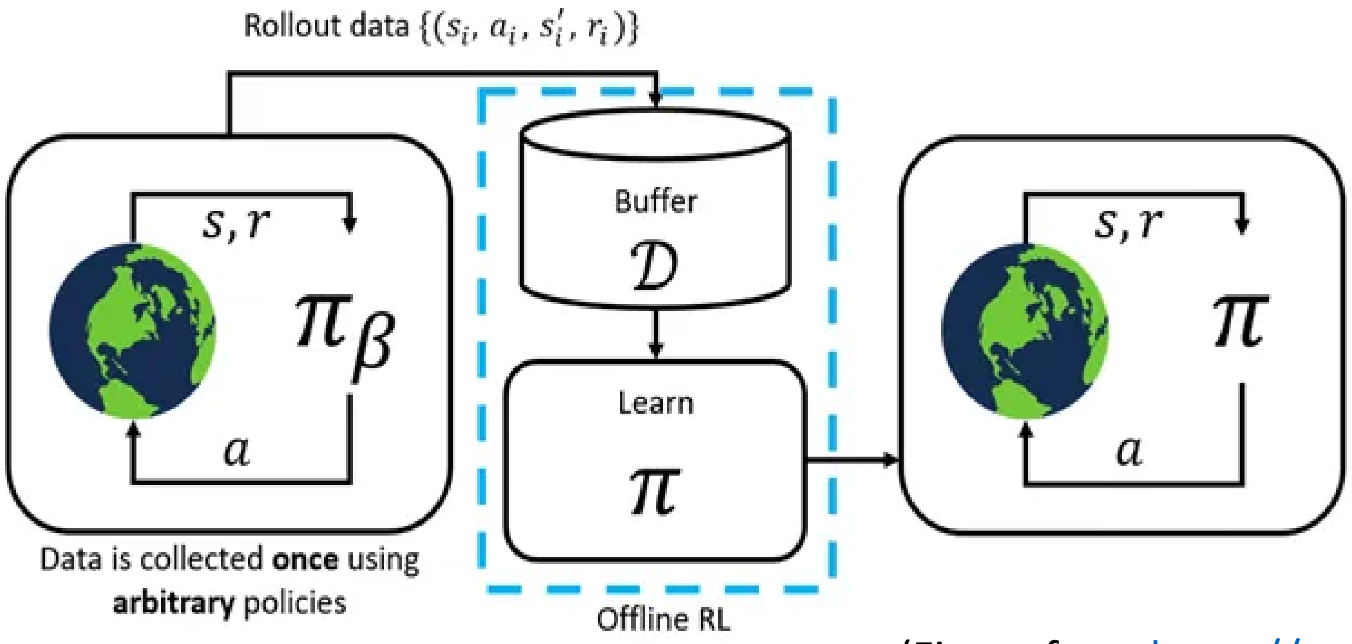
Free from Bellman Completeness: Trajectory Stitching via Model-based Return-conditioned Supervised Learning

Zhaoyi Zhou[1], Chuning Zhu[2], Runlong Zhou[2], Qiwen Cui[2], Abhishek Gupta[2], Simon Shaolei Du[2]

[1]Tsinghua University, [2]University of Washington

Offline Reinforcement Learning

- Dataset collected by sub-optimal behavior policy
- No interactions with environment during policy training
- Goal: Learn a (near-)optimal policy



(Figure from https://towardsdatascience.com/the-power-of-offline-reinforcement-learning-5e3d3942421c)

Dynamic Programming (DP)

- Bellman operation

$$\widehat{Q}_h \leftarrow \mathcal{B}\widehat{Q}_{h+1} = \{r(s,a) + \mathbb{E}_{s' \sim \mathcal{T}(\cdot|s,a)} \sup_{a'} \widehat{Q}_{h+1}(s',a')\}_{(s,a)\in\mathcal{S}\times\mathcal{A}}.$$

- Pro: Enable trajectory stitching
- Con: Could diverge in practice, due to Bellman completeness

Definition 2.2 (Bellman complete)**.** A function approximation class $\mathcal{F}$ is Bellman complete under Bellman operator $\mathcal{B}$ if $\max_{Q\in\mathcal{F}} \min_{Q'\in\mathcal{F}} \|Q' - \mathcal{B}Q\| = 0$.

