Offline Imitation Learning with Model-based Reverse Augmentation

***Jie-Jing Shao*, Hao-Sen Shi, Lan-Zhe Guo, Yu-Feng Li**

Nanjing University, LAMDA Group

KDD2024
BARCELONA, SPAIN
RESEARCH TRACK

What is this Talk about

Model-based solution is one of the main development modes of reinforcement learning.

In Imitation Learning, it suffers one serious issue

- Without reward supervision, it is difficult to determine what action an agent should take when outside the state distribution of the expert demonstrations.

Contribution of this work

We present the **reverse model-based augmentation** for offline imitation learning, revealing it could provide the guidance on out-of-expert states and be more efficiently utilized than forward-based rollouts. We formulate the idea as a solution with self-paced data augmentation, enhancing the long-term returns.

Model-based RL

Success in game world

Game environments provide ideal conditions

for trial-and-error learning.

DeepMind

2023-9-26

A Generalist Dynamics Model for Control

Ingmar Schubert[*,1], Jingwei Zhang[2], Jake Bruce[2], Sarah Bechtle[2], Emilio Parisotto[2], Martin Riedmiller[2], Jost Tobias Springenberg[2], Arunkumar Byravan[2], Leonard Hasenclever[2] and Nicolas Heess[2]
[1]TU Berlin, [2]DeepMind, [*]Work done at DeepMind

TD-MPC2:
Scalable, Robust World Models for Continuous Control

Nicklas Hansen[*], **Hao Su**[*†], **Xiaolong Wang**[*†]
[*]University of California San Diego, [†]Equal advising
{nihansen,haosu,xiw012}@ucsd.edu

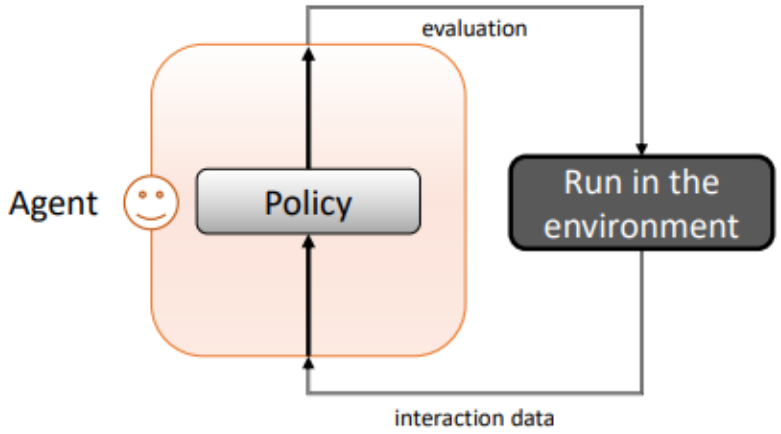
World model, a simulation for real world

Trial-and-error learning

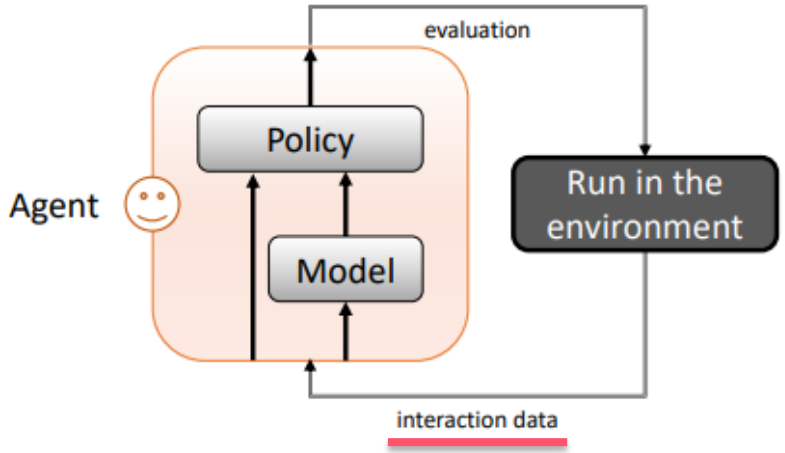
KDD2024
BARCELONA, SPAIN

Model-based RL

On-policy RL



Model-based RL



Model provides the transition and reward feedback.
More data-efficient and safe!

Offline Imitation Learning

What if the reward function is unavailable?

