# Supervised Off-Policy Ranking

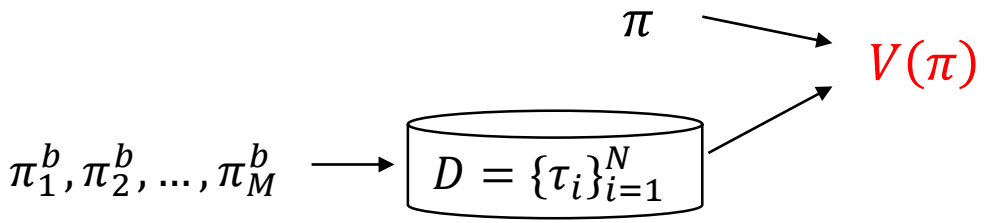
**Yue Jin [1] Yue Zhang [2] Tao Qin [3] Xudong Zhang [1] Jian Yuan [1] Houqiang Li [2] Tie-Yan Liu [3]**

[1]Tsinghua University [2]University of Science and Technology of China [3]Microsoft Research Asia

https://github.com/SOPR-T/SOPR-T

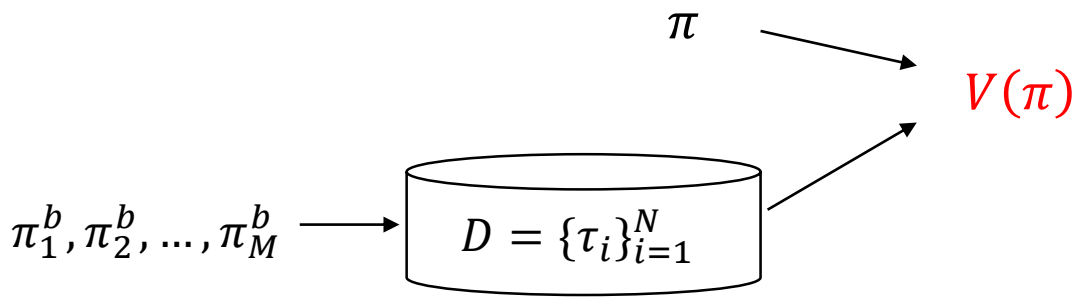
# Off-Policy Evaluation (OPE)

- Evaluating a policy with a pre-collected dataset $D = \{\tau_i\}_{i=1}^N$ consisting of trajectories $\tau_i = s_0^i, a_0^i, r_0^i, \cdots, s_T^i, a_T^i, r_T^i$ generated by other policies.
- OPE is critical to many real-world applications.
  - E.g., trading, advertising, autonomous vehicles, drug trials.
  - Online evaluation might be prohibitively expensive or dangerous.



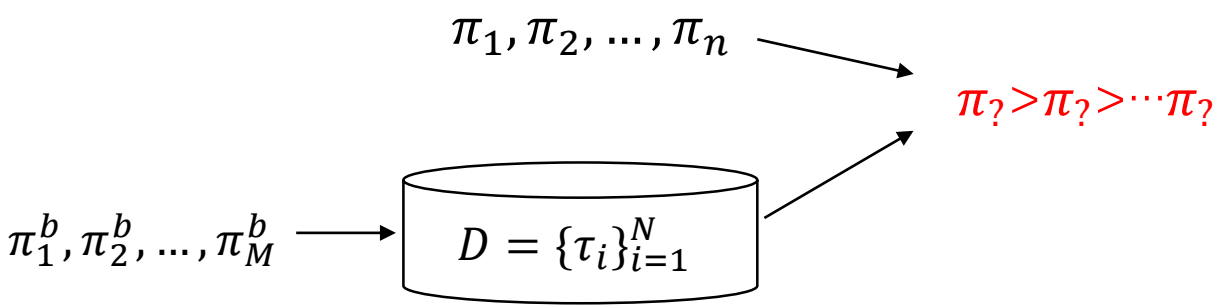

# Off-Policy Evaluation (OPE)

- Previous OPE methods:
  - Distribution correction, model estimation, and Q-estimation based methods
  - Mainly:
    - focus on **precisely estimating the expected return**, $V(\pi) = \mathrm{E}\left[\sum_{t=0}^{T} \gamma^t r_t\right]$, of a policy.
    - perform **unsupervised estimation** without directly leveraging the online performance of previously deployed policies.