Question

Can we design offline reinforcement learning algorithms that:

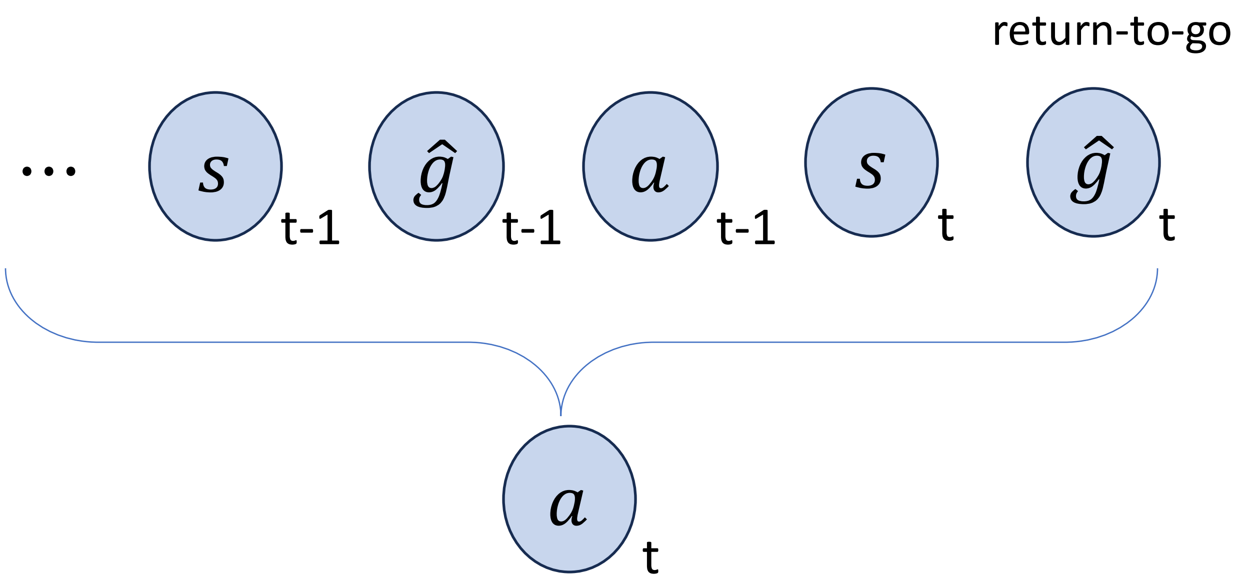
1) avoid Bellman completeness (Pro of RCSL);

2) do trajectory stitching (Pro of DP)?

RCSL (Training)

- Return-conditioned supervised learning (RCSL)

- Learn a return-conditioned policy



RCSL (Evaluation)

