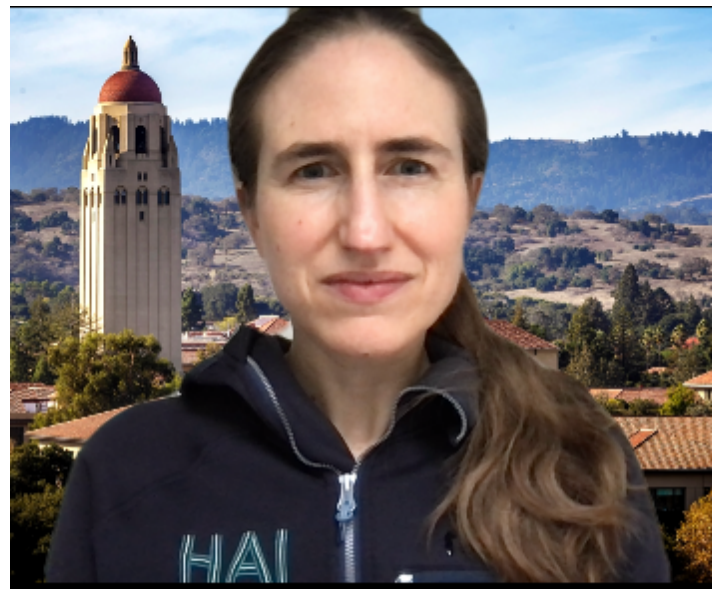
Data-Efficient Pipeline for Offline Reinforcement Learning with Limited Data

Allen Nie, Yannis Flet-Berliac, Deon R. Jordan, William Steenbergen, Emma Brunskill



Stanford University

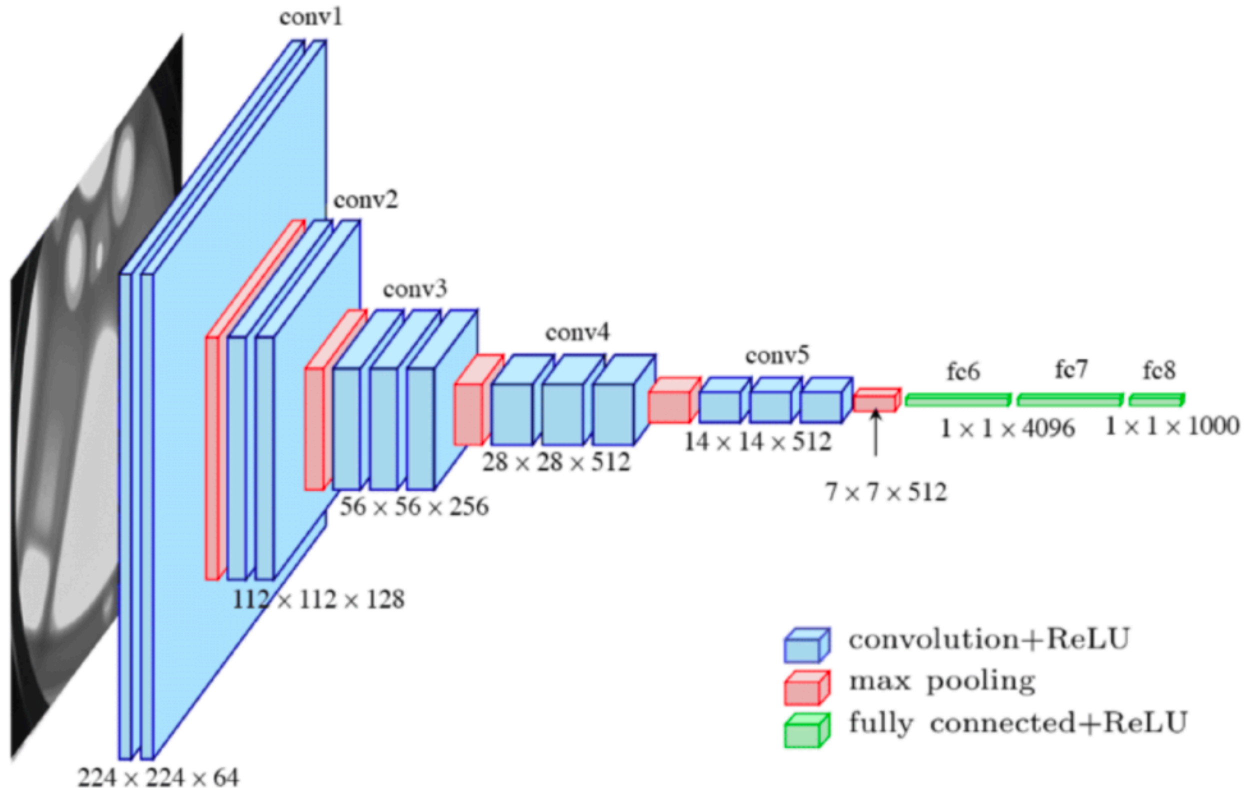
NeurIPS 2022
AAAI 2023 RL4Production Workshop

Modern Machine Learning Workflow

Architecture / Model / Hyperparameter selection using validation set

Training Data

$\{x_i, y_i\}$



Output Predictor

$\hat{f}$

Validation Data

$\{x_i, y_i\}$

$\hat{f}$

Evaluation Functions

$$\frac{\sum_i 1[y_i = \hat{f}(x_i)]}{N}$$

Predictor	Accuracy
256-dim CNN	82%
512-dim CNN	**91%**

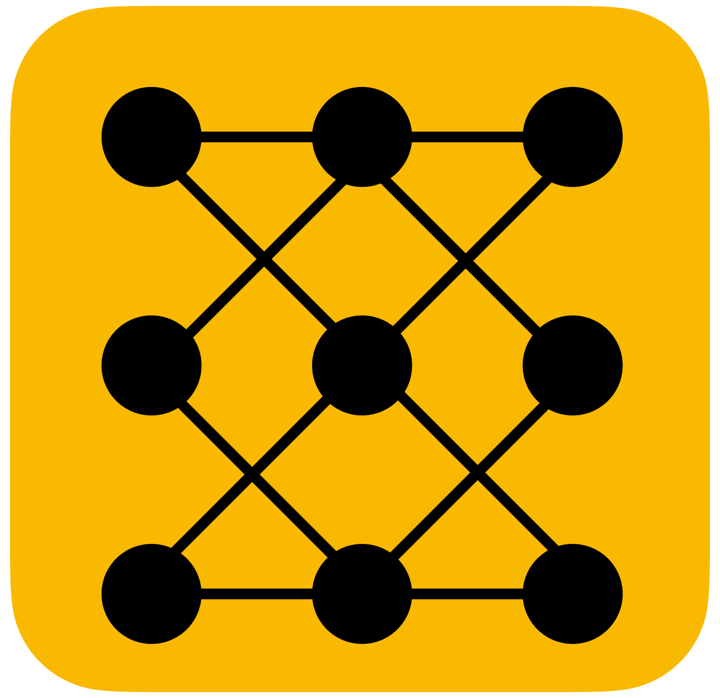
Credit: Inspired by Jonathan N. Lee’s slides

Common Offline RL Workflow: Policy Selection

Offline RL **Training**

Logged Dataset of Interactions

$$\{s_i, a_i, \tilde{s}_i, r_i\} \longrightarrow$$



Learning rate = 1e-4

NN hidden dimension = 256

$\longrightarrow$ Output Policy $\hat{\pi}$

- Offline RL leverages logged/historical datasets.
- Decouples RL policy training from deployment
- Safety, more stable training for larger policy models, etc.
- But, how to choose a hyperparameter and algorithm for $\hat{\pi}$?

