Imitation Learning with Imperfections

Presented by Shengyi Jiang

Jan 19, 2020

Introduction

Why this topic?

- A generalization of my previous two works
- Some (limited) connections with offline learning
- Importance in real-world imitation learning tasks

Introduction

What Imperfections

- in observation space O
- in transition function T
- in expert dataset D_E
- in action space A (?)

And their combinations

- in observation space O
  - Third Person Imitation Learning (ICLR 2017)
  - ~~Cross-Modal Domain Adaptation for Reinforcement Learning (ICLR 2021 submission)~~
  - Visual Imitation with Reinforcement Learning using Recurrent Siamese Networks (ICLR 2021 submission)

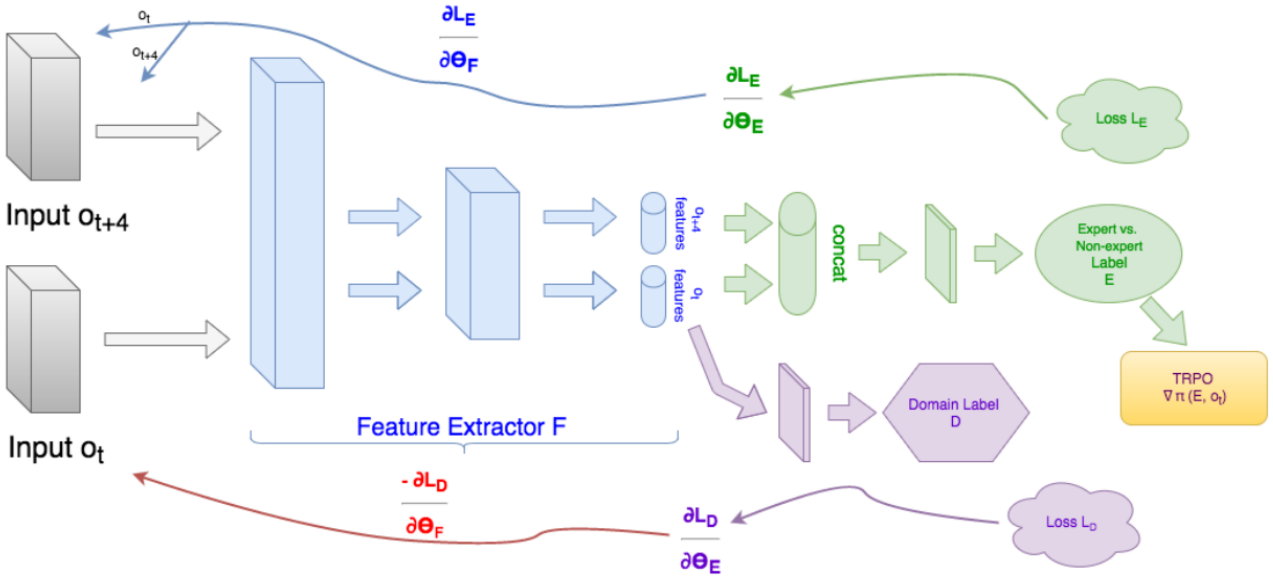
- in transition function T
  - Transfer from Simulation to Real World through Learning Deep Inverse Dynamics Model
  - State-only Imitation with Transition Dynamics Mismatch (ICLR 2020)
  - ~~Offline imitation Learning with a Misspecified Simulator (NeurIPS 2020)~~

- in expert dataset D_E
  - Imitation Learning from Imperfect Demonstration (ICML 2019)
  - Extrapolating Beyond Suboptimal Demonstrations via Inverse Reinforcement Learning from Observations (ICML 2019)
  - Inverse Reinforcement Learning from Failure (AAMAS 2016)
  - Reinforcement Learning from Imperfect Demonstrations under Soft Expert Guidance (AAAI 2019)
  - Behavioral Cloning from Noisy Demonstrations (ICLR 2021 submission)
- in action space A (?)
  - Generalization to New Actions in Reinforcement Learning (ICML 2020)
- combinations
  - Domain Adaptive Imitation Learning (ICML 2020)

Third Person Imitation Learning (ICLR 2017)

$$\max_{\pi_\theta} \min \mathcal{L}_R = \sum_i CE(\mathcal{D}_R(\mathcal{D}_F(o_i)), c_{\ell_i})$$

$$\text{s.t. } \mathrm{MI}(D_F(o_i); d_l) = 0$$



Algorithm 1 A third-person imitation learning algorithm.

Let CE be the standard cross entropy loss.
Let $\mathcal{G}$ be a function that flips the gradient sign during backpropogation and acts as the identity map otherwise.
Initialize two domains, E and N for the expert and novice.
Initialize a memory bank Ω of expert success and of failure in domain E. Each trajectory $\omega \in \Omega$ comprises a rollout of images $o = o_1, \ldots, o_t, \ldots o_n$, a class label c_ℓ, and a domain label d_ℓ.
Initialize $\mathcal{D} = \mathcal{D}_F, \mathcal{D}_R, \mathcal{D}_D$, a domain invariant discriminator.
Initialize a novice policy π_θ.
Initialize numiters, the number of inner policy optimization iterations we wish to run.
for iter in numiters **do**
  Sample a set of successes and failures ω_E from Ω.
  Collect on policy samples ω_N
  Set $\omega = \omega_E \cup \omega_N$.
  Shuffle ω
  for o, c_ℓ, d_ℓ in ω **do**
    for o_t in o **do**
      $\sigma_t = \mathcal{D}_F(o_t)$
      $\sigma_{t+4} = \mathcal{D}_F(o_{t+4})$
      $\mathcal{L}_R = CE(\mathcal{D}_R(\sigma_t, \sigma_{t+4}), c_\ell)$
      $\mathcal{L}_d = CE(\mathcal{D}_D(\mathcal{G}(\sigma_t)), d_\ell)$
      $\mathcal{L} = \lambda \cdot \mathcal{L}_d + \mathcal{L}_R$ [1]
      minimize $\mathcal{L}$ with ADAM.
    end for
  end for
  Collect on policy samples ω_N from π_θ.
  for ω in ω_N **do**
    for ω_t in ω **do**
      $\sigma_t = \mathcal{D}_F(o_t)$
      $\sigma_{t+4} = \mathcal{D}_F(o_{t+4})$
      $\hat{c}_\ell = \mathcal{D}_R(\sigma_t, \sigma_{t+4})$
      $r = \hat{c}_\ell[0]$, the probability that o_t, o_{t+4} were generated via expert rollouts.
      Use r to train π_θ with via policy gradients (TRPO).
    end for
  end for
end for
return optimized policy π_θ