Algorithm 1 Return-conditioned evaluation process

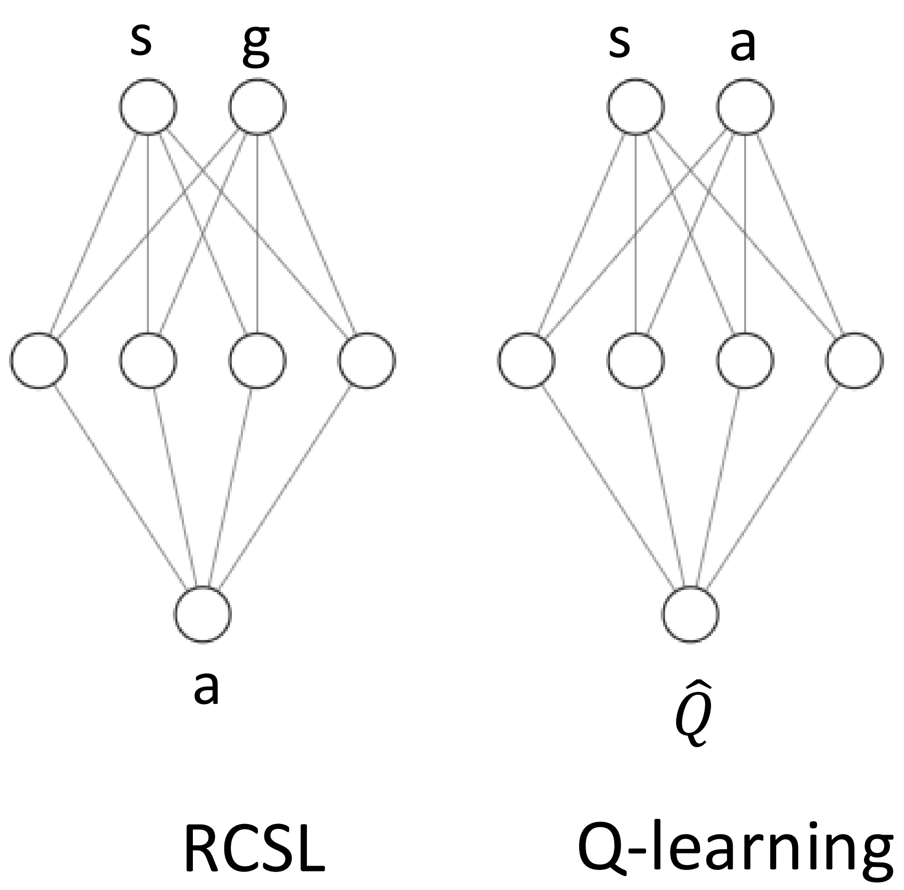
Input: Return-conditioned policy $\pi : \mathcal{S} \times \mathbb{R} \to \Delta(\mathcal{A})$, desired RTG $\tilde{g}_1$.
Observe s_1.
for $h = 1, 2, \cdots, H$ **do**
    Sample action $a_h \sim \pi(\cdot | s_h, \tilde{g}_h)$.
    Observe reward r_h and next state s_{h+1}.
    Update RTG $\tilde{g}_{h+1} = \tilde{g}_h - r_h$.
Return: Actual RTG $g = \sum_{h=1}^{H} r_h$.

Contributions

- We theoretically show how RCSL can outperform DP in deterministic environments.
- We theoretically show the shortcomings of RCSL when the off-policy dataset does not cover optimal trajectories.
- We propose **MBRCSL**, a novel framework that achieves trajectory stitching without suffering from the challenge of Bellman completeness.

Strength of RCSL: Setup

- Compare RCSL with Q-learning
- Similar network architecture: MLP
- Analyze hidden neurons needed
  - RCSL: represent optimal policy
  - Q-learning: satisfy Bellman completeness



Strength of RCSL: Result

Thm: There exists a series of MDPs (state space $|\mathcal{S}| \to \infty$), and associated datasets such that:

1) RCSL represents optimal policy with $O(1)$ hidden neurons

2) Q-learning cannot satisfy Bellman completeness with $O(|\mathcal{S}|)$ hidden neurons.

RCSL scales with state space; Q-learning does not!

Strength of RCSL: Proof Idea

- Q: What happens when Bellman completeness is satisfied?

- A: The reward function must be representable!

- $\mathcal{B}Q(s,a) = r(s,a)$ if $Q \equiv 0$.

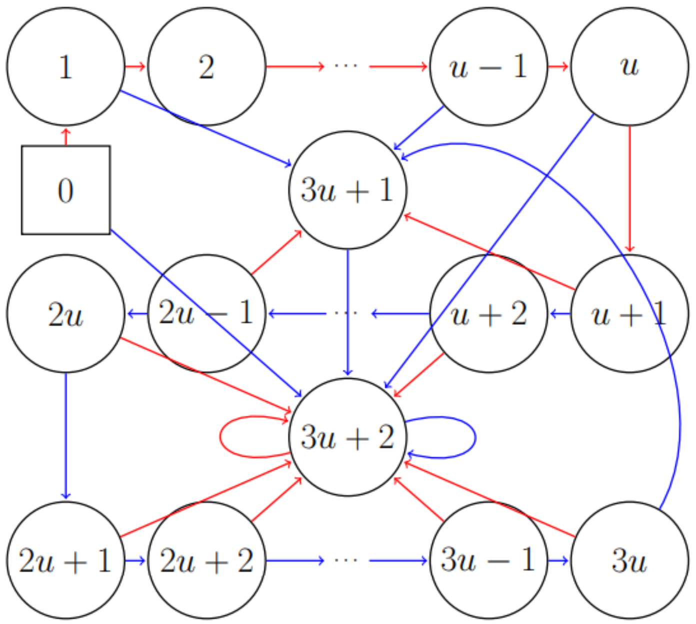
Let it be 0 function!