Common Offline RL Workflow: TD-Error or Q-value

Logged Dataset of Interactions

$\{s_i, a_i, \tilde{s}_i, r_i\}$

$\hat{\pi}$

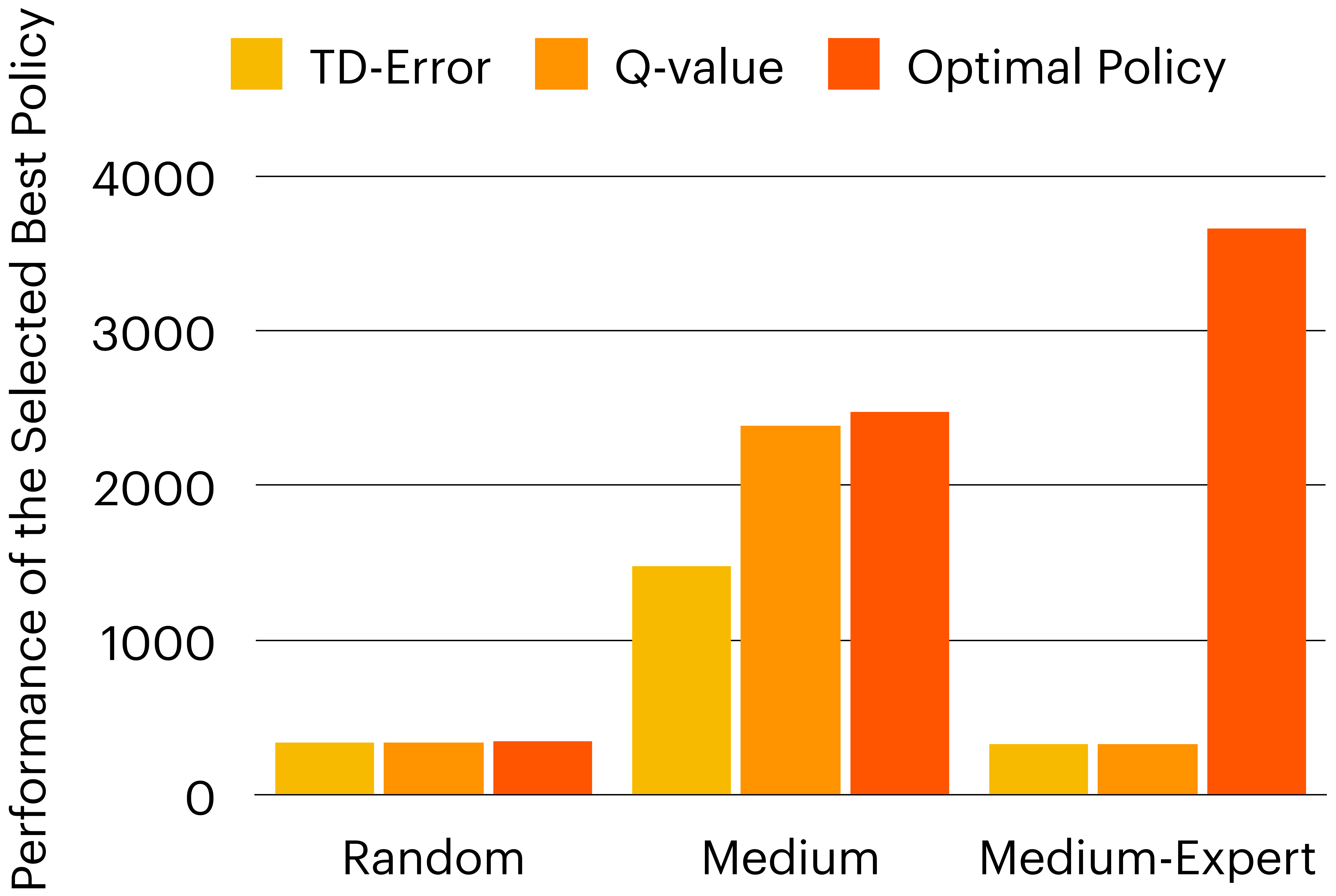
TD-Error

$$\frac{1}{N}\sum_{i=0}^{N}\sum_{t=1}^{L}[r_t + \gamma V^{\hat{\pi}}(s_{t+1}] - V^{\hat{\pi}}(s_t)$$

Policy	TD-Error
BC 256d NN	50
CQL 512d NN	**45**

- TD error is a sample-based approximation to Bellman error, and we know that $Q = Q^\star \Leftrightarrow ||Q - TQ||_\infty = 0.$
- This does not extend easily to policy optimization or non-actor-critic methods. Other efforts include:
- Selecting best policy from a set through pairwise comparison of value functions (BVFT) [Xie, Jiang 2021] [Zhang, Jiang 2021].
- Early stopping during conservative Q-function training [Kumar, Levine, 2021].

TD-Error or Q-value on the full dataset is a poor proxy



- Training on D4RL Hopper full dataset, if we use TD-error and Q-value to pick "best" policy and report their true performance.
- In a mixture quality dataset (medium-expert), TD-Error and Q-value cannot select a good policy.

Potential Offline RL Workflow: Offline Policy Evaluation

Validation Data

$$\{s_i, a_i, \tilde{s}_i, r_i\}$$

$$\hat{\pi}$$

Offline Policy **Evaluation**

$$\sum_{i=1}^{|\mathcal{D}|}\left(\prod_{t=1}^{L}\frac{\pi_\theta(a_t\,|\,s_t)}{p(a_t\,|\,s_t)}\right)R_i$$

Policy	OPE (IS)
BC 256d NN	1.5
CQL 512d NN	**6.3**

- Use Offline Policy Evaluation and a holdout validation dataset
- Not a good idea:
  - Amount of data available can impact ***both*** policy learning and quality of evaluation (due to data distribution shift, harder than in supervised learning)

Data Coverage Assumption

Offline Policy Evaluation

Evaluation data coverage **assumption**:

For all $s \in S$ and $a \in A$, the ratio $\frac{\pi_e(a \mid s)}{\pi_b(a \mid s)} < \infty$

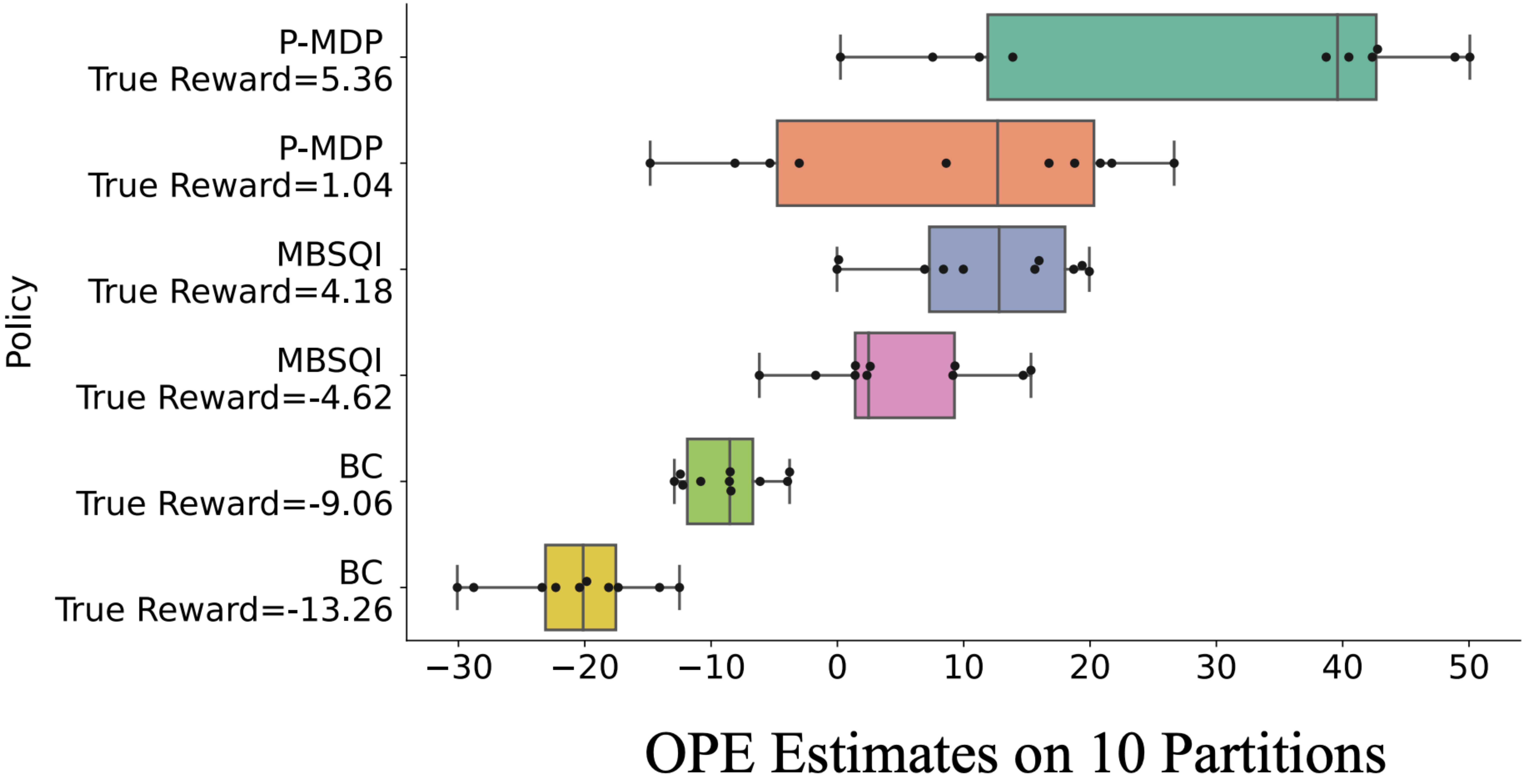
Offline Policy Training

Single-policy concentrability **assumption**:

