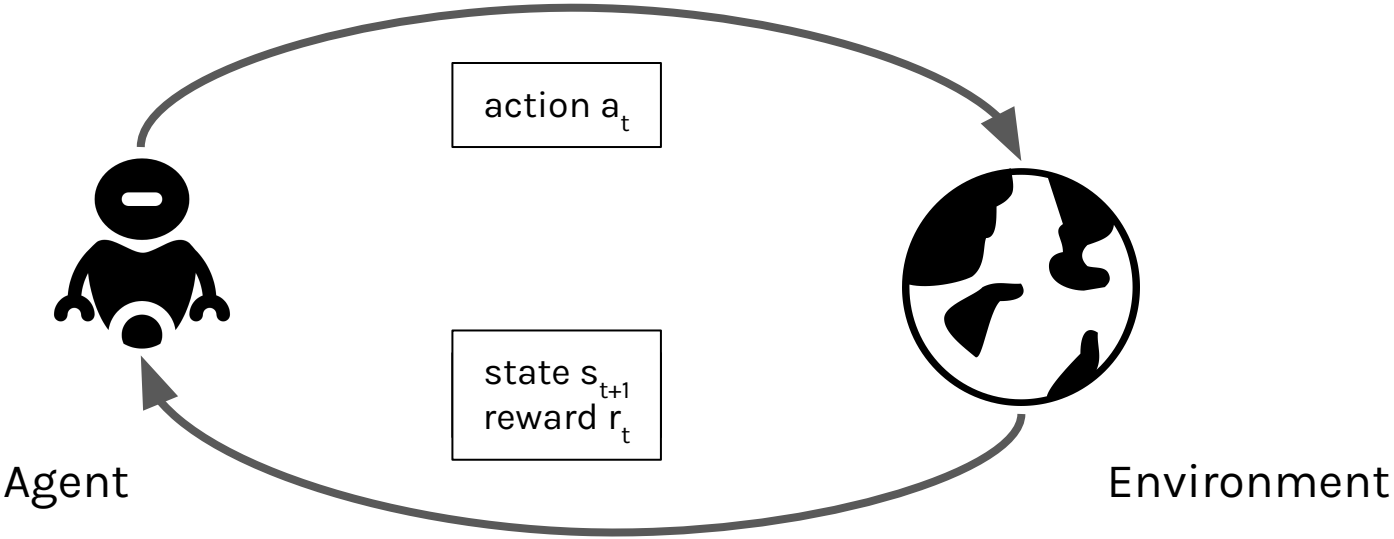
Ishan Durugkar

Estimation and Control of Visitation Distributions for Reinforcement Learning

Committee:

Peter Stone, Qiang Liu, Philipp Krähenbühl
Scott Niekum, Marc Bellemare

Reinforcement Learning (RL)[1]

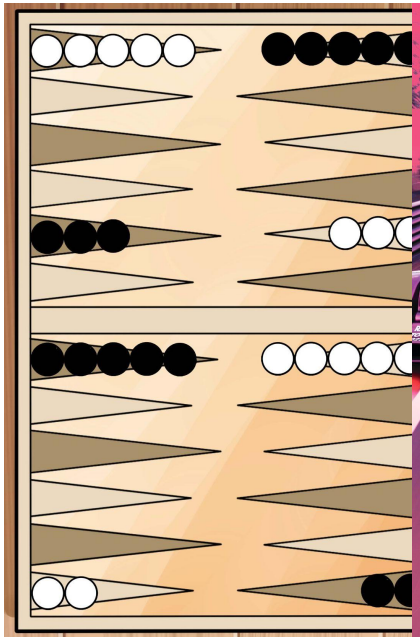


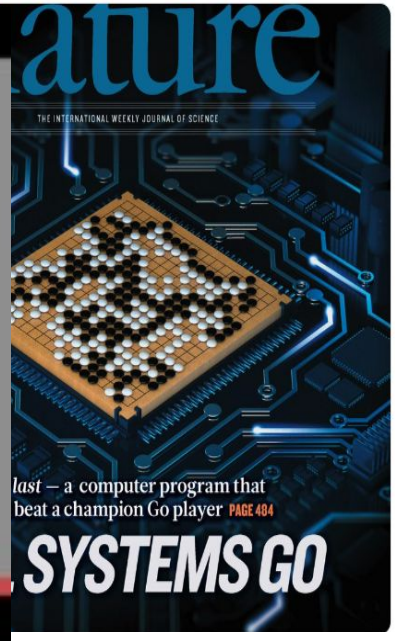
Return: $R(s) = \sum_{t=0}^{\infty} \left[\gamma^t r(s_t, a_t, s_{t+1}) | s_0 = s\right]$

Policy: $\pi : \mathcal{S} \longmapsto \Delta(\mathcal{A})$

[1] Reinforcement Learning: An Introduction, Sutton and Barto, 2018

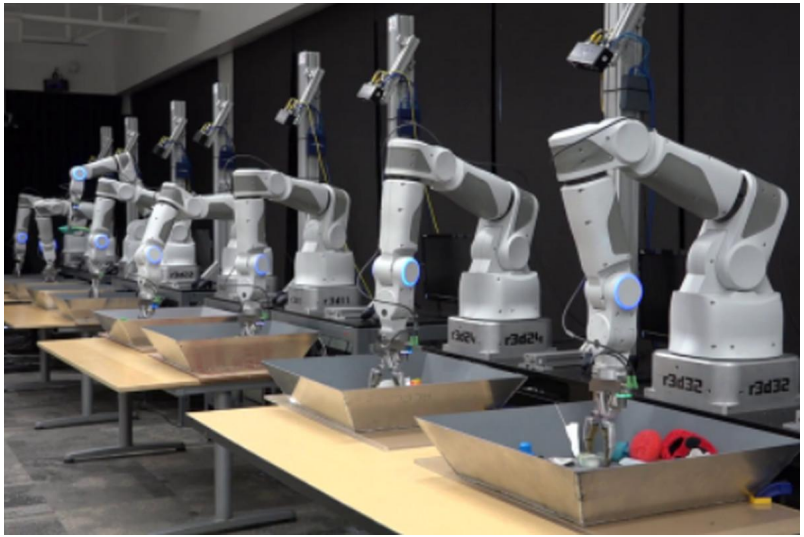
Successes





[1] TD-Gammon, a self-teaching backgammon program, achieves master-level play; Tesauro, G.; Neural Computation 1994
[2] Mastering the game of go without human knowledge; Silver et al.; Nature 2017
[3] Human-level control through deep reinforcement learning; Mnih et al.; Nature 2015
[4] Outracing champion Gran Turismo drivers with deep reinforcement learning; Wurman et al.; Nature 2022

Challenges

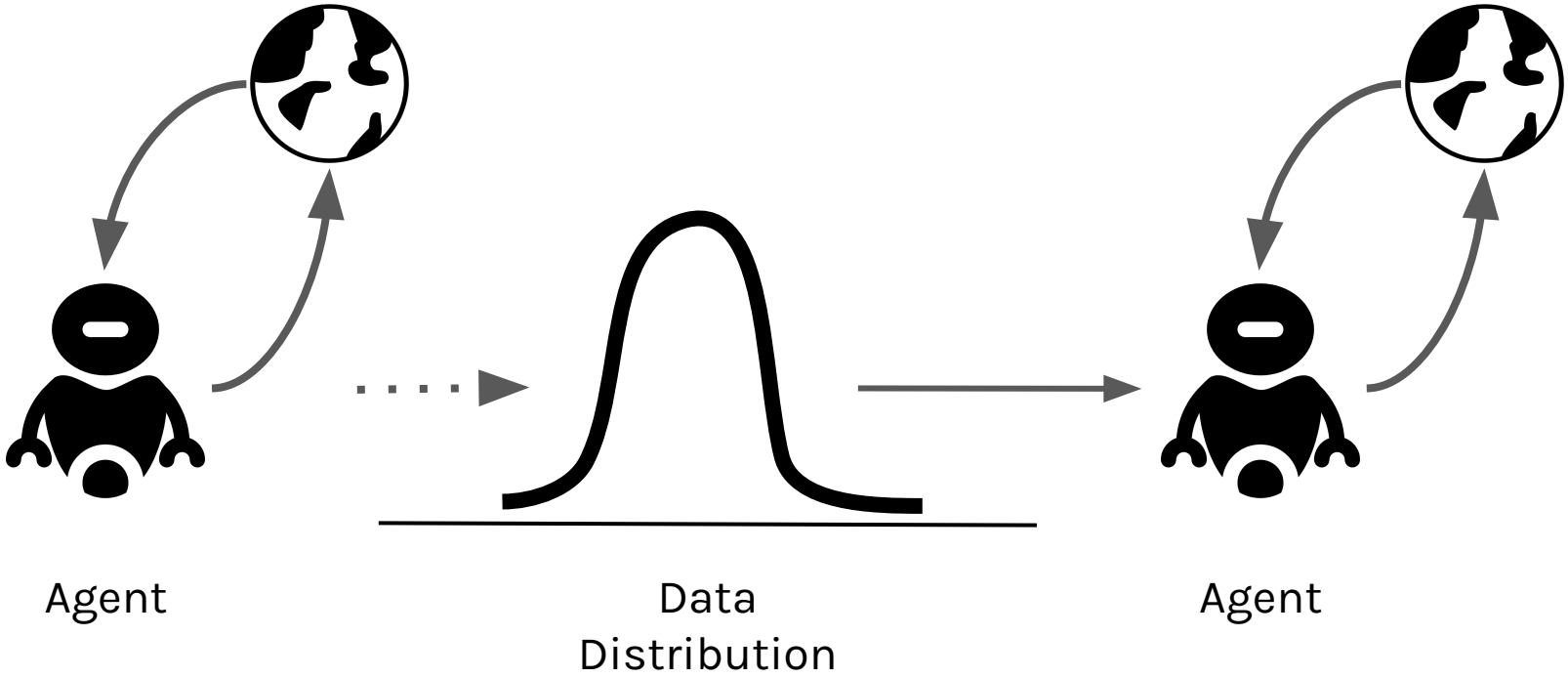


Google's Robot Farm[1]

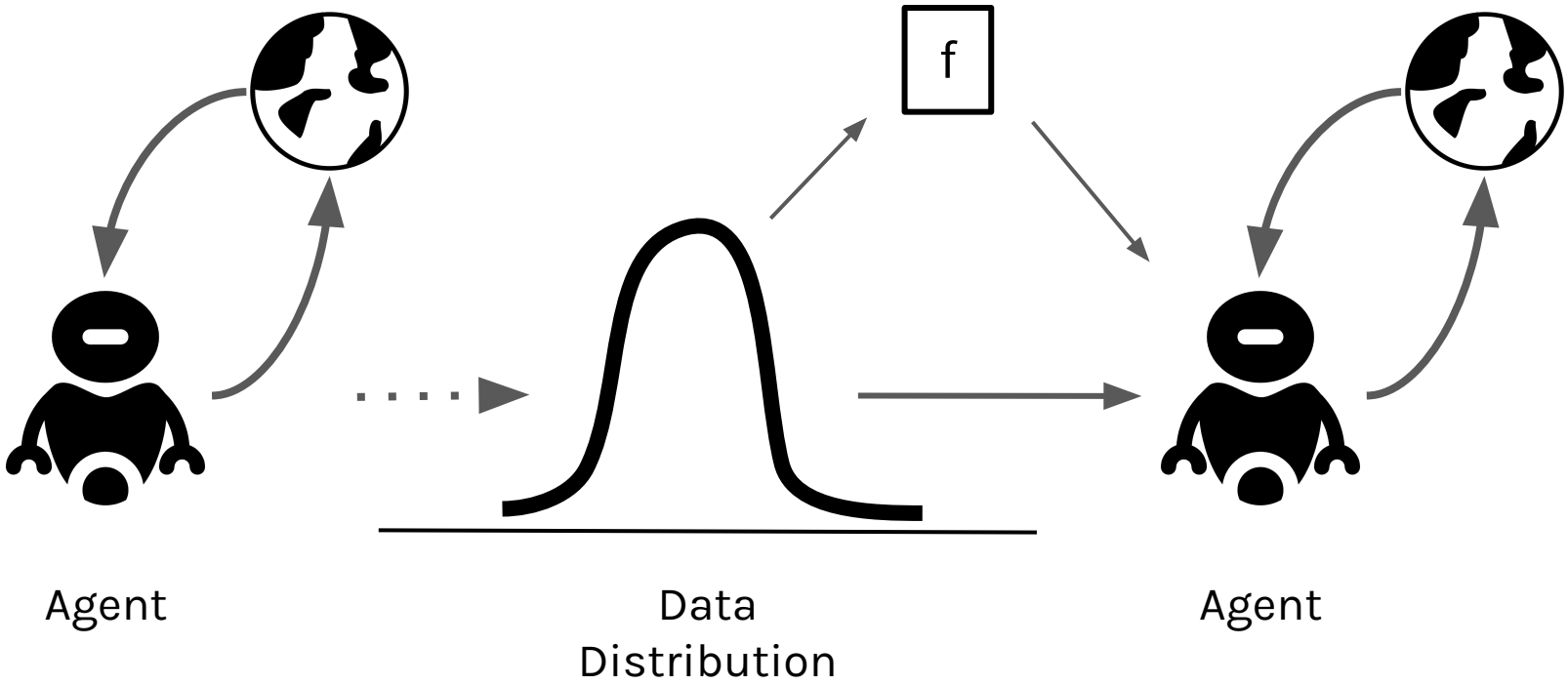
Autonomous Driving[2]

[1] Learning Hand-Eye Coordination for Robotic Grasping with Deep Learning and Large-Scale Data Collection; Levine et al.; ArXiv 2016
[2] Reward (Mis)design for Autonomous Driving; Knox et al.; ArXiv 2021

Why is RL Difficult?



Estimate and Control the Data Distribution



The Thesis Question

How can methods for estimating and controlling an agent's visitation distribution be beneficially incorporated into RL algorithms?

Distribution Matching for RL

How can methods for estimating and controlling an agent's visitation distribution be beneficially incorporated into RL algorithms?

Definitions

Policy

$$\pi : \mathcal{S} \longmapsto \Delta(\mathcal{A})$$

State visitation distribution

$$\rho_\pi(s) := \mathbb{E}_{s_0 \sim \rho_0} \left[(1-\gamma) \sum_{t=0}^{\infty} \gamma^t P(s_t = s | \pi) \right]$$

Transition visitation distribution

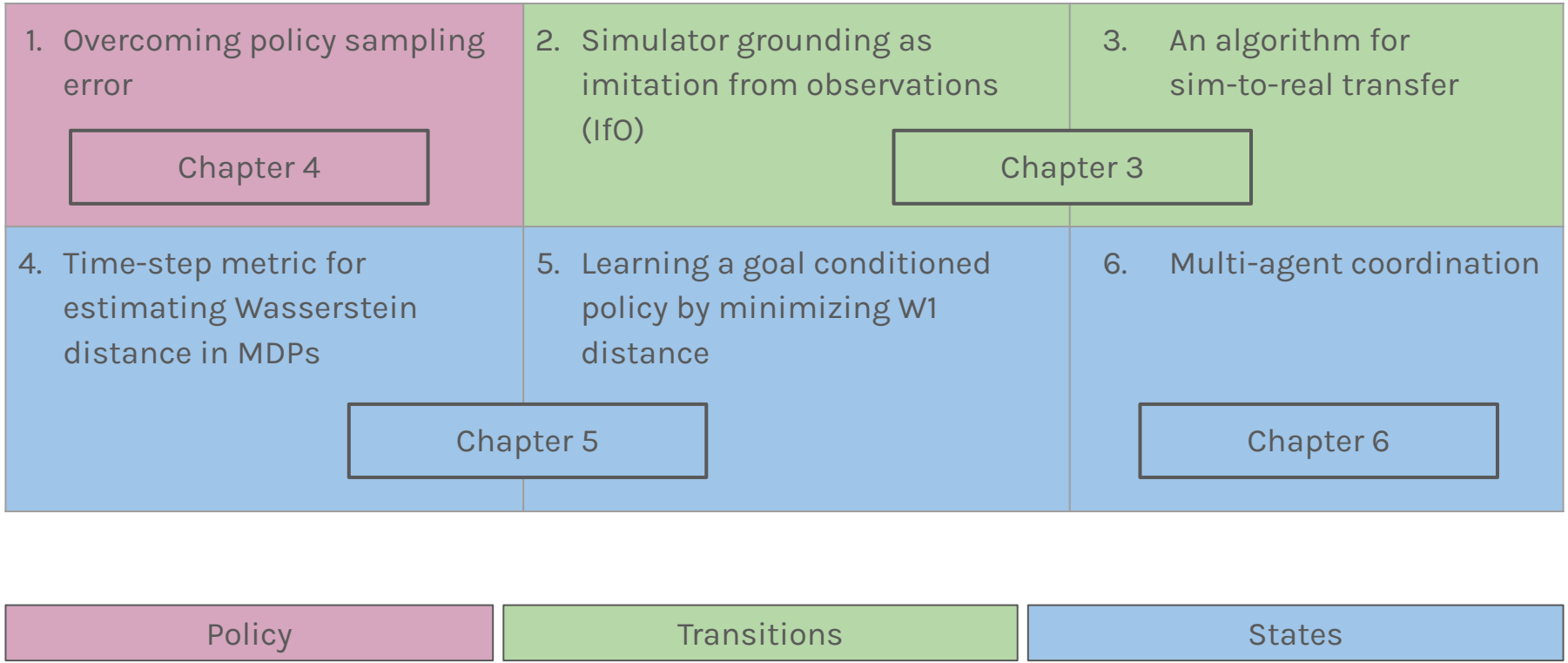
$$\rho_\pi(s, a, s') := \mathbb{E}_{s_0 \sim \rho_0} \left[(1-\gamma) \pi(a|s) P(s'|s, a) \sum_{t=0}^{\infty} \gamma^t P(s_t = s | \pi) \right]$$

Overview

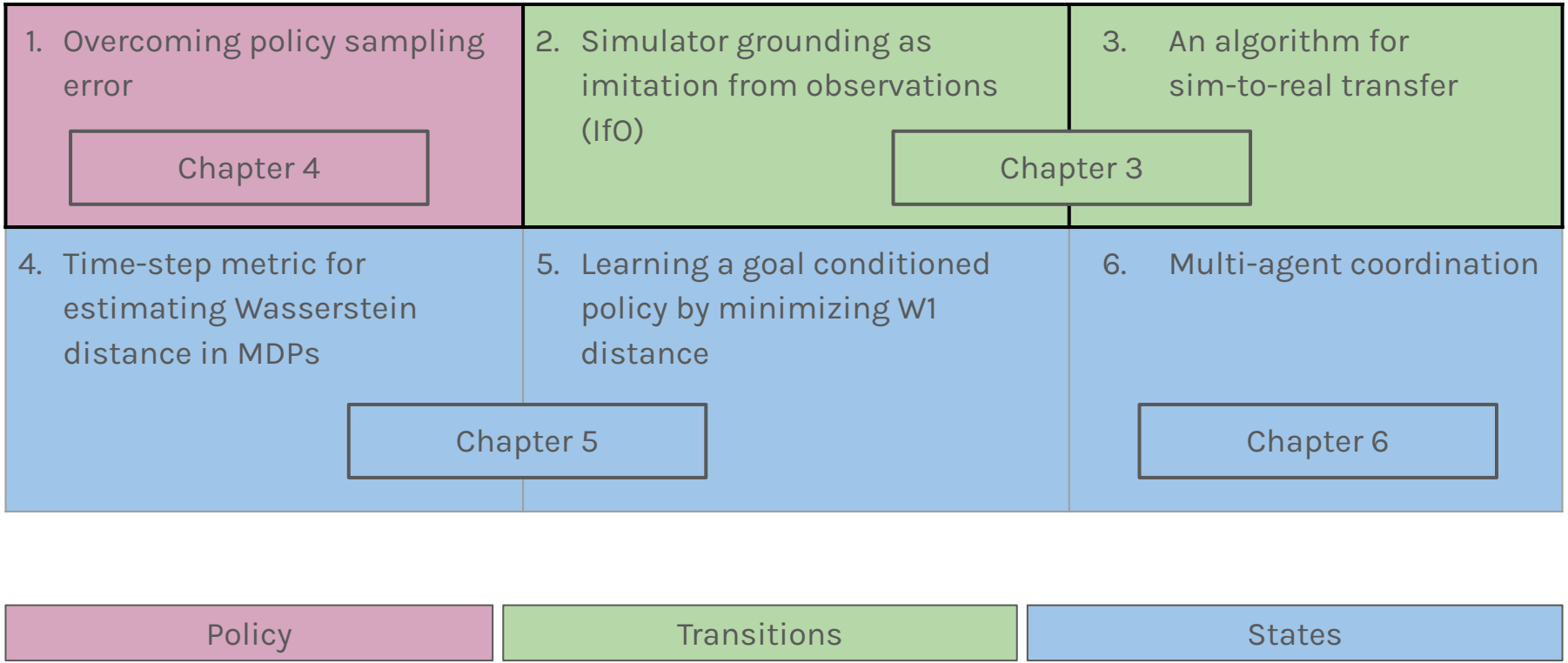
1. Overcoming policy sampling error	2. Simulator grounding as imitation from observations (IfO)	3. An algorithm for sim-to-real transfer
4. Time-step metric for estimating Wasserstein distance in MDPs	5. Learning a goal conditioned policy by minimizing W1 distance	6. Multi-agent coordination