- The design of the reward function is typically difficult

One promising way: offline IL

Learning for sequential decision-making from the demonstrations, without the difficult design of reward function.

Children mimic adults

Learning from demonstrations

KDD2024
BARCELONA, SPAIN

Problem Formulation

Consider the MDP: $M = \{S, A, T, r, d_0, \gamma\}$ with state space S and action space A, transition $T: S \times A \to \Delta(S)$, initial state distribution d_0 and discount factor γ.

- **Goal** [maximizing the excepted cumulative return]:

$$J(\pi) = \mathbb{E}_{s_0 \sim d_0, s_{t+1} \sim T(\cdot|s_t, \pi(s_t))}[\sum_{t=0}^{\infty} \gamma^t r(s_t, \pi(s_t))]$$

- **Expert data D^E from expert policy**: $D^E = \{(s_0, a_0, s_1, a_1, \dots,)|a_t \sim \pi^E(a_t|s_t)\}$ limited
- **Offline data from any behavior policy**: $D^o = \{(s_0, a_0, s_1, a_1, \dots,)|a_t \sim \pi^o(a_t|s_t)\}$ rich but low-quality
- **Model-based solution**: Learning a dynamics model $\hat{T} \to T(s_{t+1}|s_t, a_t)$ from offline data D^O

Model-based Offline Imitation Learning

Challenge: Difficult to determine the action on out-of-expert states.

Previous key idea: Keep conservative in the out-of-expert area.

MILO [NeurIPS'21] **:**

$$\min_r \max_\pi \mathbb{E}_{s,a\sim\pi}[r(s,a) + \beta(s,a)] - \mathbb{E}_{s,a\sim\pi^E}[r(s,a)]$$

Penalty functions on out-of-expert area.

CLARE [ICLR'23] **:**

$$\min_r \max_\pi (Z_\beta \, \mathbb{E}_{s,a\sim\pi}[r(s,a)] - \mathbb{E}_{s,a\sim\pi^\beta}[\beta(s,a) r(s,a)] - \mathbb{E}_{s,a\sim\pi^E}[r(s,a)])$$

They want to avoid the agent visiting states that do not appear in the expert dataset.

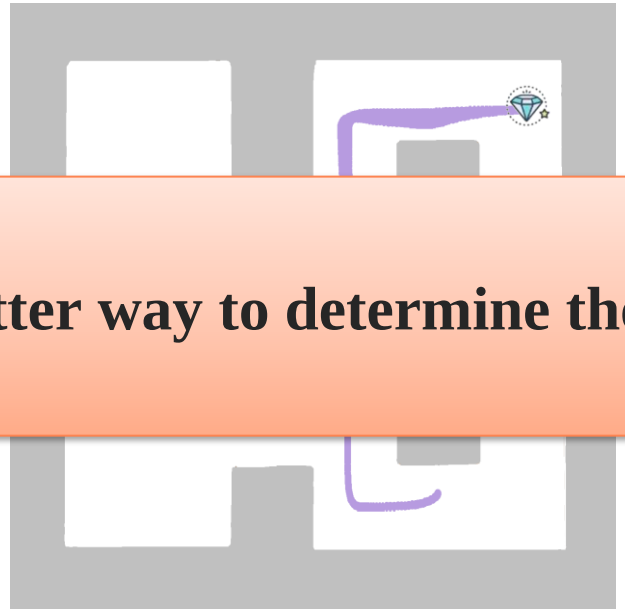
KDD2024 BARCELONA, SPAIN

Mitigating Covariate Shift in Imitation Learning via Offline Data With Partial Coverage. NeurIPS'21
CLARE: Conservative Model-Based Reward Learning for Offline Inverse Reinforcement Learning. ICLR'23