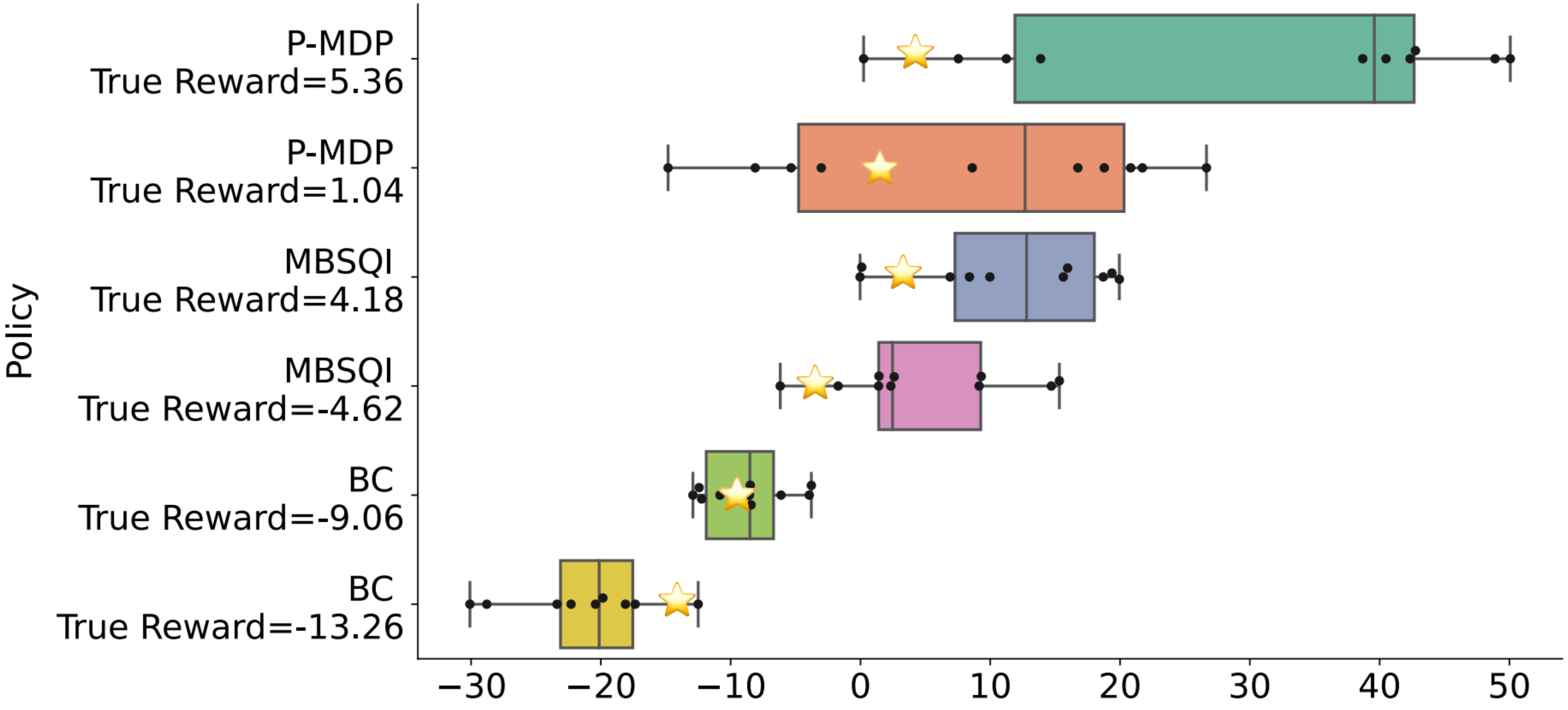
For all $s \in S$ and $a \in A$, the ratio $\frac{d_{\pi^\star}(s, a)}{d^D(s, a)} \leq B$

💡 When we have one shared dataset for training and evaluation, we have a **high chance of violating one of the two assumptions**.

Policy Evaluation is sensitive to Validation Data



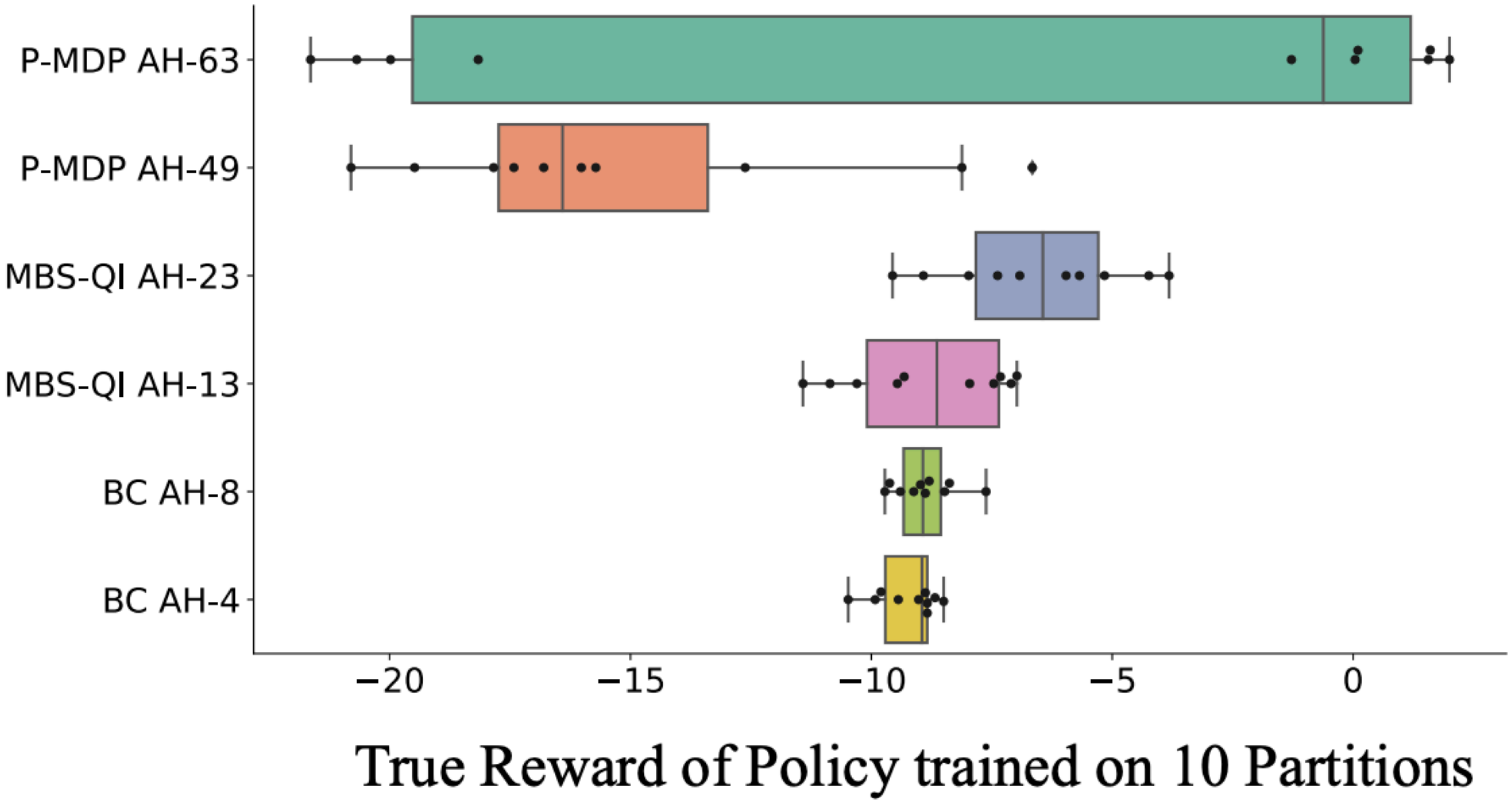
Policy Evaluation is sensitive to Validation Data