# Our Observations

- There are some mismatches between the settings of previous OPE methods and the OPE problem in real-world applications.
  1. In many applications, the end goal is to **select the better policies** from several candidate policies other than to estimate their performance precisely.
  2. We usually know the true **performance of the polices** that have been deployed into real-world systems. Such information is **not well exploited** in previous OPE methods.

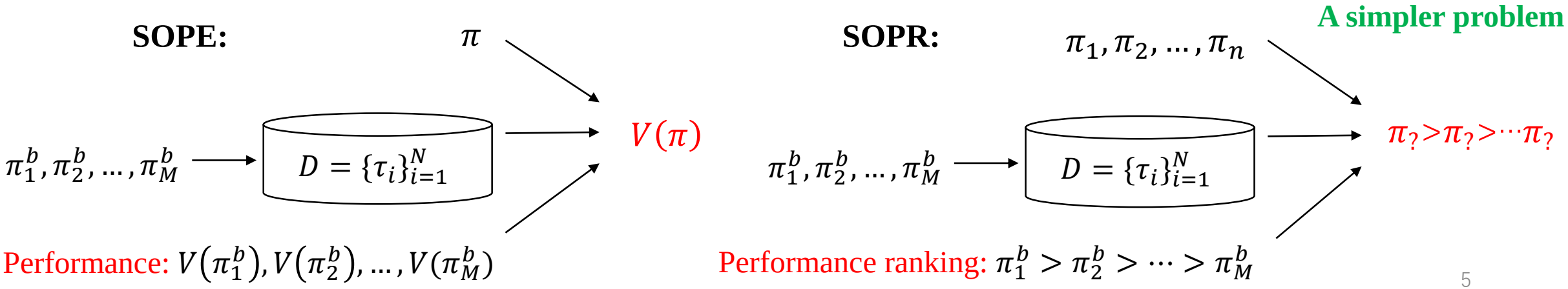



Performance: $V(\pi_1^b), V(\pi_2^b), \dots, V(\pi_M^b)$

# New Problems: Supervised OPE/OPR

- We define new problems: Supervised Off-Policy Evaluation/ Ranking
  - Supervised off-policy evaluation (SOPE):
    - estimate the performance of a target policy, using a pre-collected dataset $D = \{\tau_i\}_{i=1}^N$ and policies $\{\pi_i\}_{i=1}^M$ **with known performance**.
  - Supervised off-policy ranking (SOPR):
    - rank a set of target policies, using a pre-collected dataset $D = \{\tau_i\}_{i=1}^N$ and policies $\{\pi_i\}_{i=1}^M$ together **with their performance ranking**.

Leveraging more informative (supervised) signals

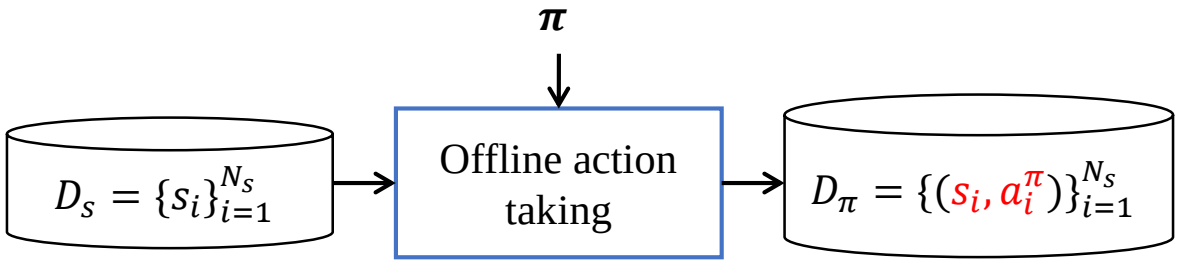


# A Solution to SOPR

- We propose a method to solve SOPR
  - Train a scoring model and rank a set of policies based on their scores