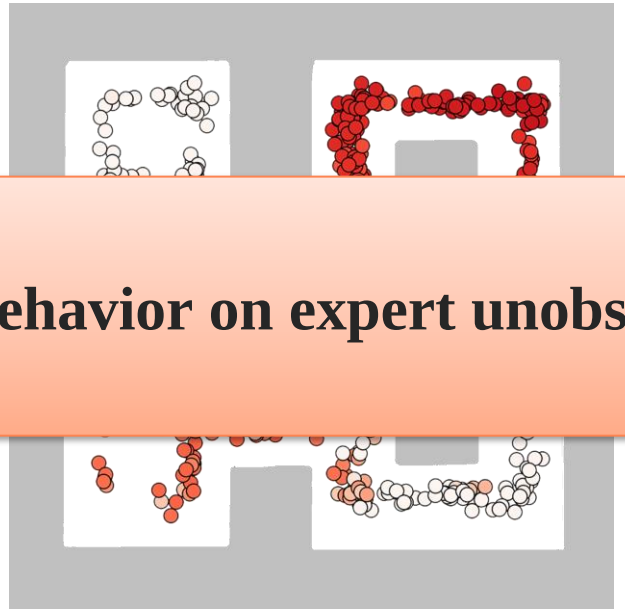
Model-based Offline Imitation Learning

These methods performs well in the expert-observed states but perform poorly in the rest of the states.

As an example, in navigation task,



Expert dataset



Cumulative return of MILO [NeurIPS'21]

Do we have a better way to determine the behavior on expert unobserved states?

r

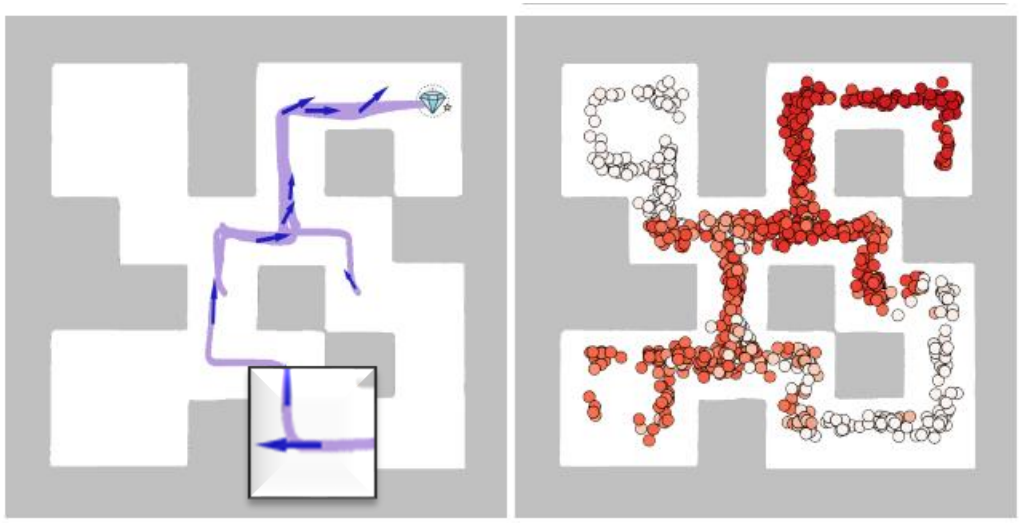
s.

Our Proposal: Model-based Reverse Augmentation

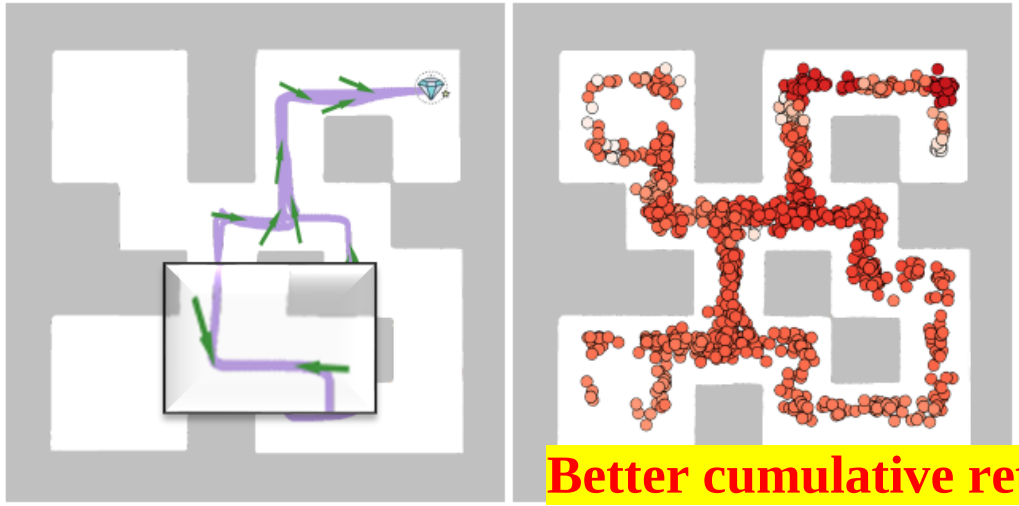
Our idea: If an action can lead an agent from expert-unobserved states to expert-observed states, we regard it as good action, as it could enhance subsequent returns.