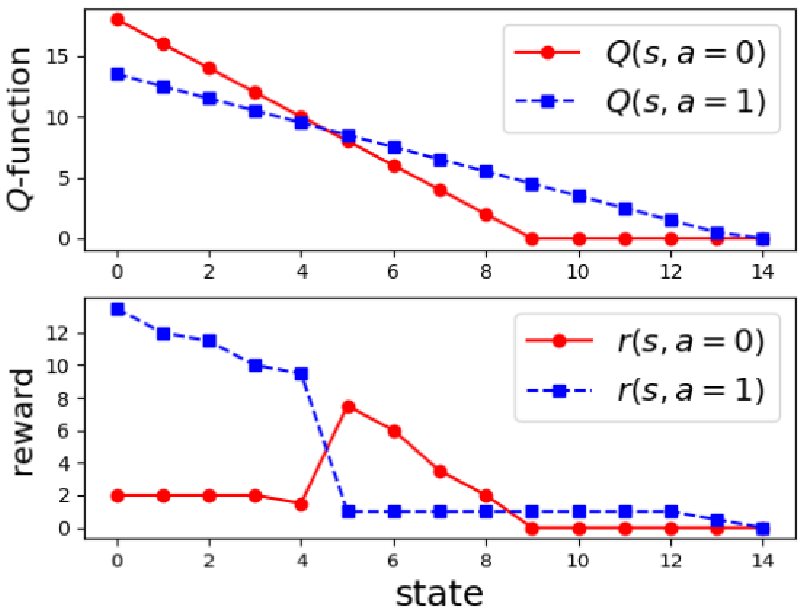
$$\widehat{Q}_h \leftarrow \mathcal{B}\widehat{Q}_{h+1} = \{r(s,a) + \mathbb{E}_{s' \sim \mathcal{T}(\cdot|s,a)} \sup_{a'} \widehat{Q}_{h+1}(s',a')\}_{(s,a)\in\mathcal{S}\times\mathcal{A}}.$$

Strength of RCSL: Proof Idea

Construct a class of MDPs (“LinearQ”):

- Optimal Q-function representable with constant hidden neurons
- Reward function must use $O(|\mathcal{S}|)$ hidden neurons to represent





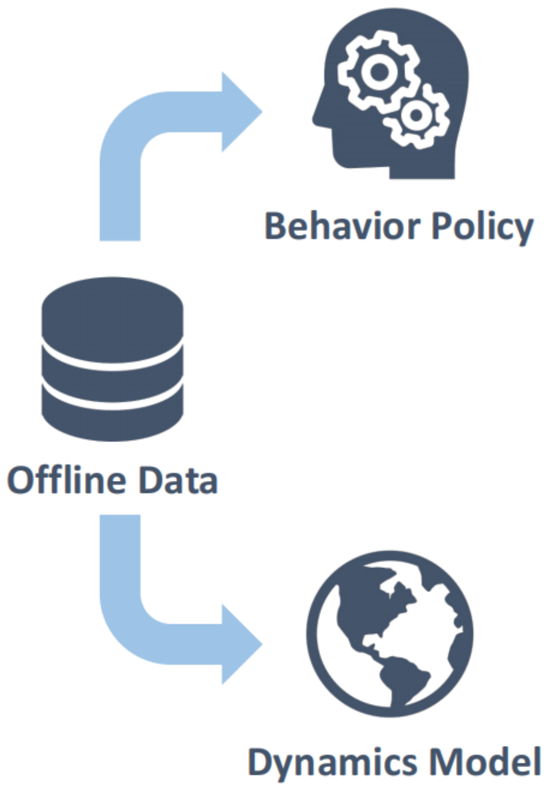
Weakness of RCSL

- No Trajectory Stitching
- For either special case of RCSL:
  - Markovian RCSL
  - Decision transformer with any context length
- Even with deterministic transition & uniform coverage of dataset.

