DualSMC: Tunneling Differentiable Filtering and Planning under Continuous POMDPs

IJCAI 2020

Yunbo Wang*, *Shanghai Jiao Tong University*

Bo Liu*, *The University of Texas at Austin*

Jiajun Wu, *Stanford University*

Yuke Zhu, *The University of Texas at Austin*

Simon S. Du, *University of Washington*

Li Fei-Fei, *Stanford University*

Joshua B. Tenenbaum, *Massachusetts Institute of Technology*

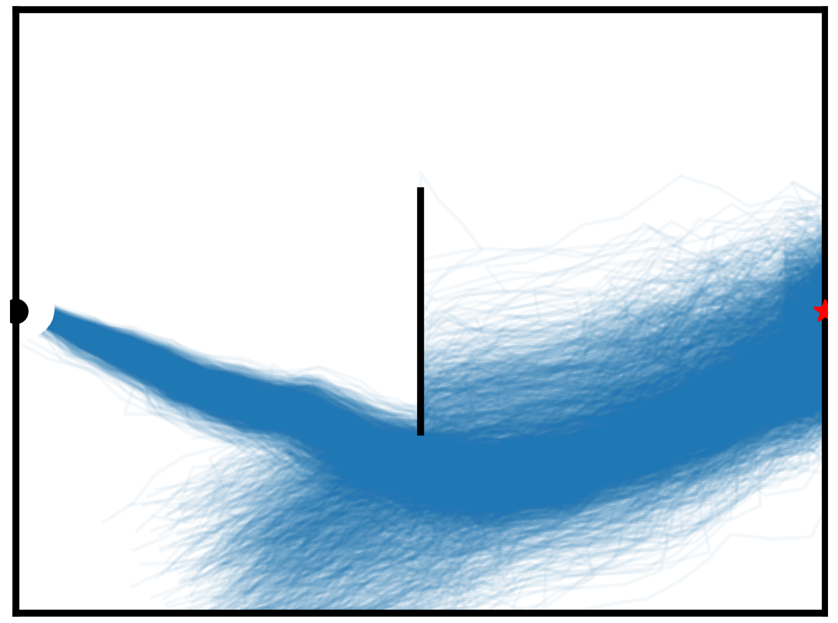
*both authors contributed equally

Code: https://github.com/Cranial-XIX/DualSMC

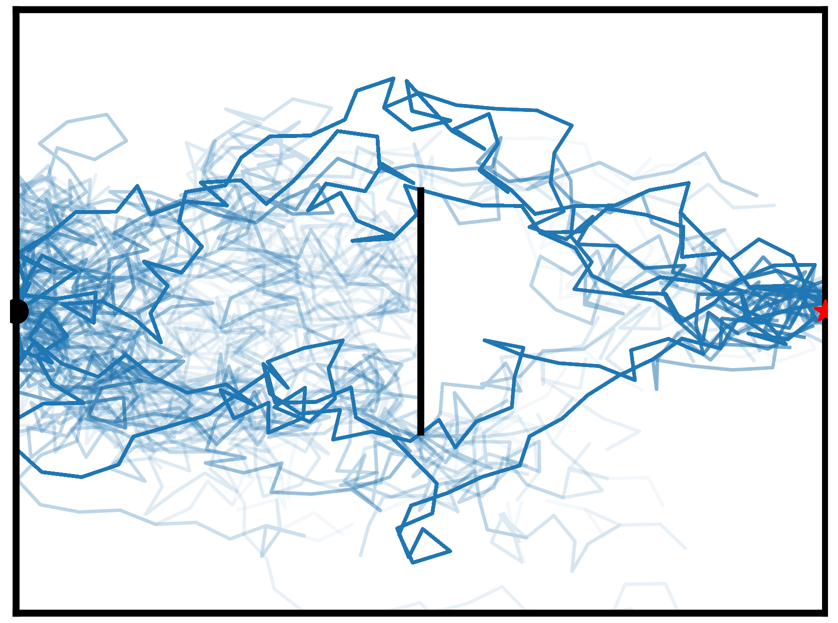
Planning algorithms

hard to deal with unknown dynamics and partial observations

- Monte Carlo tree search, MCTS
  - *Standard MCTS applies only to discrete spaces and also requires a black-box transition oracle*
- Cross entropy methods, CEM & Iterative linear quadratic regulator, iLQR
  - *Assume the distribution over future trajectories to be Gaussian, i.e. unimodal*
- **Sequential Monte Carlo planning, SMCP**
  - *Follows the framework of "control as probabilistic inference": selecting the optimal action is equivalent to finding the maximum posterior over actions conditioned on an optimal future*



CEM: here the agent samples all the actions at once from a Gaussian with learned mean and covariance. It found one solution, but forgot the other one.