OPE Estimates on 10 Partitions

⭐: True performance of the policy

Policy Learning is Sensitive to Training Data



Dataset Partitioning Has a Substantial Impact on Offline RL Workflow

- Policy selection does not allow us to take repeated measurements.
- Algorithm-Hyperparameter selection allows us to repeat measurements.
- We prove a theorem that in a chain-MDP, with fairly small number of unique states, relying on a single train-validation split will have a probability of selecting sub-optimal alg-hyp for policy $P(\pi_{\hat{j}^*} \neq \pi_{j^\star}) \geq C$.
- If we allow N_s repeated experiments, $\lim_{N_s \to \infty} P(\pi_{\hat{j}^*} \neq \pi_{j^\star}) \to 1$

Policy Selection → Alg-Hyp Selection

Properties of Ideal Offline RL Workflow

1. Compare across Offline Policy Learning Algorithms (BC, CQL, BC+TD3, IQL, MOPO, etc.)
2. Considers Evaluation Partition Variations
3. Considers Policy Learning Variations
4. Data-Efficient in small-dataset (allow using all data to get a final policy)

Common Offline RL Practices

	Compares Across OPLs	Considers Evaluation Variation	Considers Policy Learning Variation	Data Efficient (re-training)
Internal Objective / TD-Error (Thomas et al., 2015b, 2019)	❌			
OPE methods (Komorowski et al. 2018; Paine et al. 2020)	✅			
OPE + Bootstrapped Validation (HCOPE) (Thomas et al., 2015b)	✅			
Batch Value Function Tournament (Xie and Jiang, 2021)	❌			
Batch Value Function Tournament + OPE (Zhang and Jiang, 2021)	✅			
Q-Function Workflow (Kumar et al., 2021)	❌			

Common Offline RL Practices

	Compares Across OPLs	Considers Evaluation Variation	Considers Policy Learning Variation	Data Efficient (re-training)
Internal Objective / TD-Error (Thomas et al., 2015b, 2019)	❌	❌		
OPE methods (Komorowski et al. 2018; Paine et al. 2020)	✅	❌		
OPE + Bootstrapped Validation (HCOPE) (Thomas et al., 2015b)	✅	✅		
Batch Value Function Tournament (Xie and Jiang, 2021)	❌	❌		
Batch Value Function Tournament + OPE (Zhang and Jiang, 2021)	✅	❌		
Q-Function Workflow (Kumar et al., 2021)	❌	❌		

Common Offline RL Practices

	Compares Across OPLs	Considers Evaluation Variation	Considers Policy Learning Variation	Data Efficient (re-training)
Internal Objective / TD-Error (Thomas et al., 2015b, 2019)	❌	❌	❌	
OPE methods (Komorowski et al. 2018; Paine et al. 2020)	✅	❌	❌	
OPE + Bootstrapped Validation (HCOPE) (Thomas et al., 2015b)	✅	✅	❌	
Batch Value Function Tournament (Xie and Jiang, 2021)	❌	❌	❌	
Batch Value Function Tournament + OPE (Zhang and Jiang, 2021)	✅	❌	❌	
Q-Function Workflow (Kumar et al., 2021)	❌	❌	❌	

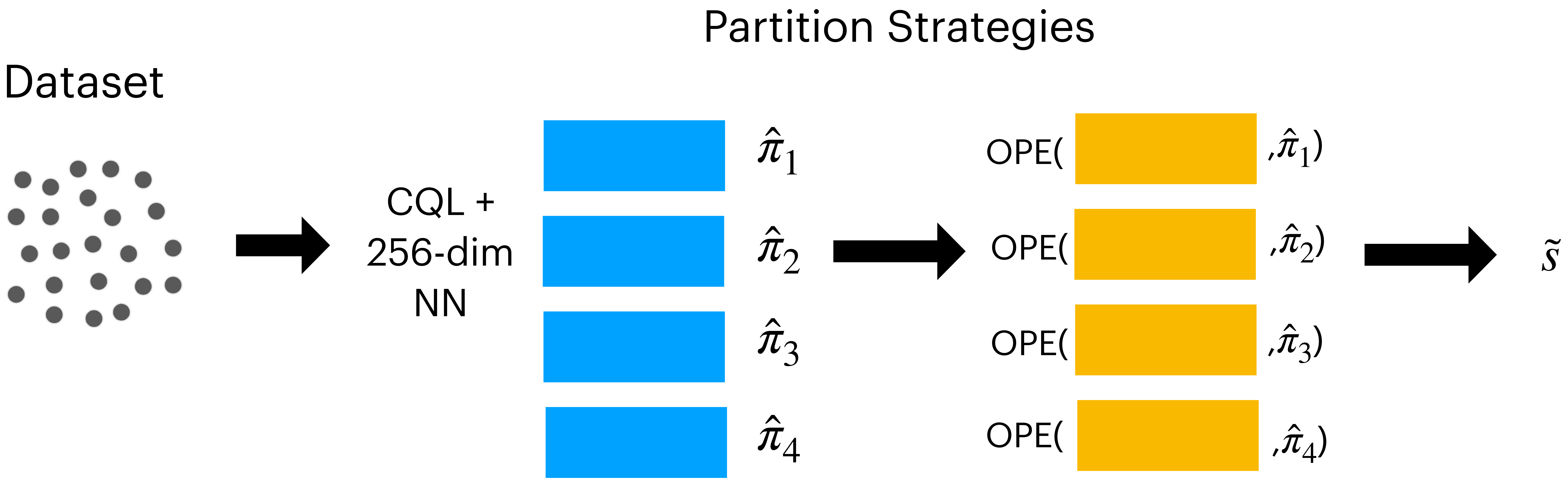
Common Offline RL Practices

	Compares Across OPLs	Considers Evaluation Variation	Considers Policy Learning Variation	Data Efficient (re-training)
Internal Objective / TD-Error (Thomas et al., 2015b, 2019)	❌	❌	❌	❌
OPE methods (Komorowski et al. 2018; Paine et al. 2020)	✅	❌	❌	❌
OPE + Bootstrapped Validation (HCOPE) (Thomas et al., 2015b)	✅	✅	❌	❌
Batch Value Function Tournament (Xie and Jiang, 2021)	❌	❌	❌	❌
Batch Value Function Tournament + OPE (Zhang and Jiang, 2021)	✅	❌	❌	❌
Q-Function Workflow (Kumar et al., 2021)	❌	❌	❌	✅

Common Offline RL Practices

	Compares Across OPLs	Considers Evaluation Variation	Considers Policy Learning Variation	Data Efficient (re-training)
Internal Objective / TD-Error (Thomas et al., 2015b, 2019)	❌	❌	❌	❌
OPE methods (Komorowski et al. 2018; Paine et al. 2020)	✅	❌	❌	❌
OPE + Bootstrapped Validation (HCOPE) (Thomas et al., 2015b)	✅	✅	❌	❌
Batch Value Function Tournament (Xie and Jiang, 2021)	❌	❌	❌	❌
Batch Value Function Tournament + OPE (Zhang and Jiang, 2021)	✅	❌	❌	❌
Q-Function Workflow (Kumar et al., 2021)	❌	❌	❌	✅
Split-Select-Retrain (SSR) (This work) (Nie et al., 2022)	✅	✅	✅	✅