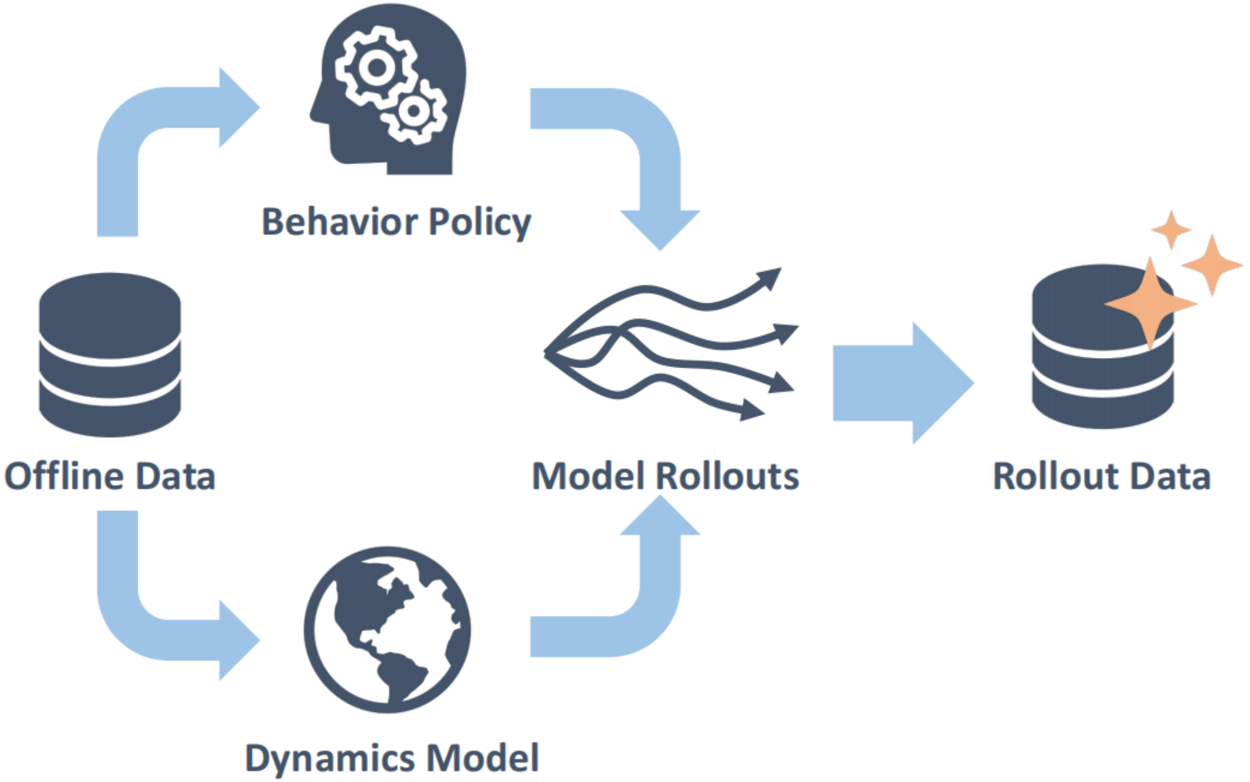
Can we design off-policy reinforcement learning algorithms that are:
1) Free from Bellman completeness
2) Retaining trajectory stitching ability of DP-based methods?

Model-based RCSL (MBRCSL)