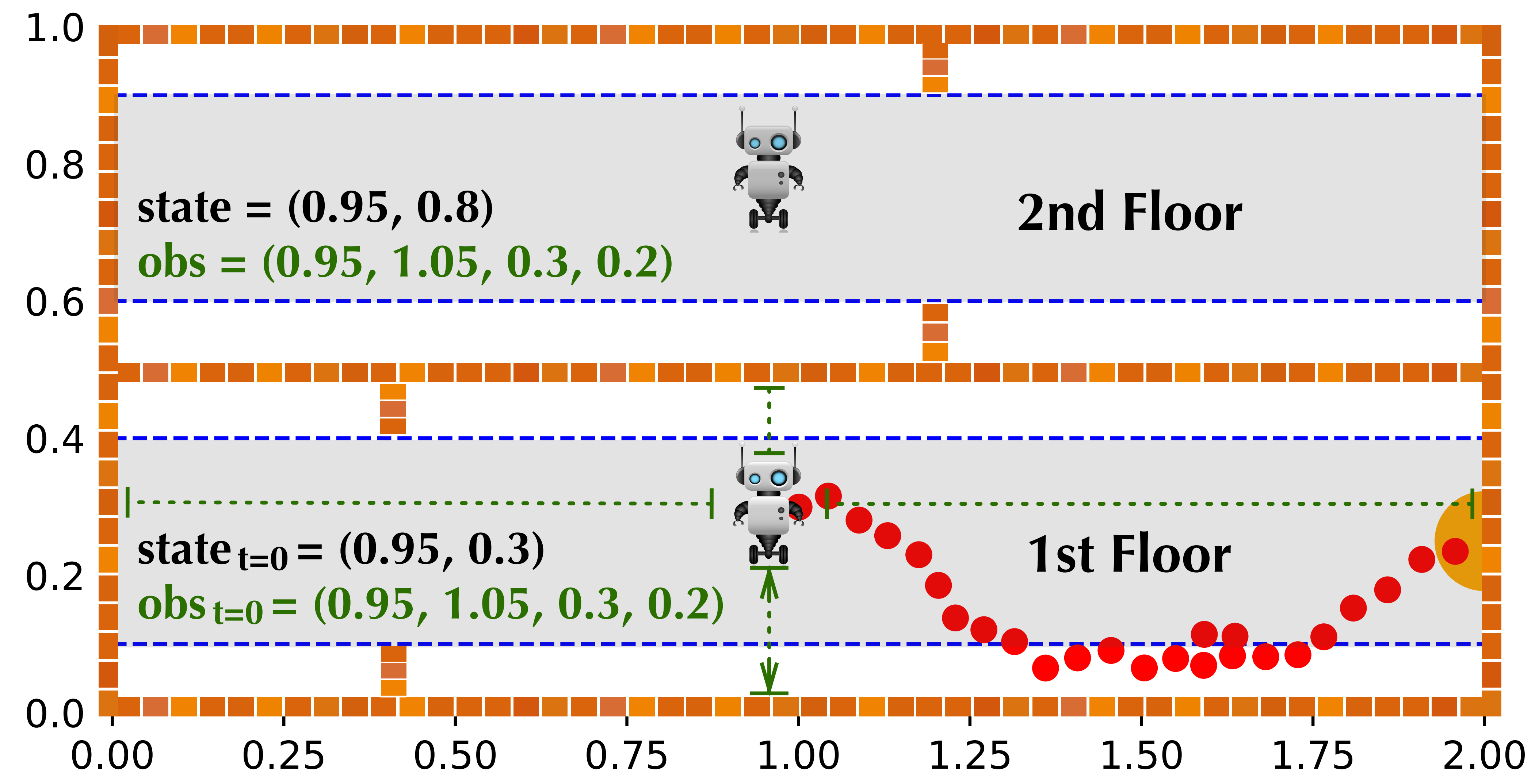


SMCP: the agent is able to focus on the promising trajectories and does not collapse on a single mode. **But, how to deal with partial observations?**

From [Piche et al. 2019]

A toy POMDPs problem

that previous methods cannot solve



- **Planning under uncertainty with continuous actions**

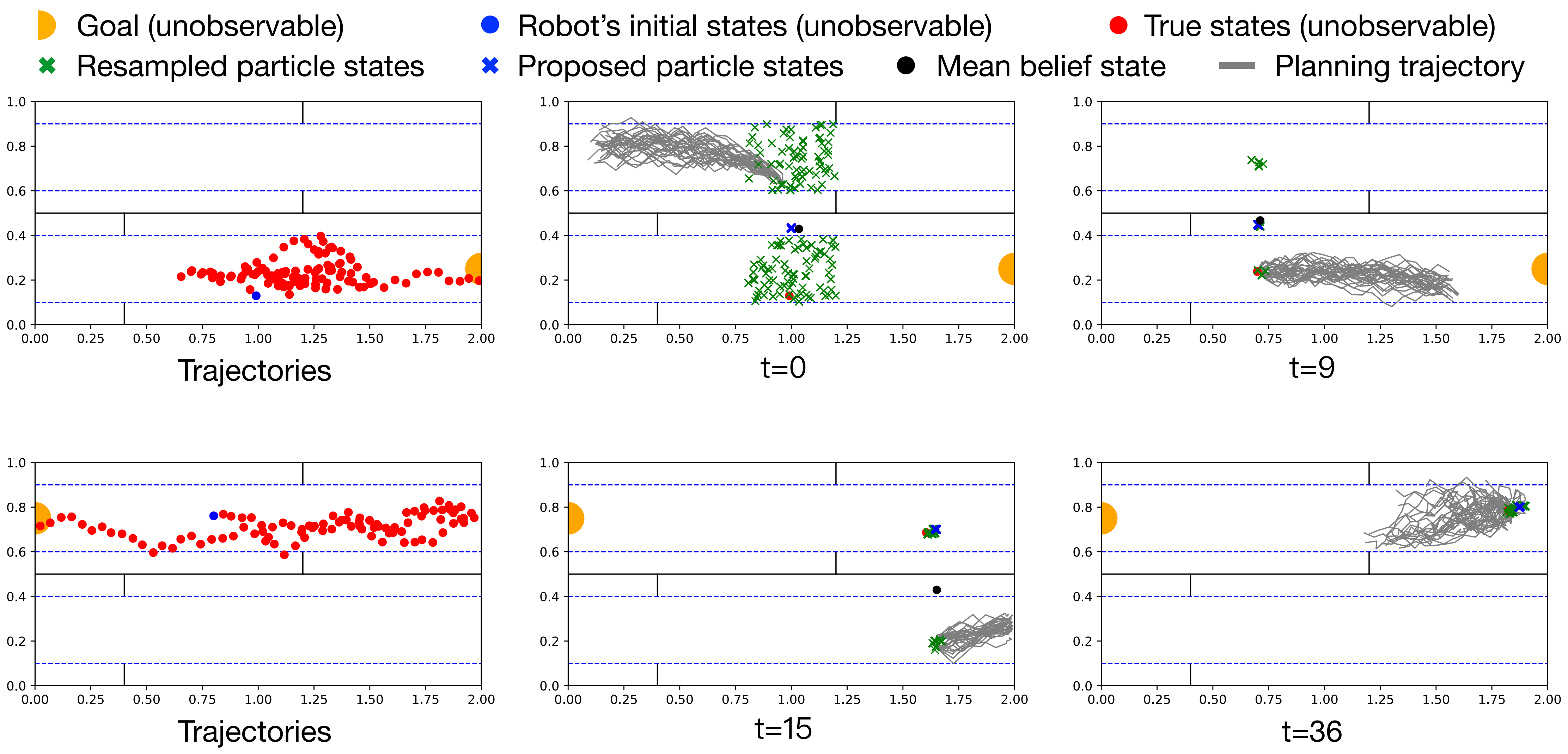
- **A straightforward solution:**

✦ State estimation: Particle filter net [Karkus et al. 2018]

