上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

SJTU

Time Reversal Symmetry for Efficient Robotic Manipulations in Deep Reinforcement Learning

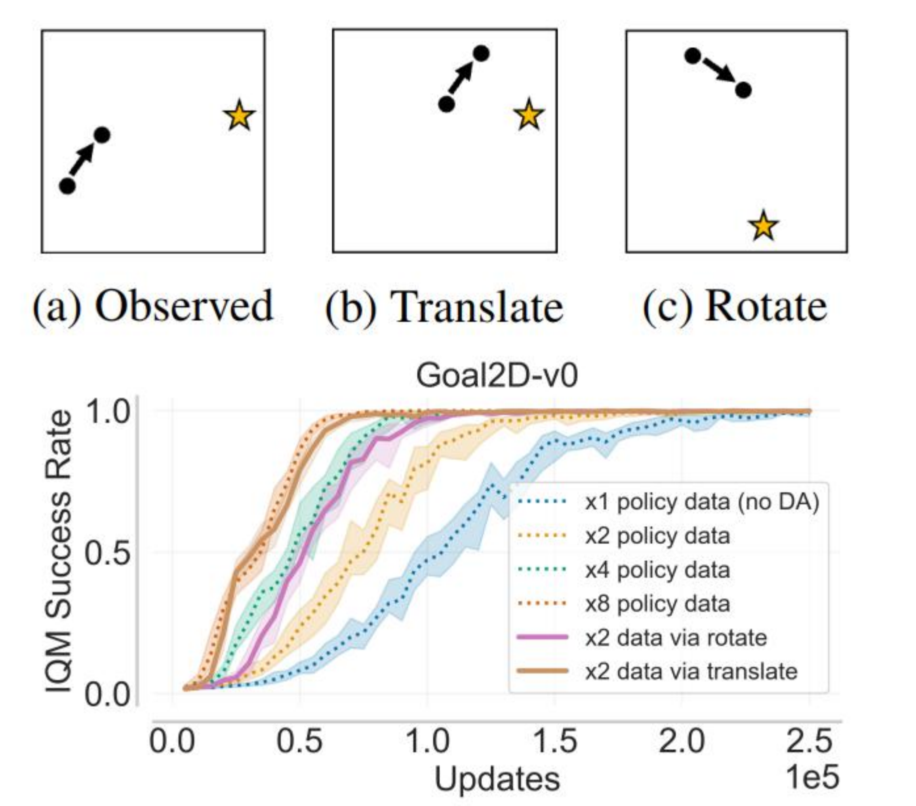
Yunpeng Jiang[1], Jianshu Hu[1], Paul Weng[2], Yutong Ban[1]

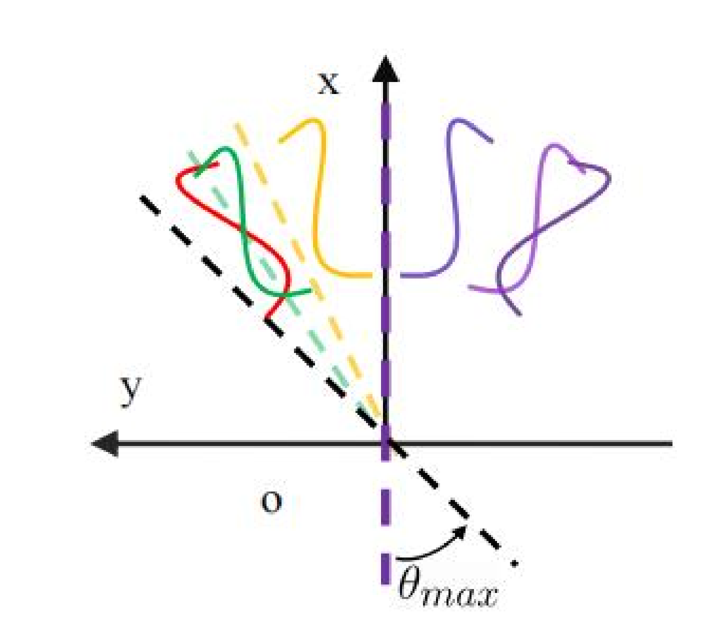
[1]Global College, Shanghai Jiao Tong University

[2]Digital Innovation Research Center, Duke Kunshan University

饮 水 思 源 · 爱 国 荣 校

Symmetry in Deep RL (spatial)