Policy	Transitions	States

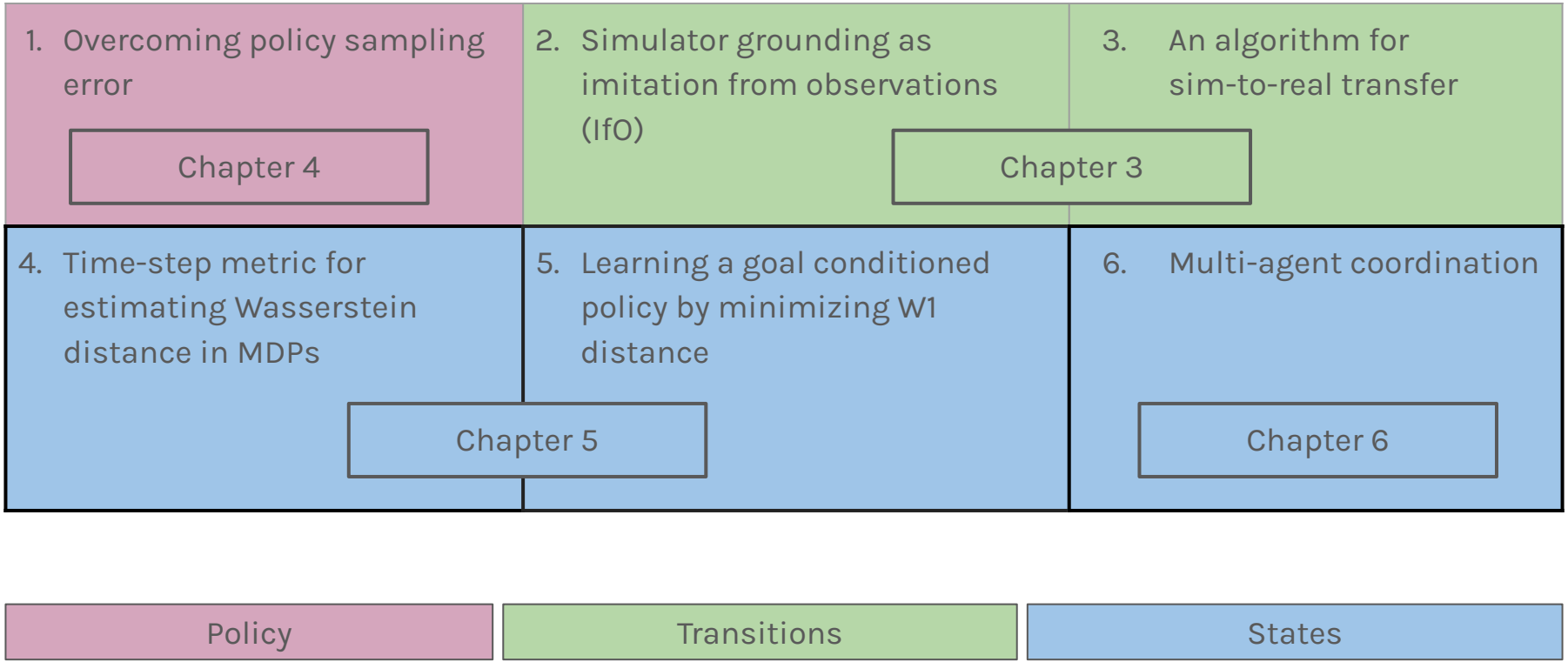
Overview


1. Overcoming policy sampling error
Chapter 4
2. Simulator grounding as imitation from observations (IfO)
3. An algorithm for sim-to-real transfer
Chapter 3
4. Time-step metric for estimating Wasserstein distance in MDPs
5. Learning a goal conditioned policy by minimizing W1 distance
Chapter 5
6. Multi-agent coordination
Chapter 6
Policy
Transitions
States

Overview - Completed Before Proposal



Overview - After Proposal



Distribution Estimation and Control for RL

Actions

- Reducing Sampling Error in Batch Temporal Difference Learning; Pavse, B., **Durugkar, I.**, Hanna, J., Stone, P.; ICML 2021

Transitions

- An Imitation from Observation Approach to Transfer Learning with Dynamics Mismatch;
  *Desai, S., ***Durugkar, I.**, *Karnan, H., Warnell, G., Hanna, J. and Stone, P; NeurIPS 2020

States

- Adversarial Intrinsic Motivation for Reinforcement Learning; **Durugkar, I.**, Tec, M., Niekum S., Stone, P.; NeurIPS 2021
- DM^2: Distributed Multi-Agent Reinforcement Learning by Distribution Matching;
  *Wang, C., ***Durugkar, I.**, *Liebman, E., Stone, P.; AAAI 2023

* - joint first authors

Overview

1. **Overcoming policy sampling error**	2. Simulator grounding as imitation from observations (IfO)	3. An algorithm for sim-to-real transfer
4. Time-step metric for estimating Wasserstein distance in MDPs	5. Learning a goal conditioned policy by minimizing W1 distance	6. Multi-agent coordination

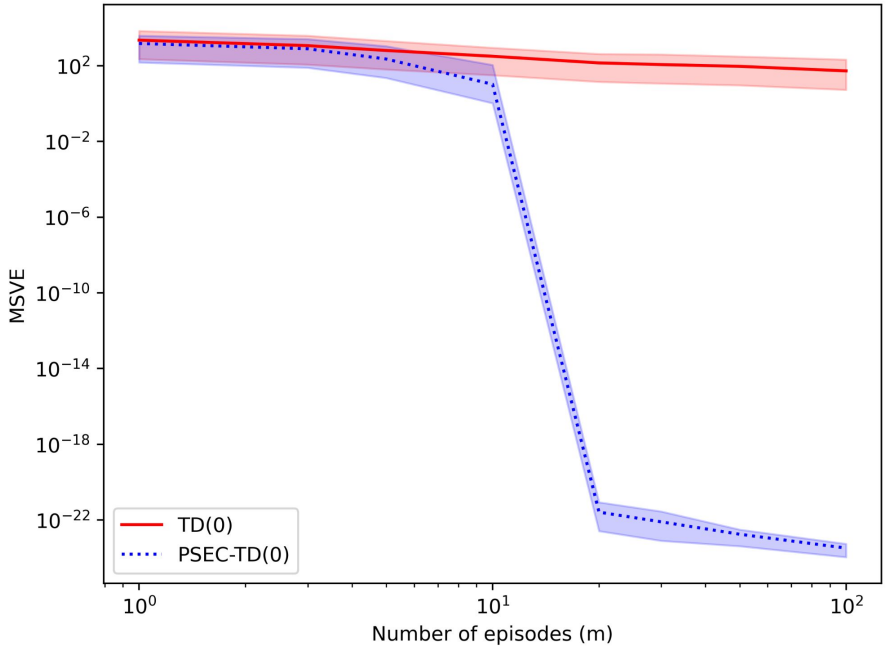
Policy	Transitions	States

Policy Sampling Error Corrected (PSEC) - TD Learning

- Batch temporal difference (TD) learning will have some sampling error
- Contribution[1]: Estimating the maximum likelihood policy implied by the dataset allows me to eliminate sampling error
- Analysis shows that PSEC-TD(0) converges to a fixed point with no policy sampling error

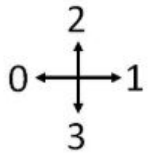
[1] Reducing sampling error in batch temporal difference learning; Pavse, B., **Durugkar, I.**, Hanna, J. and Stone, P.; ICML 2020

Results - Grid World



- 4 x 4 grid world, deterministic transitions
- Tabular representation
- equiprobable policy being evaluated
- PSEC-TD uses correction on TD error

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15



Potential Future Work - Mixed Batches

- How to deal with batches made up of data from different policies?
- Behavior policy estimation stays the same

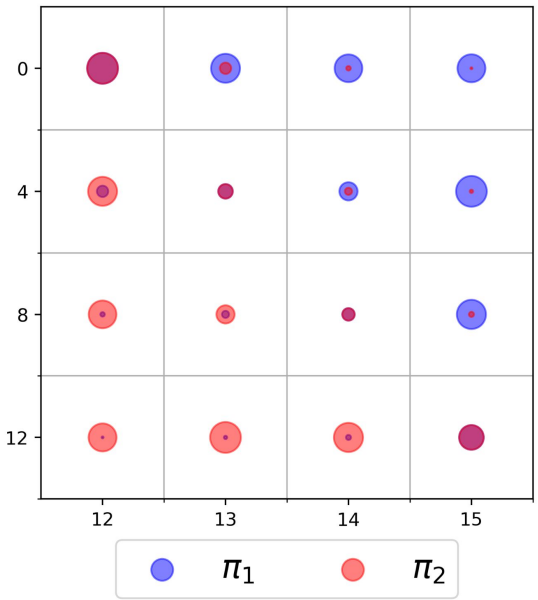
- What evaluation policy to use?

$$\pi_{mix}(a|s) = \frac{\sum_{i=1}^{K} \lambda_i \rho_{\pi_i}(s,a)}{\sum_{i=1}^{K} \lambda_i \rho_{\pi_i}(s)}$$

where the batch was obtained by executing K policies $\pi_1, \ldots, \pi_K$

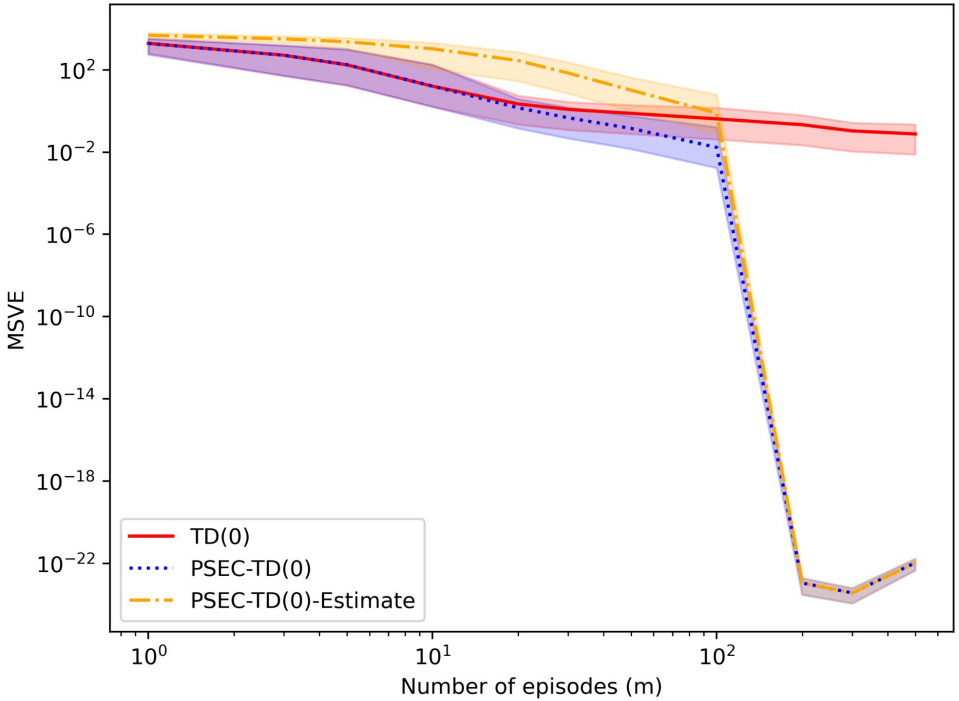
and policy *i* is executed with likelihood λ_i

Experiment



- 4x4 grid world
- Two policies, data collected from them equally
- Size of bubbles shows relative likelihood of visitation under corresponding policy

Results - Mixed Batches



- Two policies with data 50% from each policy
- Evaluation policy calculated with DP
- Takes more episodes to eliminate policy sampling error

Policy Sampling Error – Summary

- Estimating the empirical distribution of the policy can help eliminate sampling error
- Analysis shows that PSEC-TD(0) converges to a more desirable fixed point compared to TD(0)
- Experiments show that PSEC-TD(0) eliminates sampling error
- Introduce a potential avenue for future work

Overview

1. Overcoming policy sampling error	**2. Simulator grounding as imitation from observations (IfO)**	**3. An algorithm for sim-to-real transfer**
4. Time-step metric for estimating Wasserstein distance in MDPs	5. Learning a goal conditioned policy by minimizing W1 distance	6. Multi-agent coordination

Policy | Transitions | States

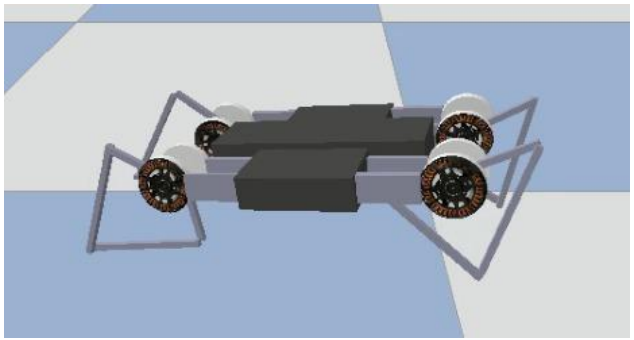
Simulator Grounding and GARAT [1]

- Transfer with dynamics mismatch seen through the transition distributions induced
- Contribution 1: Show that simulator grounding via grounded action transformation (GAT)[2] is equivalent to imitation from observations (IfO) where the expert is the target environment (real world)
- Contribution 2: Derive an adversarial distribution matching algorithm, generative adversarial reinforced action transformation (GARAT), to train the action transformation function

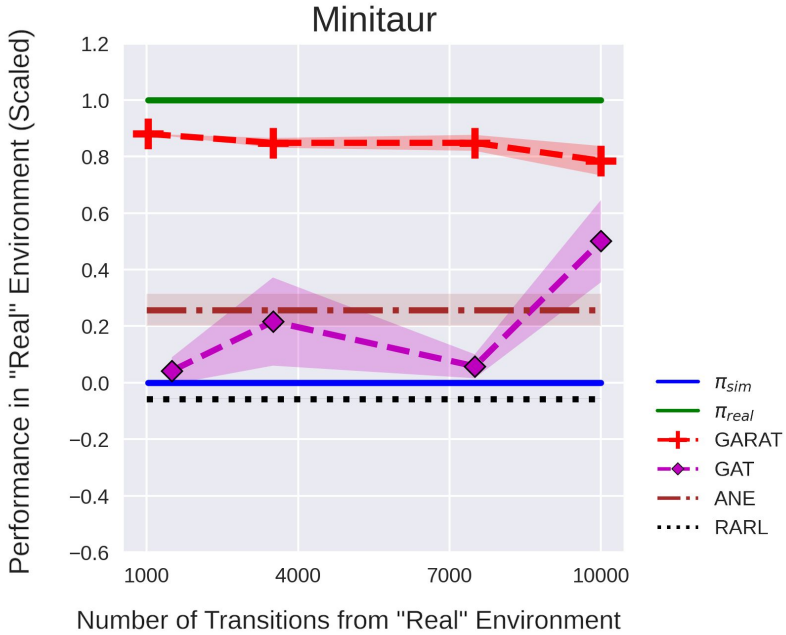
[1] An Imitation from Observation Approach to Transfer Learning with Dynamics Mismatch; *Desai, S., ***Durugkar, I.**, *Karnan, H., Warnell, G., Hanna, J. and Stone, P; NeurIPS 2020
* - joint first authors
[2] Grounded action transformation; Hanna et al.; AAAI 2017

Results - Evaluating Transfer

- Transfer between two simulators for Minitaur[1]
- Baselines trained for 1 million time-steps
- Results scaled to set performance of π_{sim} to 0 and π_{real} to 1



Minitaur domain



[1] Sim-to-Real: Learning Agile Locomotion For Quadruped Robots, Tan et al., RSS 2018

ANE - Noise and the reality gap: The use of simulation in evolutionary robotics, Morán et al., Advances in Artificial Life, 1995
RARL - Robust adversarial reinforcement learning, Pinto et al., ICML 2017

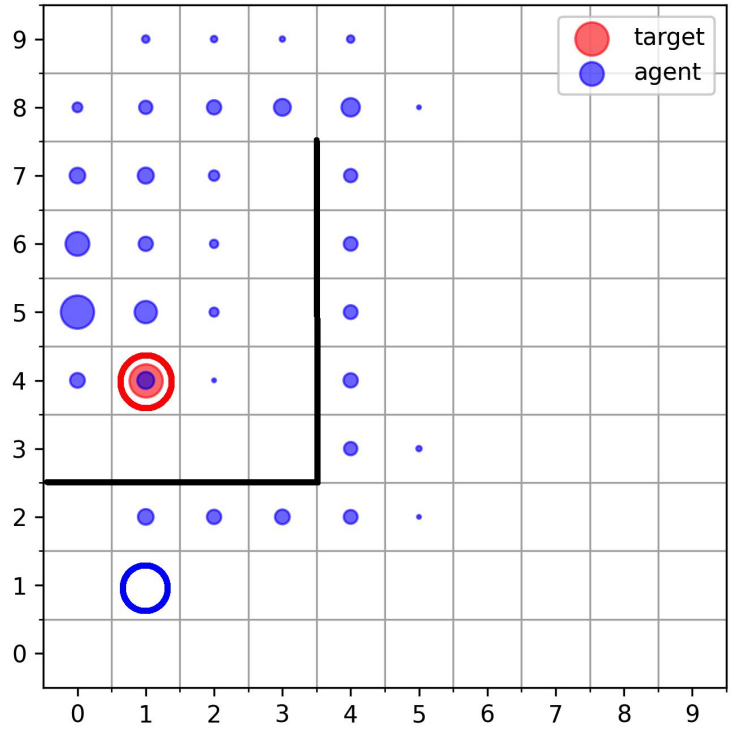
Overview

1. Overcoming policy sampling error	2. Simulator grounding as imitation from observations (IfO)	3. An algorithm for sim-to-real transfer
4. Time-step metric for estimating Wasserstein distance in MDPs	**5. Learning a goal conditioned policy by minimizing W1 distance**	6. Multi-agent coordination

Policy	Transitions	States

Minimize distribution mismatch for Goal-conditioned RL

- **Goal-conditioned RL**[1]: Agent needs to reach a goal given to it at the beginning of its episode.
- Blue circle - start state
- Red circle - goal state
- Target distribution can be specified as **Dirac distribution at the goal**
- Agent needs to **minimize mismatch of its state visitation distribution** to this target distribution.



[1] Learning to achieve goals; Kaelbling; IJCAI 1993

Minimize distribution mismatch in Goal-conditioned RL[1]

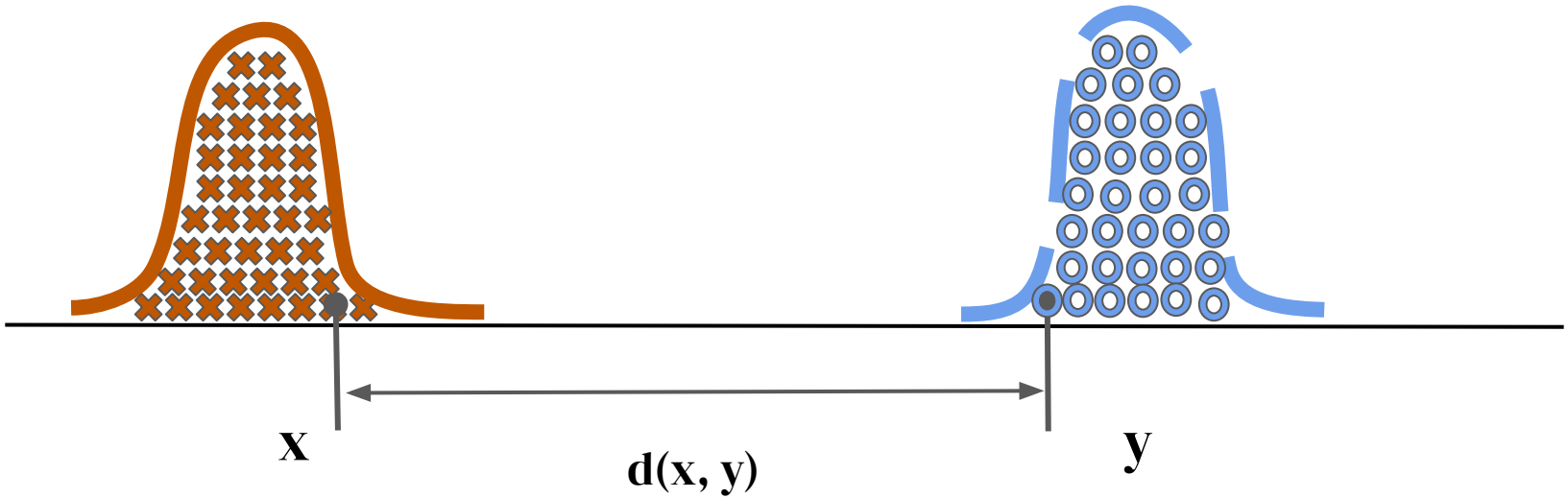
- Contribution 1: Study use of Wasserstein distance to minimize state visitation distribution mismatch. Propose use of time-step metric as ground metric for Wasserstein distance
- Contribution 2: Propose an adversarial procedure, adversarial intrinsic motivation (AIM) to learn a reward function which results in a policy that minimizes Wasserstein distance to a goal.