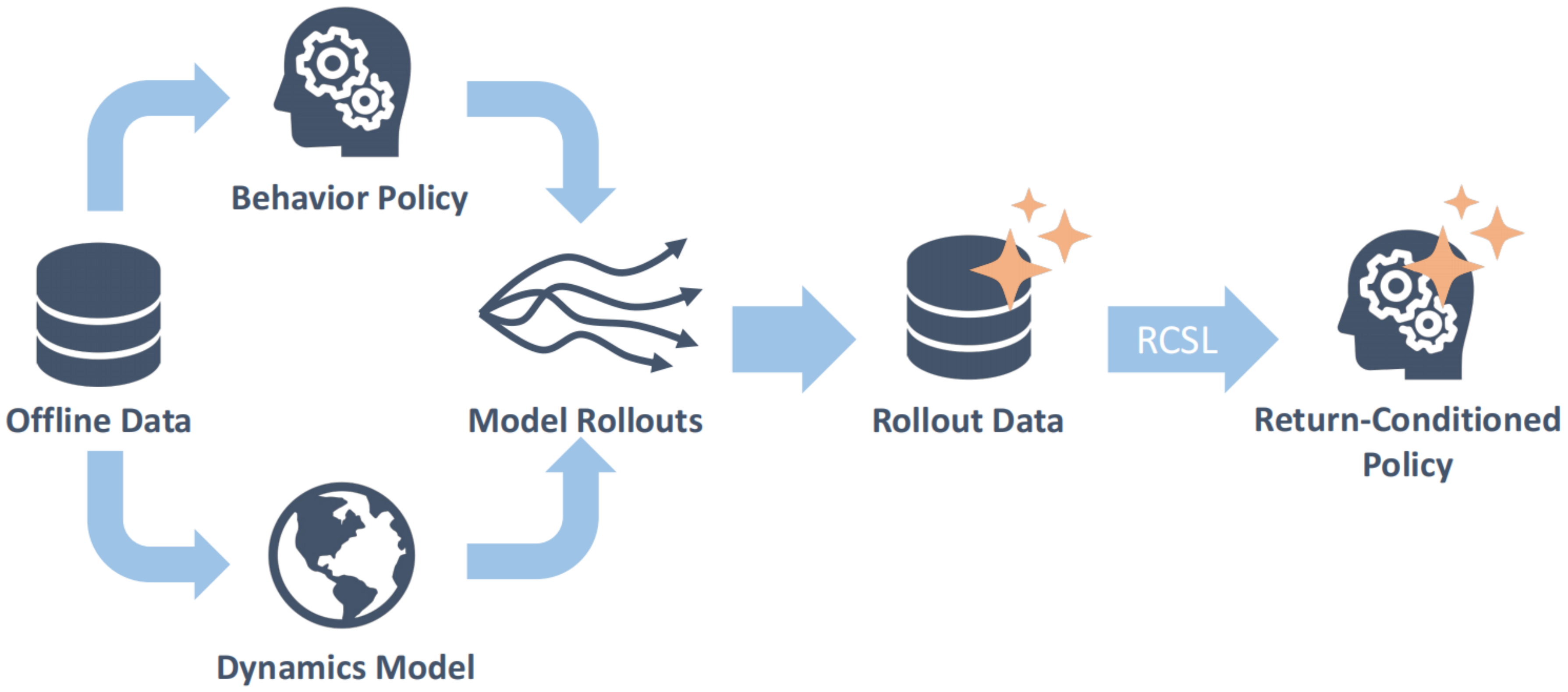
Model-based RCSL (MBRCSL)



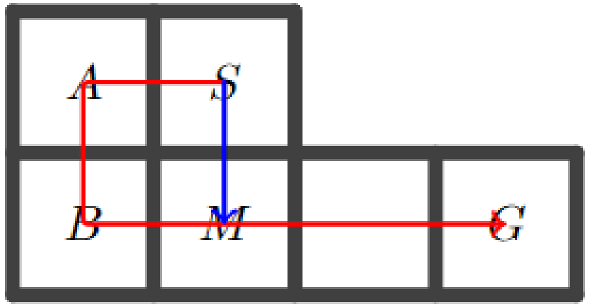
Model-based RCSL (MBRCSL)

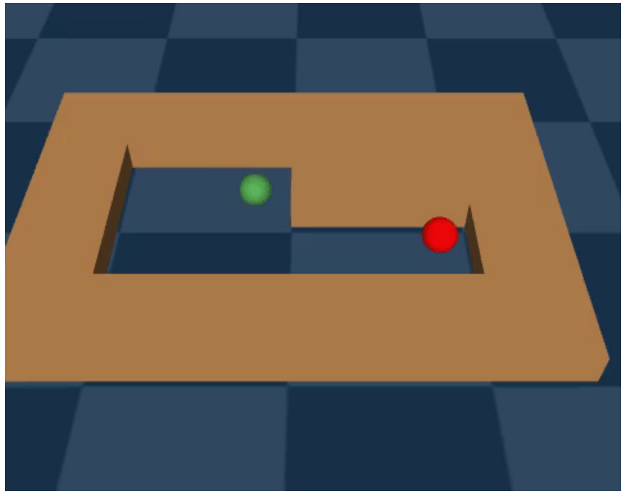


Model-based RCSL (MBRCSL)