To generate trajectories from expert-unobserved states to expert-observed states, we build reverse models to provide reverse augmentation.

- **Forward Model: expert-observed → expert-unobserved**
- **Reverse Model: expert-unobserved → expert-observed**



Forward Model



Better cumulative return!

Reverse Model (Ours)

KDD2024 BARCELONA, SPAIN

How to build Reverse Models?

Reverse dynamic model $\widehat{T}_r$: supervised learning, approximating $T(s_t|s_{t+1}, a_t)$:

$$\max_{\hat{T}_r} \sum_{(s_t, a_t, s_{t+1})} \log \hat{T}_r\,(s_t|s_{t+1}, a_t)$$

Reverse behavior policy π_r: a VAE-based actor, approximating $p(a_t|s_{t+1})$:

$$\log \pi_r\,(a|s) \geq \mathbb{E}_{z\sim\pi_r^e(\cdot|s,\,a)} \log \pi_r^d\,(a|z, s) - KL(\pi_r^e(z|s, a)||p(z|s))$$

Training: learn the conditional variational auto-encoder through maximizing the above lower bound.

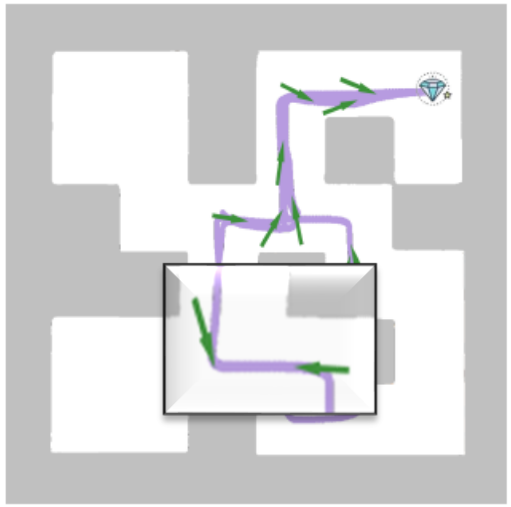
Inferencing: sample the reverse action $a \sim \pi_r(a|s)$

How to generate reverse trajectories?

- **Reverse augmentation from expert data**:

$$\{s_{-h'}, a_{-h'}, s_{-h'+1}, a_{-h'+1}, \dots, s_{-2}, a_{-2}, s_{-1}, a_{-1}, s_0\}$$

Target states $s_0 \sim D^E, a_i \sim \pi_r(s_{i+1}), s_i \sim \hat{T}_r(s_{i+1}, a_i)$ **Reverse Models**



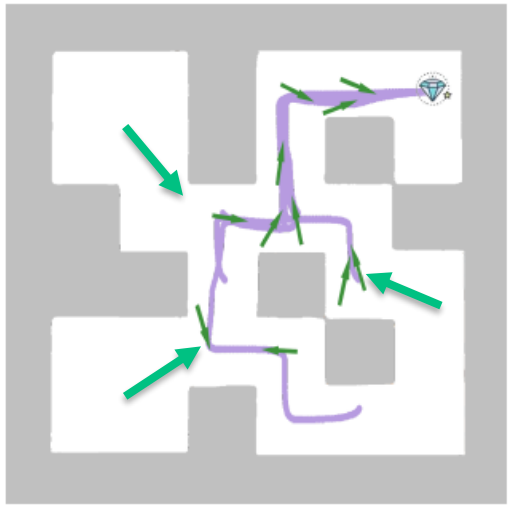
- **Beyond expert-observed area**:
  - expand the target states G

$$Conf_\pi(s) = \pi(\mathbb{E}[\pi(a|s)]|s)$$

$$s_0 \sim G = \left\{s \middle| Conf_\pi(s) \geq \mathbb{E}_{s' \sim D^E} Conf_\pi(s')\right\} \cup \{s' | s' \in D^E\}$$

  - re-sample the augmented instances

$$p(s) = 1/Conf_\pi(s)$$



Overall Algorithm

Simple but effective **S**elf-paced **R**everse **A**ugmentation.