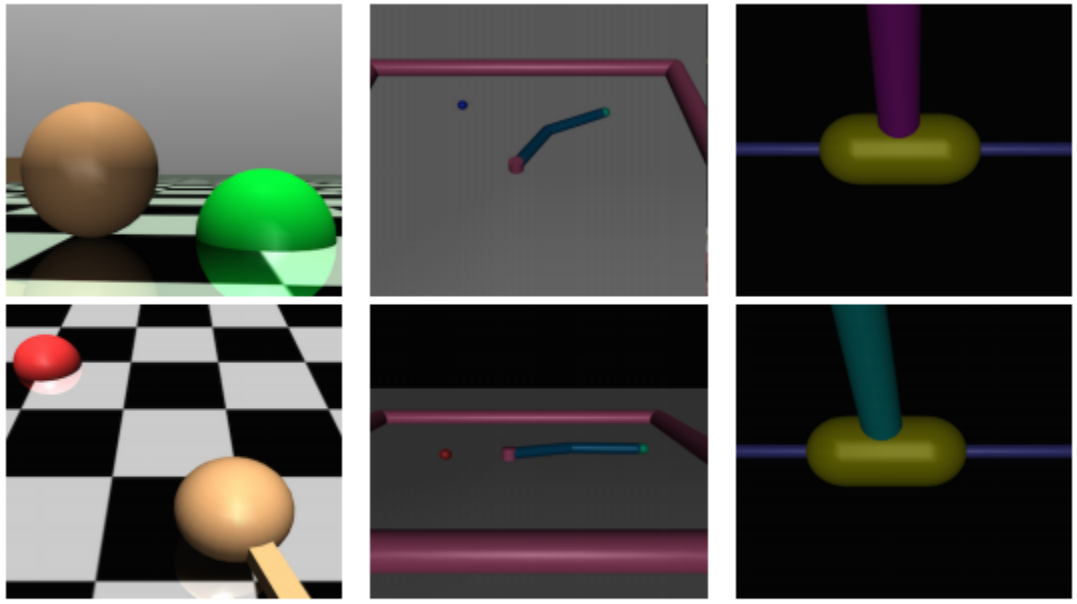
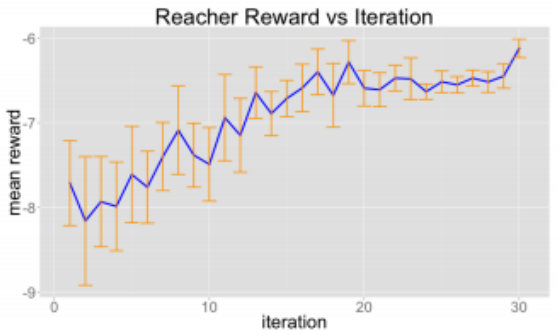
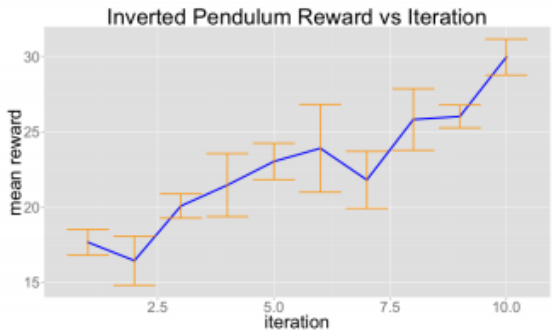
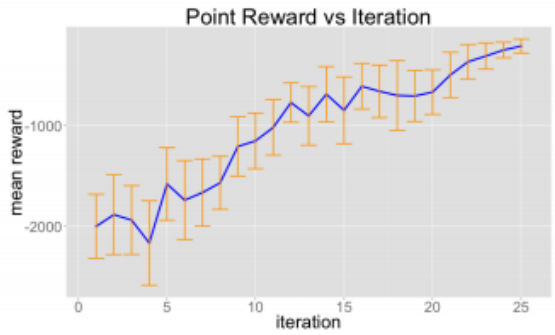
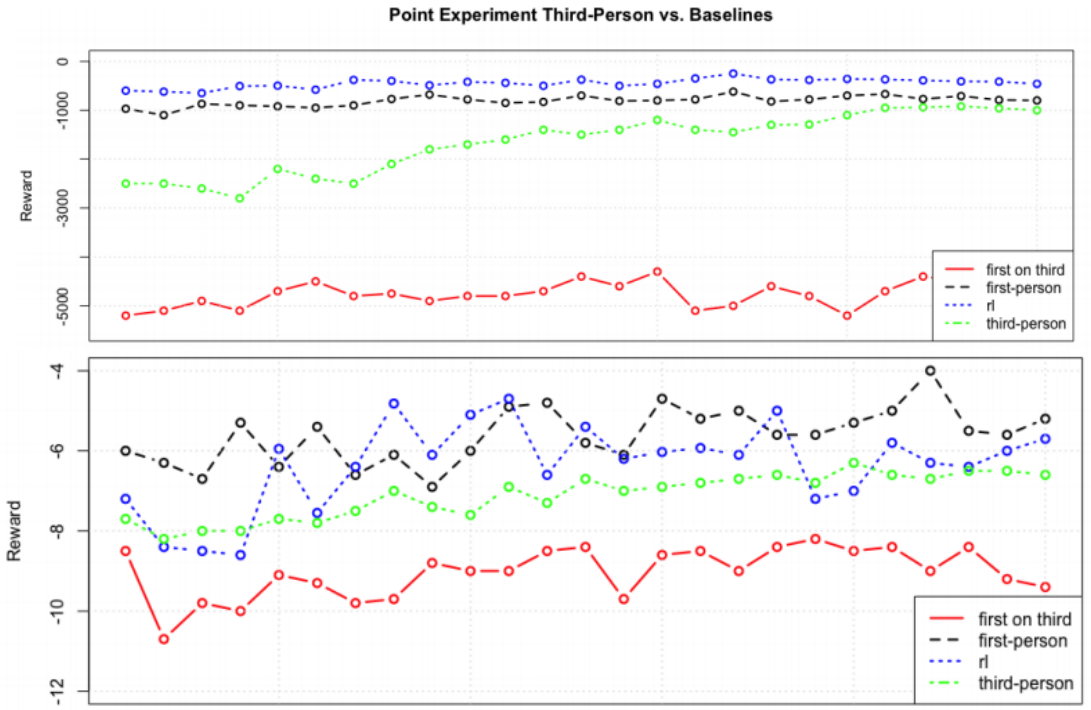


Figure 1: From left to right, the three domains we consider in this paper: pointmass, reacher, and pendulum. Top-row is the third-person view of a teacher demonstration. Bottom row is the agent's view in their version of the environment. For the point and reacher environments, the camera angles differ by approximately 40 degrees. For the pendulum environment, the color of the pole differs.