✦ Planning: SMCP [Piche et al. 2019]

- **What's wrong with the filtering part?**
  The regressed true state can be meaningless (at the center of the two floors).

- **What's wrong with the planning part?**
  The planner does not learn to help the filter (Based on the regressed state, it might go either left or right).



DualSMC Network

Our approach

- **Two sequential Monte Carlo processes - Interlinked via belief states**

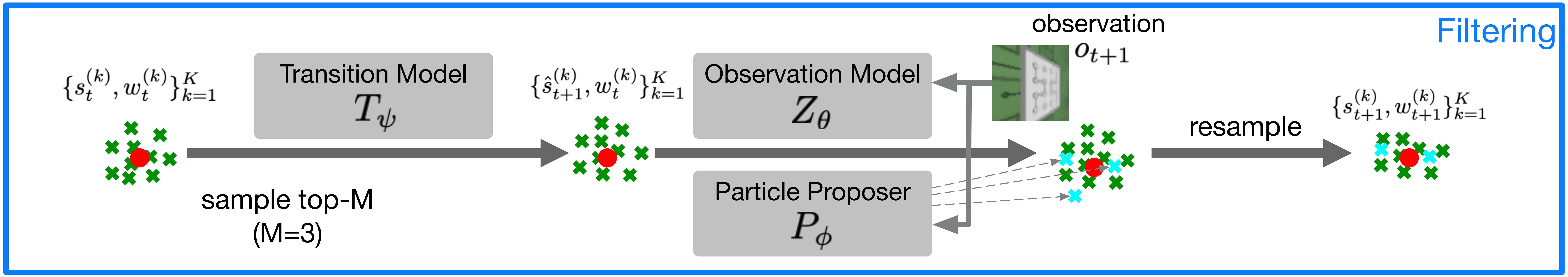
✦ Adversarial particle filtering - better capture the multi-modality of the belief

✦ DualSMC planning - control based on perceived uncertainty (so can learn to reduce it)

- **Modules**

- Proposer (***P***) to generate plausible states
- Transition model (***T***) to simulate dynamics
- Observation model (***Z***) to update Bayesian beliefs
- Policy network
- Critic network

Adversarial particle filtering

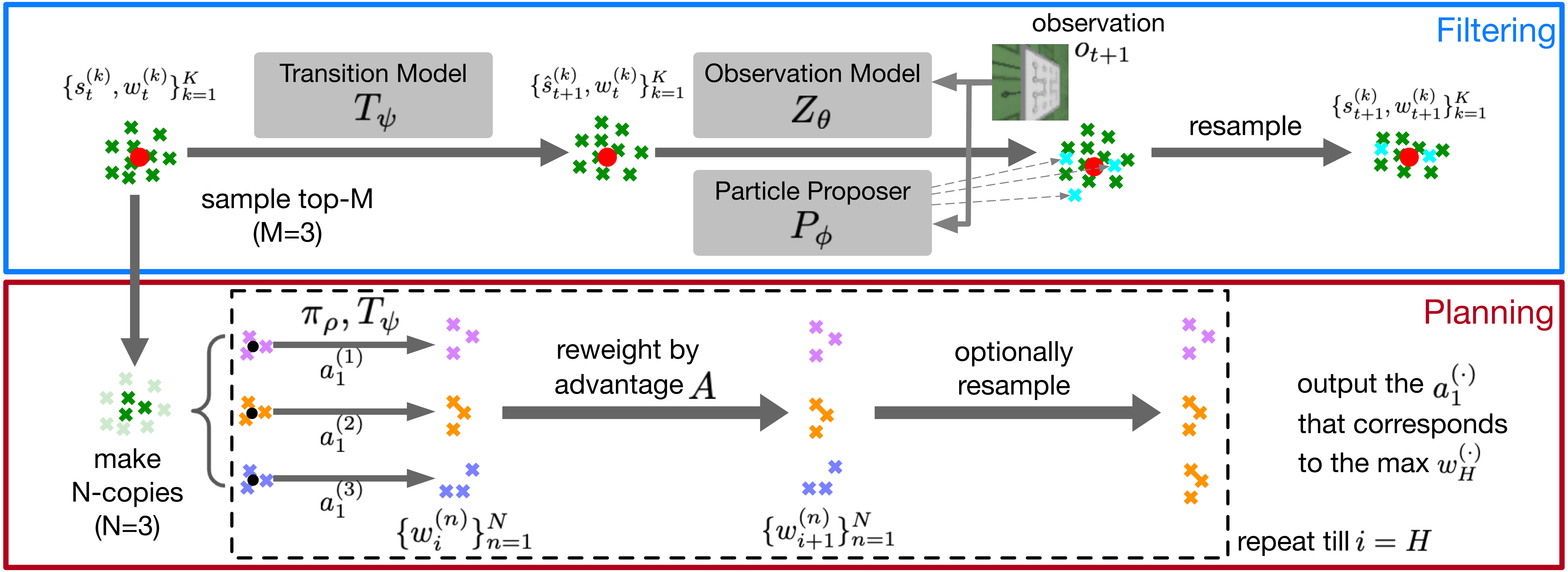


$$\mathrm{bel}(s_{t+1}) = \eta \int \mathrm{bel}(s_t)\mathcal{Z}(o_{t+1}|s_{t+1})\mathcal{T}(s_{t+1}|s_t, a_t)ds_t$$