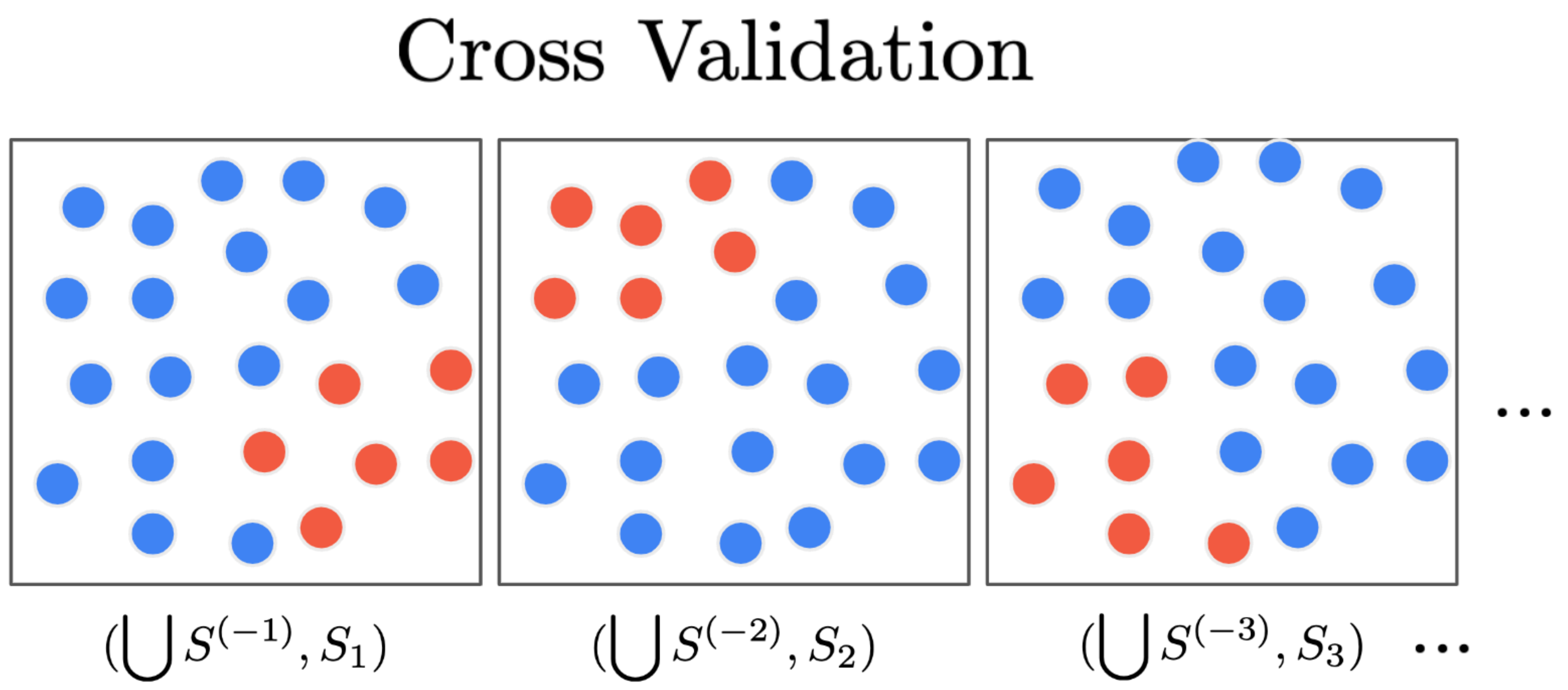
Split-Select-Retrain: Repeated Data Partitioning for More Robust Offline Policy learning



- Shifting from **policy selection** to **alg-hyp selection** allows us to do **repeated data splitting** on a single dataset.

Using Data Partition for Repeat Measurements

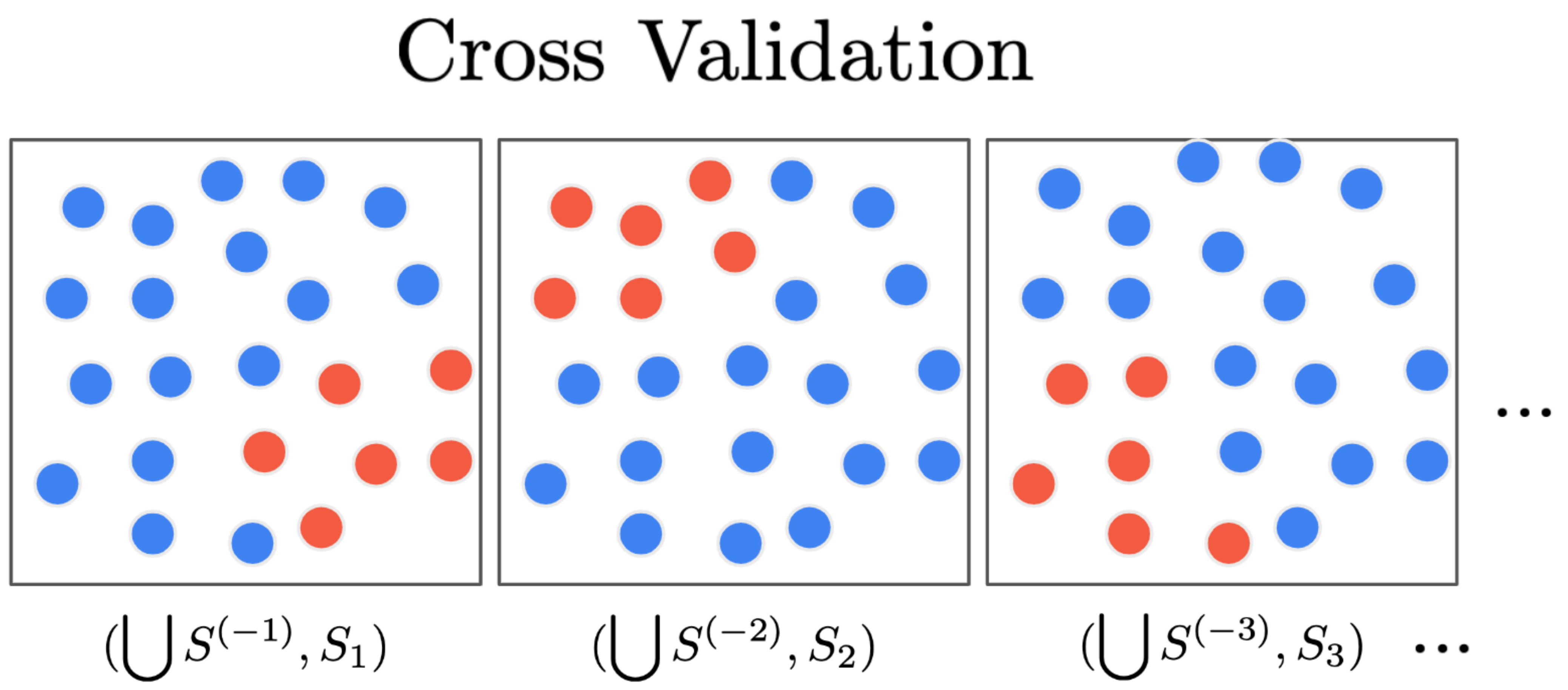
A straightforward and commonly used data partition technique in supervised learning is cross-validation.



🤔🤨 ? ?

Using Data Partition for Repeat Measurements

A straightforward and commonly used data partition technique in supervised learning is cross-validation.