    $\pi_1^b, \pi_2^b, \ldots, \pi_M^b \longrightarrow$ scores: $x_1, x_2, \cdots, x_M \longrightarrow$ ranking: $\pi_3 > \pi_1 > \cdots > \pi_5$

  - Policy representation
    - Policy parameters? × **The policies may have different forms, or do not have parameters.**
    - State-action pairs √ **The policies share the same state (input) and action (output) spaces.**
      - States come from the off-policy dataset
      - Actions are taken by the target policy on the states in an offline manner



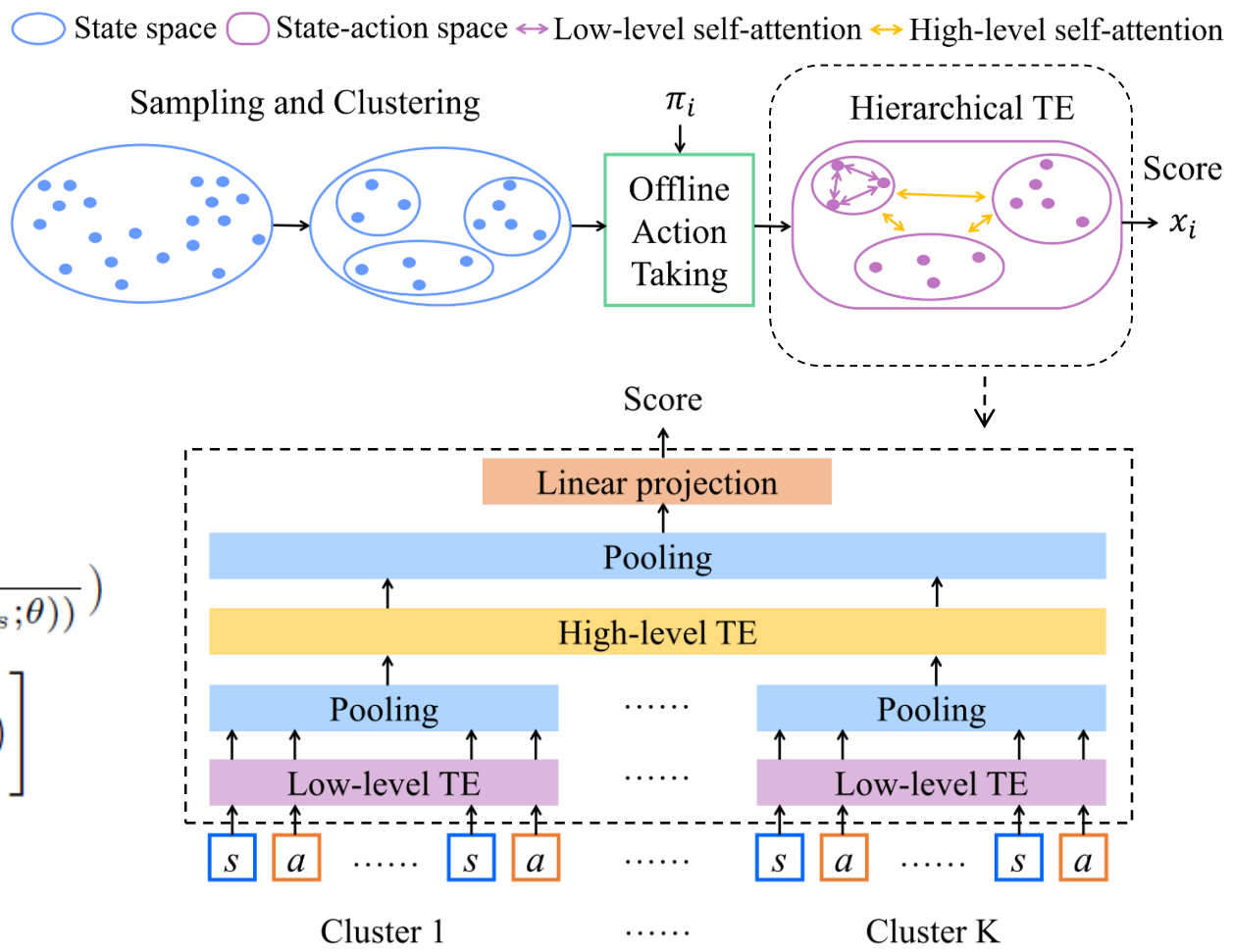
# SOPR-T

- Mapping $D_\pi$ to a score
  - Transformer encoder-based scoring model
  - A computational challenge: Millions of states
    - Down sampling and hierarchical encoding
- SOPR-T algorithm
  - Pairwise ranking loss:

$$l(\pi_i, \pi_j; \theta) = -\left[ y_{ij} \log\left(\frac{1}{1 + e^{-(f(\pi_i, D_s; \theta) - f(\pi_j, D_s; \theta))}}\right) + (1 - y_{ij}) \log\left(1 - \frac{1}{1 + e^{-(f(\pi_i, D_s; \theta) - f(\pi_j, D_s; \theta))}}\right)\right]$$

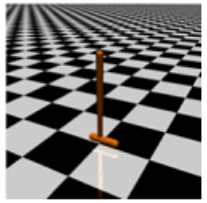
Pairwise ranking label