- **Proposer and observation model are opposite yet dependent on each other**

$$\min_\phi \max_\theta F(Z_\theta, P_\phi) = \mathbb{E}_{o_{1:t}}\left[\underbrace{\mathbb{E}_s \log Z_\theta(o_t, s)}_{\text{true states}} + \underbrace{\mathbb{E}_{s'\sim s_{\text{old}}} \log(1 - Z_\theta(o_t, s'))}_{\text{transitioned states}} + \underbrace{\mathbb{E}_{\epsilon_P} \log(1 - Z_\theta(o_t, P_\phi(o_t, \epsilon_P)))}_{\text{proposed new states}}\right]$$

Planning explicitly on belief states

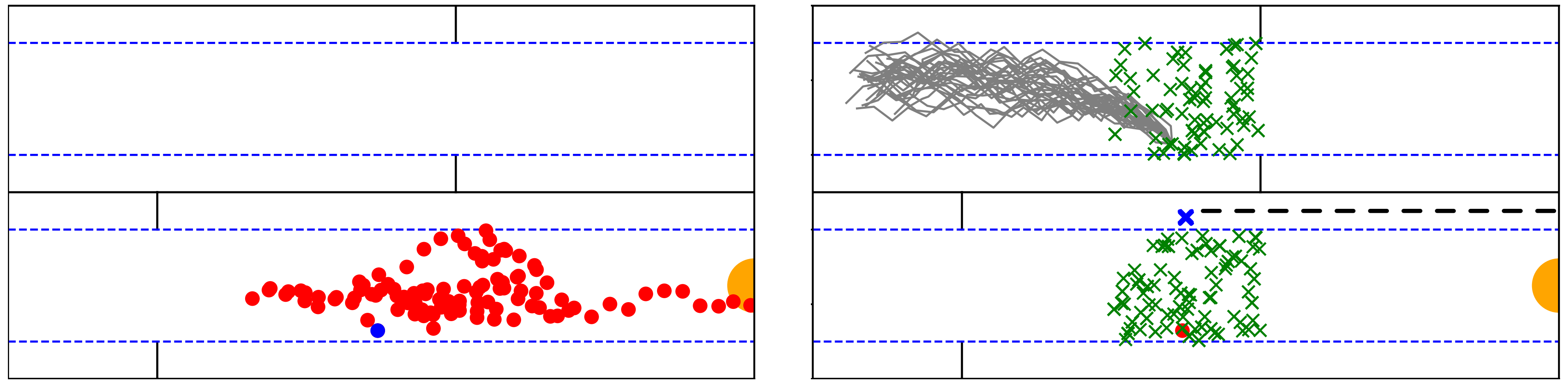


- **Plan conditioned on the top candidates of the belief particles**

Floor positioning

A toy POMDPs problem that previous methods cannot solve

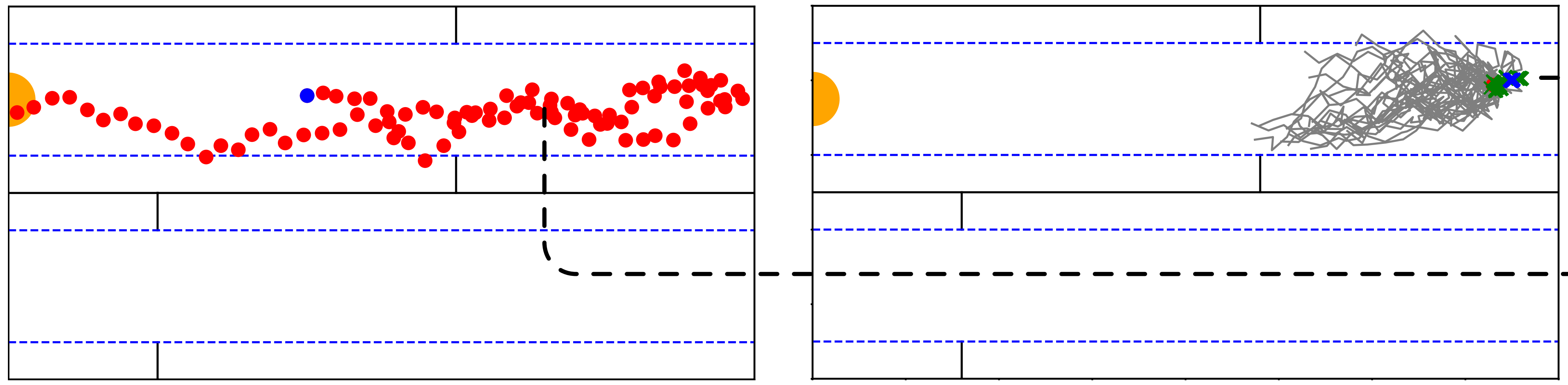
Moving trajectory	Particle distribution & planning trajectory	Explaination	Legend



Regressive PF + SMCP　　t=0

① Proposed particles by **regressive PF** were close to the *mean of the particles*

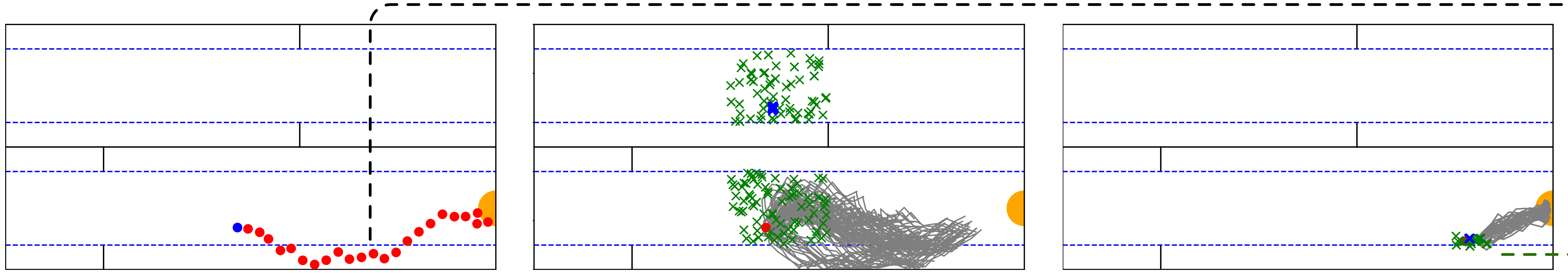
② The **SMCP** robot made *very chaotic* decisions upon the filtered belief state