SRA:

1) It gradually expands the targeted region with the reliable behavior.

2) During each training session, it explores the out-of-expert states using data augmentation with reverse models

3) The policy utilizes Q-Learning methods, such as IQL, to enhance long-term performance on augmented states.

Algorithm 1 Offline IL with Self-Paced Reverse Augmentation

Require: Expert dataset D^E, supplementary dataset D^S, length of reverse rollout h', The number of iterations N_D, N_R, N_P.

for $t = 1$ to N_D **do**
  Update the reverse dynamic model $\hat{T}_r$ with gradient descent.
end for
for $t = 1$ to N_R **do**
  Update the reverse behavior policy π_r with gradient descent.
end for
for $t = 1$ to N_P **do**
  Collect the goals G via Eq. 4.
  Rollout the reverse trajectories D^A with $G, \hat{T}_r, \pi_r$, via Eq. 2.
  Re-sample the augmented samples to obtain D^W, with the sampling weights(Eq. 5).
  Get the union dataset $D^U = D^E \cup D^S \cup D^R$
  Train the policy π with model-free reinforcement learning methods on the union dataset.
end for
return Policy π.

KDD2024
BARCELONA, SPAIN

Empirical Results

Experiments on D4RL benchmark

DataSet	BC-exp	DemoDICE	DWBC	OTIL	CLARE	MILO	ROMI	UDS	SRA
maze2d-umaze-sparse-v1	100.±11.6	88.7±10.4	25.8±5.65	128.±8.22	-3.08±6.02	75.0±11.2	154.±5.96	64.9±8.51	**155.±6.20**
maze2d-medium-sparse-v1	44.6±11.1	15.4±7.83	22.7±4.77	98.2±11.0	33.5±7.82	47.9±13.5	123.±10.5	83.0±8.84	**147.±5.67**
maze2d-large-sparse-v1	15.5±7.98	8.68±3.55	35.1±4.18	129.±14.4	18.6±9.12	51.2±17.1	101.±20.2	108.±16.7	**150.±14.9**
maze2d-umaze-dense-v1	70.6±9.55	69.1±9.21	39.2±4.77	100.±6.98	5.84±6.61	54.9±6.96	111.±6.23	62.3±6.99	**113.±5.80**
maze2d-medium-dense-v1	45.0±10.2	34.3±7.08	39.1±3.34	95.7±8.66	46.3±7.81	44.4±11.0	112.±9.10	87.3±7.80	**138.±5.29**
maze2d-large-dense-v1	18.2±8.57	21.7±6.30	56.1±5.56	120.±11.0	26.5±8.78	40.7±14.0	101.±16.6	109.±14.4	**140.±11.4**
hopper-medium	72.9±5.50	54.1±1.67	88.1±4.71	26.2±2.28	82.2±6.56	75.0±7.46	67.3±4.82	59.5±4.51	**90.2±4.93**
halfcheetah-medium	13.3±2.74	41.1±1.00	22.5±3.94	38.7±0.75	32.2±3.14	41.9±0.92	43.6±1.53	43.6±5.15	**43.7±1.72**
walker2d-medium	99.1±3.66	73.0±2.09	84.8±5.65	86.9±3.63	49.9±5.37	67.9±3.13	96.6±3.76	97.6±2.85	**101.±3.60**
ant-medium	51.3±6.87	91.2±3.79	37.5±5.95	72.4±5.68	68.5±7.35	92.0±3.55	**92.7±6.46**	87.3±5.10	88.9±7.18
hopper-medium-expert	72.9±5.50	98.6±4.32	99.4±4.43	42.5±3.70	93.9±5.81	90.9±5.42	100.±3.40	97.4±3.35	**104.±3.37**
halfcheetah-medium-expert	13.3±2.74	48.9±5.46	**82.3±3.79**	43.7±2.76	31.4±5.15	44.5±1.57	58.8±3.29	67.1±2.63	63.4±3.52
walker2d-medium-expert	99.1±3.66	93.1±5.49	**106.±1.57**	82.5±2.76	39.9±7.66	95.4±3.87	103.±2.12	103.±2.32	104.±4.88
ant-medium-expert	51.3±6.87	69.8±7.97	58.2±8.81	79.2±7.40	3.61±2.86	**115.±4.63**	105.±6.90	92.2±8.14	94.1±7.86

Across 14 settings, SRA achieves consistent improvement over both model-free and model-based baselines.