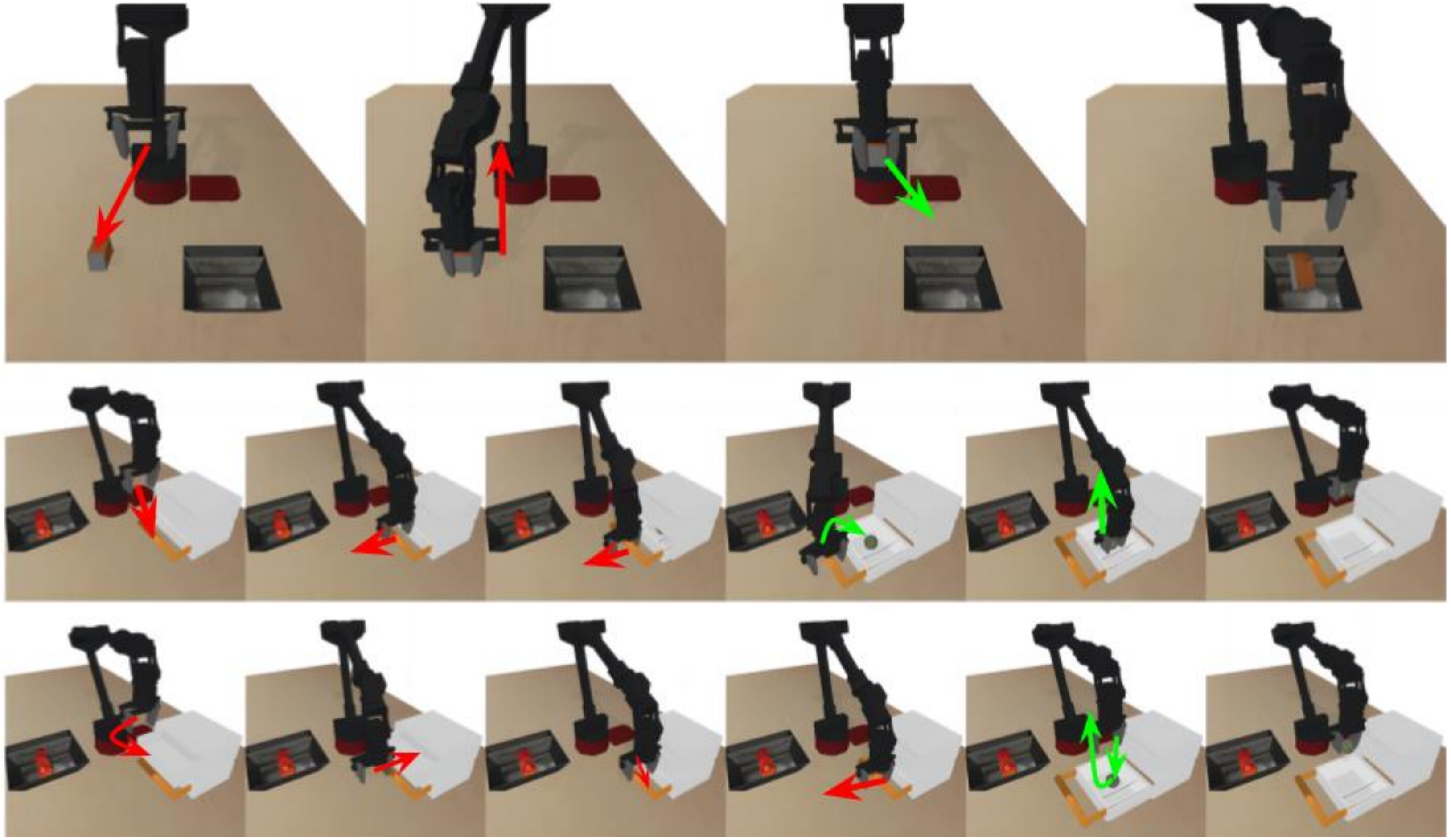
Experiments: Point Maze





Dataset	MBRCSL (ours)	COMBO	MOPO	CQL	DT	%BC
Pointmaze	**91.5±7.1**	56.6±50.1	77.9±41.9	34.8± 24.9	57.2±4.1	54.0±9.2