Adversarial PF + SMCP　　t=32

① Proposed particles by **adversarial PF** were closer to the plausible true state

② The **SMCP** robot went rightwards, being *unaware of its position* until reaching the wall



DualSMC　　t=0　　t=16

The **DualSMC** robot *made a detour* to reduce uncertainty: by stepping across the dashed blue line, it localized itself quickly with *converged belief states*

Legend

Goal　***Unobserved***

Initial states

True states

Proposed particles

Resampled particles

Planning trajectory

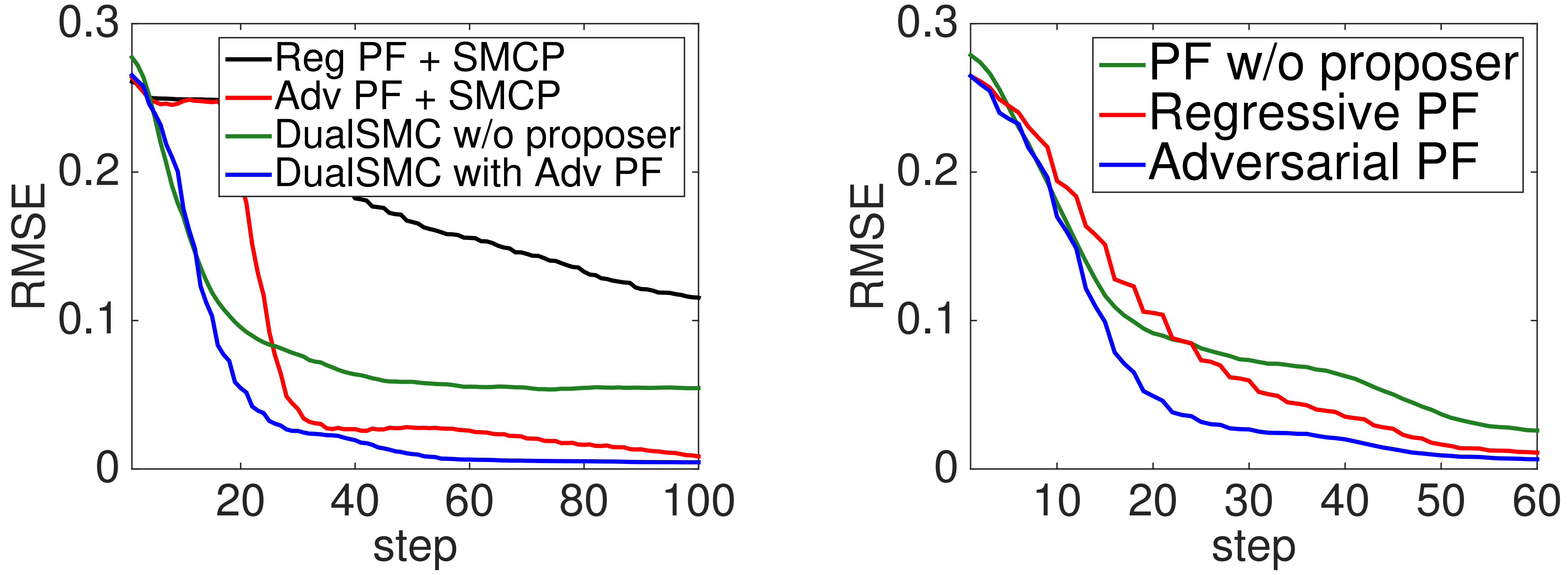
Floor positioning

A toy POMDPs problem that previous methods cannot solve

Method	Success	# Steps
DVRL [Igl *et al.*, 2018]	38.3%	162.0
LSTM filter + SMCP [Piche *et al.*, 2018]	23.5%	149.1
Regressive PF (ℓ_2, top-1) + SMCP	25.0%	107.9
Regressive PF (density, top-3) + PI-SMCP	25.0%	107.9
Adversarial PF (top-1) + SMCP	95.0%	73.3
Adversarial PF (top-3) + PI-SMCP	82.7%	86.9
DualSMC with regressive PF (ℓ_2)	45.1%	114.9
DualSMC with regressive PF (density)	58.3%	107.0
DualSMC w/o proposer	78.6%	62.1
DualSMC with adversarial PF	**99.4%**	**26.9**

Table 2: The success rate and the average number of steps of 1,000 tests in the floor positioning domain (PF is short for particle filter)

Does *adversarial training* improve the previous particle filter nets?