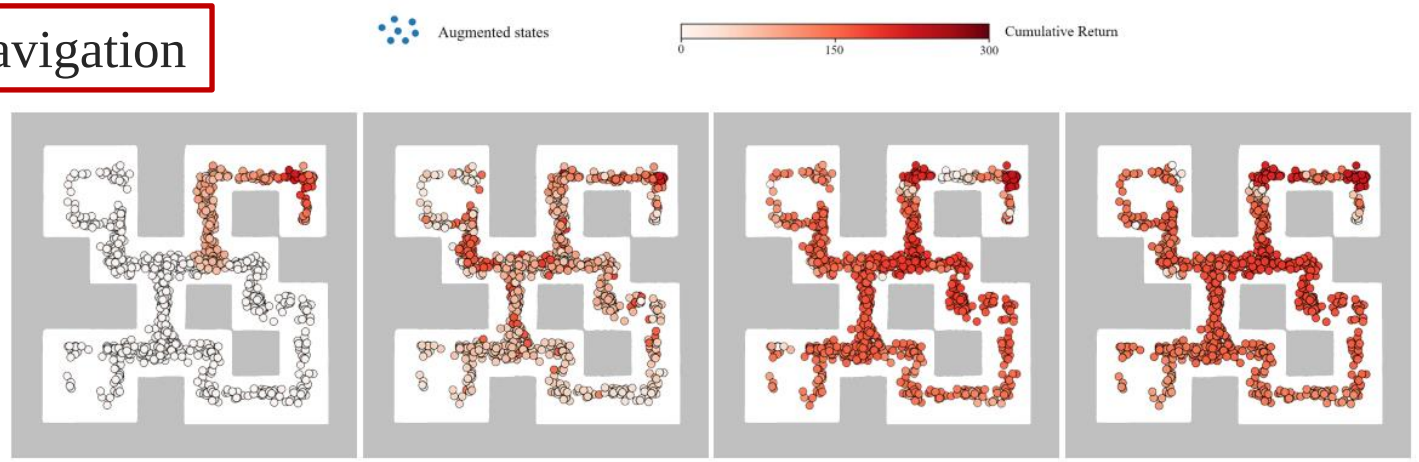
KDD2024 BARCELONA, SPAIN

Self-Paced Augmentation

Visualization on Navigation

State-wise Cumulative Return

State Distribution of Augmented data



(a) State-wise cumulative return of policy π after 0, 50000, 100000, and 150000 iterations.

(b) Re-sampled augmented dataset D^R after 0, 50000, 100000, and 150000 iterations.

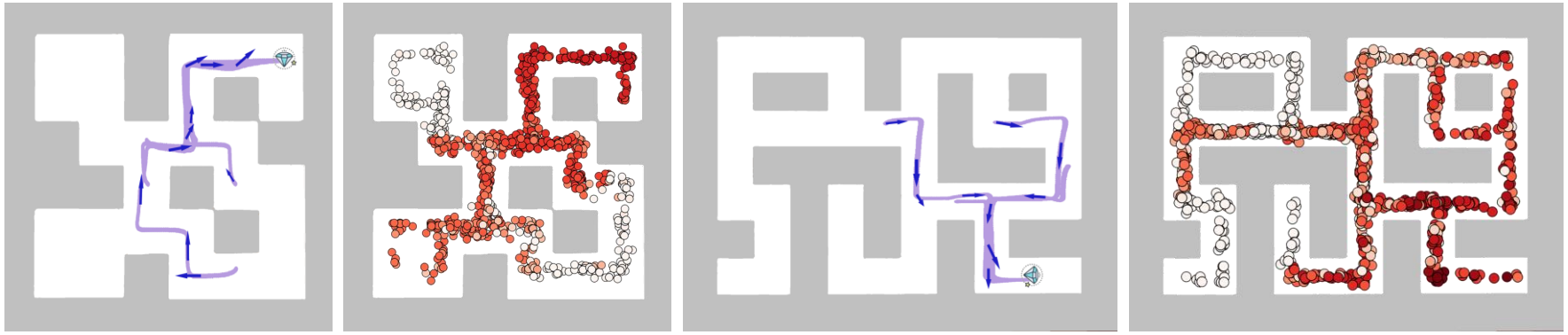
SRA gradually explores the out-of-expert area, and effectively benefits from the augmented data, expanding the red area.