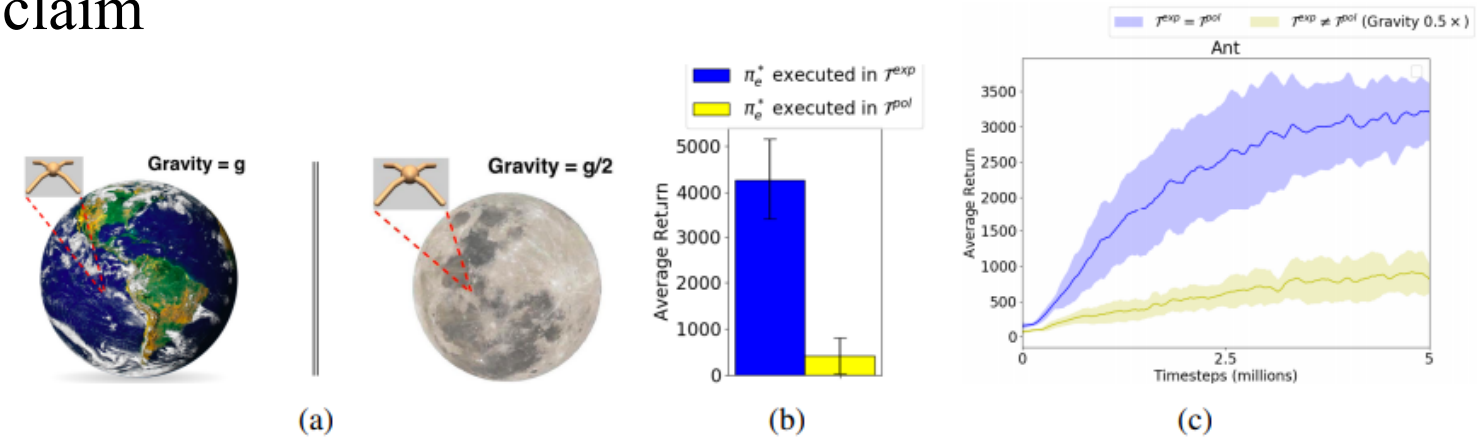

Reacher Reward vs Iteration
mean reward
iteration


Inverted Pendulum Reward vs Iteration
mean reward
iteration


Point Reward vs Iteration
mean reward
iteration


Point Experiment Third-Person vs. Baselines
Reward
first on third
first-person
rl
third-person
Reward
first on third
first-person
rl
third-person

Its claim



Actually

How to do AIRL without expert action

AIRL

$$\min_{\theta} \max_{\omega} \mathbb{E}_{(s,a)\sim\rho_E}\left[\log D_\omega(s,a)\right] + \mathbb{E}_{(s,a)\sim\pi_\theta}\left[\log(1 - D_\omega(s,a))\right] - \lambda\mathcal{H}(\pi_\theta)$$

$$D_\omega(s,a) = \frac{e^{f_\omega(s,a)}}{e^{f_\omega(s,a)} + \pi_\theta(a|s)}$$

Two-stage optimization:

surrogate expert buffer: $D_s = \{(s,a)_i\}$

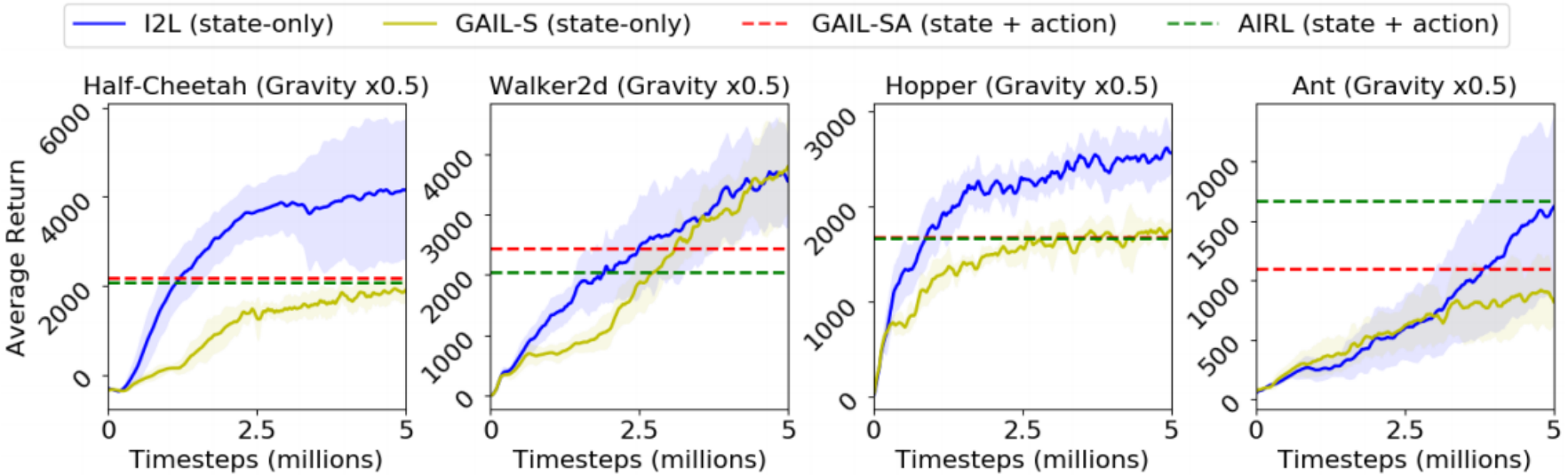
$$W_1(\rho^*(s), \tilde{\rho}(s)) = \sup_{\|g_\phi\|_L \leq 1} \mathbb{E}_{s\sim\rho^*}[g_\phi(s)] - \mathbb{E}_{s\sim\tilde{\rho}}[g_\phi(s)]$$

perform AIRL on D_S

$$\max_{\omega} \mathbb{E}_{(s,a)\sim\mathcal{B}}\left[\log \frac{e^{f_\omega(s,a)}}{e^{f_\omega(s,a)} + \pi_\theta(a|s)}\right] + \mathbb{E}_{(s,a)\sim\pi_\theta}\left[\log \frac{\pi_\theta(a|s)}{e^{f_\omega(s,a)} + \pi_\theta(a|s)}\right]$$

Algorithm 1: Indirect Imitation Learning (I2L)

Networks: Policy (θ), Discriminator (ω), Wasserstein critic (ϕ)
$\mathcal{B} \leftarrow$ empty buffer
$\tau^*_{\text{states}} := \{s_0, s_1, \ldots, s_T\}$ `/* State-only expert demonstration */`
for *each iteration* **do**
    Run π_θ in environment and collect few trajectories τ
    Update Wasserstein critic ϕ using $\mathcal{B}$ and τ^*_{states} `/* Equation 7 */`
    Obtain trajectory score $\frac{1}{|\tau|}\sum_{s\in\tau} g_\phi(s)$ for each τ using ϕ
    Add τ to $\mathcal{B}$ with the priority-based protocol, using the score as priority
    Update the AIRL discriminator ω using τ and $\mathcal{B}$ `/* Equation 6 */`
    Update policy θ with PPO using $\log D_\omega - \log(1 - D_\omega)$ as rewards
end


I2L (state-only)
GAIL-S (state-only)
GAIL-SA (state + action)
AIRL (state + action)
Half-Cheetah (Gravity x0.5)
Walker2d (Gravity x0.5)
Hopper (Gravity x0.5)
Ant (Gravity x0.5)
Average Return
Timesteps (millions)
0
2.5
5
0
2000
4000
6000
1000
3000
500
1500