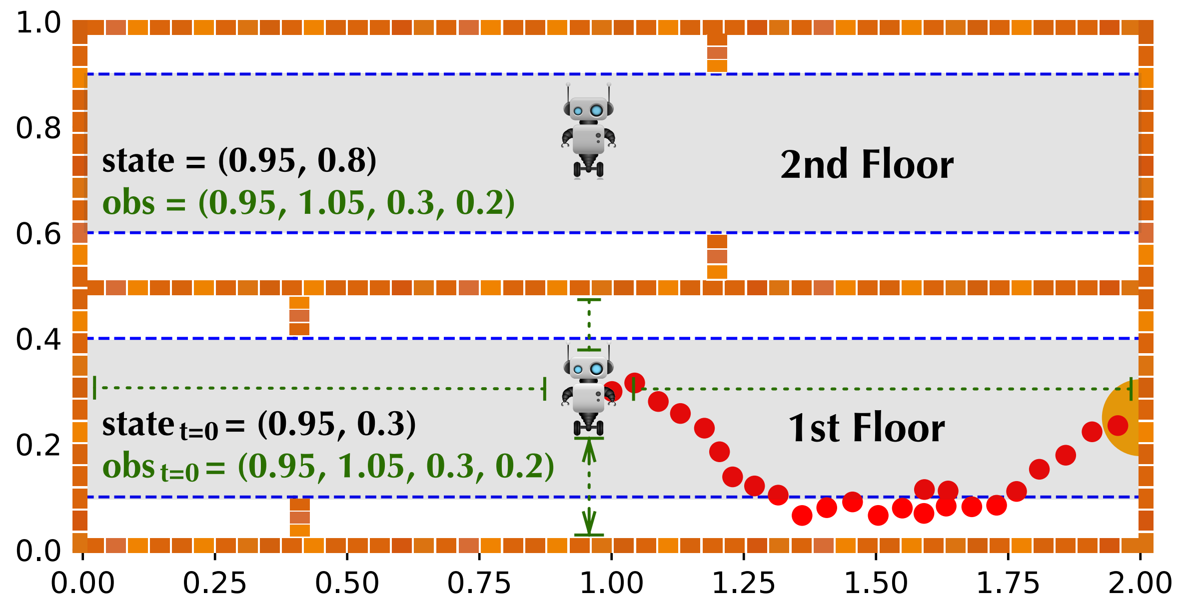
(a) Different POMDP planners (b) Different particle filters

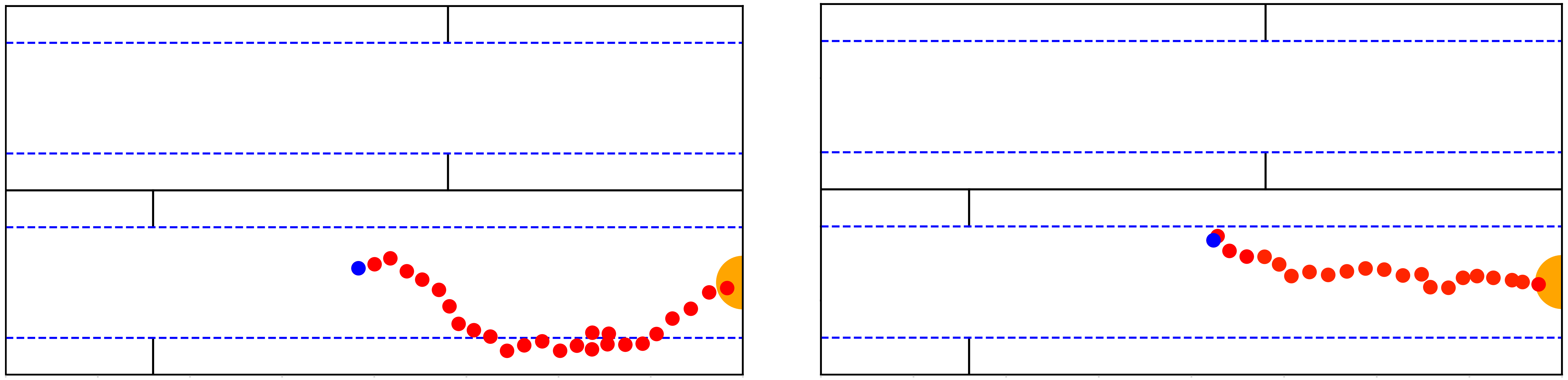
Figure 4: The state filtering error with respect to the number of steps which the robot has taken in the floor positioning domain

Floor positioning

A toy POMDPs problem that previous methods cannot solve

How does DualSMC adapt to different uncertainties?