KDD2024 BARCELONA, SPAIN

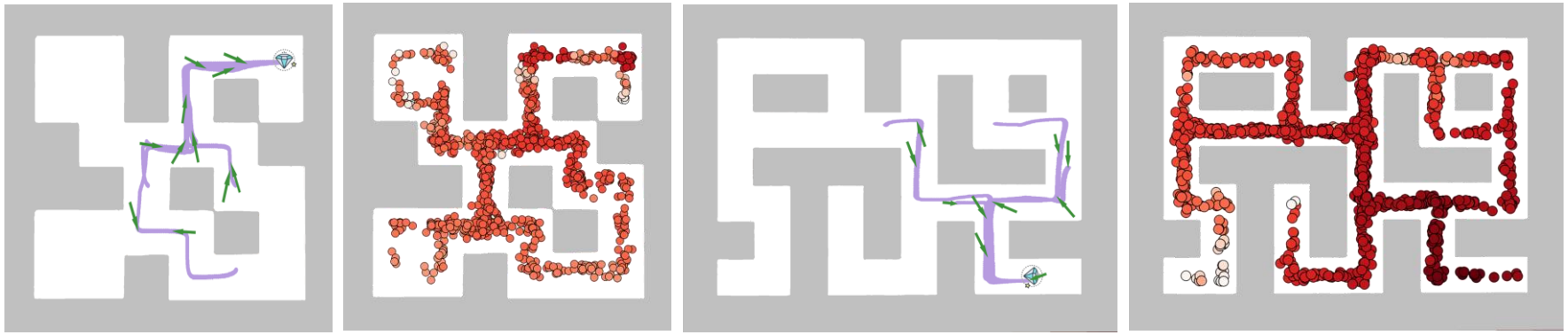
LAMDA Learning And Mining from DatA

Reverse model vs. Forward model

Forward-based MILO



Reverse-based SRA



SRA effectively explore the out-of-expert area and improve the corresponding long-term returns.

Combination with different RL methods

Scalability with different RL

Table 4: Combination of SRA and different offline reinforcement learning methods.

DataSet	IQL	SRA+IQL	TD3BC	SRA+TD3BC	AWAC	SRA+AWAC	SAC-N	SRA+SAC-N
maze2d-umaze-sparse-v1	64.9±8.51	155.±6.20 ↑	38.1±12.9	145.±7.27 ↑	68.6±14.5	135.±10.1 ↑	151.±6.44	150.±6.47
maze2d-medium-sparse-v1	83.0±8.84	147.±5.67 ↑	22.4±9.64	140.±8.55 ↑	100.±13.5	87.8±16.4	147.±10.3	153.±7.36 ↑
maze2d-large-sparse-v1	108.±16.7	150.±14.9 ↑	57.9±13.2	143.±17.7 ↑	74.8±16.8	89.9±24.1 ↑	128.±20.7	158.±18.0 ↑
hopper-medium	59.5±4.51	90.2±4.93 ↑	57.2±1.90	96.4±5.74 ↑	38.3±3.88	85.6±6.04 ↑	3.33±0.49	107.±2.31 ↑
halfcheetah-medium	43.6±5.15	43.7±1.72 ↑	43.2±0.82	1.30±1.25	42.0±1.77	44.9±1.95 ↑	-.15±0.08	7.49±2.71 ↑
walker2d-medium	97.6±2.85	101.±3.60 ↑	89.6±3.40	103. ±3.58 ↑	90.8±5.91	94.3±4.46 ↑	4.19±0.42	86.5±7.72 ↑
ant-medium	87.3±5.10	88.9±7.18 ↑	90.4±5.42	42.0±8.30	57.0±8.39	82.2±9.29 ↑	-27.±3.40	47.2±7.43 ↑
Win/Tie/Loss	7/0/0		5/0/2		6/0/1		6/0/1	

SRA has scalability and can support different RL methods.
The performance of methods have been enhanced through SRA.

THANK YOU

Poster Presentation:

Tue, Aug 27 @ 18:30-21:30

#141

Project Page:

Code:

KDD2024

BARCELONA, SPAIN