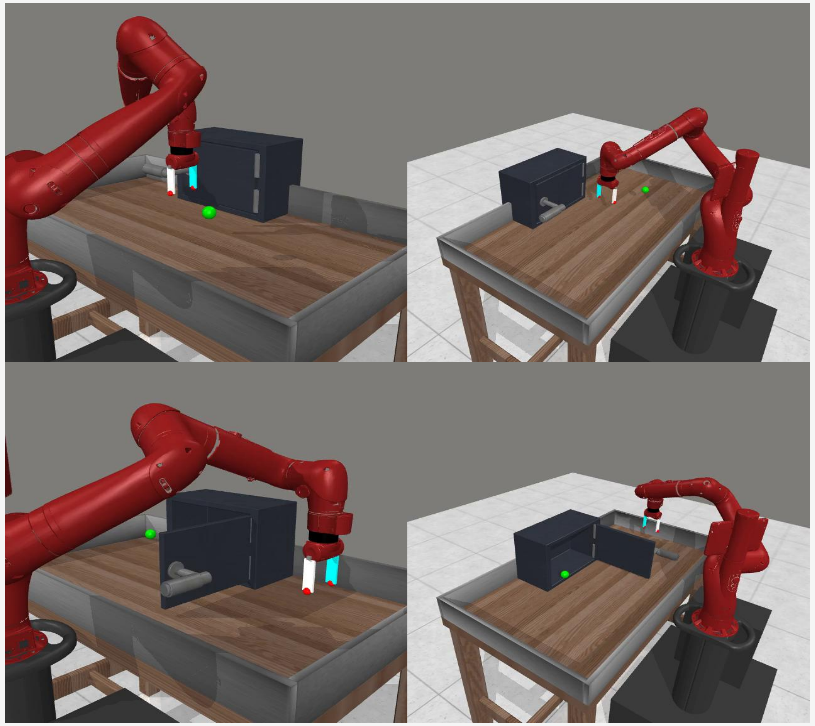


Reflection

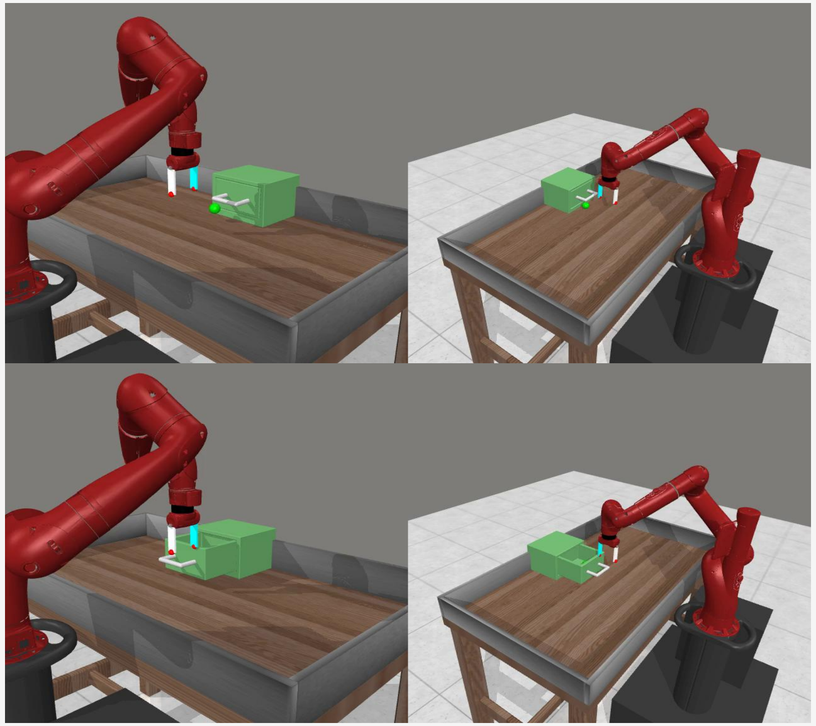
Given the effectiveness of spatial symmetry, can we leverage symmetry in time domain to boost the learning efficiency of deep RL?

Time Reversal Symmetry

- Observation: Tasks often exist as a pair.
- Examples: door opening/closing, drawer opening/closing.



door opening/closing



drawer opening/closing

Time Reversal Symmetry

- Observation: Tasks often exist as a pair.
- Examples: plate slide side (back), window opening/closing.