[1] Adversarial intrinsic motivation for reinforcement learning; **Durugkar, I.**, Tec, M., Niekum S., and Stone, P.; NeurIPS 2021

Wasserstein Distance

- Distance between distributions (say μ and ν)

$$W_d^p(\mu,\nu) := \inf_{\zeta \in Z(\mu,\nu)} \mathop{\mathbb{E}}_{(x,y)\sim\zeta} [d(x,y)^p]^{\frac{1}{p}}$$

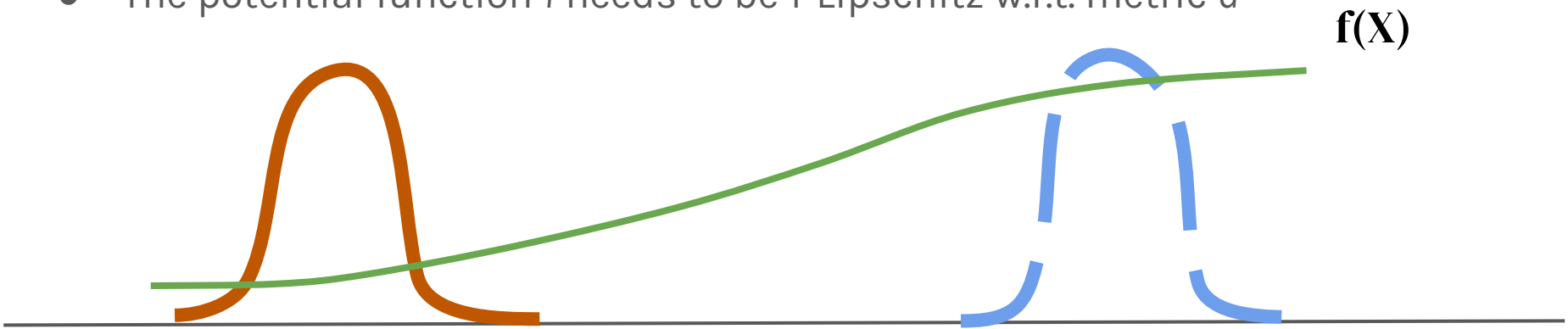


Wasserstein Distance using Kantorovich Duality

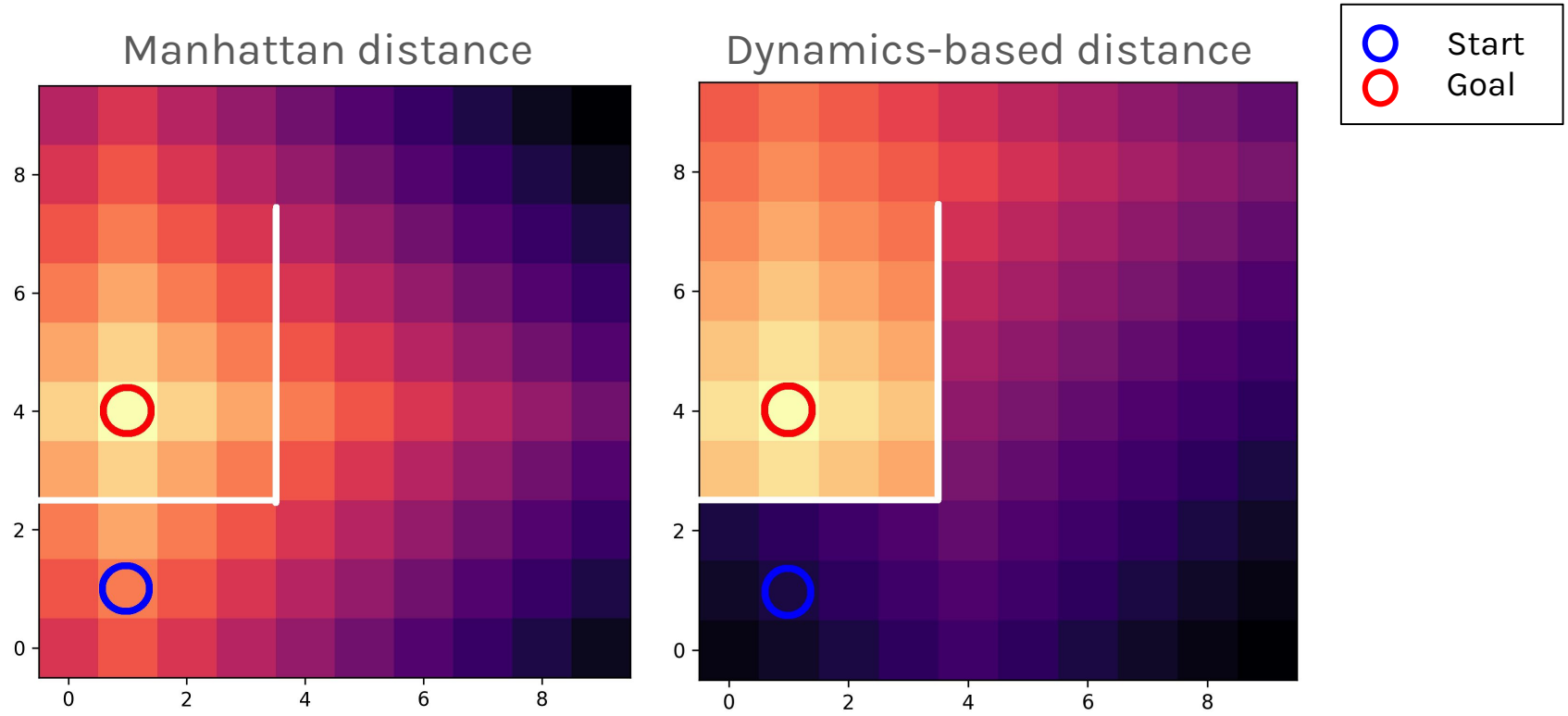
- If estimating Wasserstein-1 distance, the dual form can be used

$$W_d^1(\mu, \nu) = \sup_{\text{Lip}(f) \leq 1} \mathbb{E}_{y \sim \nu}[f(y)] - \mathbb{E}_{x \sim \mu}[f(x)]$$

- The potential function *f* needs to be 1-Lipschitz w.r.t. metric *d*



Why the Ground Metric Matters



Wasserstein Distance using Kantorovich Duality

- In most previous work, *d* is assumed to be L2 distance between features.
- I propose the use of the time-step metric for *d* in MDPs

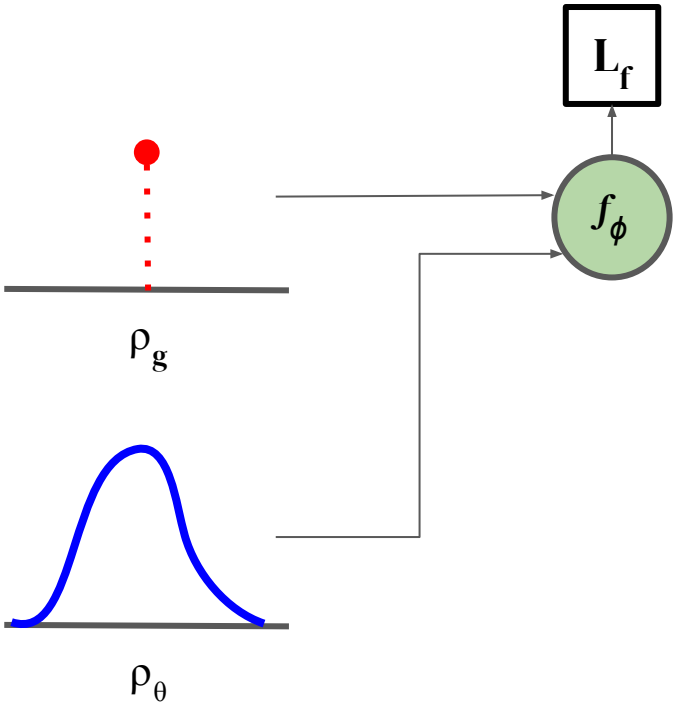
$$d_T^{\pi}(x, y) = \mathbb{E}\left[T(y|\pi, x)\right]$$

- Lipschitz continuity can be enforced as follows:

$$\mathop{\mathbb{E}}_{s' \sim \pi, P}\left[\left|f(s) - f(s')\right|\right] \leq 1 \quad \forall s \in \mathcal{S}$$

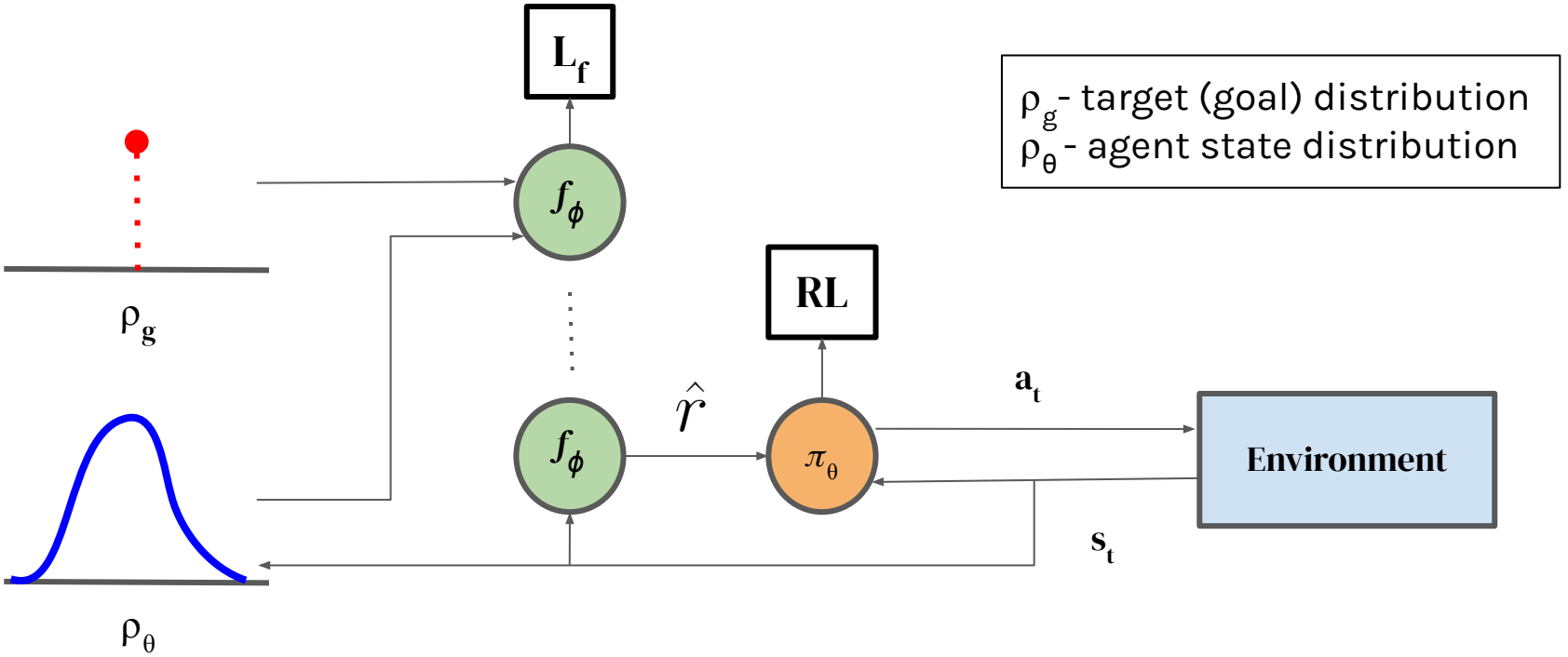
- In practice, we enforce it using samples from the environment

Adversarial Intrinsic Motivation (AIM)



ρ_g - target (goal) distribution
ρ_θ - agent state distribution

Adversarial Intrinsic Motivation (AIM)



Analysis

Analysis - Discounted Setting

Comparing the optimal policy (π^*) and the policy that minimizes Wasserstein distance to goal (π ♦)

Proposition 4: A lower bound on the value of any state under a policy π can be expressed in terms of the time-step distance from that state to the goal:

$$v^{\pi}(s|s_g) = \mathbb{E}\left[\gamma^{T(s_g|\pi,s)}\right] \geq \gamma^{d_T^{\pi}(s,s_g)} \quad \forall\ s \in \mathcal{S}$$

Analysis - Discounted Setting

Comparing the optimal policy (π^*) and the policy that minimizes Wasserstein distance to goal (π ♦)

Proposition 4: A lower bound on the value of any state under a policy π can be expressed in terms of the time-step distance from that state to the goal:

$$v^{\pi}(s|s_g) = \mathbb{E}\left[\gamma^{T(s_g|\pi,s)}\right] \geq \gamma^{d_T^{\pi}(s,s_g)} \quad \forall\ s \in \mathcal{S}$$

Theorem 5: If the transition dynamics are deterministic, the policy that minimizes the Wasserstein distance over the time-step metric in a goal-conditioned MDP is the optimal policy.

Analysis - Undiscounted Setting

- Undiscounted setting (γ = 1), with reward function

$$r(s_t, a_t, s_{t+1}|s_g) := \begin{cases} 0 & \text{if } s_{t+1} = \bar{s} \\ -1 & \text{otherwise} \end{cases}$$

- Assume agent reaches goal state from any start state within T steps

Analysis - Undiscounted Setting

- Undiscounted setting (γ = 1), with reward function

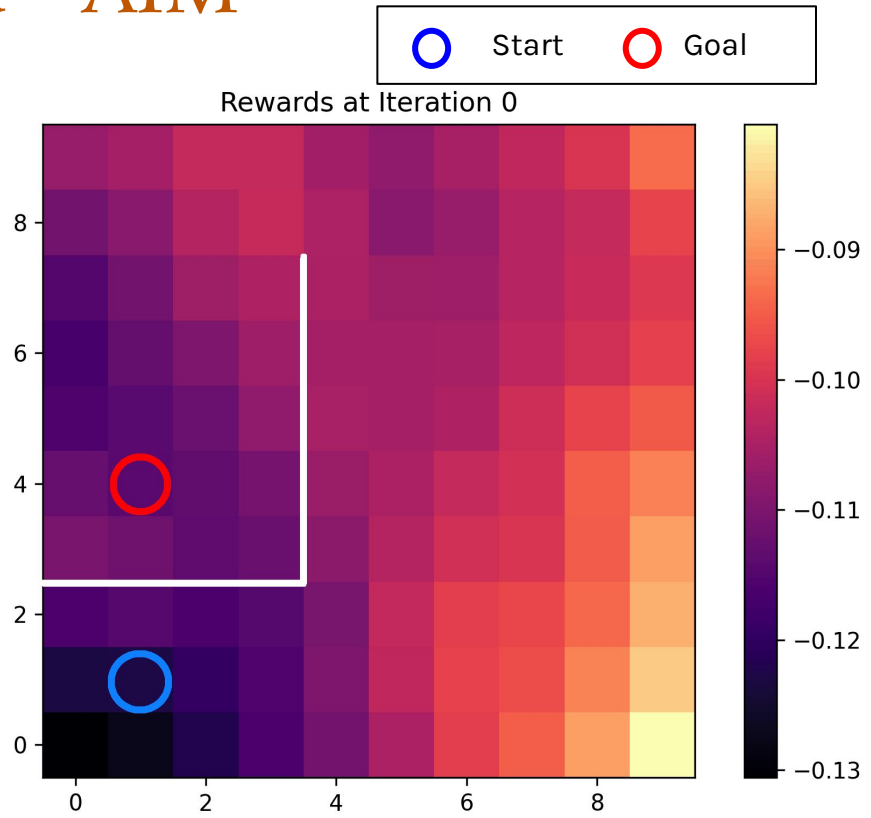
$$r(s_t, a_t, s_{t+1}|s_g) := \begin{cases} 0 & \text{if } s_{t+1} = \bar{s} \\ -1 & \text{otherwise} \end{cases}$$

- Assume agent reaches goal state from any start state within T steps

Proposition 8: Assuming non-zero measure for all states s under the agent's state visitation distribution ρ_π, and considering s_g as the given goal state, the difference in potentials $f(s) - f(s_g) = v_\pi(s|s_g)$

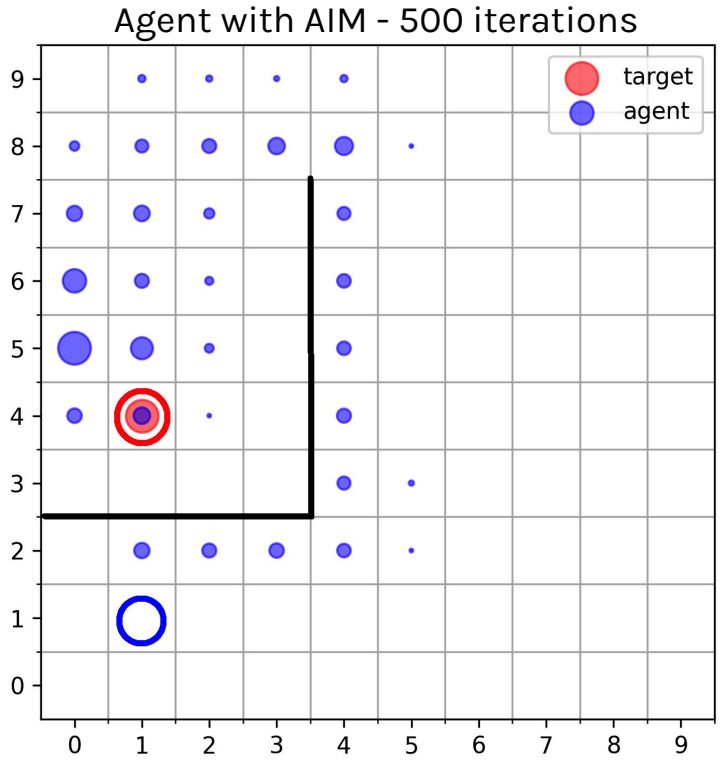
Experiments: Grid World - AIM

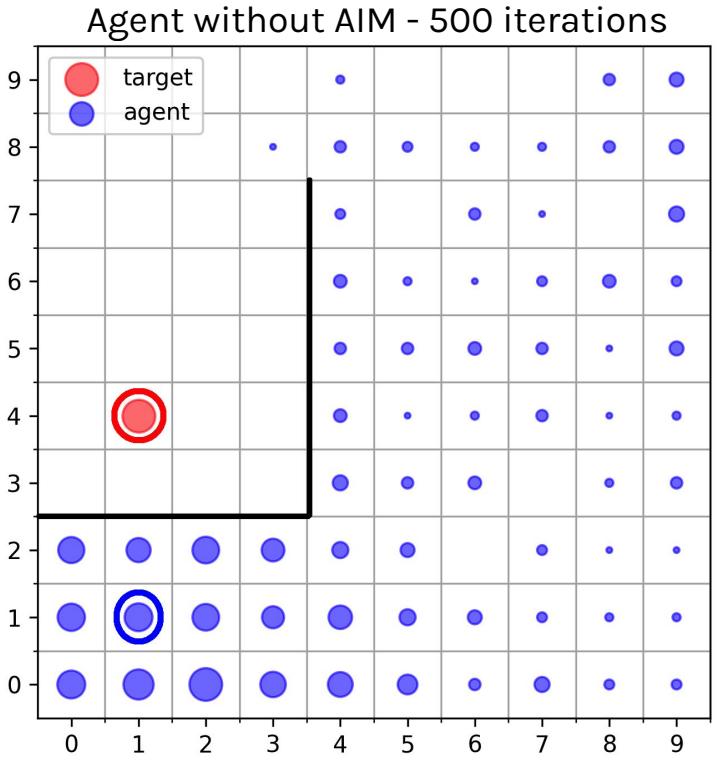
- 10 x 10 grid world, 4 actions, deterministic transitions
- Bold white lines are walls that agent cannot cross
- Features - (x, y) coordinates of agent state
- Agent algorithm – soft Q-learning[1]
- Every iteration involves data collection, 5 potential function update steps, and 10 Q-function update steps



[1] Reinforcement learning with deep energy-based policies; Haarnoja et al.; ICML 2017