Imitation Learning from Imperfect Demonstration (ICML 2019)

$$p(x) = \alpha p_{\text{opt}}(x) + \sum_{i=1}^{n} \nu_i p_i(x)$$
$$= \alpha p_{\text{opt}}(x) + (1-\alpha) p_{\text{non}}(x),$$

confidence scores $r(x) \triangleq p(y = +1|x)$

$$\mathcal{D}_{\text{c}} \triangleq \{(x_{\text{c},i}, r_i)\}_{i=1}^{n_{\text{c}}} \overset{\text{i.i.d.}}{\sim} q(x, r),$$
$$\mathcal{D}_{\text{u}} \triangleq \{x_{\text{u},i}\}_{i=1}^{n_{\text{u}}} \overset{\text{i.i.d.}}{\sim} p(x),$$

Algorithm 1 2IWIL

Input: Expert trajectories and confidence $\mathcal{D}_{\mathrm{c}} = \{(x_{\mathrm{c},i}, r_i)\}_{i=1}^{n_{\mathrm{c}}}$, $\mathcal{D}_{\mathrm{u}} = \{x_{\mathrm{u},i}\}_{i=1}^{n_{\mathrm{u}}}$
Estimate the class prior by $\widehat{\alpha} = \frac{1}{n_{\mathrm{c}}} \sum_{i=1}^{n_{\mathrm{c}}} r_i$
Train a probabilistic classifier by minimizing Eq. (7) with $\beta = \frac{n_u}{n_u + n_c}$
Predict confidence scores $\{\widehat{r}_{\mathrm{u},i}\}_{i=1}^{n_{\mathrm{u}}}$ for $\{x_{\mathrm{u},i}\}_{i=1}^{n_{\mathrm{u}}}$
for $i = 0, 1, 2, \ldots$ **do**
    Sample trajectories $\{x_i\}_{i=1}^{n_a} \sim \pi_\theta$
    Update the discriminator parameters by maximizing Eq. (5)
    Update π_θ with reward $-\log D_w(x)$ using TRPO
end for

$$\min_{\theta} \max_{w} \mathbb{E}_{x \sim p_\theta} \left[\log D_w(x)\right] + \mathbb{E}_{x,r \sim q} \left[\frac{r}{\alpha} \log(1 - D_w(x))\right]. \qquad (5)$$

Algorithm 2 IC-GAIL

Input: Expert trajectories, confidence, and weight threshold $\{x_{\mathrm{u},i}\}_{i=1}^{n_\mathrm{u}}, \{(x_{\mathrm{c},i}, r_i)\}_{i=1}^{n_\mathrm{c}}, \tau$
Estimate the class prior by $\widehat{\alpha} = \frac{1}{n_\mathrm{c}} \sum_{i=1}^{n_\mathrm{c}} r_i$
$\lambda = \max\{\tau, \widehat{\alpha}\}$
for $i = 0, 1, 2, \ldots$ **do**
  Sample trajectories $\{x_i\}_{i=1}^{n_a} \sim \pi_\theta$
  Update the discriminator parameters by maximizing Eq. (10)
  Update π_θ with reward $-\log D_w(x)$ using TRPO
end for

$$p = \alpha p_{\mathrm{opt}} + (1-\alpha) p_{\mathrm{non}}$$

$$p' = \alpha p_\theta + (1-\alpha) p_{\mathrm{non}}$$

$$\min_\theta \max_w \mathbb{E}_{x\sim p}[\log(1 - D_w(x))] + \lambda \mathbb{E}_{x\sim p_\theta}[\log D_w(x)] + (1-\lambda)\mathbb{E}_{x,r\sim q}\left[\frac{(1-r)}{(1-\alpha)} \log D_w(x)\right], \quad (10)$$

Theorem 4.1. *The classification risk (6) can be equivalently expressed as*

$$R_{\mathrm{SC},\ell}(g) = \mathbb{E}_{x,r\sim q}[r(\ell(g(x)) - \ell(-g(x))) + (1-\beta)\ell(-g(x))] + \mathbb{E}_{x\sim p}[\beta\ell(-g(x))], \tag{7}$$

Theorem 4.3. *Let $\mathcal{G}$ be the hypothesis class we use. Assume that the loss function ℓ is ρ_ℓ-Lipschitz continuous, and that there exists a constant $C_\ell > 0$ such that $\sup_{x\in\mathcal{X},y\in\{\pm1\}} |\ell(yg(x))| \leq C_\ell$ for any $g \in \mathcal{G}$. Let $\widehat{g} \triangleq \arg\min_{g\in\mathcal{G}} \widehat{R}_{\mathrm{SC},\ell}(g)$ and $g^* \triangleq \arg\min_{g\in\mathcal{G}} R_{\mathrm{SC},\ell}(g)$. For $\delta \in (0,1)$, with probability at least $1-\delta$ over repeated sampling of data for training $\widehat{g}$,*

$$\begin{aligned} R_{\mathrm{SC},\ell}(\widehat{g}) - R_{\mathrm{SC},\ell}(g^*) &\leq 16\rho_\ell((3-\beta)\mathfrak{R}_{n_\mathrm{c}}(\mathcal{G}) + \beta\mathfrak{R}_{n_\mathrm{u}}(\mathcal{G})) \\ &\quad + 4C_\ell\sqrt{\frac{\log(8/\delta)}{2}}\left((3-\beta)n_\mathrm{c}^{-\frac{1}{2}} + \beta n_\mathrm{u}^{-\frac{1}{2}}\right). \end{aligned}$$

Theorem 4.4. *Denote that*

$$V(\pi_\theta, D_w) = \mathbb{E}_{x\sim p}[\log(1 - D_w(x))] + \mathbb{E}_{x\sim p'}[\log D_w(x)],$$

and that $C(\pi_\theta) = \max_w V(\pi_\theta, D_w)$. *Then,* $V(\pi_\theta, D_w)$ *is maximized when* $D_w = \frac{p'}{p+p'} (\triangleq D_{w^*})$, *and its maximum value is* $C(\pi_\theta) = -\log 4 + 2\mathrm{JSD}(p\|p')$. *Thus,* $C(\pi_\theta)$ *is minimized if and only if* $p_\theta = p_{\mathrm{opt}}$ *almost everywhere.*

Theorem 4.4 implies that the optimal policy can be found by solving the following objective,

$$\min_\theta \max_w \mathbb{E}_{x\sim p}[\log(1 - D_w(x))] + \mathbb{E}_{x\sim p'}[\log D_w(x)].$$

Theorem 4.5. $V(\pi_\theta, D_w)$ *can be transformed to* $\widetilde{V}(\pi_\theta, D_w)$, *which is defined as follows:*

$$\widetilde{V}(\pi_\theta, D_w) = \mathbb{E}_{x\sim p}[\log(1 - D_w(x))] + \alpha\mathbb{E}_{x\sim p_\theta}[\log D_w(x)] + \mathbb{E}_{x,r\sim q}[(1 - r)\log D_w(x)].$$

