Robust Imitation of a Few Demonstrations with a Backwards Dynamics Model

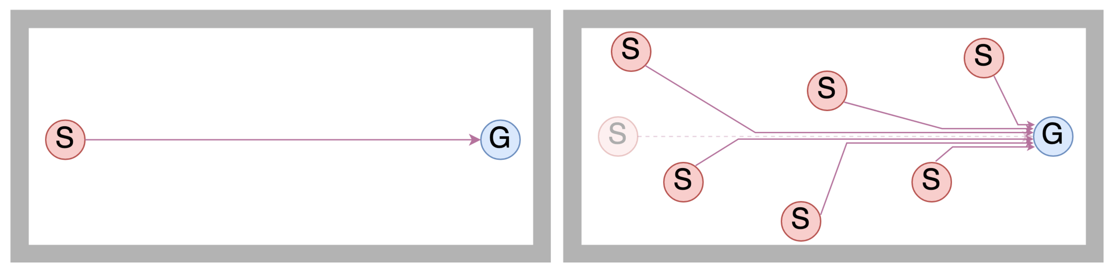
Jung Yeon Park   Lawson L.S. Wong

Khoury College of Computer Sciences, Northeastern University

Motivation

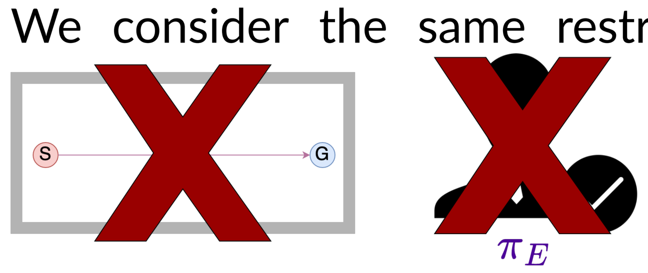
- Expert demonstrations can help RL solve difficult tasks, but naive cloning suffers from covariate shift
- Demonstrations are costly to obtain in many real world applications

Question: Given a limited number of demonstrations from a single start state, how to learn a policy that can solve the task from new start states?



Problem Setting

We consider the same restrictive setting as BC.



$\tau_E = \{s_0, a_0, s_1, a_1, \ldots\}$

Objective: Learn a robust policy that can solve the task from start states unseen in the training data. Robustness is measured as

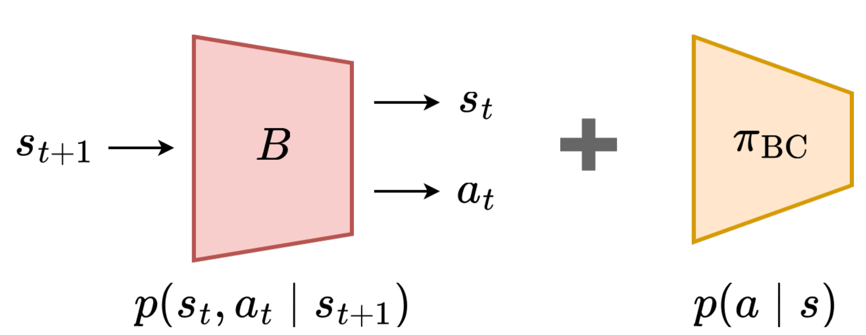
$$R(\pi_\theta) = \mathbb{E}_{s_0 \in S_R}\left[\mathbb{1}\{\exists t \leq T, s_t \in \mathcal{G}\}\right], \quad (1)$$

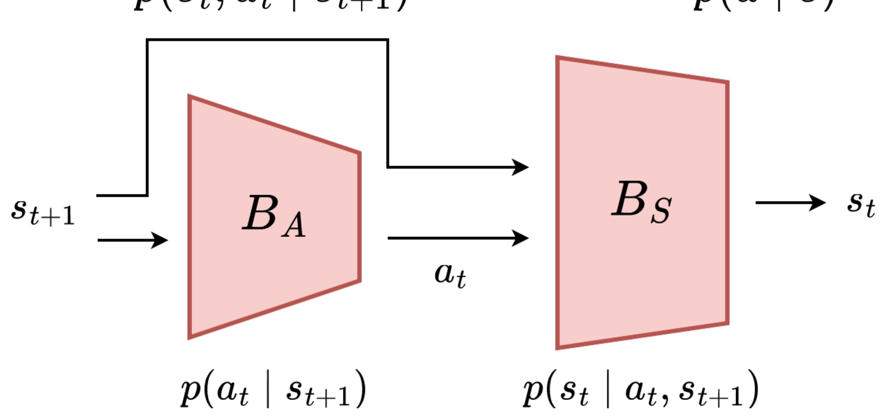
where $S_R \supset S_0$.