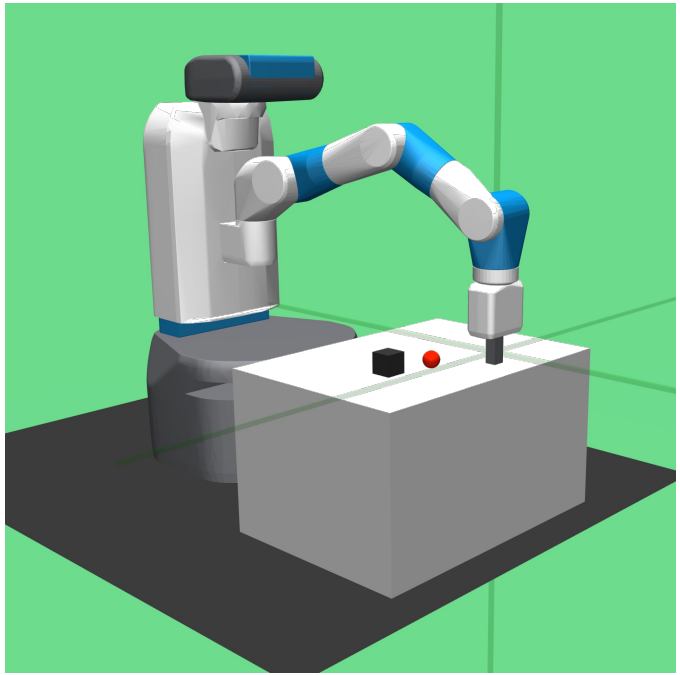
Experiments: Grid World





Experiments: Fetch Domain

- MuJoCo gym environment
- Continuous state and action space
- Various tasks: Reach, Push, Slide, and Pick and Place
- AIM combined with HER[1] (AIM + HER)
- Policy trained with TD3[2]

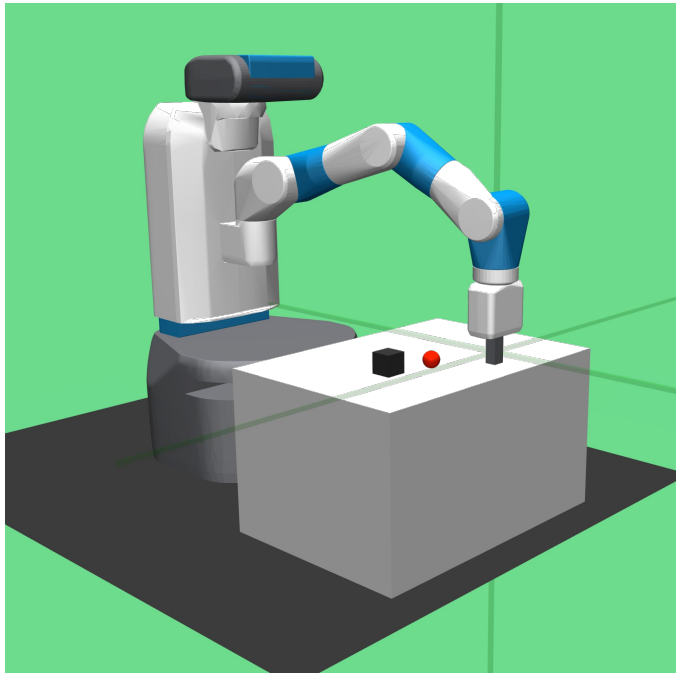


[1] Hindsight experience replay; Andrychowicz et al.; NeurIPS 2017
[2] Addressing function approximation error in actor-critic methods; Fujimoto et al.; ICML 2018

Experiments: Fetch Domain

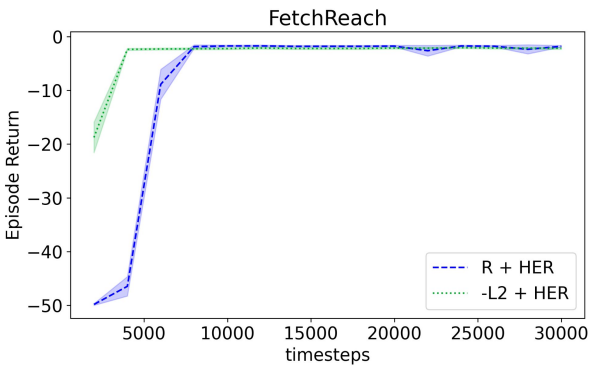
Baselines:

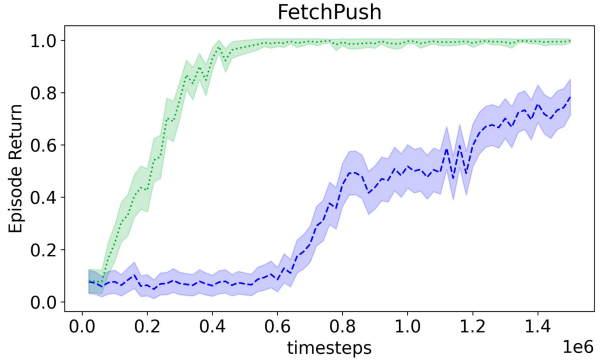
- Only sparse reward (R + HER)
- Exact distance to goal (-L2 + HER)
  - Oracle reward
- Distance learned via regression from MC rollouts[1] (MC + R + HER)
- General exploration bonus[2] (RND + R + HER)
- GAIL[3] reward from hindsight trajectories (GAIL + R + HER)

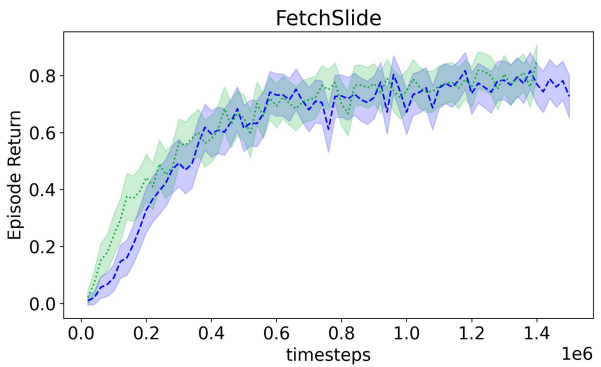


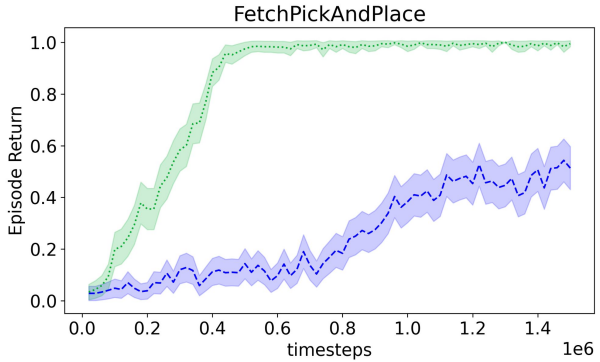
[1] Dynamical distance learning for semi-supervised and unsupervised skill discovery; Hartikainen et al.; ICLR 2020
[2] Exploration by random network distillation; Burda et al.; ICLR 2019
[3] Generative adversarial imitation learning; Ho and Ermon; NeurIPS 2016

Experiments: Fetch Domain

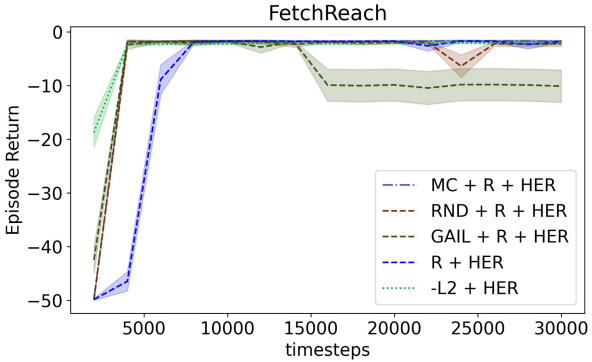


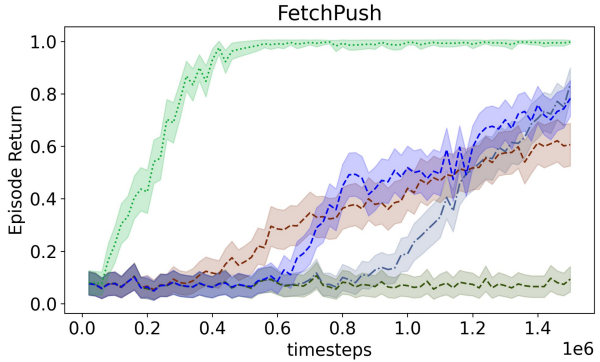


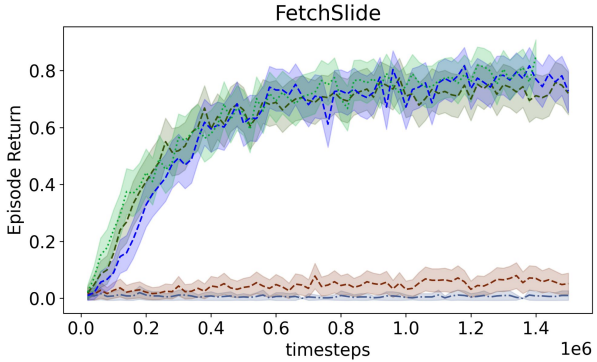


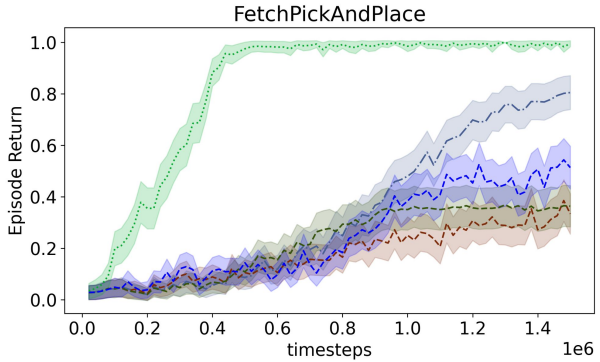


Experiments: Fetch Domain

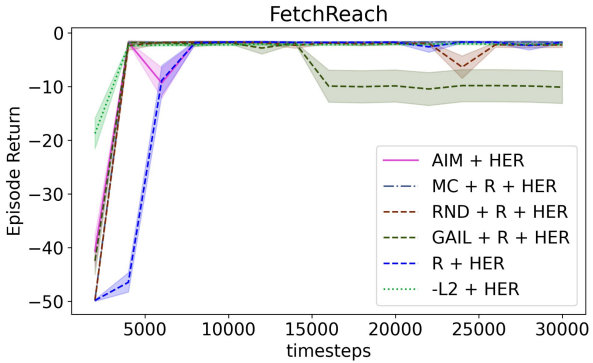


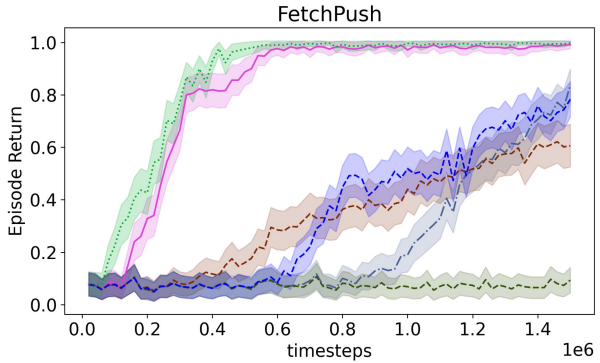


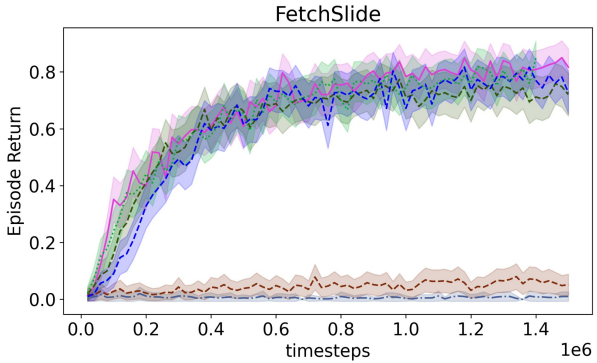


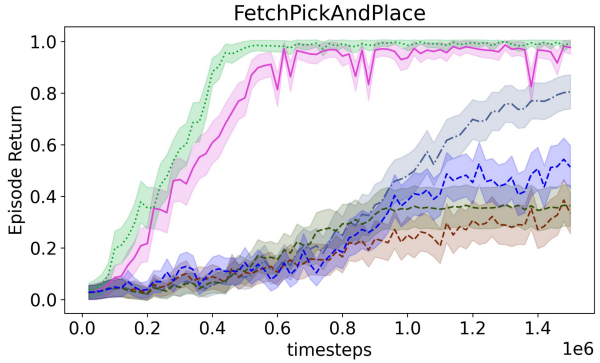


Experiments: Fetch Domain









Adversarial Intrinsic Motivation – Summary

- Considering the goal-conditioned RL problem through a perspective of distribution mismatch minimization.
- Requires use of the Wasserstein distance.
- Introduce a novel regularization objective for estimating Wasserstein distance in MDPs
- Compare learning under AIM with learning with sparse reward in goal-conditioned RL
- Experimental validation

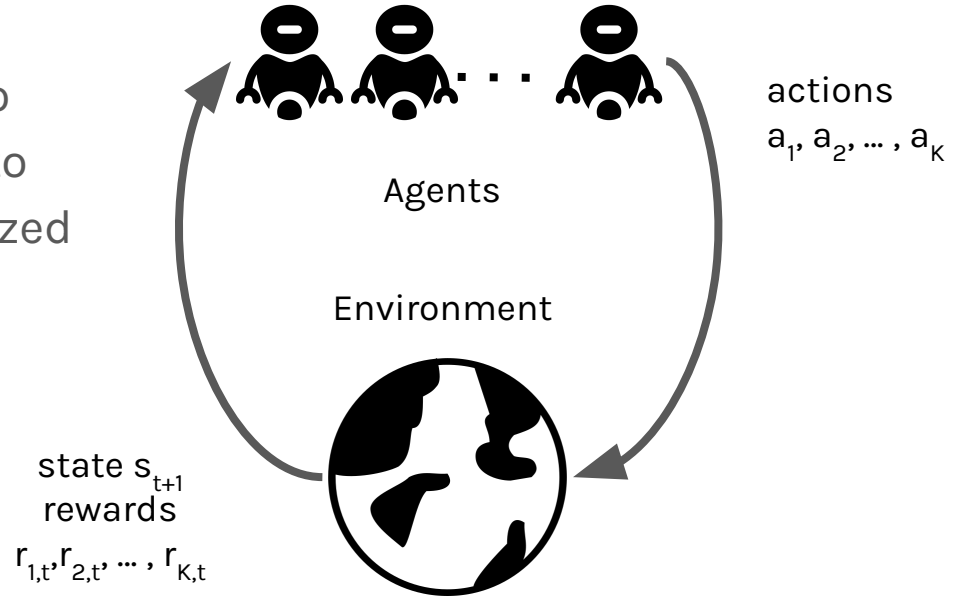
Overview

1. Overcoming policy sampling error	2. Simulator grounding as imitation from observations (IfO)	3. An algorithm for sim-to-real transfer
4. Time-step metric for estimating Wasserstein distance in MDPs	5. Learning a goal conditioned policy by minimizing W1 distance	**6. Multi-agent coordination**

Policy	Transitions	States

Multi-agent Coordination via Distribution Matching[1]

- Contribution: Distribution matching as a novel method to present a coordination signal to agents learning in a decentralized manner

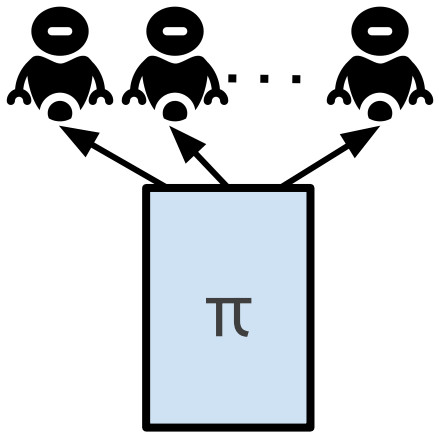


[1] DM^2: Decentralized Multi-Agent Reinforcement Learning via Distribution Matching; *Wang, C., ***Durugkar, I.***, *Liebman, E., Stone P.; AAAI 2023

* - joint first authors

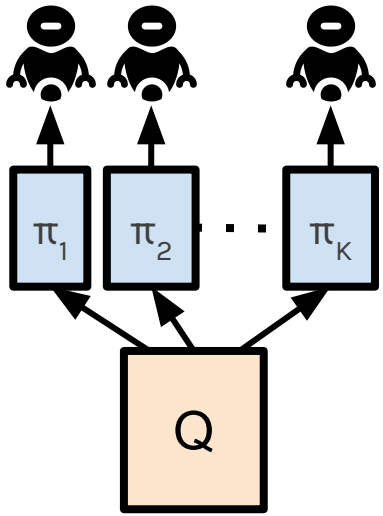
Motivation

- Multi-agent reinforcement learning (MARL) is challenging — agents learning simultaneously makes the environment nonstationary
- Strategies:
  - Fully centralized learning



Motivation

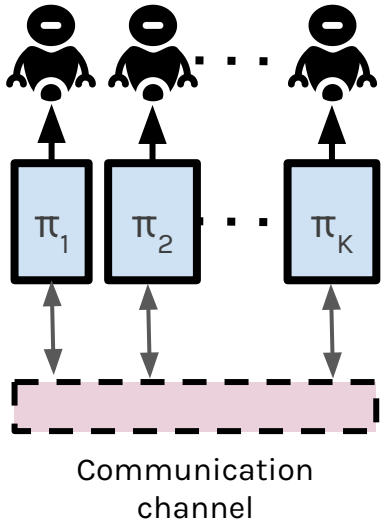
- Multi-agent reinforcement learning (MARL) is challenging — agents learning simultaneously makes the environment nonstationary
- Strategies:
  - Fully centralized learning
  - Centralized training, decentralized execution (CTDE) [1]



[1] Sunehag et al., Value Decomposition Networks for Cooperative Multiagent learning, AAMAS 2018.

Motivation

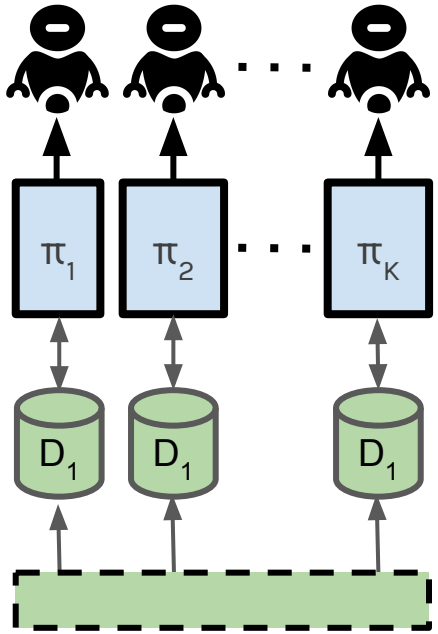
- Multi-agent reinforcement learning (MARL) is challenging — agents learning simultaneously makes the environment nonstationary
- Strategies:
  - Fully centralized learning
  - Centralized training, decentralized execution (CTDE) [1]
  - Decentralized learning + communication[2]



[1] Sunehag et al., Value Decomposition Networks for Cooperative Multiagent learning, AAMAS 2018.
[2] Jaques et al., Social Influence as Intrinsic Motivation for Multi-Agent Deep Reinforcement Learning, ICML 2019.

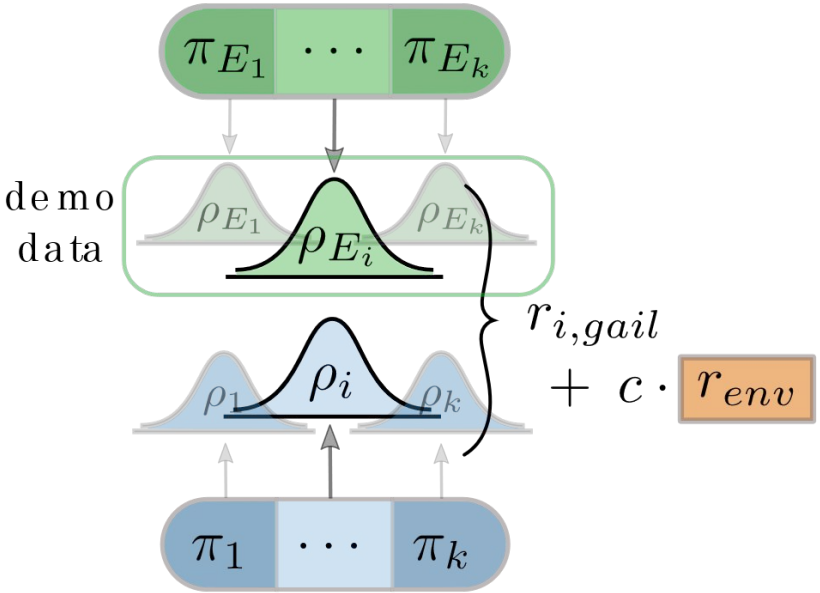
DM²: Decentralized MARL with Distribution Matching

- Control agent visitation distributions to induce coordination between agents learning independently
- The target distribution acts as coordination signal



DM²: Decentralized MARL with Distribution Matching

- Individual agents distribution matching to target distributions induced by demonstrations from coordinated expert demonstrations
- Distribution matching reward combined with task reward



Analysis

Theorem 7: Each agent maximizing its individual return over the individual distribution matching rewards r_{ϕ_i} will converge to the joint expert policy π_E

Analysis

Theorem 7: Each agent maximizing its individual return over the individual distribution matching rewards r_{ϕ_i} will converge to the joint expert policy π_E

If the expert policies are optimal with respect to the shared task, then π_E is a Nash equilibrium for rewards that are a linear combination of the task and distribution matching reward.