Scoring function

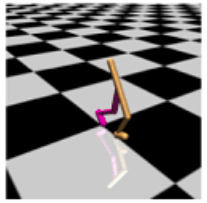


# Experiments

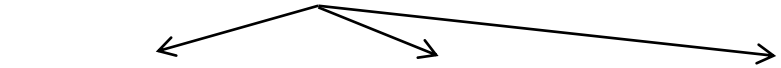
- Tasks:
  - Mujoco games
  - Public datasets: D4RL


Hopper


Half Cheetah


Walker

- Training/ validation/ test policy sets
  - Collecting 50 policy snapshots during online SAC training



Training set: 30 policies

Validation set: 10 policies

Test set II: 10 policies

  - Two test sets
    - **Offline trained policies (Test Set I)**
      - Trained by offline RL algorithms: BEAR, CQL, CRR
      - Different policy network architectures
    - **Online trained policies (Test Set II)**
  - Ranking labels: based on Monte-Carlo evaluation

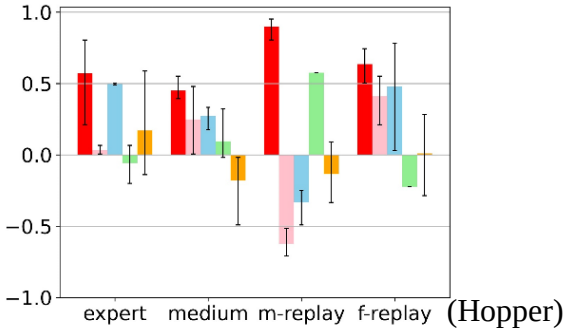
- Baselines:
  - Fitted Q evaluation (FQE)
  - Weighted step-wise importance sampling (IW)
  - Model-based Monte-Carlo estimation (MB)
  - DualDICE
- Evaluation metrics:
  - Rank correlation
  - Normalized regret @ k $= \frac{|V_{\max} - V_{\max\_topk}|}{V_{\max} - V_{min}}$

# Experiments

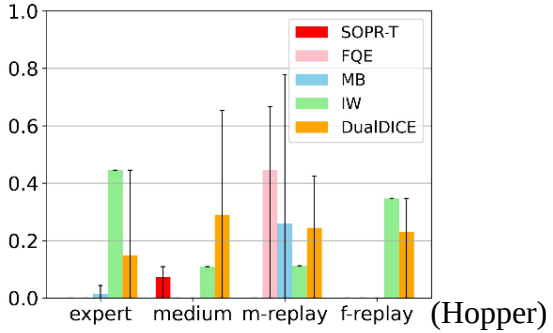
*SOPR-T performs the best in most of the tasks.*