Experiments: Simulated Robotics



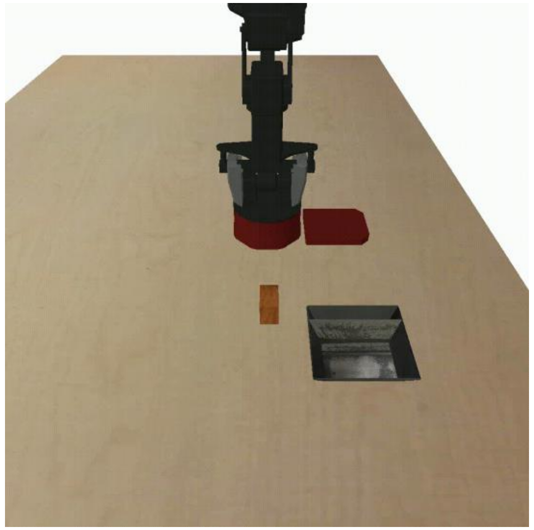
Pick Place

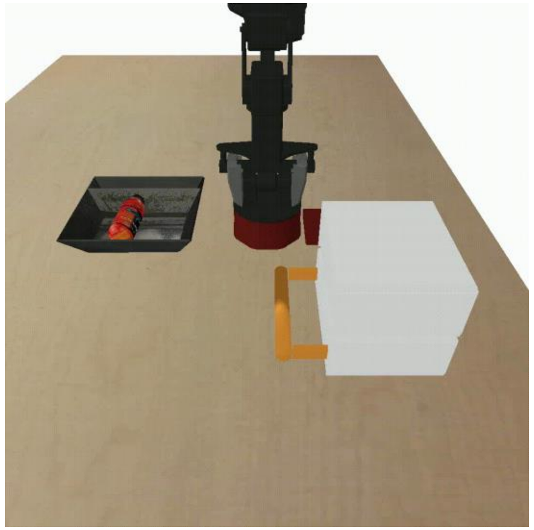
Closed Drawer

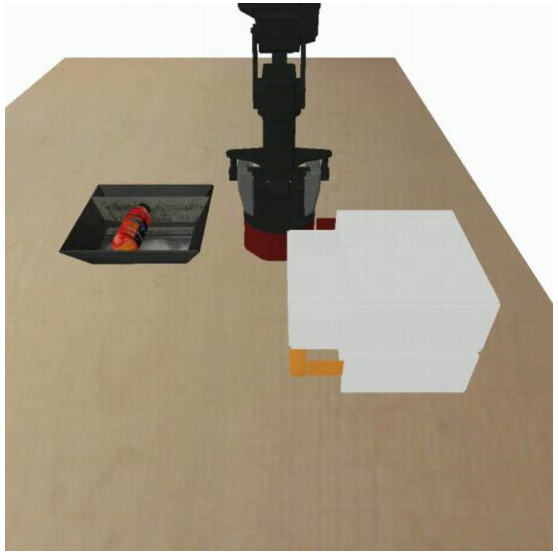
Blocked Drawer

Experiments: Simulated Robotics

Task	MBRCSL (ours)	CQL	COMBO	DT	BC	MBCQL
PickPlace	**0.40±0.16**	0.22±0.35	0±0	0±0	0.07± 0.03	0.08±0.05
ClosedDrawer	**0.51±0.12**	0.11±0.08	0±0	0±0	0.38±0.02	0±0
BlockedDrawer	**0.68±0.09**	0.34±0.23	0±0	0±0	0.61±0.02	0±0







Takeaways

- RCSL outperforms DP-based off-policy algorithms given expert dataset, due to freedom from Bellman-completeness

- RCSL cannot do trajectory stitching as DP-based methods can

- MBRCSL retains trajectory stitching of DP-based methods, while avoiding Bellman-completeness.