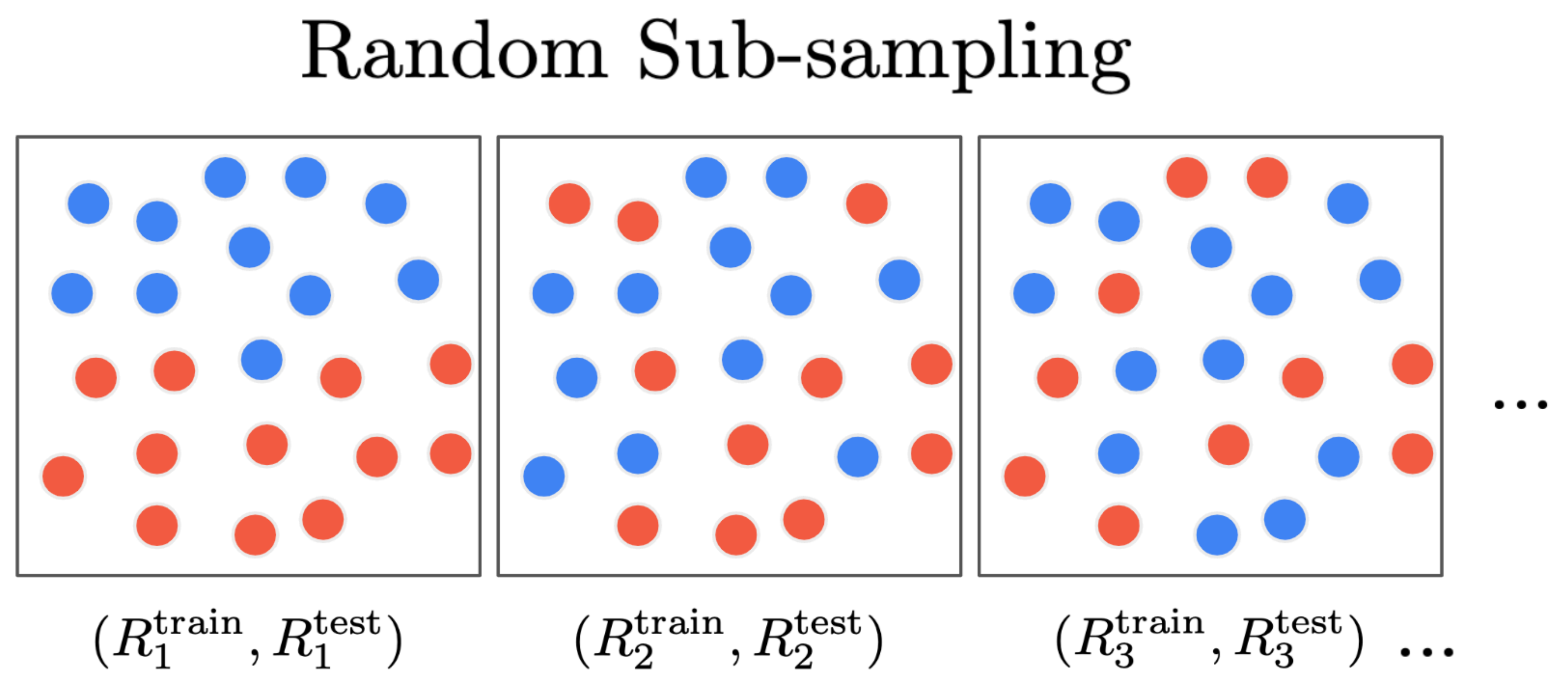
Cross-validation does not work well as a data partition technique because:

1. We want N_s to be large, according to **Theorem 1**.

2. For cross-validation, when N_s is large, the size for evaluation dataset is small, violating **OPE data coverage assumption**.

Using Data Partition for Repeat Measurements

Instead, we (re-)introduce random sub-sampling, originally proposed in 1981.

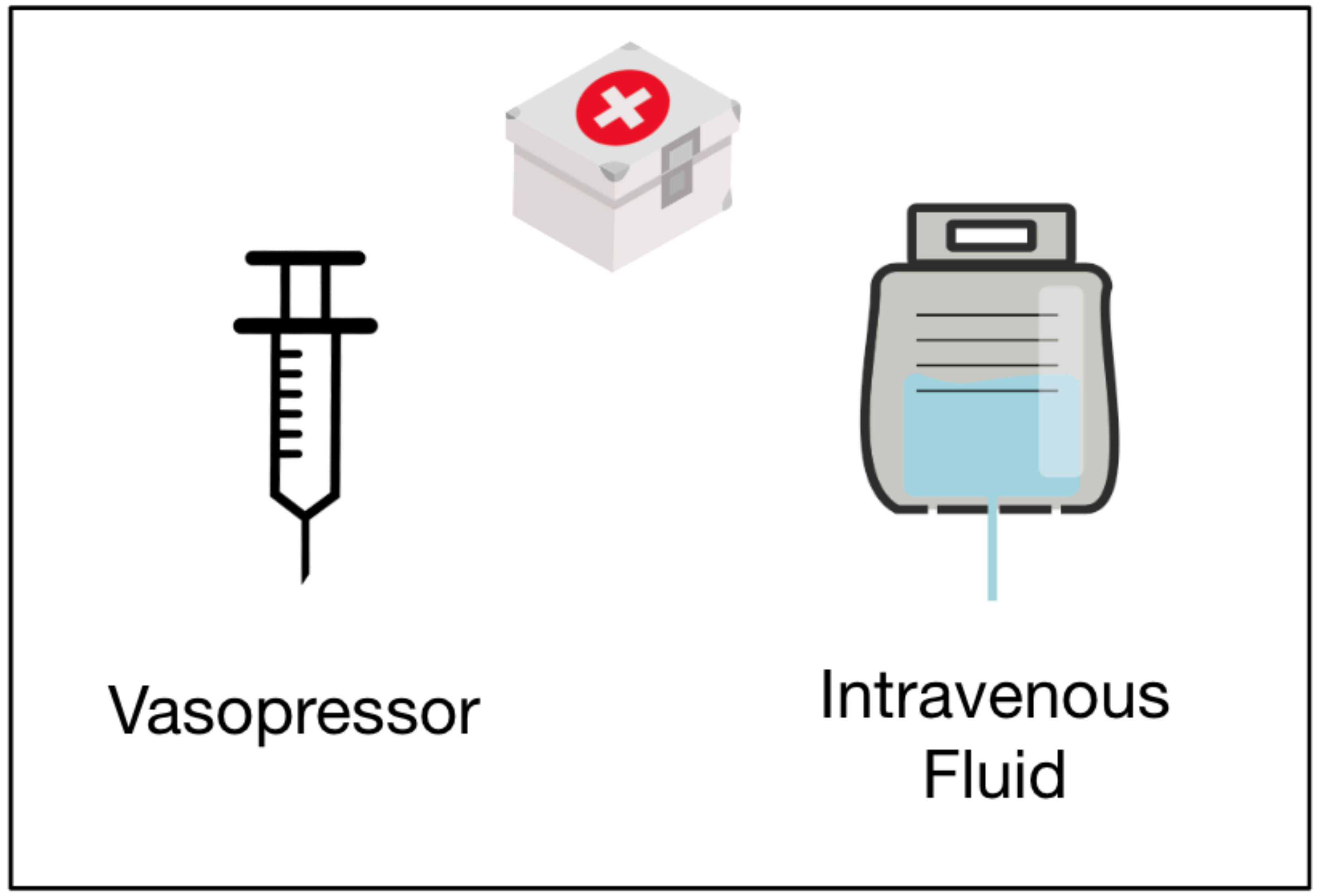


K Times

Random sub-sampling allows us to split the data into training/validation with each repeat.