Experimental Setting

- StarCraft II Multi-Agent Challenge[1] tasks
  - 5m vs 6m (5v6)
  - 3s vs 4z (3sv4z)
- Baselines w/environment reward alone
  - IPPO (decentralized)
  - QMIX[2] (CTDE)
  - R-MAPPO[3] (CTDE)
- Distribution Matching Baseline: DM^2 w/SIL [4]

[1] Samvelyan et al., The StarCraft Multi-Agent Challenge, AAMAS 2019.
[2] Rashid et al., Qmix: Monotonic Value Function Factorisation for Deep Multi-agent Reinforcement Learning, ICML 2018.
[3] Yu et al., The Surprising Effectiveness of PPO in Cooperative, Multi-Agent Games, ArXiv 2021.
[4] Oh et al., Self-Imitation Learning, ICML 2018.

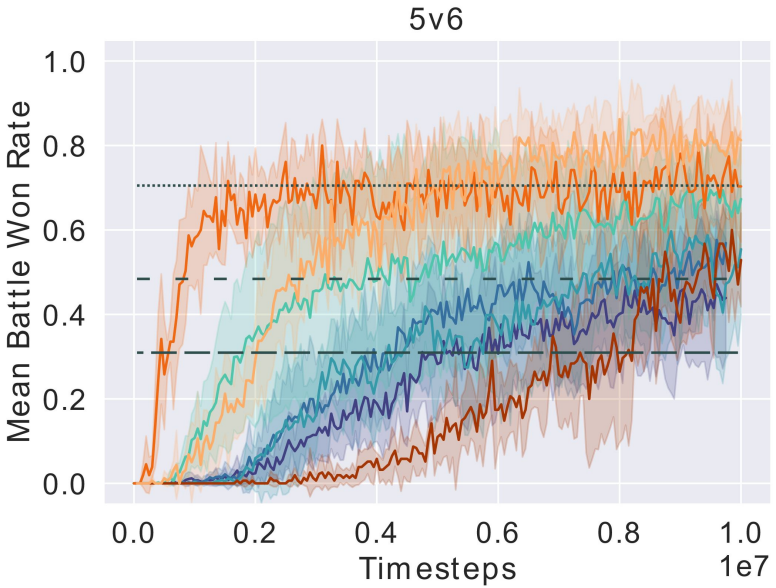
Experimental Setting

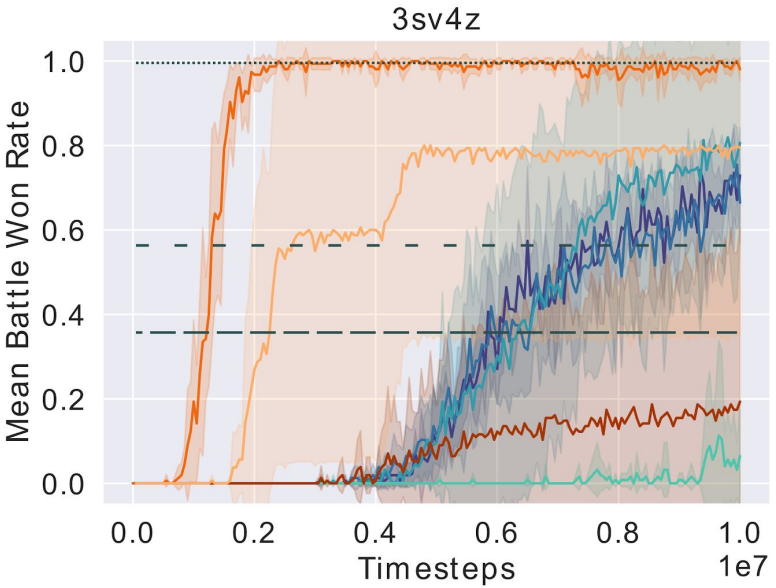
- MARL algorithm: Independent PPO (IPPO)[1]
- Demonstrations from K experts
  - State-only demonstrations sampled from saved IPPO **and** QMIX checkpoints
- Per-agent reward function:

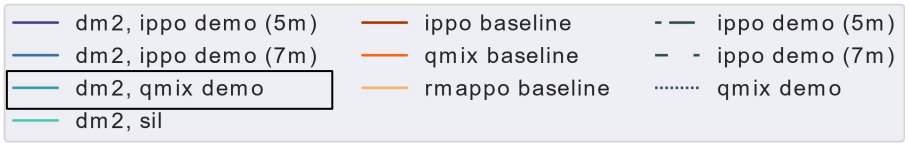
$$r_{i,mix} = r_{env} + r_{i,GAIL} * c$$

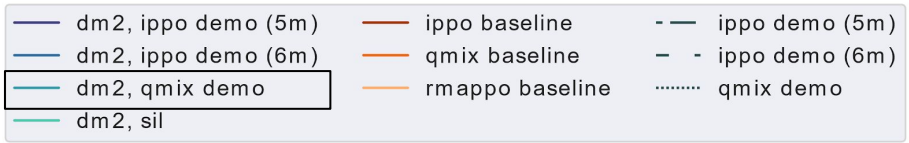
[1] Yu et al., The Surprising Effectiveness of PPO in Cooperative, Multi-Agent Games, ArXiv 2021.

Results – DM²


5v6
Mean Battle Won Rate
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
1e7
Timesteps


3sv4z
Mean Battle Won Rate
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
1e7
Timesteps


dm2, ippo demo (5m)
dm2, ippo demo (7m)
dm2, qmix demo
dm2, sil
ippo baseline
qmix baseline
rmappo baseline
ippo demo (5m)
ippo demo (7m)
qmix demo


dm2, ippo demo (5m)
dm2, ippo demo (6m)
dm2, qmix demo
dm2, sil
ippo baseline
qmix baseline
rmappo baseline
ippo demo (5m)
ippo demo (6m)
qmix demo

Multi-agent Coordination – Summary

- Controlling the state visitation distributions of individual agents can be a strategy for multi-agent coordination
- Can speed up learning for tasks, and improve upon performance of target distributions

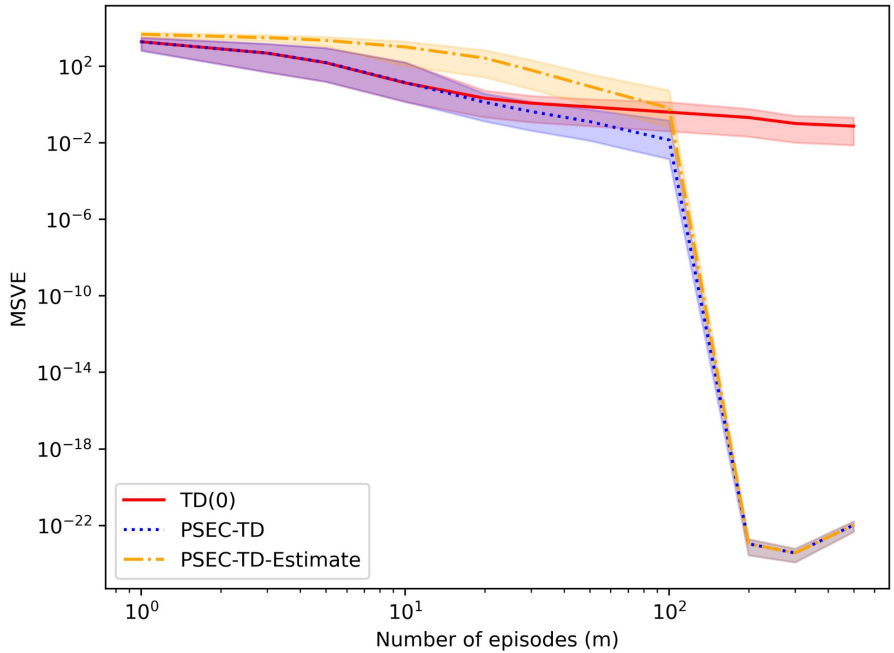
Overview

1. Overcoming policy sampling error	2. Simulator grounding as imitation from observations (IfO)	3. An algorithm for sim-to-real transfer
4. Time-step metric for estimating Wasserstein distance in MDPs	5. Learning a goal conditioned policy by minimizing W1 distance	6. Multi-agent coordination

Policy	Transitions	States

Future Work - Estimation

- PSEC-TD(0) showed that it can eliminate sampling error
- When evaluating a batch of data from a mixture of policies, initial experiments indicate that with enough data, PSEC might work as expected.
- More investigation is needed



Future Work - Estimation

- Estimation – long term research:
  - Combination of distributional RL and successor features
  - Impact of other distribution estimation techniques (diffusion, density estimation, and others)

Future Work – Minimizing Distribution Mismatch

- GARAT showed that learning an action transformation function can be seen as a distribution mismatch problem
- What other problems can benefit similarly?
- Short term avenue:
  - Other objectives for behavioral cloning[1]
- Long term avenue:
  - Learning a dynamics model of the environment

[1] ABC: Adversarial Behavioral Cloning for Offline Mode-seeking Imitation Learning; Hudson, E., **Durugkar, I.**, Warnell, G., Stone, P.; ArXiv 2022

Future Work – Extending AIM

- Use of the Wasserstein distance to measure distance between distributions
- Considering RL problems as controlling visitation distributions
- Short term:
  - Exploration
  - Beyond goal-conditioned RL
  - Skill learning
- Long term:
  - Distribution control for general reward functions

Future Work – Distribution Control in MARL

- DM^2 has opened doors for the kind of impact distribution control can have in MARL
- Potential avenues:
  - Better coordination techniques
  - Beyond cooperative tasks
  - Bootstrapping K-expert demonstrations to N agents (N > K)

Related Work

- Generative Adversarial Nets; Goodfellow et al.; NeurIPS 2014

Imitation Learning

- Generative adversarial imitation learning; Ho and Ermon; NeurIPS 2016

Off-Policy Evaluation

- Breaking the curse of horizon: infinite-horizon off-policy estimation; Liu et al.; NeurIPS 2018
- DualDICE: Behavior-agnostic estimation of discounted stationary distribution corrections; Nachum et al.; NeurIPS 2019

Distributional RL

- A distributional perspective on reinforcement learning; Bellemare et al.; ICML 2017

Exploration

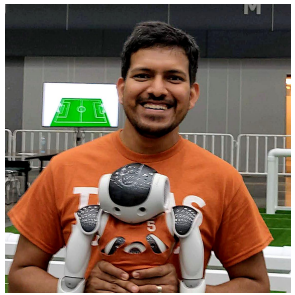
- Provably efficient maximum entropy exploration; Hazan et al.; ICML 2019
- Efficient exploration via state marginal matching; Lee et al.; ArXiv 2019

Summary

How can methods for estimating and controlling an agent's visitation distribution be beneficially incorporated into RL algorithms?

- Variety of problems benefit from estimating or controlling visitation distributions
- Various actionable insights and broader implications for future work

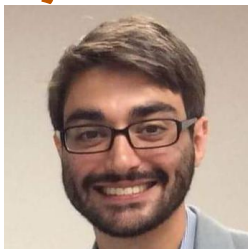
Thank you!



Ishan Durugkar

Peter Stone

Scott Niekum



Garrett Warnell

Josiah Hanna

Elad Liebman

Mauricio Tec

Siddharth Desai

Haresh Karnan

Brahma Pavse

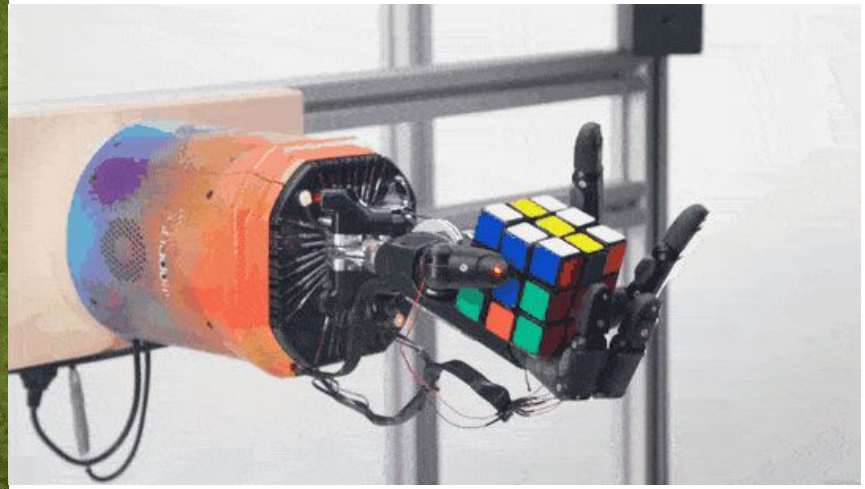
Caroline Wang

Questions?

How can methods for estimating and controlling an agent's visitation distribution be beneficially incorporated into RL algorithms?

1. Overcoming policy sampling error	2. Simulator grounding as imitation from observations (IfO)	3. An algorithm for sim-to-real transfer
4. Time-step metric for estimating Wasserstein distance in MDPs	5. Learning a goal conditioned policy by minimizing W1 distance	6. Multi-agent coordination

Successes



Distribution Estimation and Control for RL

Actions

- Reducing Sampling Error in Batch Temporal Difference Learning; Pavse, B., **Durugkar, I.**, Hanna, J., Stone, P.; ICML 2021

Transitions

- An Imitation from Observation Approach to Transfer Learning with Dynamics Mismatch;
  *Desai, S., ***Durugkar, I.**, *Karnan, H., Warnell, G., Hanna, J. and Stone, P; NeurIPS 2020

States

- Adversarial Intrinsic Motivation for Reinforcement Learning; **Durugkar, I.**, Tec, M., Niekum S., Stone, P.; NeurIPS 2021
- DM^2: Distributed Multi-Agent Reinforcement Learning by Distribution Matching
  *Wang, C., ***Durugkar, I.**, *Liebman, E., Stone, P.; AAAI 2023

* - joint first authors

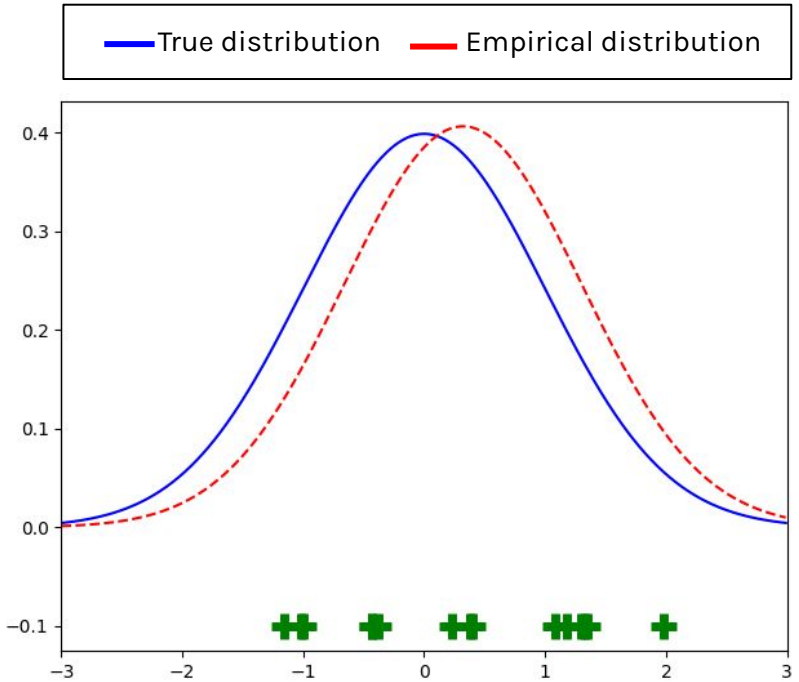
Simulator Grounding – Summary

- Show that the problem of learning an action transformation function is one of imitation from observations.
- A generative adversarial technique to minimize the transition visitation distribution mismatch between the source domain and the target domain.
- Experiments show that this perspective is more efficient and effective than previous approaches.

PSEC

Policy Sampling Error Corrected (PSEC)-TD learning

- Discrepancy between distribution that generated the data and distribution implied by this data
- Contribution[1]:
  - Improve batch temporal difference (TD) learning by estimating the policy distribution implied by the data
  - Analyze the fixed point reached by using this method



[1] Reducing sampling error in batch temporal difference learning; Pavse, B., **Durugkar, I.**, Hanna, J. and Stone, P.; ICML 2020

Analysis of TD(0) Fixed Point

- TD(0) converges to the following fixed point[1], based on the maximum likelihood MDP and policy estimated from the data.

$$\hat{v}^{\hat{\pi}}(s) = \sum_{a \in \widehat{\mathcal{A}}} \hat{\pi}(a|s)[\bar{R}(s,a) + \gamma \sum_{k \in \widehat{\mathcal{S}}} \widehat{P}(s'|s,a)\hat{v}^{\hat{\pi}}(s'))]$$

[1] Sutton, 1988

Analysis of PSEC-TD(0) Fixed Point

- PSEC corrects for the policy sampling error
- It converges to the fixed point based on the MDP estimated from the data, but following the policy that generated the data π.

$$\hat{v}^{\pi}(s) = \sum_{a \in \widehat{\mathcal{A}}} \pi(a|s)[\bar{R}(s,a) + \gamma \sum_{k \in \widehat{\mathcal{S}}} \widehat{P}(s'|s,a)\hat{v}^{\pi}(s'))]$$

PSEC-TD(0) Mechanism

- Algorithm 1 lays out batch linear temporal difference learning (TD(0)) with off-policy updates.
- For policy sampling error corrected (PSEC) TD(0), set π_e as the policy which generated the data (π), and π_b as maximum likelihood policy of the batch ($\hat{\pi}$)

Algorithm 1 Batch Linear TD(0) to estimate v^{π_e}

Input: policy to evaluate π_e, behavior policy π_b, batch $\mathcal{D}$, linear value function, $\hat{v} : \mathcal{S} \times \mathbb{R}^d \to \mathbb{R}$, step-size $\alpha > 0$, convergence threshold $\Delta > 0$
Initialize: weight vector $\boldsymbol{w}_0$ arbitrarily (e.g.: $\boldsymbol{w}_0 := \mathbf{0}$), aggregation vector $\boldsymbol{u} := \mathbf{0}$, batch process counter, $i = 0$
while $|\boldsymbol{w}_{i+1} - \boldsymbol{w}_i| \geq \mathbf{1} \cdot \Delta$ **do**
  for each episode, $\tau \in \mathcal{D}$ **do**
    for each transition, $(s, a, r, s') \in \tau$ **do**
      $\hat{y} \leftarrow r + \gamma \boldsymbol{w}_i^T \boldsymbol{x}(s')$
      $\rho \leftarrow \frac{\pi_e(a|s)}{\pi_b(a|s)}$ {for on-policy, $\pi_b = \pi_e$}
      $\boldsymbol{u} \leftarrow \boldsymbol{u} + \left[\rho \hat{y} - \boldsymbol{w}_i^T \boldsymbol{x}(s)\right] \boldsymbol{x}(s)$
    end for
  end for
  $\boldsymbol{w}_{i+1} \leftarrow \boldsymbol{w}_i + \alpha \boldsymbol{u}$ {batch update}
  $\boldsymbol{u} \leftarrow \mathbf{0}$ {clear aggregation}
  $i \leftarrow i + 1$
end while

GARAT

The Sim-to-Real Problem : Dynamics Mismatch

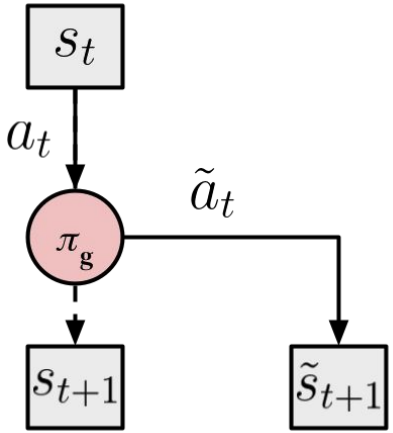
Sim-to-real transfer refers to learning the agent policy in a simulator (source environment) and transferring it to the real world (target environment).