Method

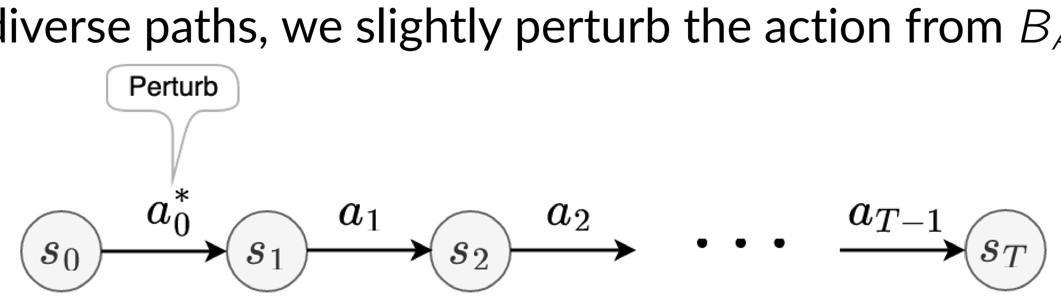
Backwards Model-based Imitation Learning (BMIL)

Key Idea: Pair a generative backwards dynamics model with an imitation learning policy.





Using B, we generate short model rollouts starting from every state in the demonstrations. To produce diverse paths, we slightly perturb the action from B_A.

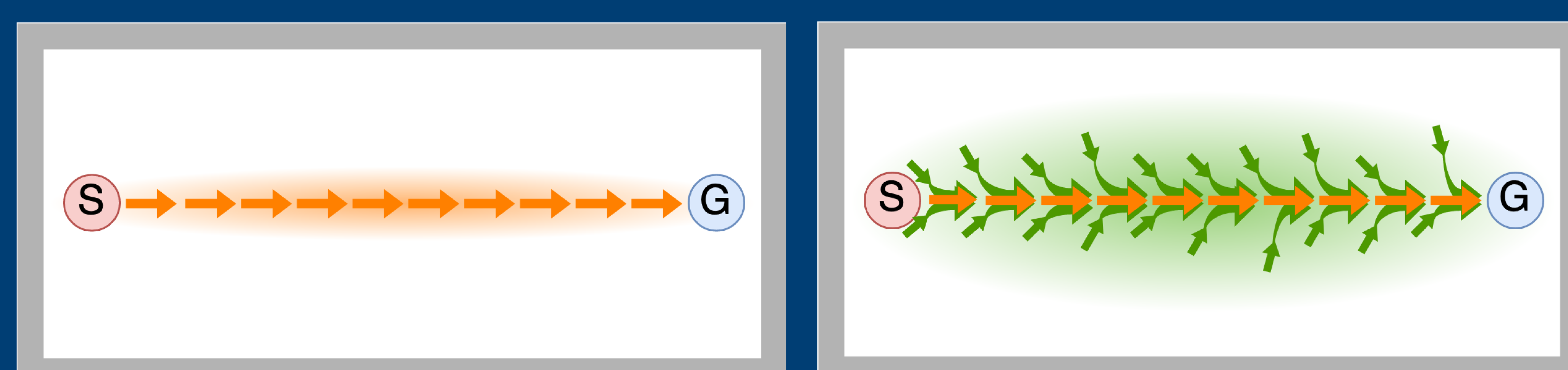


The policy is then trained on both the rollouts and demonstrations.

$$\mathcal{L} = p_d \mathcal{L}_{BC} + (1 - p_d)\mathbb{E}_{(s,a)\sim\tau_B}\left[-\log \pi_\theta(a \mid s)\right], \quad (2)$$

where p_d is the probability of sampling from demonstration data.

In **imitation learning** with no environment interactions, a **backwards dynamics model** can help provide more synthetic data to train a robust policy. By perturbing the model rollouts, the policy learns a wider region of attraction and can **generalize to start states unseen** in the demonstrations.