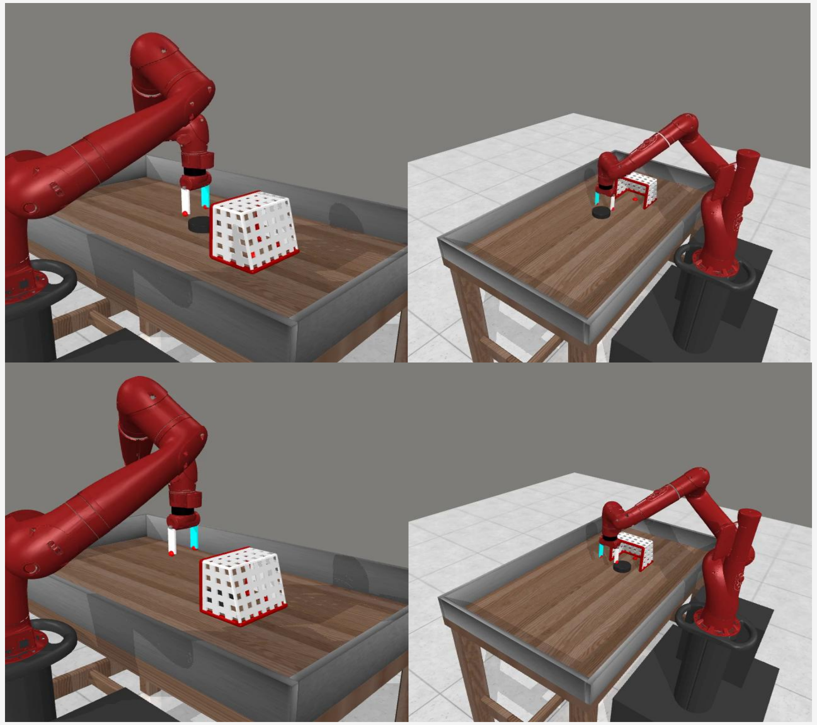
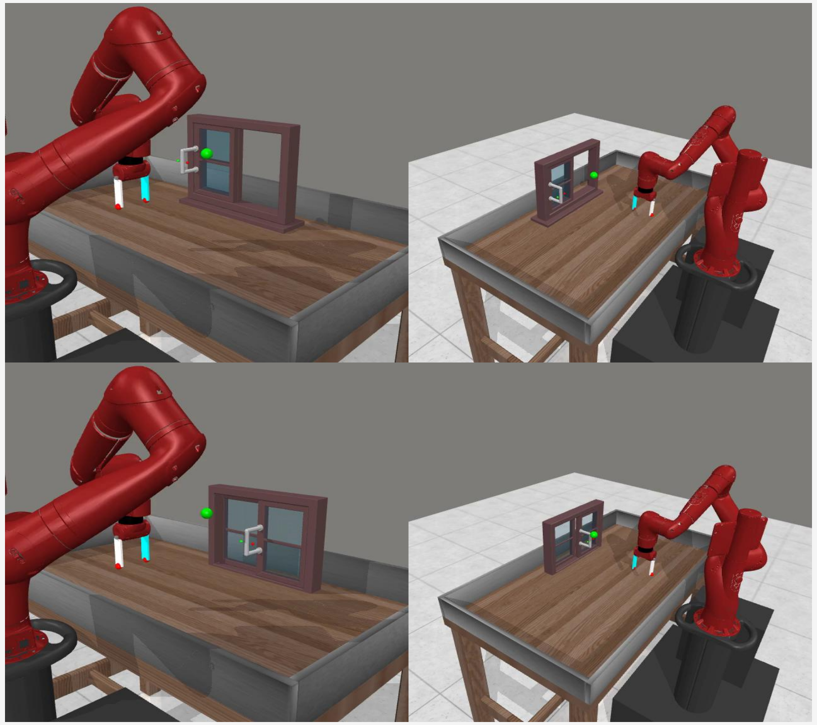
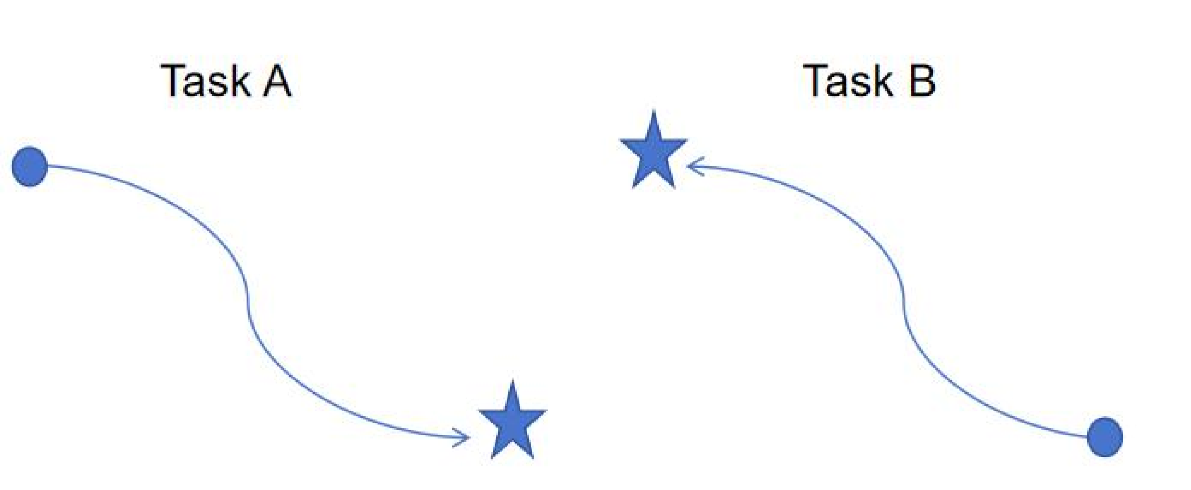


plate slide side (back)