Theorem 4.6. *Let* $\mathcal{W}$ *be a parameter space for training the discriminator and* $D_{\mathcal{W}} \triangleq \{D_w \mid w \in \mathcal{W}\}$ *be its hypothesis space. Assume that there exist a constant* $C_L > 0$ *such that* $|\log D_w(x)| \le C_L$ *and* $|\log(1 - D_w(x))| \le C_L$ *for any* $x \in \mathcal{X}$ *and* $w \in \mathcal{W}$. *Assume that both* $\log D_w(x)$ *and* $\log(1 - D_w(x))$ *for any* $w \in \mathcal{W}$ *have Lipschitz norms no more than* $\rho_L > 0$. *For a fixed agent policy* π_θ, *let* $\{x_{\mathrm{a},i}\}_{i=1}^{n_a}$ *be a sample generated from* π_θ, $D_{\widehat{w}} \triangleq \arg\max_{w\in\mathcal{W}} \widehat{V}(\pi_\theta, D_w)$, *and* $D_{w^*} \triangleq \arg\max_{w\in\mathcal{W}} V(\pi_\theta, D_w)$. *Then, for* $\delta \in (0, 1)$, *the following holds with probability at least* $1 - \delta$:

$$\begin{aligned} &V(\pi_\theta, D_{w^*}) - V(\pi_\theta, D_{\widehat{w}}) \\ &\le 16\rho_L(\mathfrak{R}_{n_{\mathrm{u}}}(D_{\mathcal{W}}) + \alpha\mathfrak{R}_{n_{\mathrm{a}}}(D_{\mathcal{W}}) + \mathfrak{R}_{n_{\mathrm{c}}}(D_{\mathcal{W}})) \\ &\quad + 4C_L\sqrt{\frac{\log(6/\delta)}{2}}\left(n_{\mathrm{u}}^{-\frac{1}{2}} + \alpha n_{\mathrm{a}}^{-\frac{1}{2}} + n_{\mathrm{c}}^{-\frac{1}{2}}\right). \end{aligned}$$

Behavioral Cloning from Noisy Demonstrations (ICLR 2021 submission)

Assumption 1. *The T-step expected reward of π_e^* satisfies $\mathcal{J}(\pi, R) \leq \mathcal{J}(\pi_e^*, R)$, $\mathcal{J}_\beta(\pi, R) \leq \mathcal{J}_\beta(\pi_e^*, R)$, and $\mathcal{J}_\beta(\pi_e^*, R) \leq \mathcal{J}(\pi_e^*, R)$ for any non-optimal policies $\pi, \beta \in \Pi \setminus \{\pi_e^*\}$.*

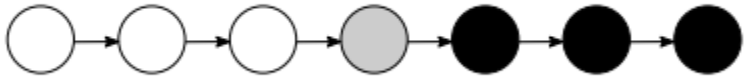
Assumption 2. *With small probability ϵ, which we call non-optimal probability, the policies π_e the noisy experts follow during demonstrations are sampled at each time step as $\pi_e = \pi \sim p_\Pi$ if $\epsilon \geq z \sim \mathcal{U}(0, 1)$, otherwise $\pi_e = \pi_e^*$, where p_Π is an unknown distribution over the set of policies, z is a random variable, and $\mathcal{U}(0, 1)$ is a uniform distribution with range $[0, 1]$.*

Assumption 3. *The reward $R_t^{\pi_e}$ is at least zero if the noisy expert has followed a policy $\pi \in \Pi \setminus \{\pi_e^*\}$ once or more so far, otherwise $R_t^{\pi_e} = \mathbb{E}_{s \sim d_t^{\pi_e}} \left[\epsilon \mathbb{E}_{\pi \sim p_\Pi}[R^\pi(s)] + (1 - \epsilon) R^{\pi_e^*}(s)\right]$.*

Assumption 4. *The sequence $\{R_1^{\pi_e}, ..., R_T^{\pi_e}\}$ has monotonically decreasing property $R_t^{\pi_e} \geq R_{t+1}^{\pi_e}$.*

Corollary 1. *If the Assumptions 1 - 4 hold and the policy π_θ has a probability of non-optimal behavior $\hat{\epsilon} = \epsilon + \zeta$, the following bound holds:*

$$\mathcal{J}(\pi_\theta, R) \geq \left\{ \sum_{t=0}^{T-1} (1 - \hat{\epsilon})^t \right\} \cdot \mathcal{J}_{\pi_e}(\pi_\theta, R) \tag{2}$$



$$
\begin{aligned}
\mathbb{E}_{s\sim d^{\pi_e}, a\sim\pi_e(\cdot|s)}[\log \pi_\theta(a|s)] &\leq -\alpha\Omega_e(\theta) - \beta\Omega_*(\theta) - \gamma\Omega_{e+*}(\theta) \\
\Omega_e(\theta) &= \mathbb{E}_{s\sim d_e^{\pi_e}}\left[D_{KL}[\pi_e(\cdot|s)||\pi_\theta(\cdot|s)]\right] \\
\Omega_*(\theta) &= \mathbb{E}_{s\sim d_*^{\pi_e}}\left[D_{KL}[\pi_e^*(\cdot|s)||\pi_\theta(\cdot|s)]\right] \\
\Omega_{e+*}(\theta) &= \epsilon\mathbb{E}_{s\sim d_*^{\pi_{e+*}}, \pi\sim p_\Pi}[D_{KL}[\pi(\cdot|s)||\pi_\theta(\cdot|s)]] \\
&+ (1-\epsilon)\mathbb{E}_{s\sim d_*^{\pi_{e+*}}}\left[D_{KL}[\pi_e^*(\cdot|s)||\pi_\theta(\cdot|s)]\right]
\end{aligned}
$$

Algorithm 1 Overview of Our Algorithm

Given the expert demonstrations $\mathcal{D}$.
Set $\hat{R}(s,a) = 1$ for $\forall (s,a) \in \mathcal{D}$.
for iteration = $1, M$ **do**
  for $k = 1, K$ **do**
    Initialize parameters θ^k.
    Optimize θ^k with the objective (1).
  end for
  Set $\hat{R}(s,a) = \eta \bar{\pi}_\theta(a|s)$ for $\forall (s,a) \in \mathcal{D}$.
end for

$$\arg\max_{\theta} \mathbb{E}_{s\sim d^{\pi_e}, a\sim\pi_e(\cdot|s)}[\log \pi_\theta(a|s)\hat{R}(s,a)]$$