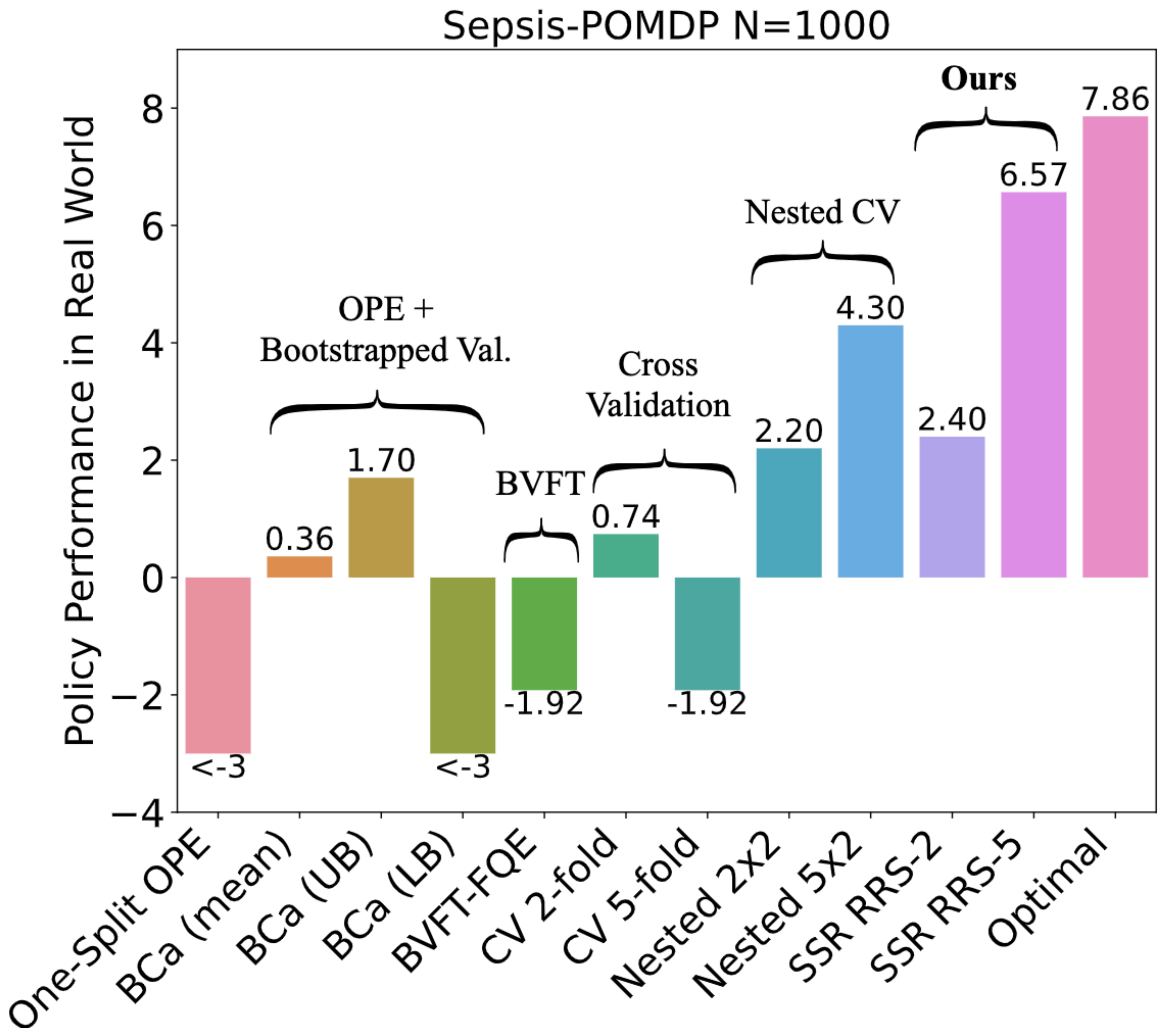
1. No limit on N_s
2. Approaches Leave-p-out cross-validation at the limit.
3. Central Limit Theorem shows it has the similar ability to discover optimal alg-hyp just like k-fold cross-valiadtion.

Experiment: Simulated Sepsis Domain



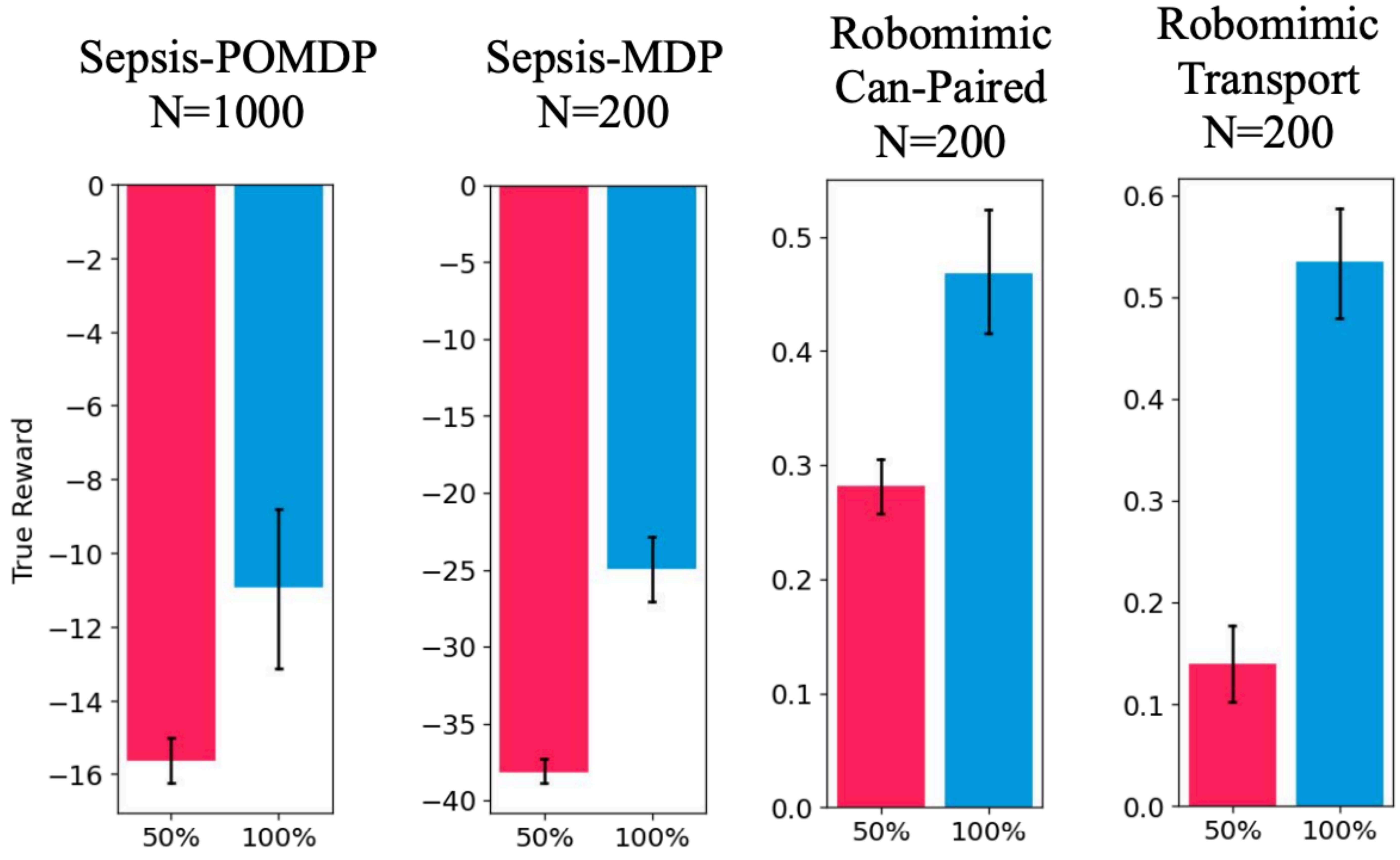
- We use Sepsis simulator created by Oberst and Sontag (2019).
- The state is 6-dim that captures biophysical state of the patient such as **heart rate**, **oxygen level**, residual level of **medication**.
- Generated 1000 patients with an existing sub-optimal policy.

Experiment: Selecting Alg-Hyp



- Compare different methods of selecting hyper-parameters and offline RL algorithms.
- K = 5 is sufficient
- We can see that <u>on average</u>, our framework **SSR-RRS** outperforms **One-split OPE**, **BCa**, **CV** and **Nested-CV**.

Is Re-training in SSR Important?



- On average, training on 100% of the dataset (if your dataset is small) will produce policies better than training on 50%.
- Caveat: could there exists a subset of data that gives a better policy? Likely yes...

Is SSR pipeline sensitive to OPEs?