window opening/closing

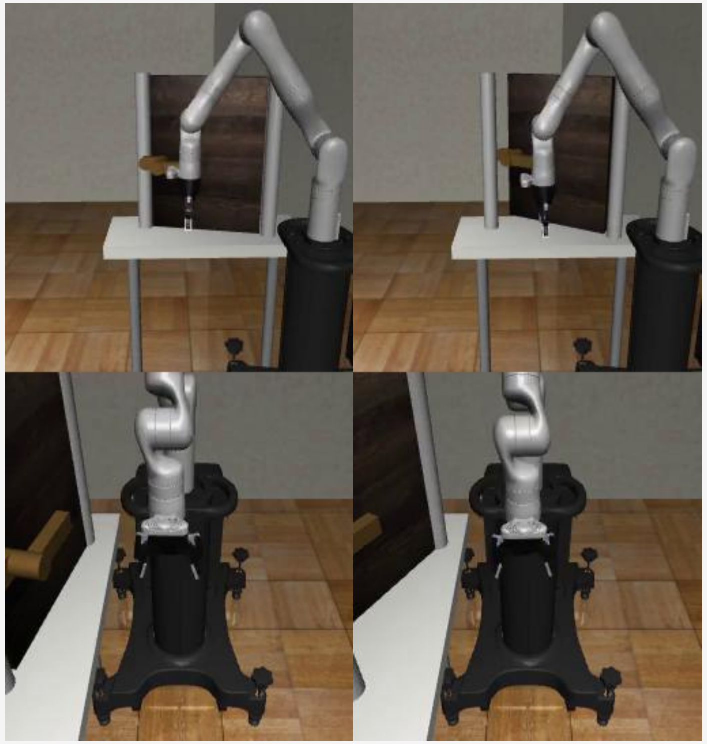
Time Reversal Symmetry



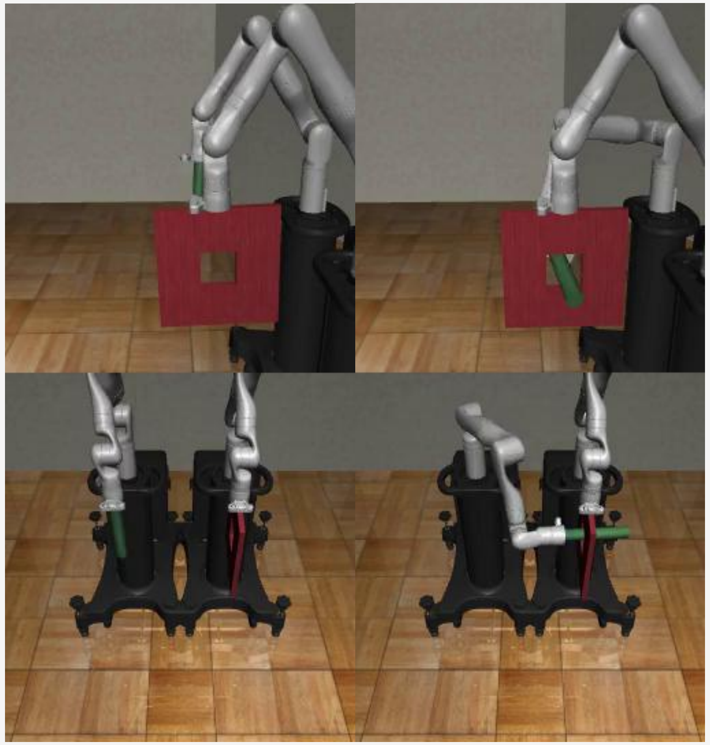
- Hypothesis: For some task pair (A, B), task A can be solved more quickly with reversed trajectories from task B.
- Objective of our work: Develop a method that enhances data efficiency of DRL agents with time reversal symmetry.
- Outcome: An agent that can solve two/multiple tasks simultaneously and much faster with the help of time reversal symmetry.

Full Time Reversal Symmetry

- For one transition (s_t, a_t, s_{t+1}) and the transition function T, if there exists $\overleftarrow{a_t}$ such that $T(s_{t+1}|s_t, a_t) = T(s_t|s_{t+1}, \overleftarrow{a_t})$, we say that the state pair (s_t, s_{t+1}) satisfies full time reversal symmetry.



door opening/closing outward



peg insertion/removal

Proposed Technique1: Trajectory Reversal Augmentation with Dynamics-Aware Filtering

- Reverse transitions: $(s_t, a_t, s_{t+1}) \rightarrow (s_{t+1}, \overleftarrow{a_t}, s_t)$.

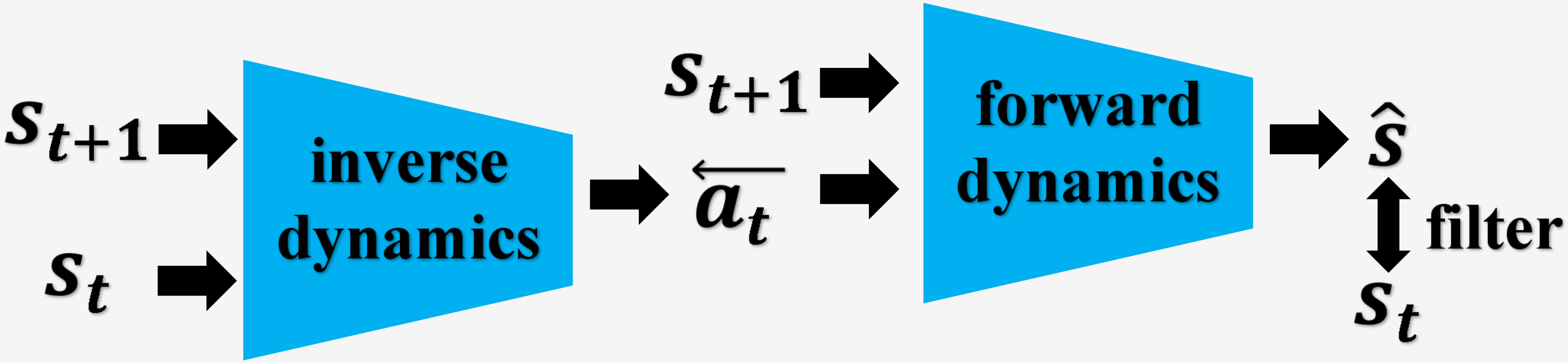
- Core idea: data augmentation using reversed transitions.

- Filter out fake (irrversible) transitions.

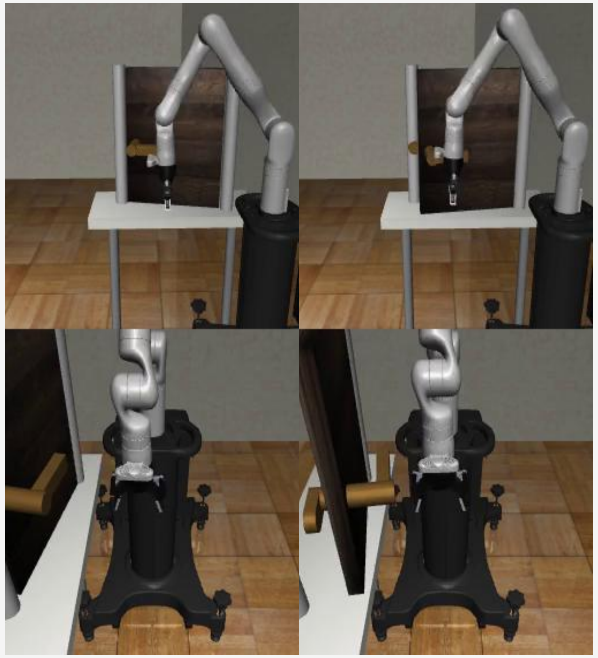
1. Train an inverse dynamics model: output $\overleftarrow{a_t}$ with s_{t+1} and s_t as inputs.