- Main results
  - Performance on offline learned policy sets (**Test Set I**)

Rank correlation

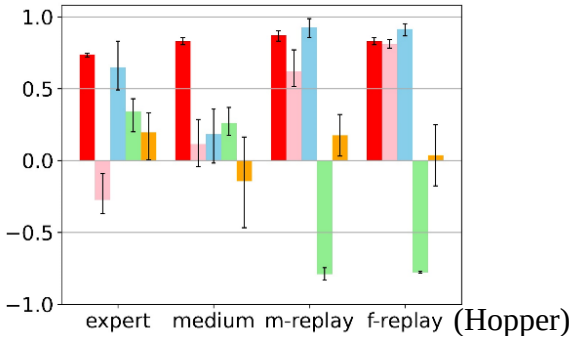



Regret@3

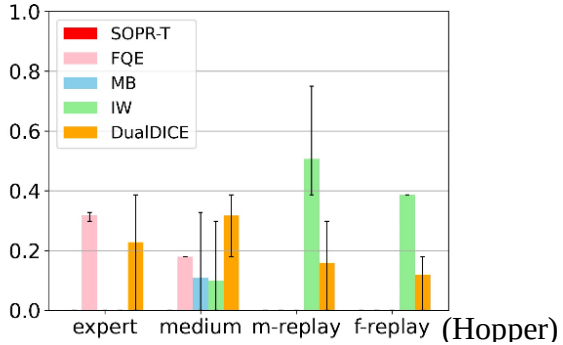



  - Performance on online learned policy sets (**Test Set II**)

Rank correlation

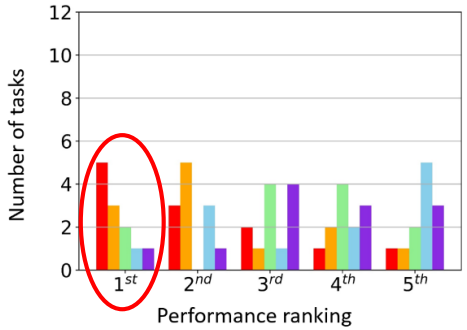



Regret@3

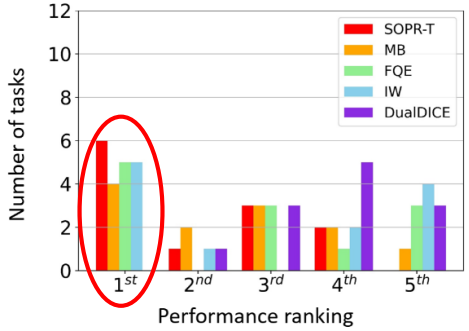



- Performance ranking statistics
- **Test Set I**

Rank correlation

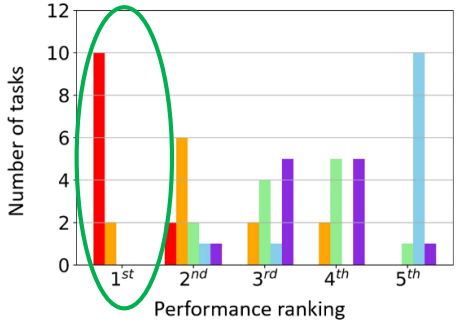



Regret@3

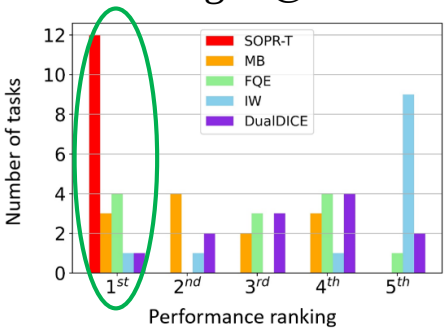



- **Test Set II**

Rank correlation




Regret@3




- Further studies: using different amount of data, and the advantage of Transformer encoder

# *Thanks!*

Contact: jiny23@126.com and taoqin@microsoft.com

Code: https://github.com/SOPR-T/SOPR-T