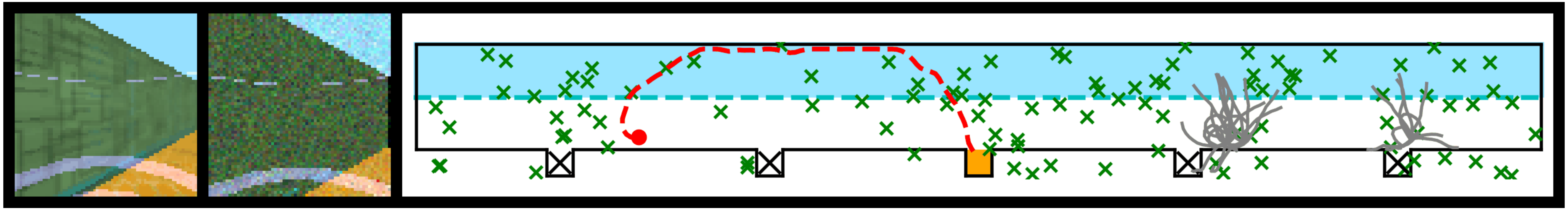


(a) Partial observation

(b) Full observation

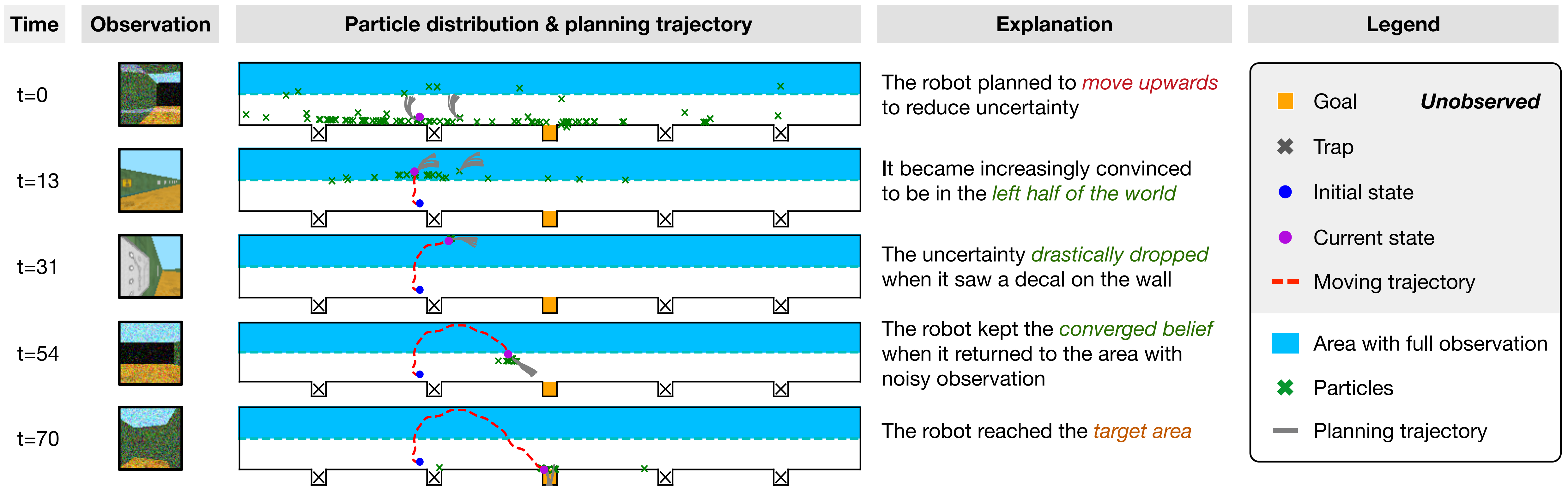
3D light-dark navigation

A visually rich domain simulated by DeepMind Lab



Method	Success	# Steps
PlaNet [Hafner *et al.*, 2019]	30%	34.24
DVRL [Igl *et al.*, 2018]	42%	98.48
LSTM + SMCP [Piche *et al.*, 2018]	59%	85.40
Adversarial PF (top-1) + SMCP	58%	56.11
Adversarial PF (top-3) + PI-SMCP	64%	64.37
DualSMC with regressive PF (ℓ_2)	92%	66.88
DualSMC with regressive PF (density)	98%	70.95
DualSMC with adversarial PF	**98%**	**67.49**