2. Train a forward dynamics model: output $\hat{s}$ with s_{t+1} and $\overleftarrow{a_t}$ as inputs.

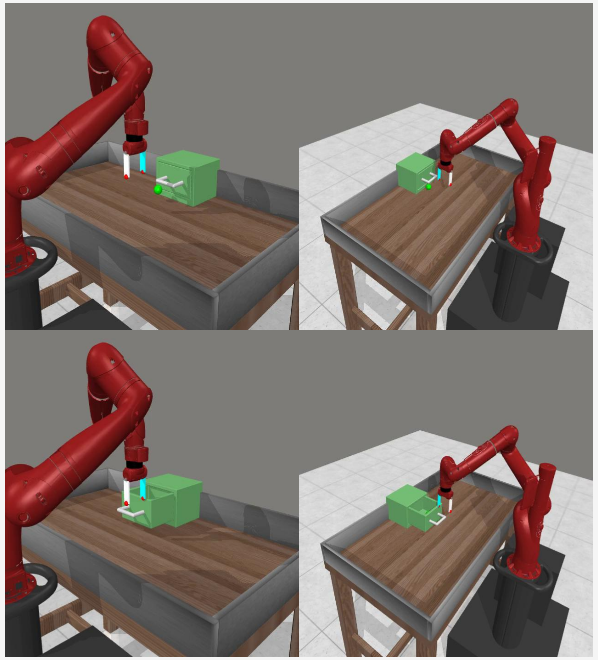
3. Filter based on the error between s_t and $\hat{s}$.



What if there are lots of irreversible transitions?



door opening/closing inward



drawer opening/closing

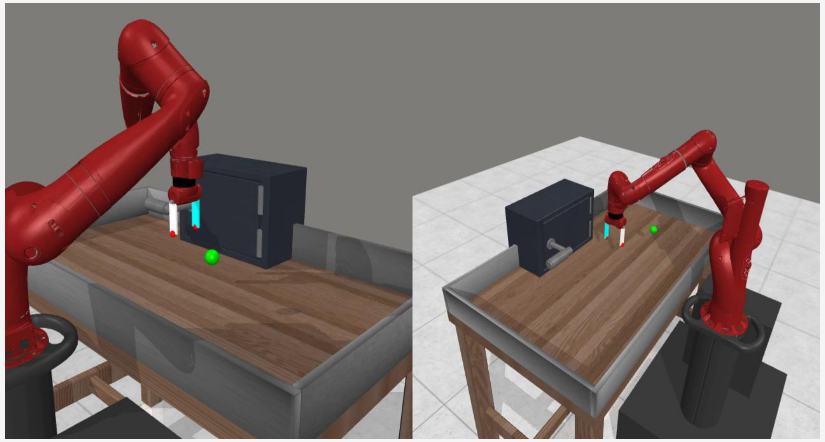
- The robotic arm is opening the door/drawer by grasping and pulling it while closing the door/drawer by pushing it, which creates irreversible transitions.
- However, we can observe that the trajectories of the objects are actually reversible.
- We need to find out how to leverage this property.

Partial Time Reversal Symmetry

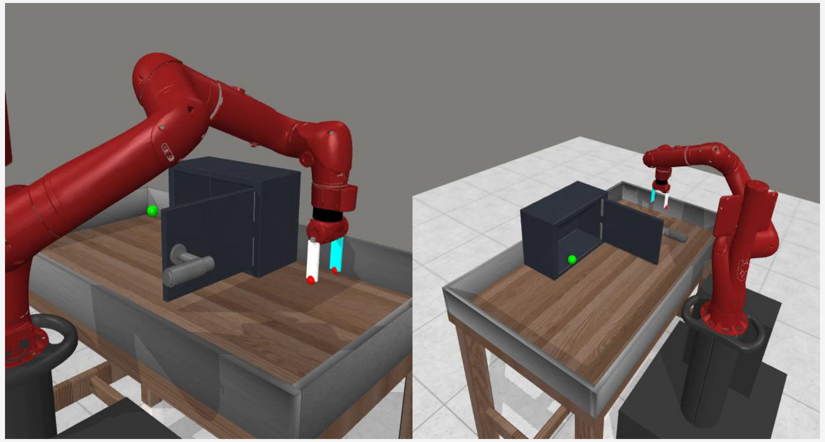
- Assume that a state s can be decomposed into two parts (x, y).

- For one state pair (s_t, s_{t+1}) and the transition function T, if there exist $(\overleftarrow{y_t}, \overleftarrow{y_{t+1}})$ and $(\overleftarrow{a_t}, \overleftarrow{a_{t+1}})$ such that $T(s_{t+1}|s_t, a_t) = T(\overleftarrow{s_t}|\overleftarrow{s_{t+1}}, a_t)$, we say that the state pair (s_t, s_{t+1}) satisfies partial time reversal symmetry.

- Intuitively, x can be the part that is reversible (eg. containing object state information).



