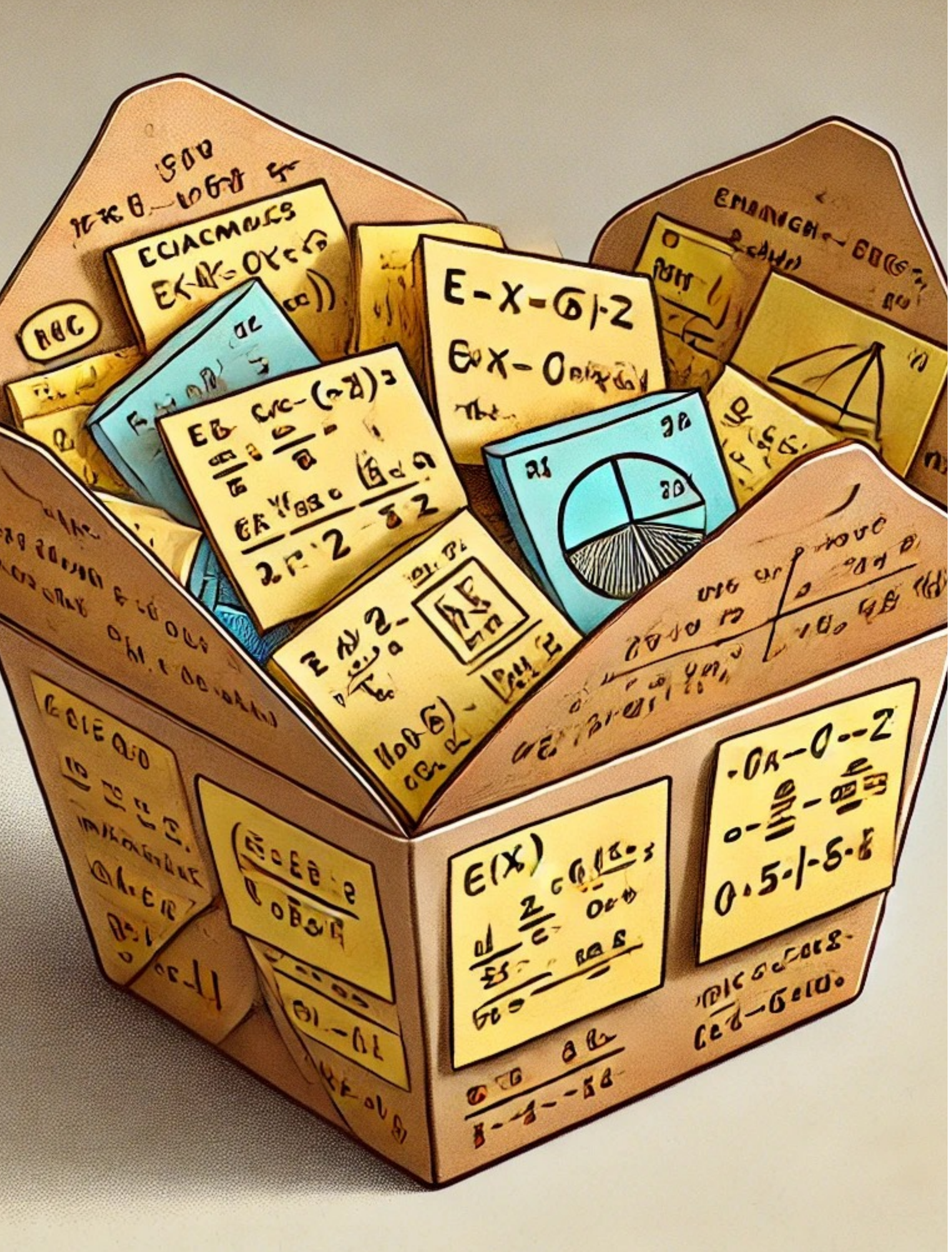
$$\hat{\theta}_{\text{DF}} = \frac{1}{N}\sum_{i\in\mathscr{D}} f(X_i) + \frac{1}{n}\sum_{i\in\mathcal{S}} (Z_i - f(X_i))$$

Result:

Var$(\hat{\theta}_{\text{HT}}) \geq$ **Var**$(\hat{\theta}_{\text{DF}})$

Difference estimator generally increases the precision of the estimates

power tuning can resolve the underperformance issues (see PPI++)



Takeaways

Always stratify by ML predictions and allocate budget proportionally

If ML predictions are accurate, Neyman allocation can help

If you use simple random sampling, estimate with the difference estimator w/ power tuning

Thank you!

Questions?

Email: fogliato@amazon.com