Sepsis-POMDP	Parameters	Best AH Performance Chosen by SSR-RRS K=5
FQE-1	[64], lr=3e-4, epoch=20	2.84
FQE-2	[64], lr=1e-5, epoch=20	-74.26
FQE-3	[64], lr=3e-4, epoch=50	-20.88
FQE-4	[64], lr=1e-5, epoch=50	-14.16
FQE-5	[128], lr=3e-4, epoch=20	-75.26
FQE-6	[128], lr=1e-5, epoch=20	-14.48
FQE-7	[128], lr=3e-4, epoch=50	-75.54
FQE-8	[128], lr=1e-5, epoch=50	-74.26
IS	N/A	4.47
CWPDIS	N/A	4.68
WIS	N/A	6.75

- On the same domain, if instead of using one OPE method, we use other.
- The pipeline is sensitive to which OPE we select.
- However:

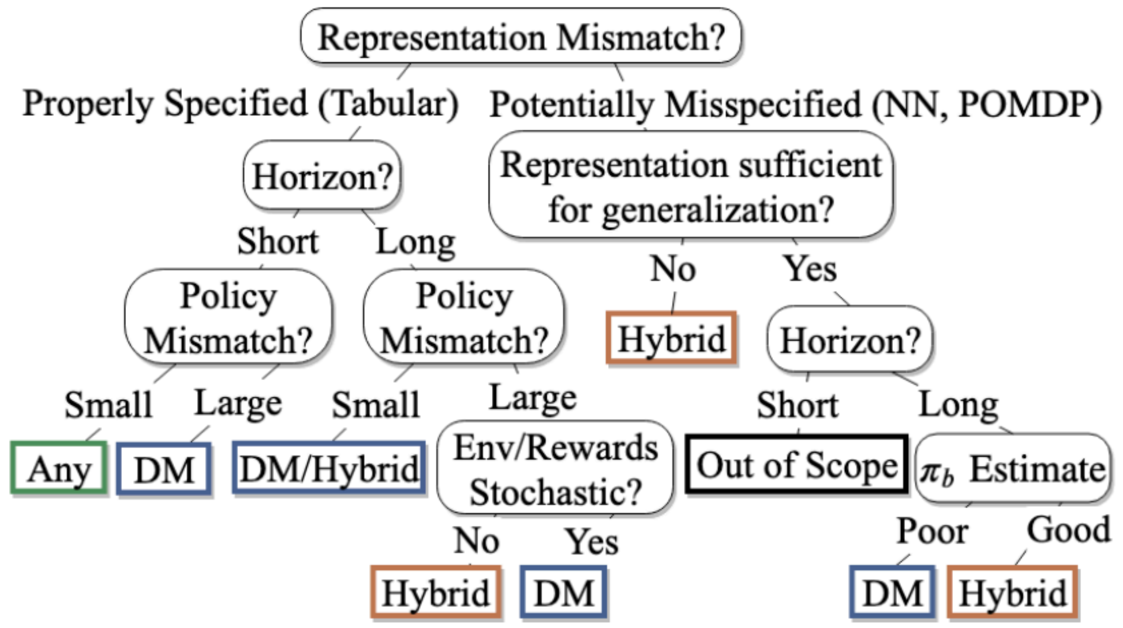


Figure 2: *General Guideline Decision Tree.*

Empirical Study of Off-Policy Policy Evaluation for Reinforcement Learning. Voloshin et al. 2021

Is SSR pipeline Robust?

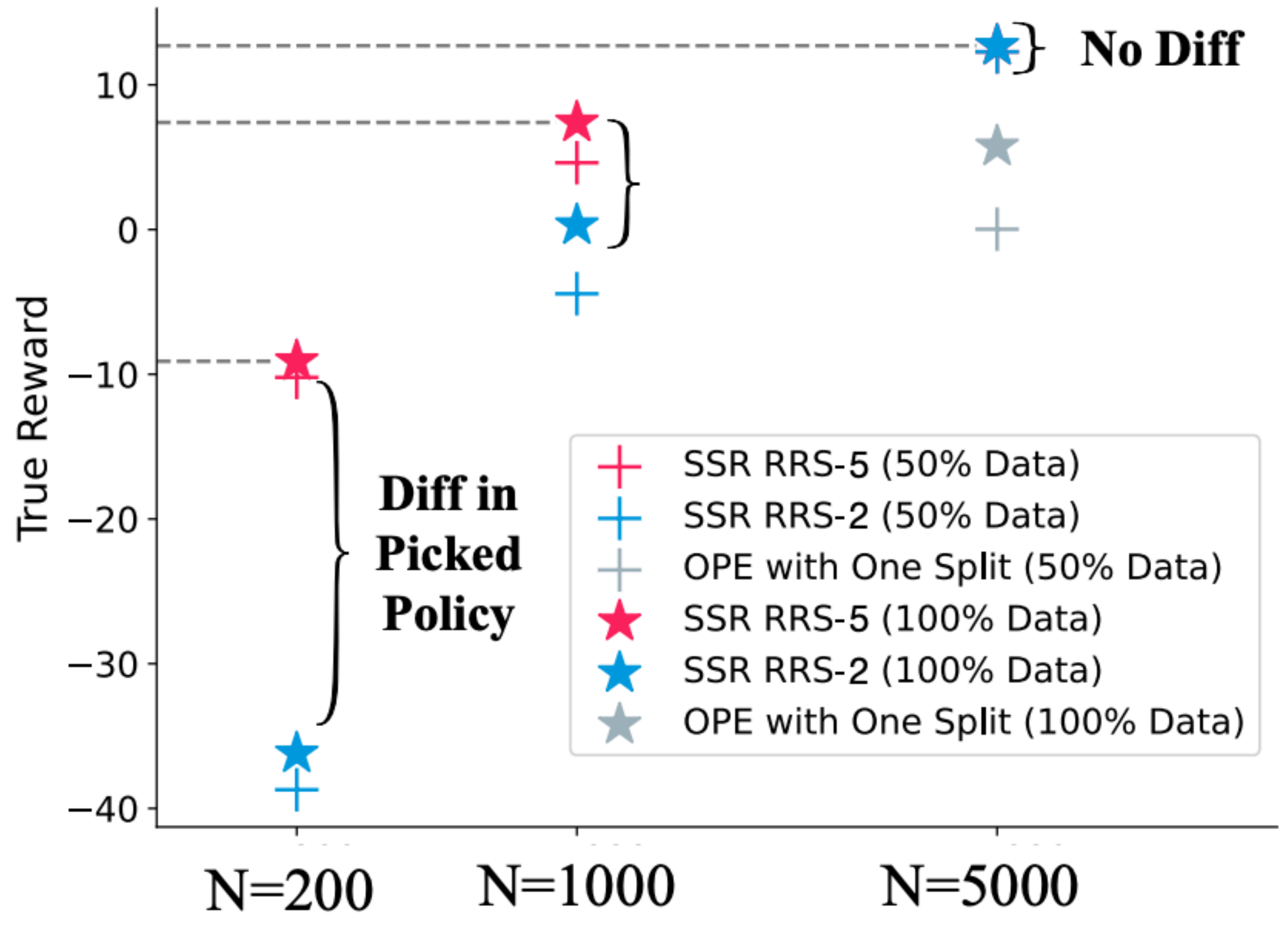
we only show the performance of the best policy among all AH pairs. Here we show that SSR-RRS can still robustly select a good hyperparameter for a given offline RL policy learning algorithm (the gap between best AH selected and true best AH is relatively small).

Sepsis-POMDP	Range of True Policy Performance (95%CI)	Percentile of AH Chosen by SSR-RRS	Performance of AH Chosen by SSR-RRS	True Best AH Performance
BCQ	[-10.8, -0.73]	94%	5.98	7.86
MBSQI	[-7.34, -2.26]	95%	6.40	7.42
BC	[-8.98, -8.37]	58%	-8.46	-7.42
BC+PG	[-5.55, -4.26]	78%	-3.68	2.52
P-MDP	[-31.17, -21.26]	83%	0.23	2.82

Table A.4: We show the relative position (percentile) of the AH selected by SSR-RRS K=5 pipeline.

What if the dataset gets large?

The number of trajectories in the dataset and the $|S| \times |A|$ space should be jointly considered to know if you have collected "enough" data.



In Sepsis-POMDP, where we only have ~20,000 unique states, when we have 5000 patients, the gap between different K is negligible.

Summary & Future Directions

- In Offline RL, we want to extract a good policy **reliably**.
- Many offline RL algorithms and model hyper-parameters to choose from. How do we select what works the best?
- **Split-Select-Retrain (SSR) allows us to:**
  - **Leverage full dataset (data efficient)**
  - **Be robust to data coverage issues in OPL and OPE.**
- Currently, number of repeats (K) is chosen heuristically. Is there an adaptive method to pick best K?
- Alternatively, can we build a strategy to select a subset of trajectories that will allow us to estimate Alg-hyp with less K?

Data-Efficient Pipeline for Offline Reinforcement Learning with Limited Data

Scan:

ArXiv: https://arxiv.org/abs/2210.08642

Twitter: @Allen_A_N
Email: anie@stanford.edu