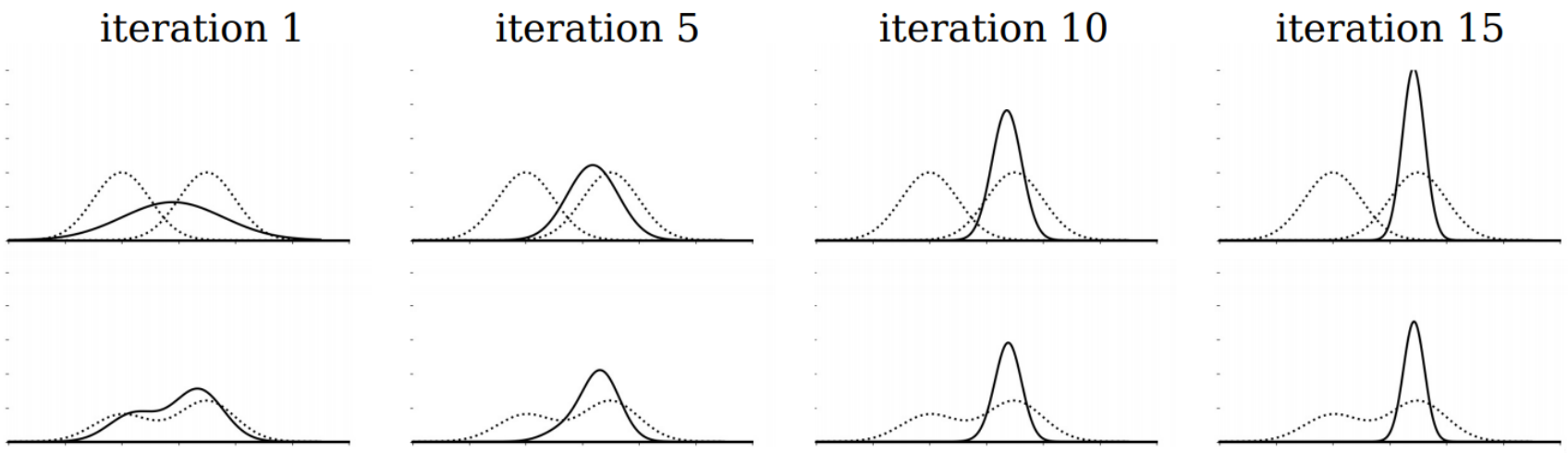


Figure 2: A toy example of the weighted action sampling procedure given a state $s \in S_*^{\pi_e}$ in our algorithm. On both rows, the horizontal line is the action domain. On the top row, the left dotted lines, the right dotted lines, and the solid line describe $\pi \in \Pi \setminus \{\pi_e^*\}$, $\pi_e^*(a|s)$, and $\pi_\theta(a|s)$, respectively. On the bottom row, the dotted lines and the solid line describe the mixture distribution $\epsilon\pi(a|s)+(1-\epsilon)\pi_e^*(a|s)$ with $\epsilon = 0.4$ and the mixture distribution weighted by $\pi_\theta(a|s)$, respectively.

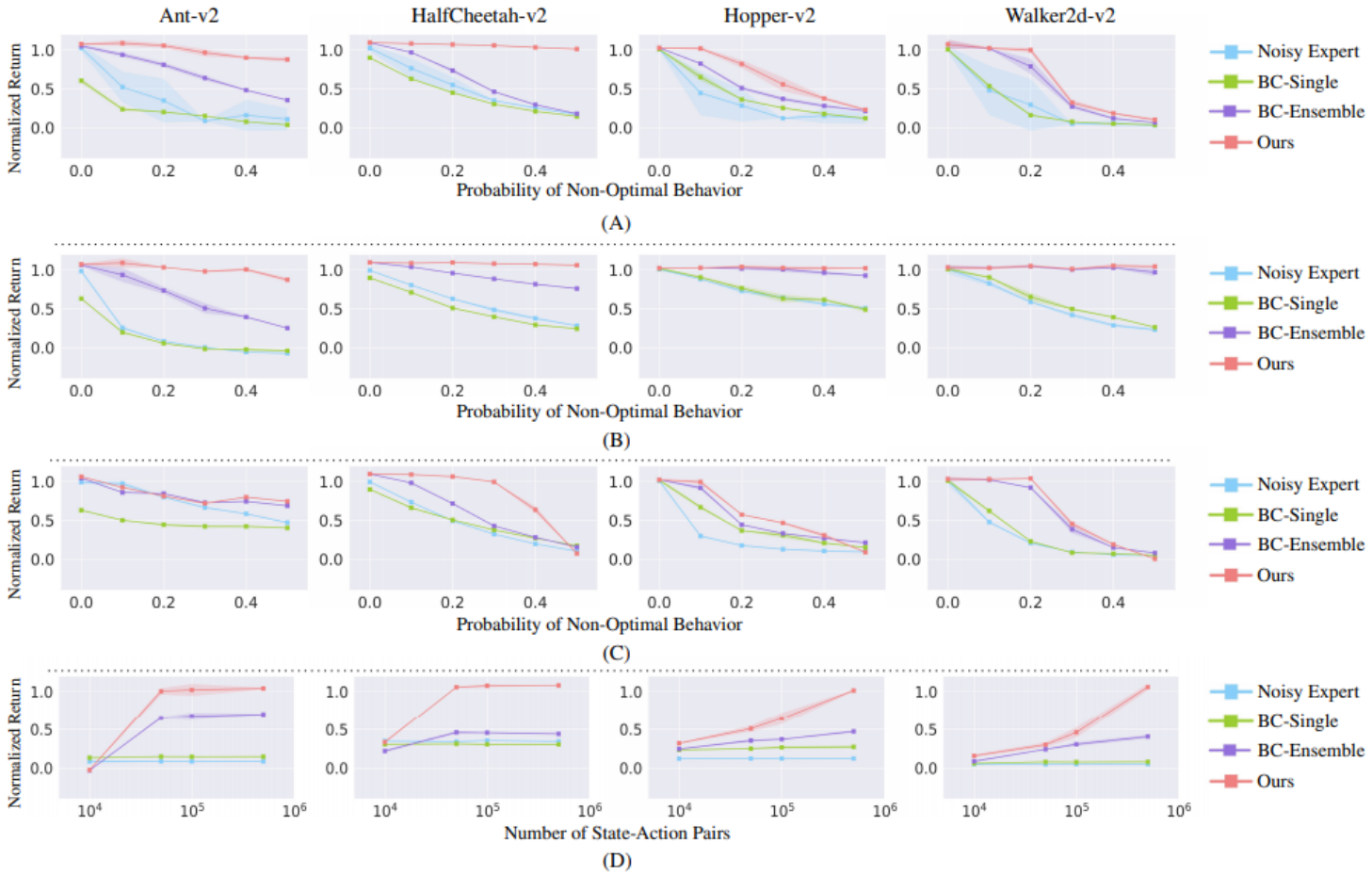
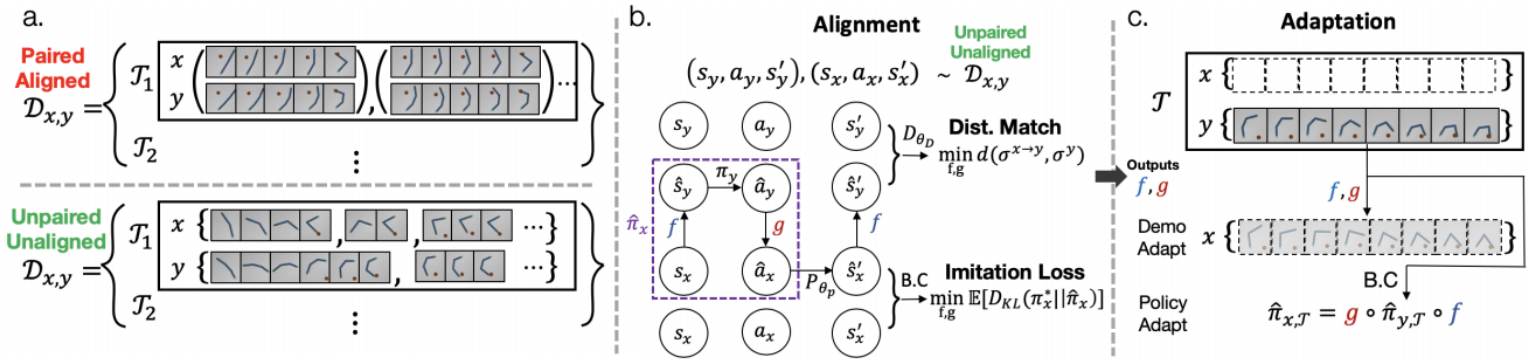