Table 3: The average result of 100 tests for 3D light-dark navigation

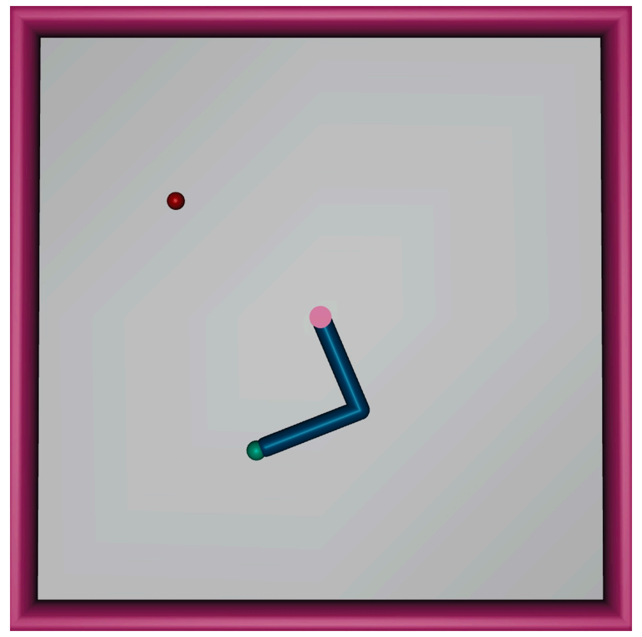


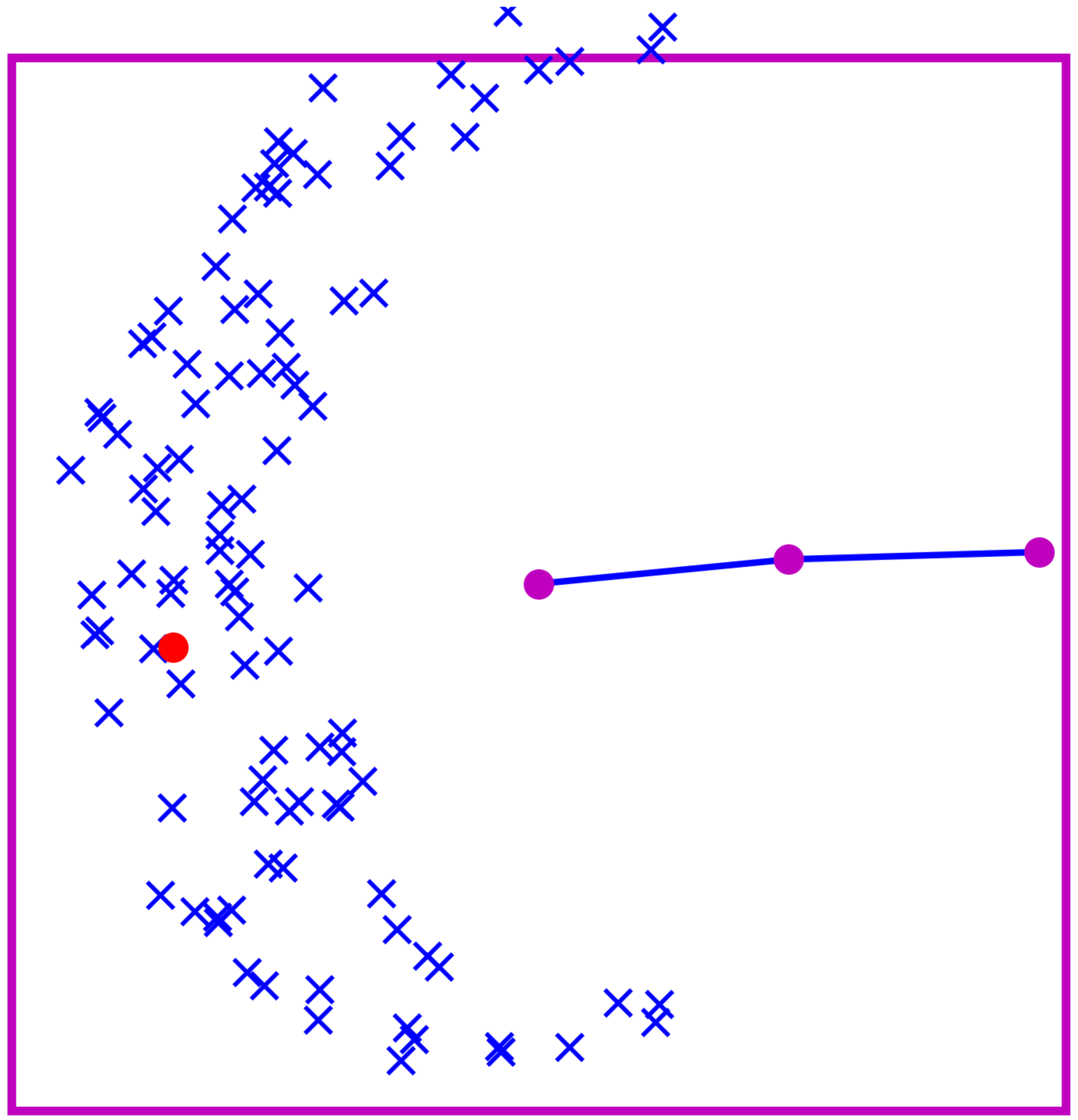
Modified Reacher

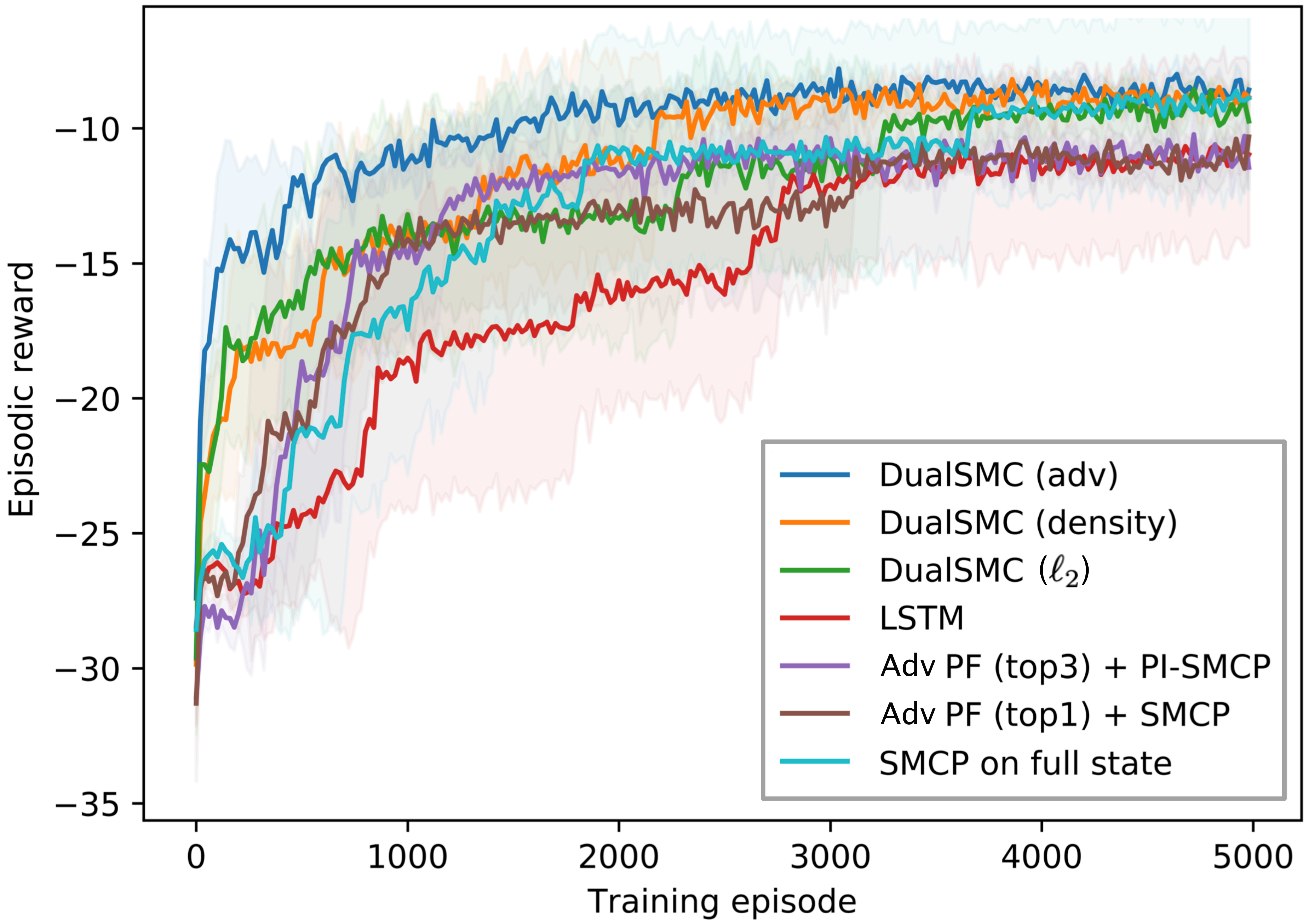
A continuous control task with partial observations

$$o = (\cos(\theta_1), \cos(\theta_2), \sin(\theta_1), \sin(\theta_2), \omega_1, \omega_2, r)$$

$$r = ||(r_x, r_y, r_z)||_2 + \epsilon_r$$







Conclusions & limitations

DualSMC is a solution to continuous POMDPs

✓ First, it learns plausible belief states for high-dimensional POMDPs with an adversarial particle filter.

✓ Second, it plans future actions by considering the distributions of the learned belief states.

✓ The filter and the planner are inter-dependent and jointly trained.

◉ However, an imperfect model of the environment **dynamics** will make accumulated errors for prediction over **long sequences**, which is an open problem for all model-based planning methods. But since DualSMC shares the transition model between filtering and planning, it may have a more severe impact.

Thanks

Code: https://github.com/Cranial-XIX/DualSMC