door opening



door closing

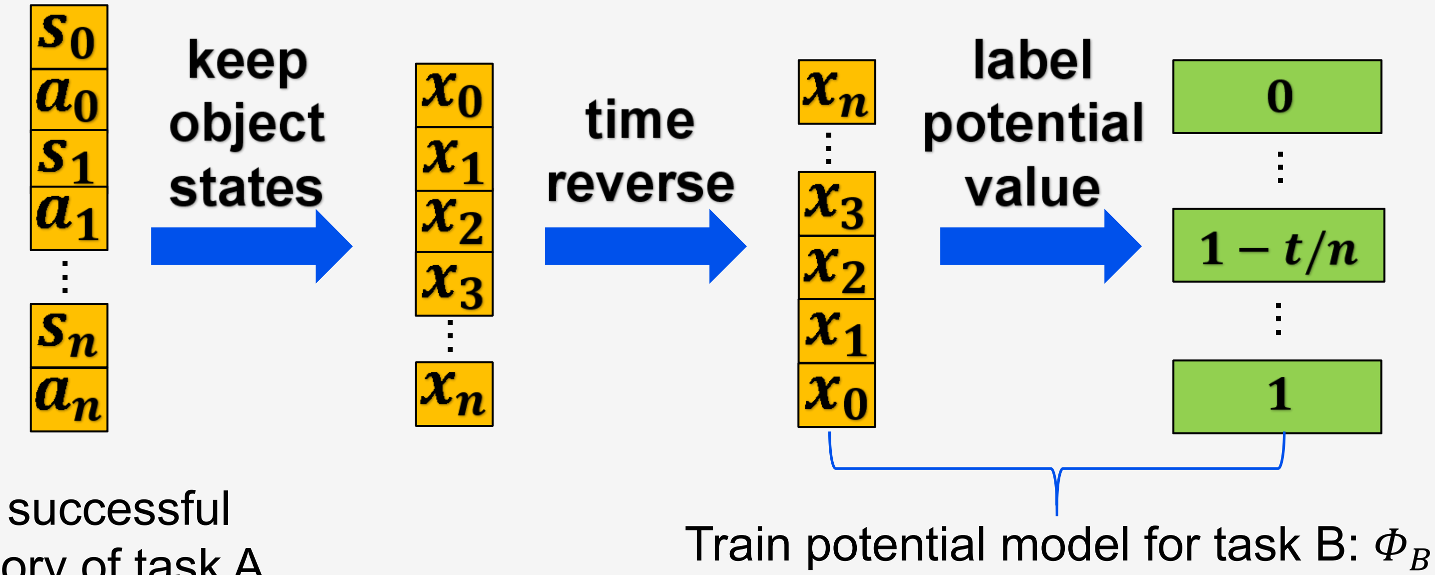
Proposed Technique2: Time Reversal Symmetry Guided Reward Shaping

- Core idea: reward shaping using reversed trajectories.

- Use reversed trajecotries of object states (reversible part) to relabel the reward terms.

1. Train a potential model: output potential value $\Phi(x_t)$ with x_t as input.

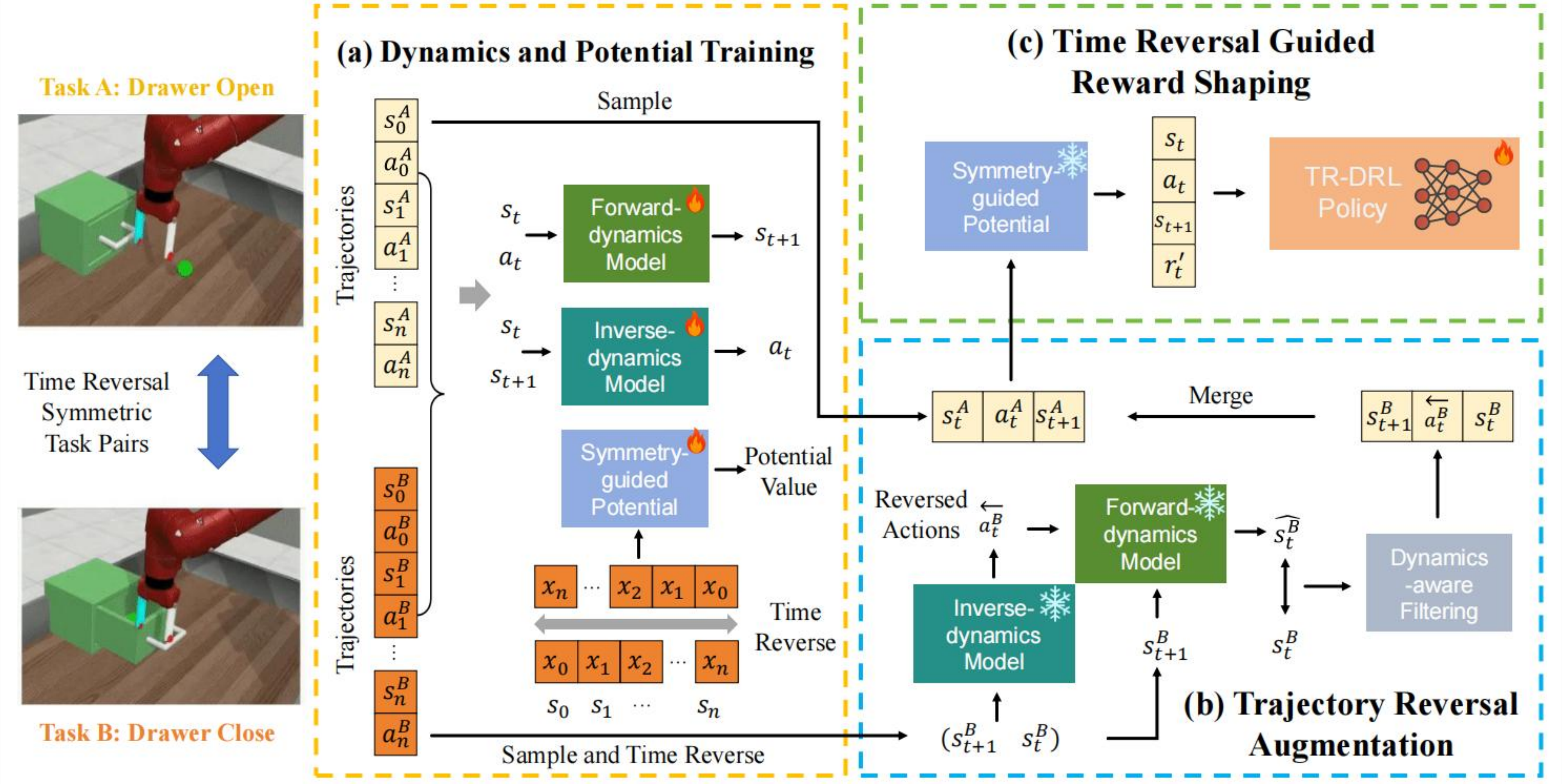
2. The reward of one transition (s_t, a_t, s_{t+1}) can be reshaped as $r_t + \Phi(x_{t+1}) - \Phi(x_t)$.