Scan for paper

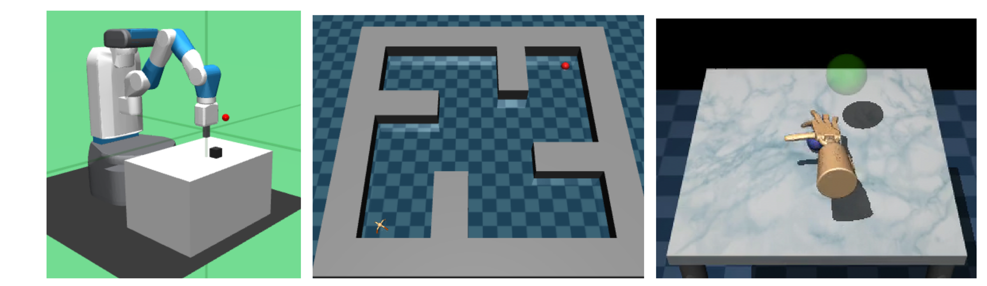
Northeastern University

Contributions

- We propose **new imitation learning method** that pairs a backwards dynamics model with a policy.
- We demonstrate that a **backwards model can improve robustness** over behavior cloning.
- On a variety of long-horizon, sparse-reward domains, BMIL noticeably **extends the region of attraction** around demonstration data.

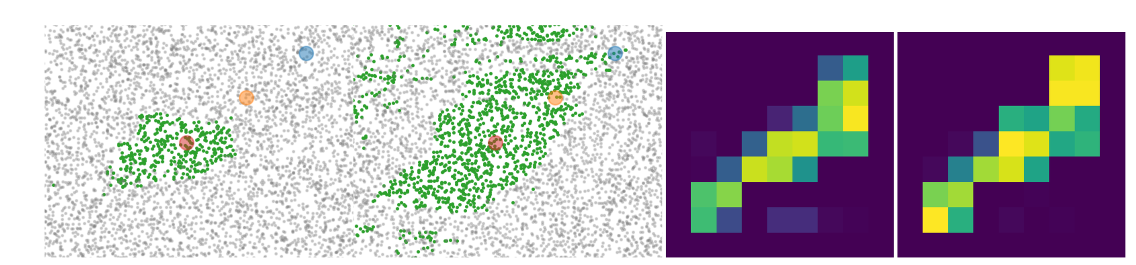
Experiments

Continuous control: 1) Fetch, 2) Maze, 3) Adroit.



The training data consists of trajectories from a single start-goal pair and/or their ε-neighborhoods. We evaluate by varying the initial states (e.g. joint positions/velocities, agent coordinates, etc.)

			Robustness (%)			Relative to BC		
			BC	VINS	BMIL	BC	VINS	BMIL
Fetch	Push (5 demos)		12.1±0.3	12.8±0.4	**14.6**±0.6	1	1.06	**1.21**
	PickAndPlace (10 demos)		4.1±0.1	3.4±0.1	**17.5**±0.9	1	0.84	**4.31**
Maze	Point (20 demos)	UMaze	**49.0**±1.9	39.5±2.1	**47.8**±3.5	**1**	0.81	**0.98**
		Room5x11	36.8±3.4	17.3±2.8	**38.6**±3.4	1	0.47	**1.05**
		Corridor7x7	33.7±1.5	**37.7**±1.2	**38.9**±2.3	1	**1.12**	**1.16**
	Ant (20 demos)	UMaze	**63.0**±1.0	44.7±2.1	**64.8**±1.5	**1**	0.71	**1.03**
		Room5x11	**33.2**±0.9	30.2±0.8	29.1±0.8	**1**	0.91	0.87
		Corridor7x7	**21.7**±0.6	19.6±0.6	17.6±0.5	**1**	0.90	0.81
Adroit	Relocate (20 demos)		7.9±0.7	3.8±0.7	**13.3**±1.0	1	0.48	**1.68**



- BMIL learns a larger region of attraction than BC and substantially increases robustness.
- BMIL still achieves close to 100$ success rates on original task.

Additional Results

Forward vs Backwards Dynamics: Using a forwards dynamics model does not increase robustness.

	Robustness (%)			Relative to BC		
	BC	BMIL (Forwards)	BMIL (Backwards)	BC	BMIL (Forwards)	BMIL (Backwards)
Push	12.1±0.3	12.4±0.6	14.6±0.6	1	1.03	1.21
PickAndPlace	4.1±0.1	4.1±0.2	17.5±0.9	1	1.03	4.31

Computation Budget: BMIL trains both the model and policy and uses more total gradient steps than BC ($\sim 6x$ on Fetch). Increasing the number of policy gradient steps for BC does not improve robustness.