The problem arises when the simulator and the real world have different dynamics. i.e. the same action leads to different states

Background: Grounded Action Transformation (GAT)

- Grounded action transformation [1] (GAT) introduces an action transformation function that changes the action that the agent picks before it is sent to the simulator

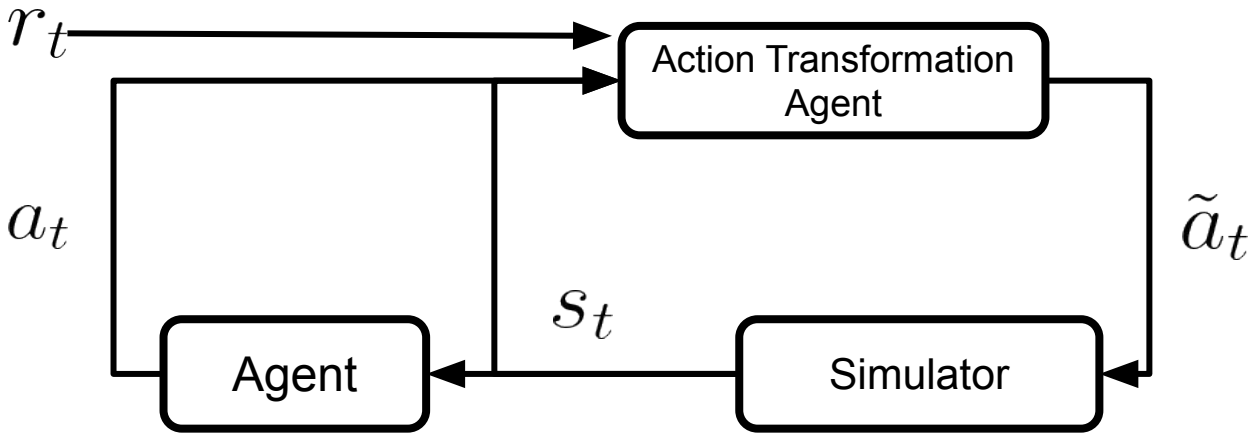
Simulator with ideal action transformation function



[1] Hanna et al., AAAI 2017

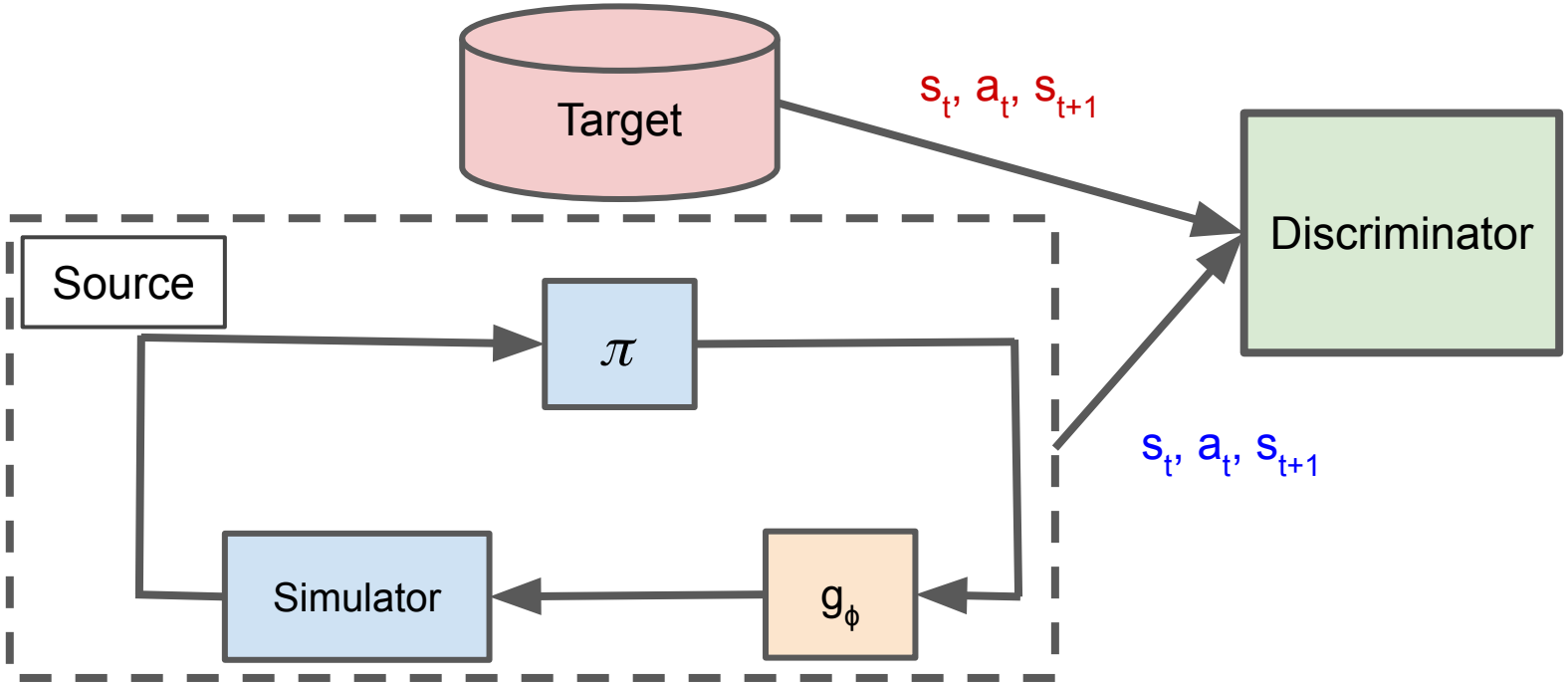
Action Transformation as Imitation from Observations[1]

- Action transformation[2] can be seen as a sequential decision making task
- The task is to take actions $\tilde{a}_t$ such that sequence of its states, $x_t = (s_t, a_t)$, are as similar to the real world (expert) as possible



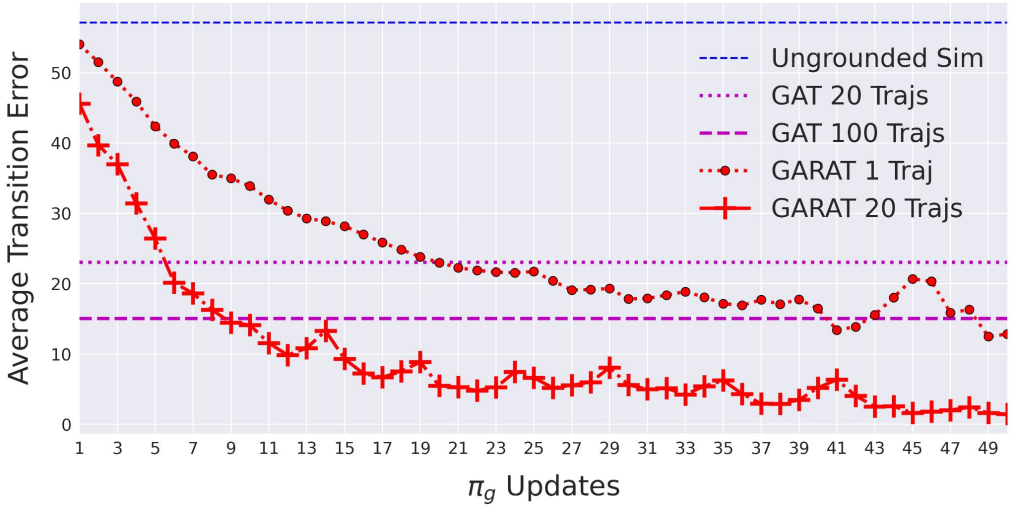
[1] Imitation from observation: Learning to imitate behaviors from raw video via context translation; Liu et al; ICRA 2018
[2] Grounded action transformation; Hanna et al.; AAAI 2017

Generative Adversarial Reinforced Action Transformation



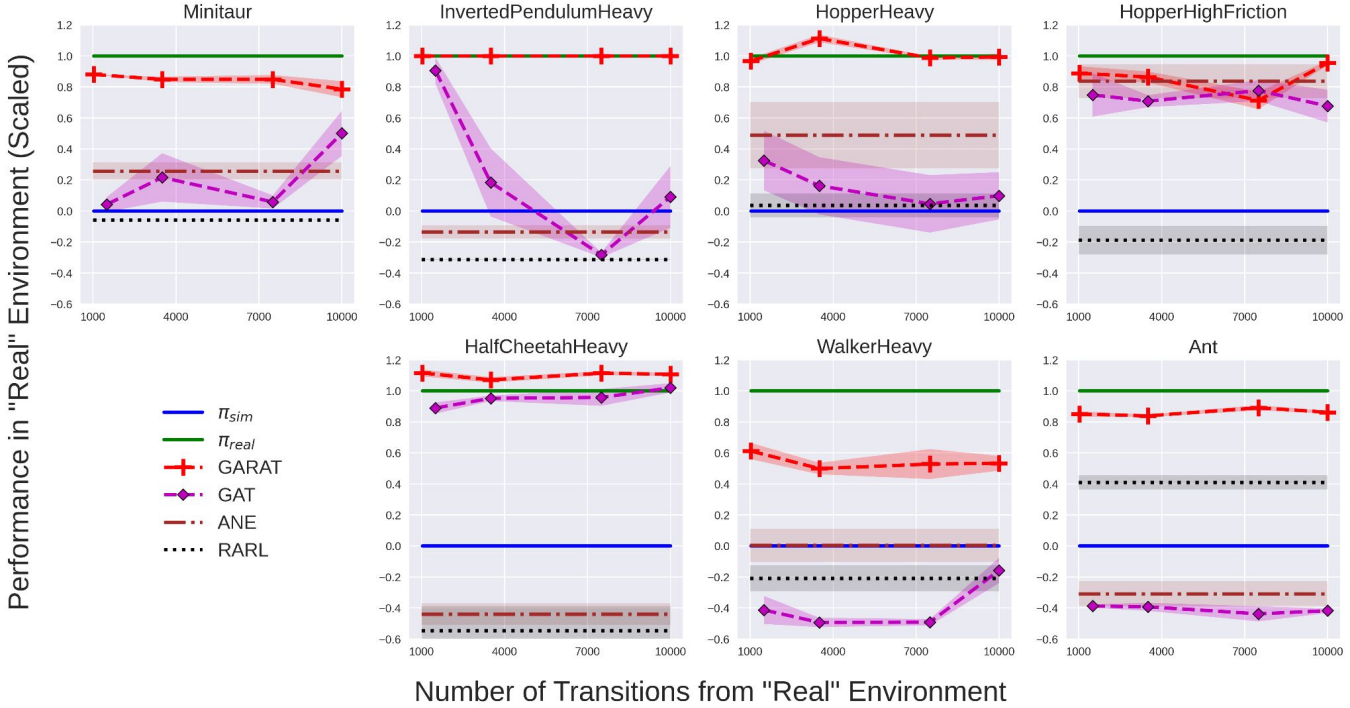
Results - Inverted Pendulum

- Transfer from standard Inverted Pendulum environment in MuJoCo to one with heavier pole
- GAT trained to convergence
- Action transformation policy in GARAT trained with PPO[1]



[1] Proximal Policy Optimization Algorithms, Schulman et al., ArXiv, 2017

Results - Evaluating Transfer



AIM Objectives

Objective

- Potential function f_ϕ will estimate Wasserstein distance, and agent will learn policy π_θ to minimize it
- Adversarial objective:

$$\min_{\theta \in \Theta} \max_{\phi \in \Phi} \mathop{\mathbb{E}}_{s_g \sim \sigma} \left[f_\phi(s_g, s_g) - \mathop{\mathbb{E}}_{s \sim \rho_\theta} [f_\phi(s, s_g)] \right]$$

- Potential function additionally needs to be 1-Lipschitz w.r.t. time-step metric

Objectives

- Given a fixed policy π_θ the objective for the potential function is:

$$L_f := \mathop{\mathbb{E}}_{s_g \sim \sigma} \left[-f_\phi(s_g, s_g) + \mathop{\mathbb{E}}_{s \sim \rho_\theta} \left[f_\phi(s, s_g) \right] \right] + \lambda \mathop{\mathbb{E}}_{(s,a,s',s_g) \sim \mathcal{D}} \left[(\max(|f_\phi(s, s_g) - f_\phi(s', s_g)| - 1, 0))^2 \right]$$

- The reward function for the agent is:

$$\hat{r}(s, a, s', s_g) = f_\phi(s', s_g) - b$$

Background: Stochastic Games

- Stochastic game[1] $\langle K, \mathcal{S}, \mathcal{A}, \rho_0, \mathcal{T}, R, \gamma \rangle$
  - Number of agents K
  - State space $\mathcal{S}$
  - Action space $\mathcal{A} \equiv A^K$
  - Initial state distribution $\rho_0 : \Delta(\mathcal{S})$
  - Transition function $\mathcal{T} : \mathcal{S} \times A_0 \times \cdots \times A_{K-1} \mapsto \Delta(\mathcal{S})$
  - Reward function $R_i : \mathcal{S} \times A_0 \times \cdots \times A_{K-1} \mapsto \mathbb{R}$
  - Discount factor γ
- Per-agent policy $\pi_i : \mathcal{S} \mapsto \Delta(A_i)$

[1] Littman, Markov Games as a Framework for Multi-agent Reinforcement Learning, ICML 1994.

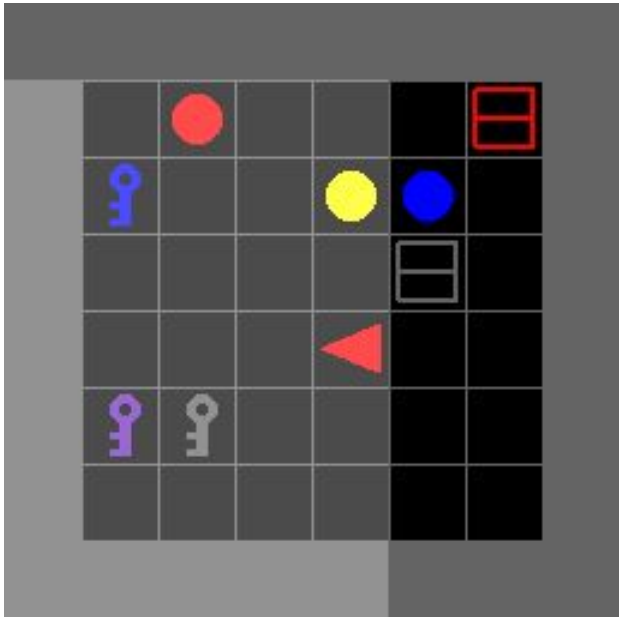
Exploration

- Go-Explore[1] provides a recipe for exploration: go to the frontier of the explored region, then explore randomly
- Returning to frontier states is done by resetting the simulator, or learning a goal-conditioned policy by behavioral cloning
- AIM can be used to learn a policy to efficiently reach fronter states

[1] First return, then explore; Ecoffet et al., Nature, 2021.

Beyond goal-conditioned RL

- AIM works well on goal-conditioned RL
- Can we extend its capabilities to other task-conditioned problem settings?
- Consider language-conditioned RL[1, 2]



The BabyAI domain

[1] BabyAI: A Platform to Study the Sample Efficiency of Grounded Language Learning; Chevalier-Boisvert et al., ICLR 2019.
[2] Using natural language for reward shaping in reinforcement learning; Goyal et al., IJCAI 2019.

Skill learning

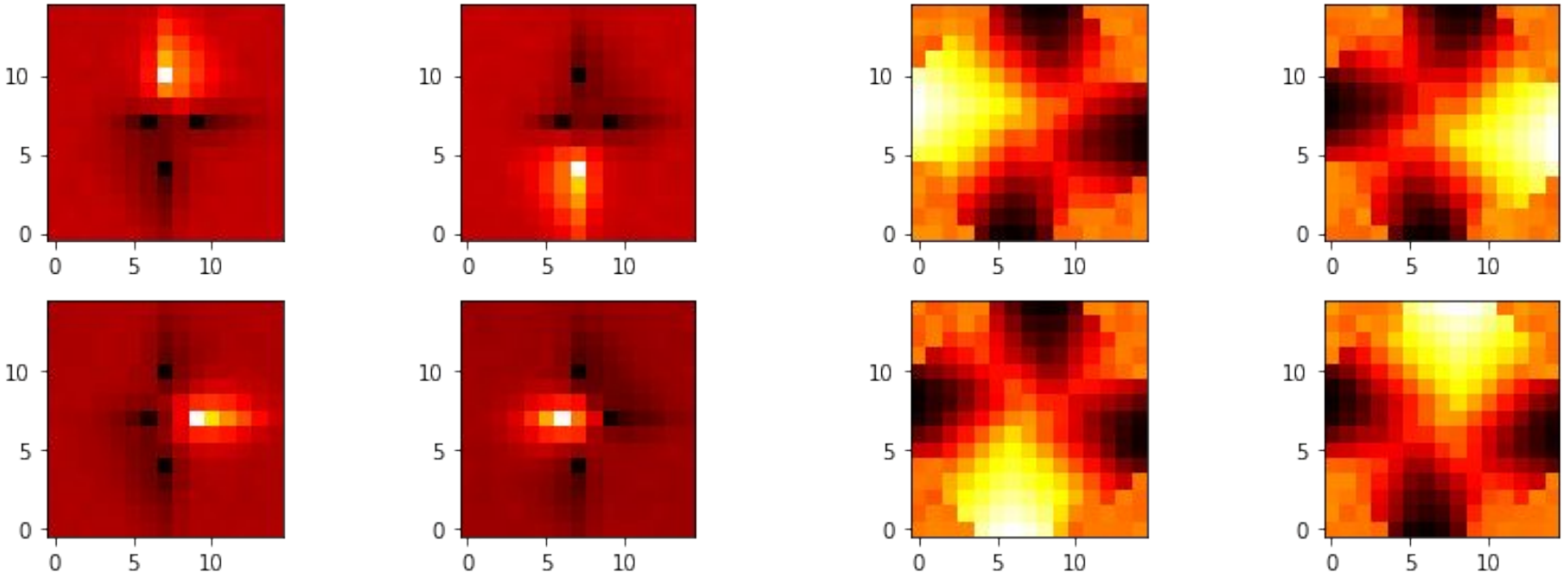
- Learning useful skills in the absence of reward
- Current approaches focus on mutual information maximization[1,2]
- Can other distribution control methods be used?

[1] Variational Intrinsic Control; Gregor et al.; ArXiv, 2016
[2] Diversity is all you need: Learning skills without a reward function; Eysenbach et al.; ArXiv, 2018

Skill learning via Wasserstein distance maximization

Mutual Information maximization

Wasserstein distance maximization[1]



[1] Wasserstein distance maximizing intrinsic control; Durugkar et al.; Deep RL workshop at NeurIPS 2021

Model learning as distribution matching

- Model-based learning learns an environment model using interactions
- Uses model to train agent policy or plan next action(s)
- Generally a simple paradigm for model learning: increase accuracy of per step transitions

Model learning as distribution matching

- GARAT shows effectiveness of minimizing distribution mismatch of transition visitation distribution
- Hypothesis: learning a model to minimize transition visitation distribution mismatch will lead to more efficient model-based learning
- Related ideas: MuZero[1] and the value equivalence principle [2]

[1] Mastering atari, go, chess and shogi by planning with a learned model; Schrittwieser et al.; Nature 2020
[2] The value equivalence principle for model-based reinforcement learning; Grimm et al.; NeurIPS 2020

Challenges

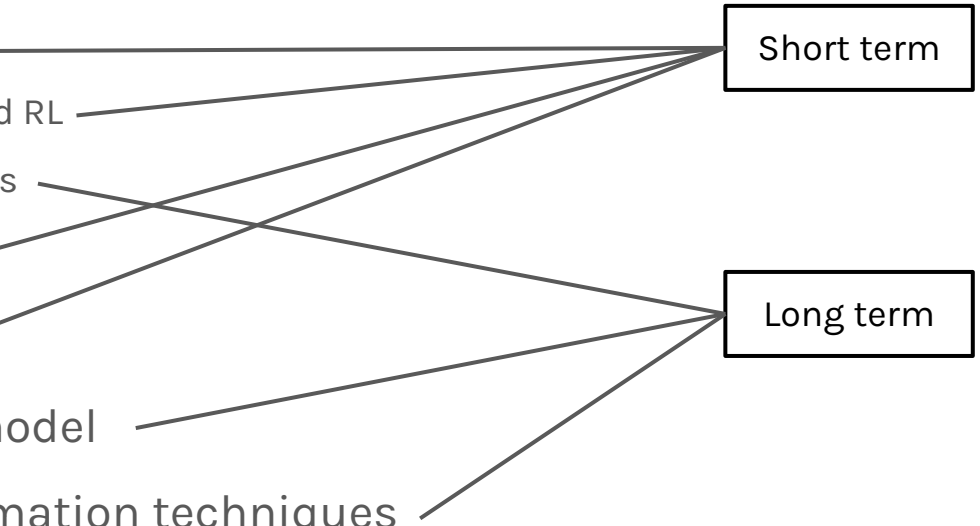
- Model will need to be trained using RL
- Model might overfit to policies used to generate data
- Model convergence will be hard to evaluate since adversarial methods search for saddle points, and not minima
- Adversarial methods can be hard to train with high-dimensional observational spaces like images

Future Work

- Extending AIM:
  - Exploration
  - Beyond goal-conditioned RL
  - General reward functions
- Imitation Learning
- Skill learning
- Learning a dynamics model
- Other distribution estimation techniques

Future Work

- Extending AIM:
  - Exploration
  - Beyond goal-conditioned RL
  - General reward functions
- Imitation Learning
- Skill learning
- Learning a dynamics model
- Other distribution estimation techniques



Distribution Matching for General Reward Functions

- Minimizing distribution mismatch was a viable strategy for goal-conditioned RL
- Can it be used for more general reward functions, or other target distributions?