Ant-v2
HalfCheetah-v2
Hopper-v2
Walker2d-v2
Normalized Return
Probability of Non-Optimal Behavior
(A)
(B)
(C)
Number of State-Action Pairs
(D)
Noisy Expert
BC-Single
BC-Ensemble
Ours

Table 1: The experimental results against IC-GAIL, 2IWIL and T-REX.

	Ant-v2	HalfCheetah-v2	Hopper-v2
IC-GAIL	0.6313 ± 0.162	0.9416 ± 0.103	1.2334 ± 0.152
2IWIL	1.0420 ± 0.021	1.0248 ± 0.059	1.2238 ± 0.135
T-REX	0.5862 ± 0.124	0.0010 ± 0.113	0.4414 ± 0.219
Ours	1.0559 ± 0.053	1.0932 ± 0.092	1.0035 ± 0.045

Domain Adaptive Imitation Learning (ICML 2020)



Definition 1. *An* ***MDP reduction*** *from* $\mathcal{M}_x = (\mathcal{S}_x, \mathcal{A}_x, P_x, \eta_x, R_x)$ *to* $\mathcal{M}_y = (\mathcal{S}_y, \mathcal{A}_y, P_y, \eta_y, R_y)$ *is a tuple* $r = (\phi, \psi)$ *where* $\phi : \mathcal{S}_x \to \mathcal{S}_y, \psi : \mathcal{A}_x \to \mathcal{A}_y$ *are maps that preserve:*

1. *(π-optimality)* $\forall (s_x, a_x, s_y, a_y) \in \mathcal{S}_x \times \mathcal{A}_x \times \mathcal{S}_y \times \mathcal{A}_y$:

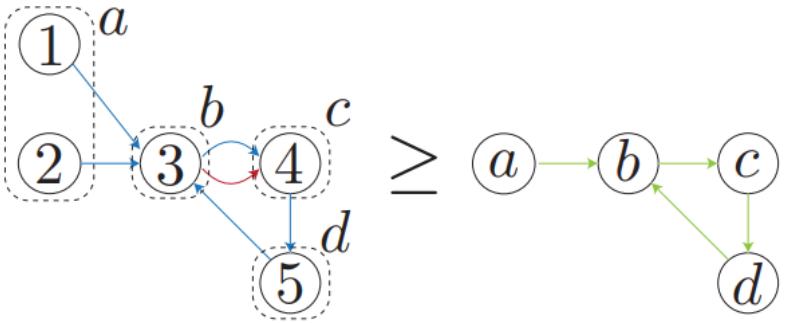
$$O_{\mathcal{M}_y}(\phi(s_x), \psi(a_x)) = 1 \Rightarrow O_{\mathcal{M}_x}(s_x, a_x) = 1 \quad (1)$$

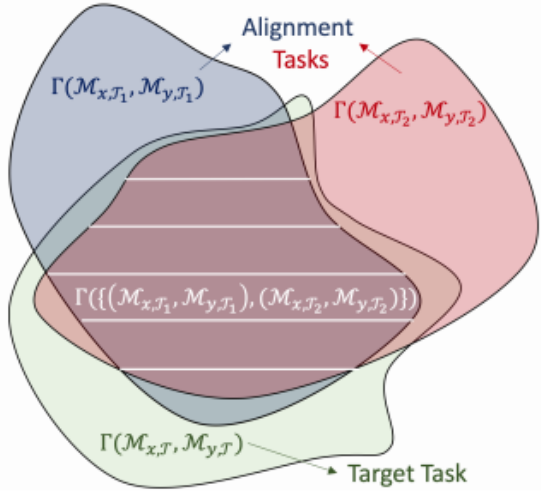
$$O_{\mathcal{M}_y}(s_y, a_y) = 1 \Rightarrow \phi^{-1}(s_y) \neq \emptyset, \psi^{-1}(a_y) \neq \emptyset \quad (2)$$

2. *(dynamics)* $\forall (s_x, a_x, s_y, a_y) \in \mathcal{S}_x \times \mathcal{A}_x \times \mathcal{S}_y \times \mathcal{A}_y$ *such that* $O_{\mathcal{M}_y}(s_y, a_y) = 1, s_x \in \phi^{-1}(s_y), a_x \in \psi^{-1}(a_y)$:

$$P_y(s_y, a_y) = \phi(P_x(s_x, a_x)) \quad (3)$$

where we define $\phi^{-1}(s_y) = \{s_x | \phi(s_x) = s_y\}$, $\psi^{-1}(a_y) = \{a_x | \psi(a_x) = a_y\}$. *Furthermore,* r *is an* ***MDP permutation*** *if and only if* ϕ, ψ *are bijective.*





Definition 4. *Let $\mathcal{M}_x$, $\mathcal{M}_y$ be two MDPs and $\hat{\pi}_x = g \circ \pi_y \circ f$ for $f : \mathcal{S}_x \to \mathcal{S}_y, g : \mathcal{A}_y \to \mathcal{A}_x$ and policy π_y. The* ***co-domain policy execution process*** *$\mathcal{P}_{\hat{\pi}_x} = \{\hat{s}_y^{(t)}, \hat{a}_y^{(t)}\}_{t \geq 0}$ is realized by running $\hat{\pi}_x$ in $\mathcal{M}_x$, i.e:*

$$s_x^{(0)} \sim \eta_x, \quad \hat{s}_y^{(t)} = f(s_x^{(t)}), \quad \hat{a}_y^{(t)} \sim \pi_y(\cdot|\hat{s}_y^{(t)}),$$
$$a_x^{(t)} = g(\hat{a}_y^{(t)}), \quad s_x^{(t+1)} = P_x(s_x^{(t)}, a_x^{(t)}) \quad \forall t \geq 0 \qquad (4)$$

$$\min_{f,g} -J(\hat{\pi}_x) + \lambda d(\sigma_{\hat{\pi}_x}^{x \to y}, \sigma_{\pi_y}^{y})$$