One successful trajectory of task A

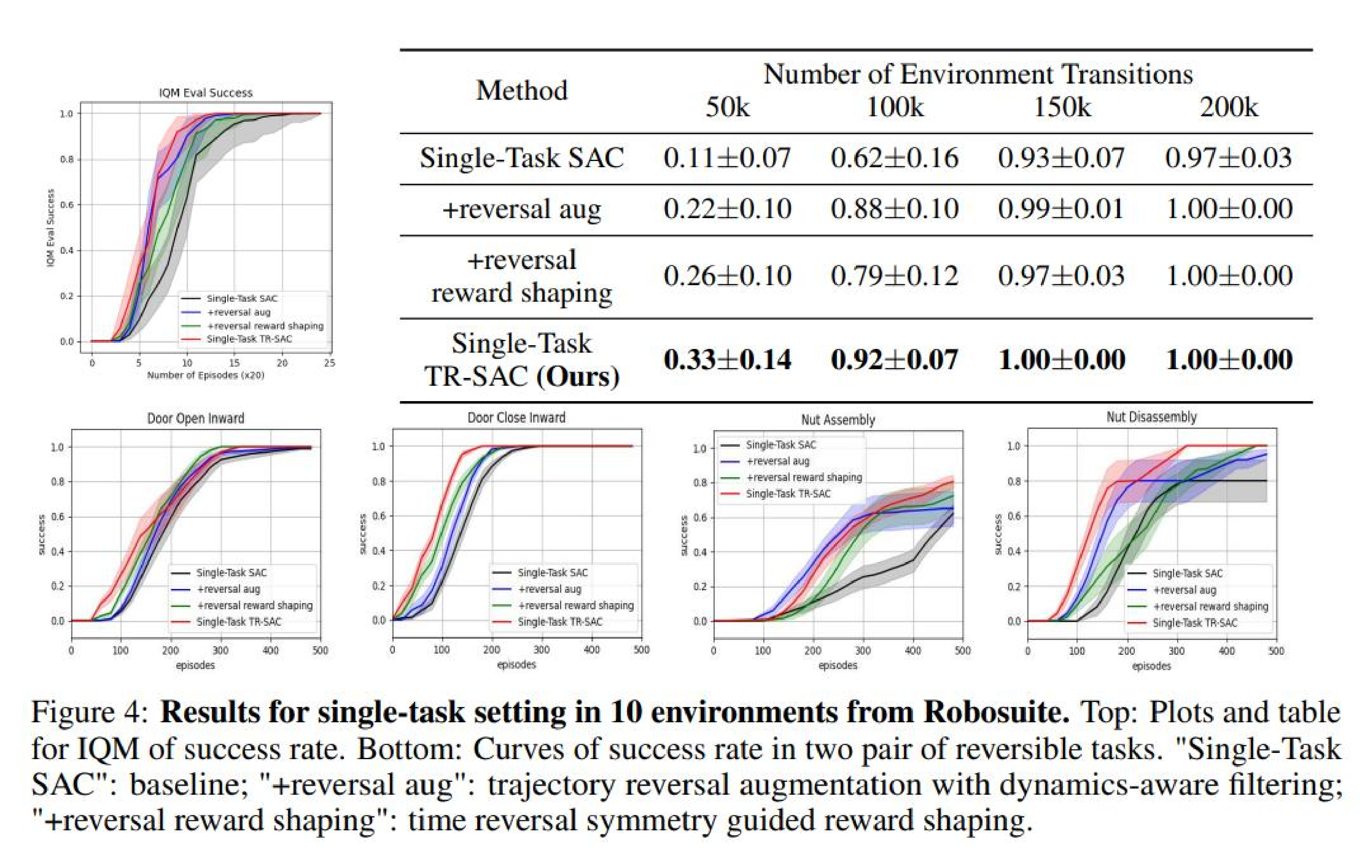
Train potential model for task B: Φ_B

Method Diagram of TR-DRL



Task A: Drawer Open
Time Reversal Symmetric Task Pairs
Task B: Drawer Close
(a) Dynamics and Potential Training
Trajectories
s_0^A a_0^A s_1^A a_1^A s_n^A a_n^A
Trajectories
s_0^B a_0^B s_1^B a_1^B s_n^B a_n^B
Sample
s_t a_t
Forward-dynamics Model
s_{t+1}
s_t s_{t+1}
Inverse-dynamics Model
a_t
Symmetry-guided Potential
Potential Value
x_n ··· x_2 x_1 x_0
Time Reverse
x_0 x_1 x_2 ··· x_n
s_0 s_1 ··· s_n
Sample and Time Reverse
(c) Time Reversal Guided Reward Shaping
Symmetry-guided Potential
s_t a_t s_{t+1} r_t'
TR-DRL Policy
s_t^A a_t^A s_{t+1}^A
Merge
s_{t+1}^B $\overleftarrow{a}_t^B$ s_t^B
Reversed Actions $\overleftarrow{a}_t^B$
Forward-dynamics Model
$\widehat{s_t^B}$
Dynamics-aware Filtering
Inverse-dynamics Model
s_{t+1}^B
s_t^B
$(s_{t+1}^B \quad s_t^B)$
(b) Trajectory Reversal Augmentation