Imitation Learning

- Matching state-action[1] and state[2] visitation distributions in imitation learning is well-studied
- Can we use distribution matching for just policies?

- Behavioral cloning minimizes mode-covering KL divergence
- Consider minimizing other divergences[3]

[1] Generative adversarial imitation learning; Ho and Ermon; NeurIPS 2016
[2] Generative adversarial imitation from observations; Torabi et al.; I3 workshop at ICML 2019
[3] ABC: adversarial behavioral cloning for offline mode-seeking imitation learning, Offline RL workshop at NeurIPS 2022

Model learning as distribution matching

- GARAT shows effectiveness of minimizing distribution mismatch of transition visitation distribution
- Hypothesis: learning a model to minimize transition visitation distribution mismatch will lead to more efficient model-based learning
- Related ideas: MuZero[1] and the value equivalence principle[2]

[1] Mastering atari, go, chess and shogi by planning with a learned model; Schrittwieser et al.; Nature 2020
[2] The value equivalence principle for model-based reinforcement learning; Grimm et al.; NeurIPS 2020

Other distribution estimation techniques

- The approaches in this thesis have compared two distributions in order to improve effectiveness at the tasks considered.
- Other approaches might open new doors

RL as Minimizing Distribution Mismatch

- AIM (Contribution 4) was able to learn goal-reaching policies
- Extend the idea to more general reward functions

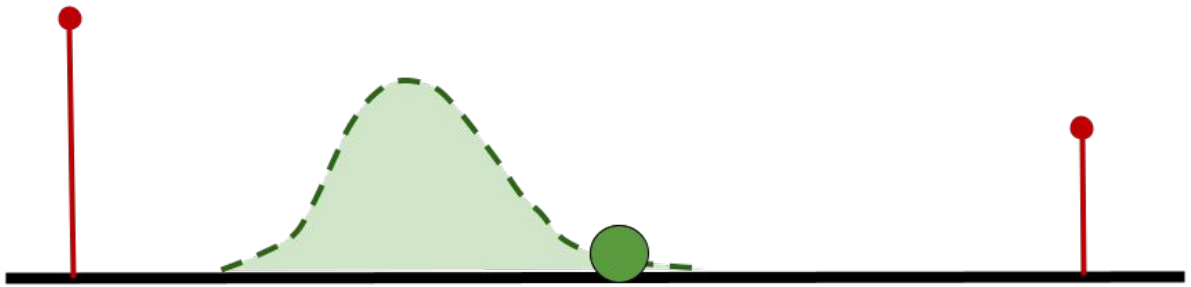
$$R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \longmapsto [R_{min}, R_{max}]$$

Initial Approaches

- Assume full knowledge of reward function
- Set target distribution as proportional to reward or exponential of reward

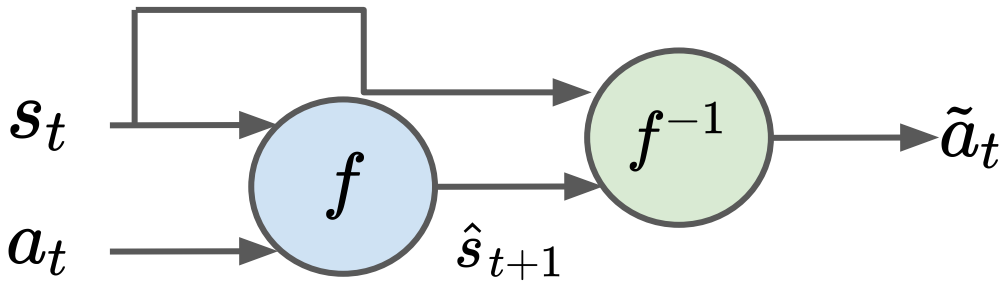
$$\rho_t(s, a, s') \propto R(s, a, s') \qquad \rho_t(s, a, s') \propto e^{R(s,a,s')}$$

- Minimizing Wasserstein distance might not lead to policy that maximizes return



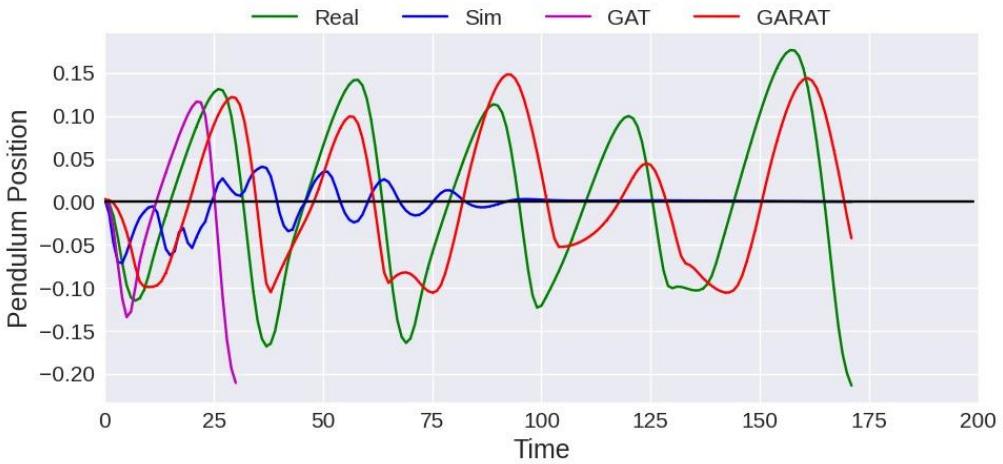
Grounded Action Transformation (GAT)

- GAT learns the action transformation function to minimize the transition mismatch at each step
- Does so by learning a forward model of the real world to predict what the next state would be and an inverse model of the simulator to predict what action needs to be taken

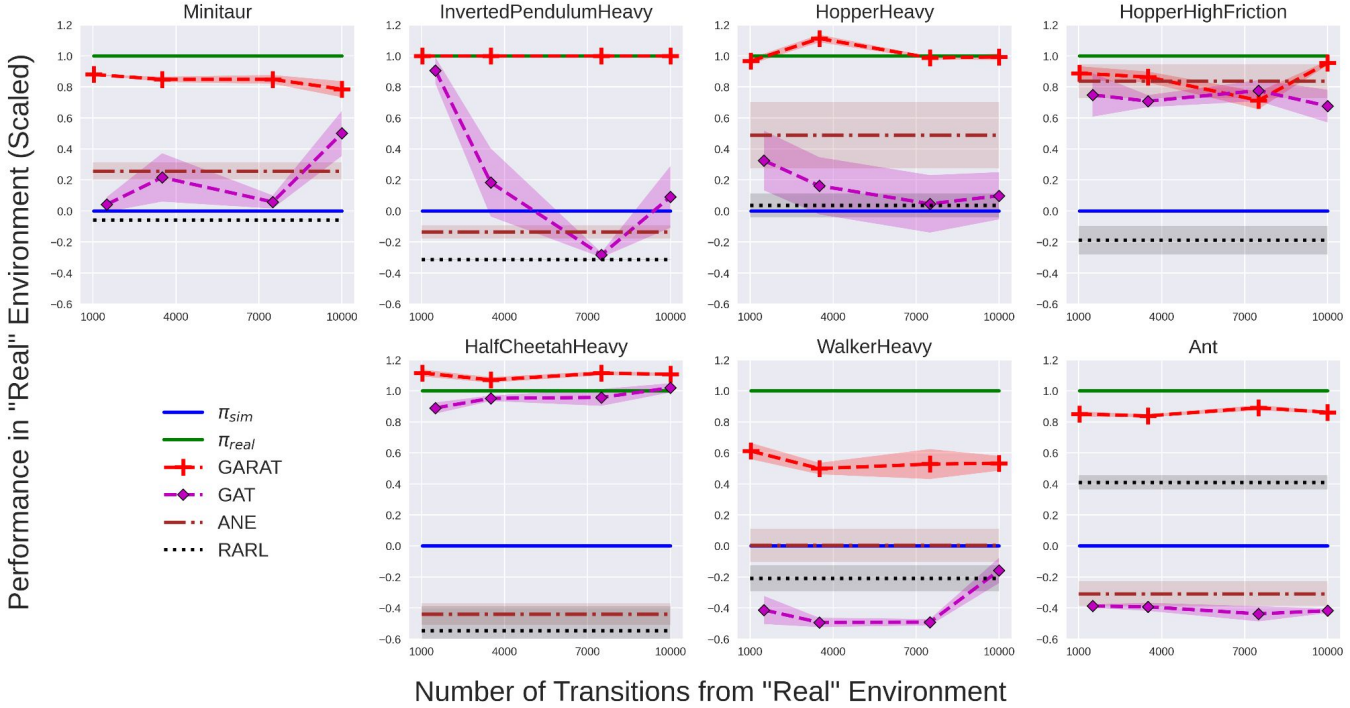


Results - CartPole

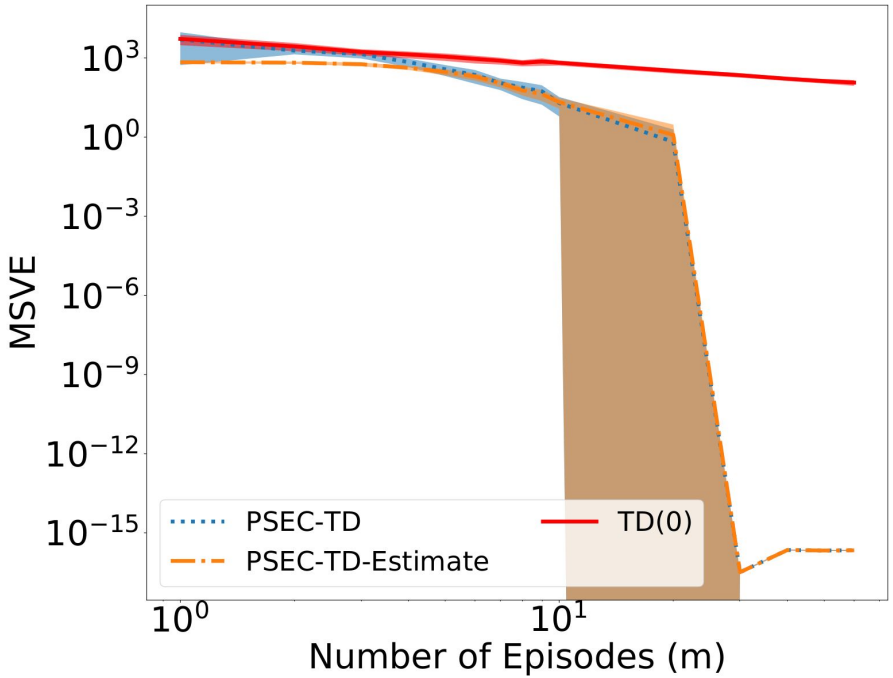
- Qualitative results
- Compares how the different
  in “Sim”, “Real”, and “Sim” g
  same agent policy