Algorithm 1 Generative Adversarial MDP Alignment (GAMA)

input: Alignment task set $\mathcal{D}_{x,y} = \{(\mathcal{D}_{\mathcal{M}_{x,\mathcal{T}_i}}, \mathcal{D}_{\mathcal{M}_{y,\mathcal{T}_i}})\}_{i=1}^{N}$ of unpaired trajectories, fitted $\pi^*_{y,\mathcal{T}_i}$

while not done **do**:

for $i = 1, ..., N$ **do**:

Sample $(s_x, a_x, s'_x) \sim \mathcal{D}_{\mathcal{M}_{x,\mathcal{T}_i}}, (s_y, a_y, s'_y) \sim \mathcal{D}_{\mathcal{M}_{y,\mathcal{T}_i}}$ and store in buffer $\mathcal{B}_x^i, \mathcal{B}_y^i$

for $j = 1, ..., M$ **do**:

Sample mini-batch j from $\mathcal{B}_x^i, \mathcal{B}_y^i$

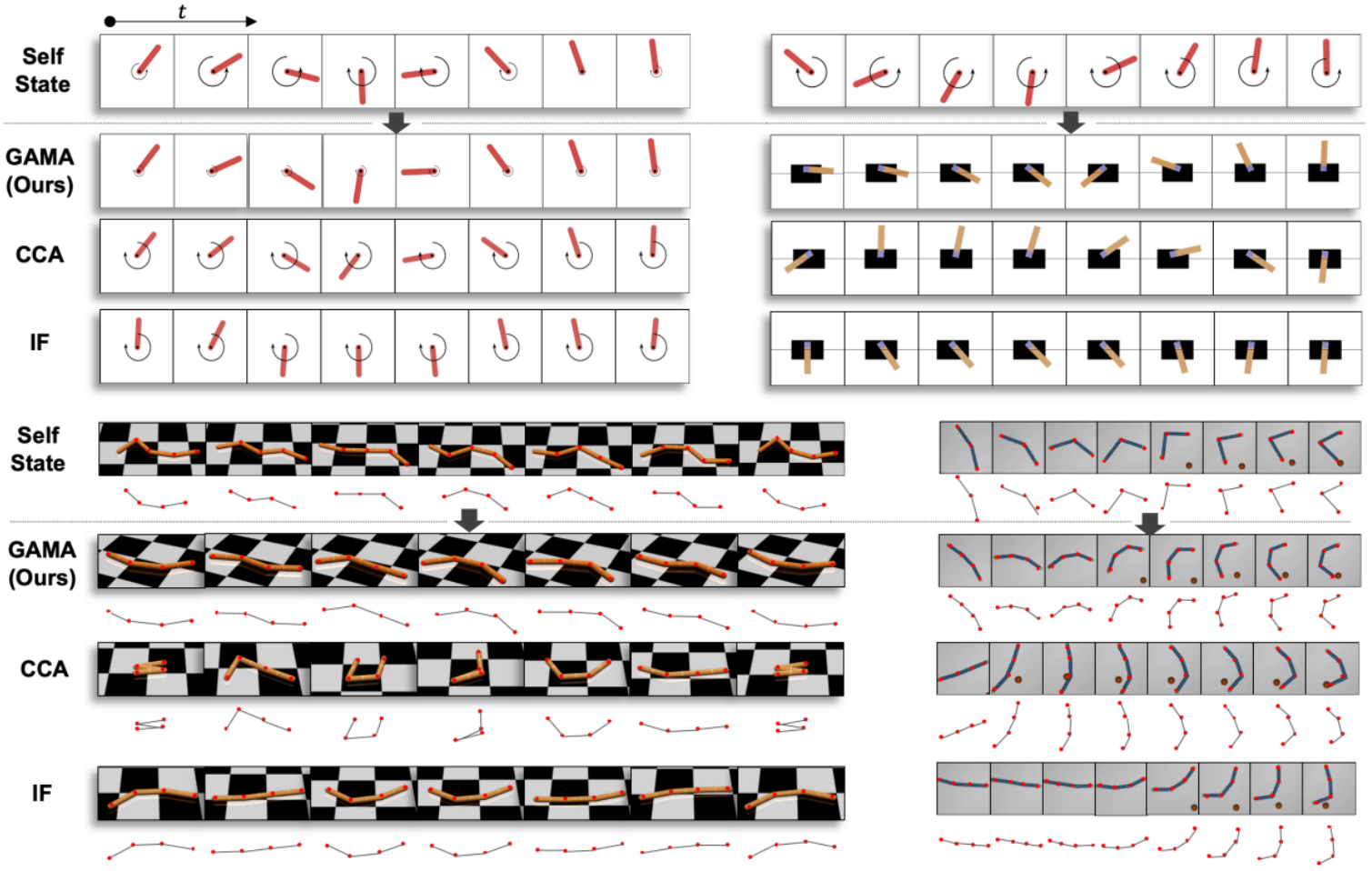
Update dynamics model with: $-\hat{\mathbb{E}}_{\pi^*_{x,\mathcal{T}_i}}[\nabla_{\theta_P}(P^x_{\theta_P}(s_x, a_x) - s'_x)^2]$

Update discriminator: $\hat{\mathbb{E}}_{\pi^*_{y,\mathcal{T}_i}}[\nabla_{\theta_D^i} \log D_{\theta_D^i}(s_y, a_y, s'_y)] + \hat{\mathbb{E}}_{\pi^*_{x,\mathcal{T}_i}}[\nabla_{\theta_D^i} \log\left(1 - D_{\theta_D^i}(\hat{s}_y, \hat{a}_y, \hat{s}'_y)\right)]$

Update alignments $(f_{\theta_f}, g_{\theta_g})$ with gradients:

$$-\hat{\mathbb{E}}_{\pi^*_{x,\mathcal{T}_i}}[\nabla_{\theta_f} \log D_{\theta_D}(\hat{s}_y, \hat{a}_y, \hat{s}'_y)] + \hat{\mathbb{E}}_{\pi^*_{x,\mathcal{T}_i}}[\nabla_{\theta_f}(\hat{\pi}_{x,\mathcal{T}_i}(s_x) - a_x)^2]$$

$$-\hat{\mathbb{E}}_{\pi^*_{x,\mathcal{T}_i}}[\nabla_{\theta_g} \log D_{\theta_D}(\hat{s}_y, \hat{a}_y, \hat{s}'_y)] + \hat{\mathbb{E}}_{\pi^*_{x,\mathcal{T}_i}}[\nabla_{\theta_g}(\hat{\pi}_{x,\mathcal{T}_i}(s_x) - a_x)^2]$$


t
Self State
GAMA (Ours)
CCA
IF
Self State
GAMA (Ours)
CCA
IF