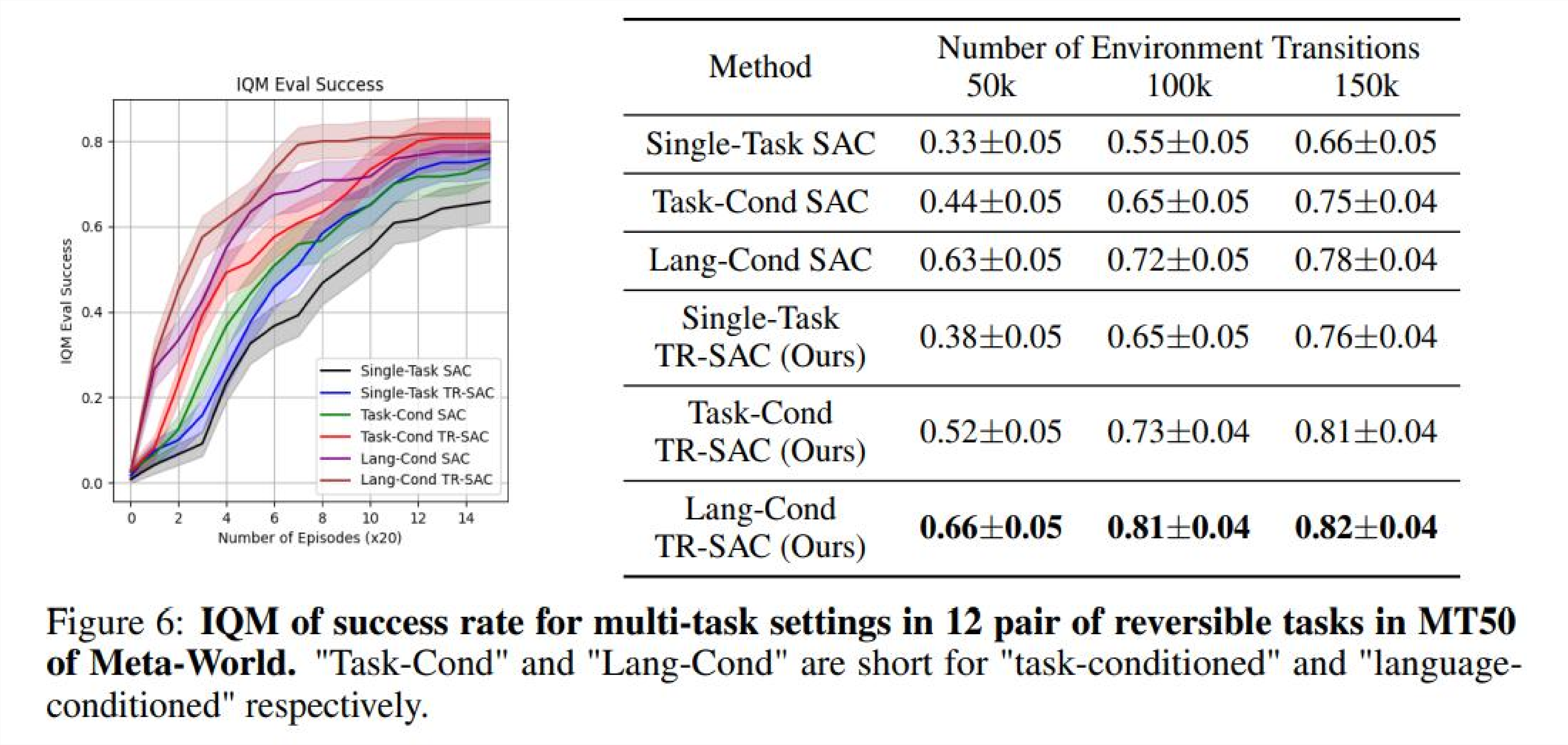
Experimental Results in Robosuite



Method	Number of Environment Transitions			
	50k	100k	150k	200k
Single-Task SAC	0.11±0.07	0.62±0.16	0.93±0.07	0.97±0.03
+reversal aug	0.22±0.10	0.88±0.10	0.99±0.01	1.00±0.00
+reversal reward shaping	0.26±0.10	0.79±0.12	0.97±0.03	1.00±0.00
Single-Task TR-SAC (**Ours**)	**0.33±0.14**	**0.92±0.07**	**1.00±0.00**	**1.00±0.00**

Figure 4: **Results for single-task setting in 10 environments from Robosuite.** Top: Plots and table for IQM of success rate. Bottom: Curves of success rate in two pair of reversible tasks. "Single-Task SAC": baseline; "+reversal aug": trajectory reversal augmentation with dynamics-aware filtering; "+reversal reward shaping": time reversal symmetry guided reward shaping.

Experimental Results in MetaWorld (MT50)



Method	Number of Environment Transitions 50k	 100k	 150k
Single-Task SAC	0.33±0.05	0.55±0.05	0.66±0.05
Task-Cond SAC	0.44±0.05	0.65±0.05	0.75±0.04
Lang-Cond SAC	0.63±0.05	0.72±0.05	0.78±0.04
Single-Task TR-SAC (Ours)	0.38±0.05	0.65±0.05	0.76±0.04
Task-Cond TR-SAC (Ours)	0.52±0.05	0.73±0.04	0.81±0.04
Lang-Cond TR-SAC (Ours)	**0.66±0.05**	**0.81±0.04**	**0.82±0.04**

Figure 6: **IQM of success rate for multi-task settings in 12 pair of reversible tasks in MT50 of Meta-World.** "Task-Cond" and "Lang-Cond" are short for "task-conditioned" and "language-conditioned" respectively.

Conclusion

- We propose TR-DRL, a framework leveraging time reversal symmetry to enhance sample efficiency of DRL algorithms.
- Key contributions include a novel notion of partial time reversal symmetry, trajectory reversal augmentation with dynamics-aware filtering, and symmetry-guided reward shaping.
- Experiments on Robosuite and Metaworld demonstrate improved agent performance and learning efficiency.

Thanks for watching

- If our work interests you, you may explore further with more video demonstrations on our project website and apply our method using our open-source code repository.

project website

code repository