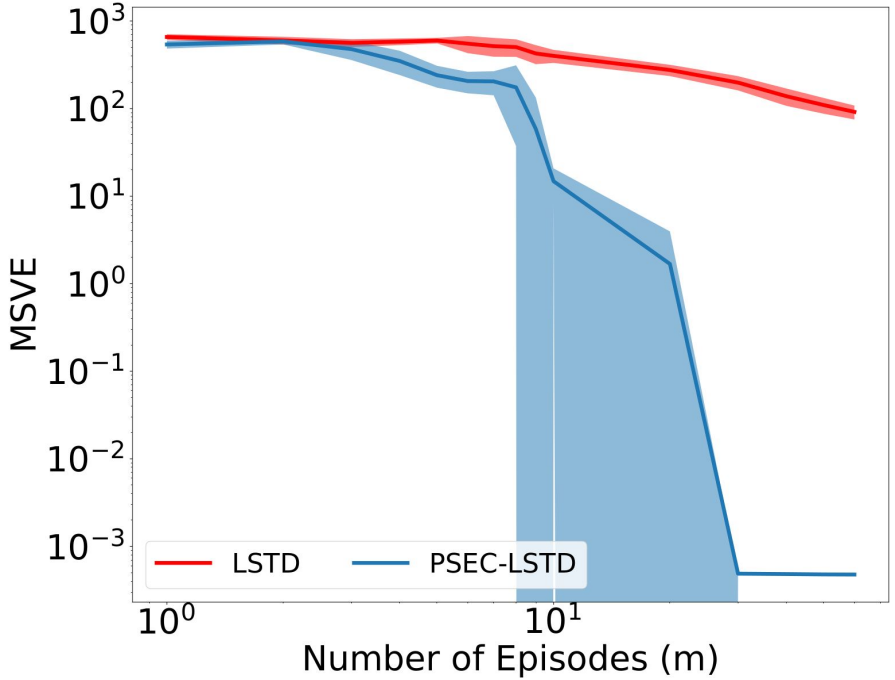


Results - Evaluating Transfer

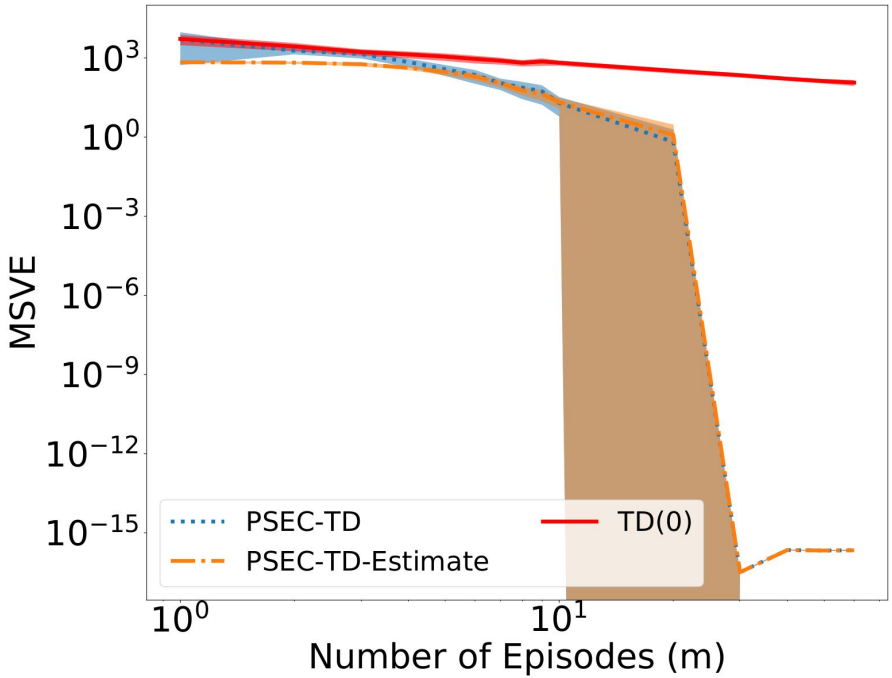


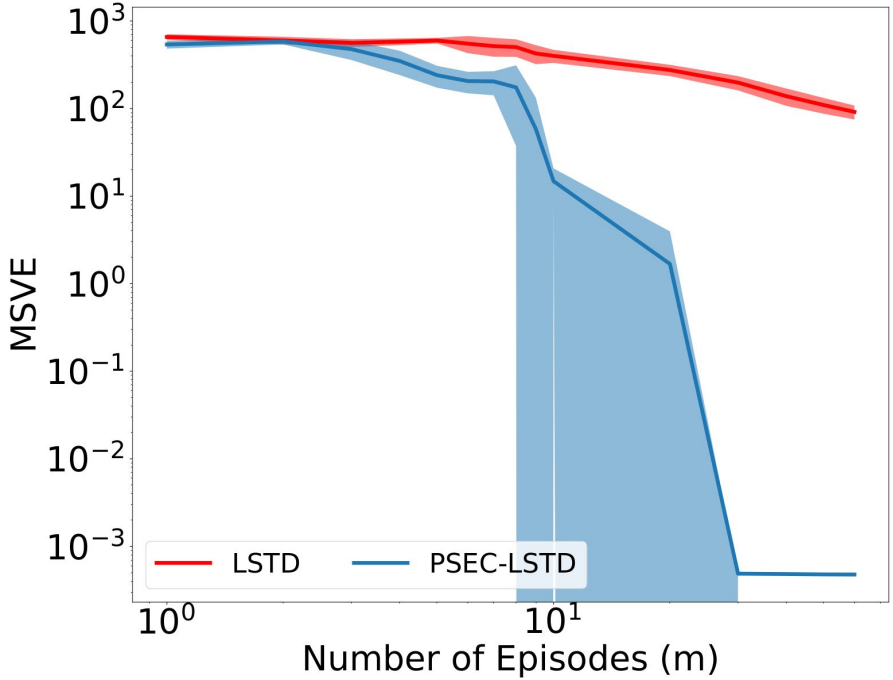
Results - Grid World



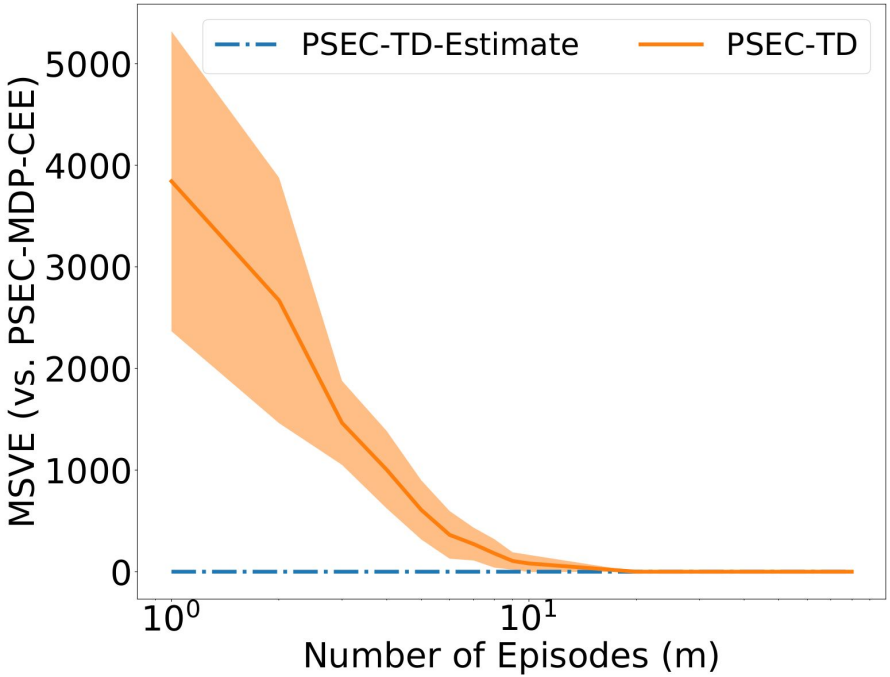


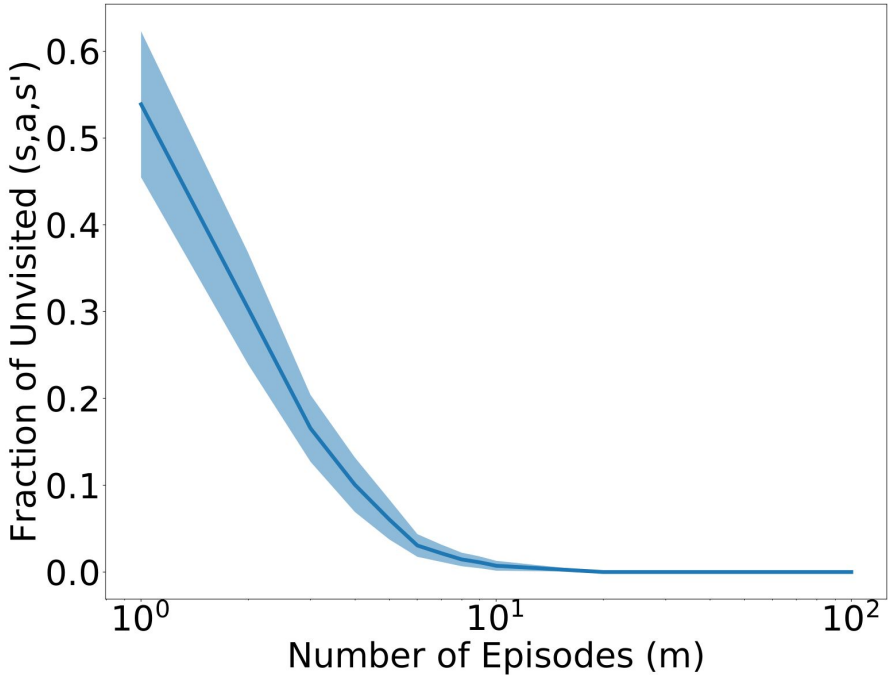
Results - Grid World



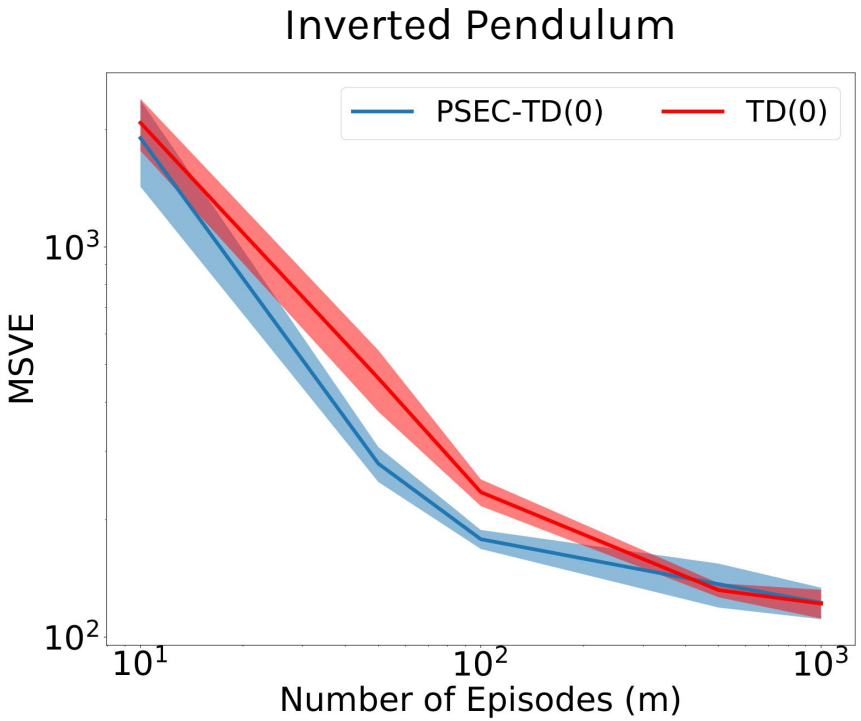


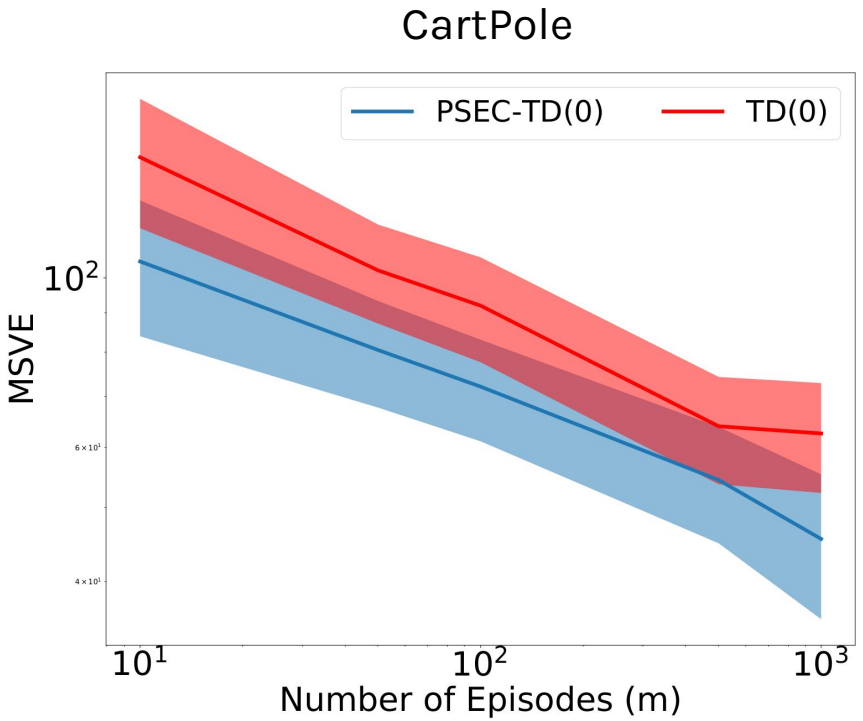
Results - Comparison to state-actions visited





Results with Function Approximation

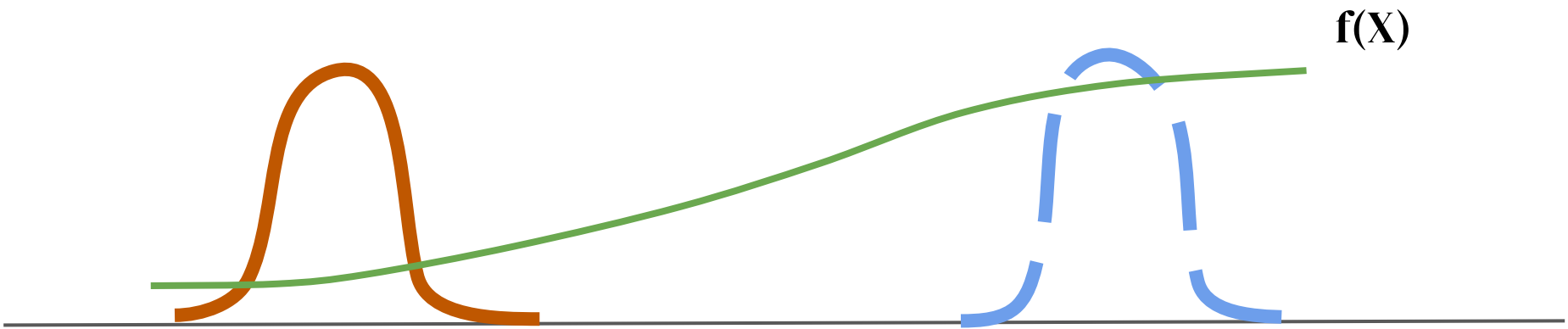




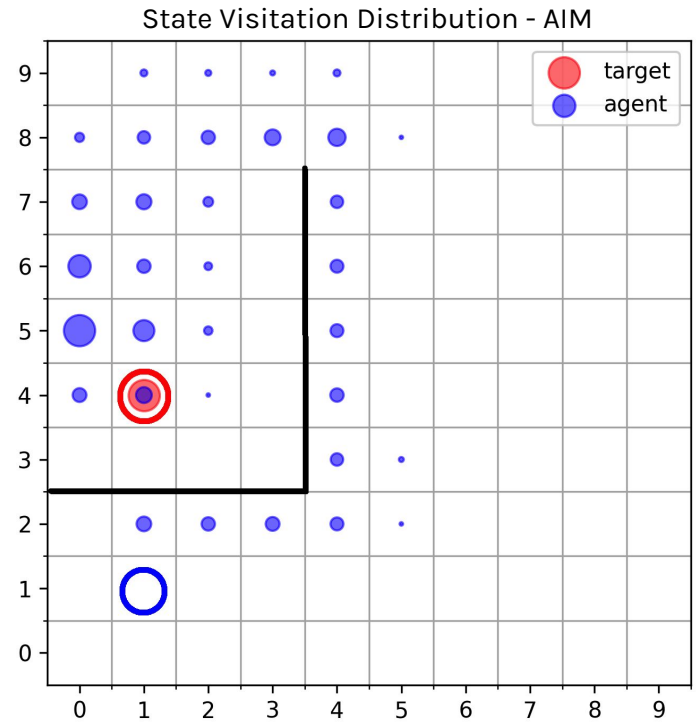
Wasserstein Distance using Kantorovich Duality

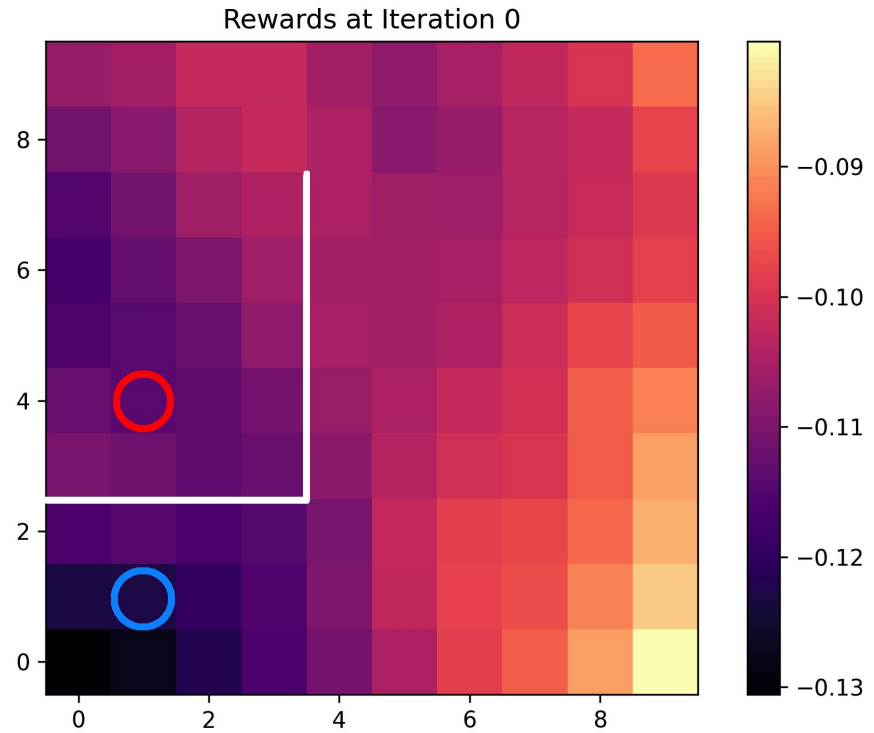
- If estimating Wasserstein-1 distance, the dual form can be used

$$W_d^1(\mu, \nu) = \sup_{\text{Lip}(f) \leq 1} \mathbb{E}_{y \sim \nu}[f(y)] - \mathbb{E}_{x \sim \mu}[f(x)]$$



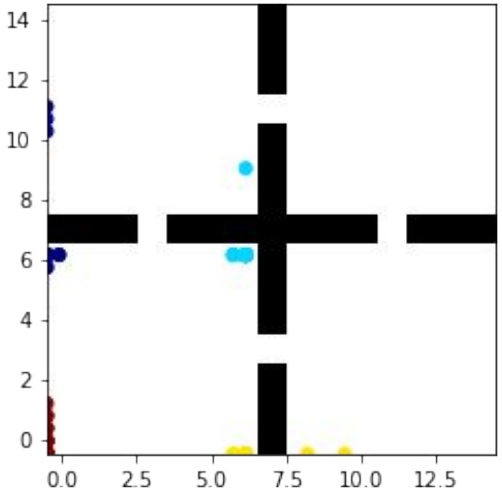
Experiments: Grid World - AIM



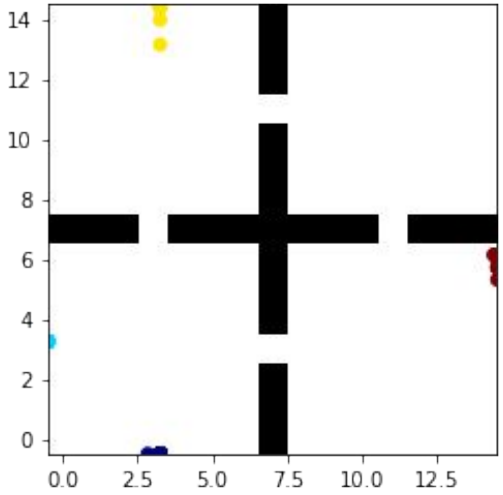


Experiments - Four Rooms

VIC



WIC



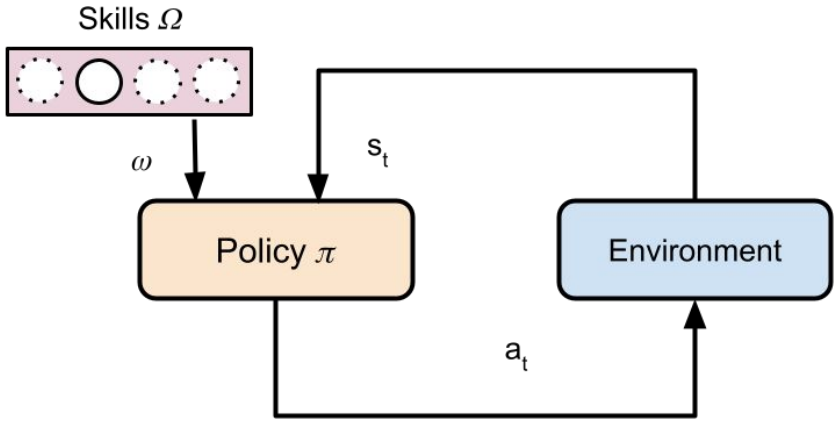
- Continuous state space
- Features are (x, y) coordinates of agent's location
- 5 actions
- Each episode starts at centre of bottom-left room
- 4 skills. Which skill to be followed sampled with uniform probability at the beginning of the episode
- Skill executed for 20 time-steps
- States at the end of the episode are plotted, and colored with different colors for each skill

Maximizing Distribution Mismatch to Learn Skills

- Can Wasserstein distance maximization be useful for RL?
- Estimate Wasserstein distance of skill's state visitation from start state
- Contribution: Train skills to maximize the Wasserstein distance, called Wasserstein distance maximizing intrinsic control (WIC)

Unsupervised RL

- MDP without a reward function
- Agent learns a skill-conditioned policy $\pi_\theta : \mathcal{S} \times \Omega \longmapsto \triangle(\mathcal{A})$
- Skills sampled uniformly at beginning of skill episode



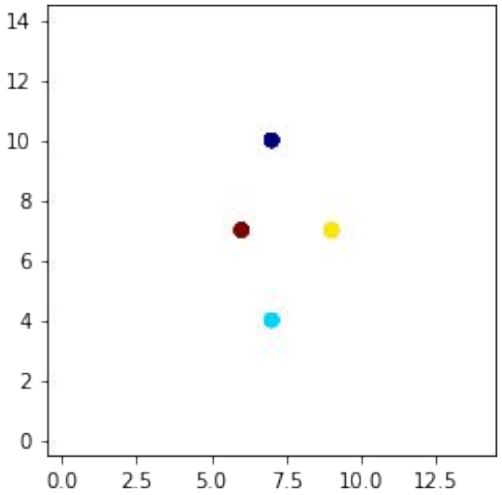
Previous **Approaches**

- Previous approaches focus on mutual information maximization
- From end-points (VIC [1]) or trajectory (DIAYN [2]) try to predict the skill that was executed
- Learn skills that can be distinguished easily
- Do NOT try to learn skills that travel far in the environment

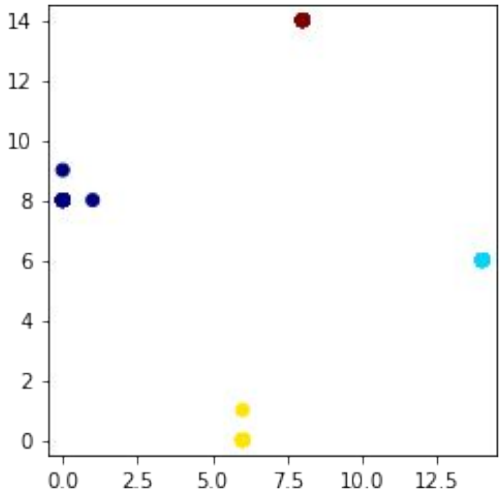
[1] Variational Intrinsic Control; Gregor et al.; ArXiv, 2016
[2] Diversity is all you need: Learning skills without a reward function; Eysenbach et al.; ArXiv, 2018

Experiments - Tabular Grid World

VIC



WIC

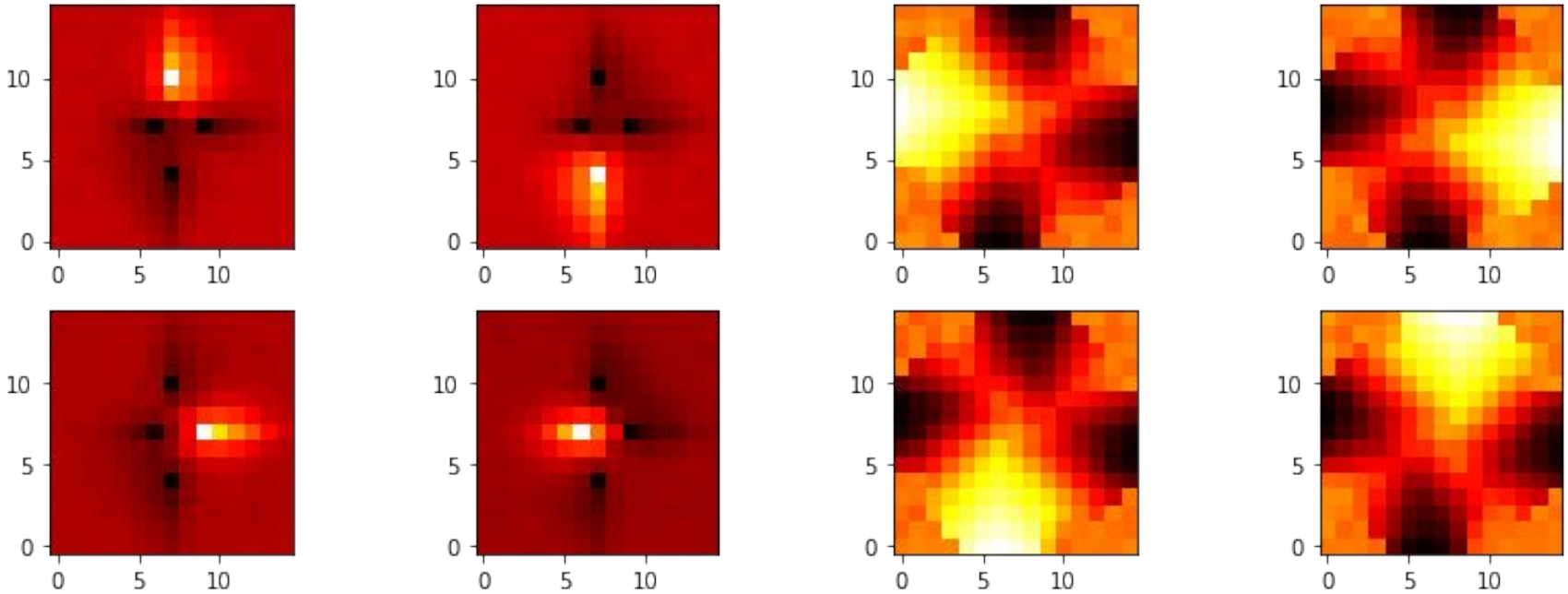


- 15 x 15 tabular grid world
- 5 actions
- Each episode starts at the state in centre
- 4 skills. Which skill to be followed sampled with uniform probability at the beginning of the episode
- Skill executed for 10 time-steps
- States at the end of the episode are plotted, and colored with different colors for each skill

Rewards - Tabular Grid World

VIC

WIC



WIC - Challenges

- How to evaluate quality of skills learned?
  - Use them in downstream task
  - State coverage
  - Some other metric?
- Evaluate on domains with more complex observations. e.g. Atari 2600
- Analyze Wasserstein distance maximizing skills and compare to mutual information maximization objectives

Model learning as distribution matching

- Model-based learning learns an environment model using interactions
- Uses model to train agent policy or plan next action(s)
- Generally a simple paradigm for model learning: increase accuracy of per step transitions

Model learning as distribution matching

- GARAT shows effectiveness of minimizing distribution mismatch of transition visitation distribution
- Hypothesis: learning a model to minimize transition visitation distribution mismatch will lead to more efficient model-based learning
- Related ideas: MuZero[1] and the value equivalence principle [2]

[1] Mastering atari, go, chess and shogi by planning with a learned model; Schrittwieser et al.; Nature 2020
[2] The value equivalence principle for model-based reinforcement learning; Grimm et al.; NeurIPS 2020

Challenges

- Model will need to be trained using RL
- Model might overfit to policies used to generate data
- Model convergence will be hard to evaluate since adversarial methods search for saddle points, and not minima
- Adversarial methods can be hard to train with high-dimensional observational spaces like images

RL as Minimizing Distribution Mismatch

- AIM (Contribution 4) was able to learn goal-reaching policies
- Extend the idea to more general reward functions

$$R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \longmapsto [R_{min}, R_{max}]$$

Initial Approaches

- Assume full knowledge of reward function
- Set target distribution as proportional to reward or exponential of reward

$$\rho_t(s, a, s') \propto R(s, a, s') \qquad \rho_t(s, a, s') \propto e^{R(s,a,s')}$$

- Minimizing Wasserstein distance might not lead to policy that maximizes return