Domain Adaptive Imitation Learning

Kuno Kim [1]   Yihong Gu [2]   Jiaming Song [1]   Shengjia Zhao [1]   Stefano Ermon [1]

Abstract

We study the question of how to imitate tasks across domains with discrepancies such as embodiment, viewpoint, and dynamics mismatch. Many prior works require paired, aligned demonstrations and an additional RL step that requires environment interactions. However, paired, aligned demonstrations are seldom obtainable and RL procedures are expensive. We formalize the Domain Adaptive Imitation Learning (DAIL) problem, which is a unified framework for imitation learning in the presence of viewpoint, embodiment, and dynamics mismatch. Informally, DAIL is the process of learning how to perform a task optimally, given demonstrations of the task in a distinct domain. We propose a two step approach to DAIL: alignment followed by adaptation. In the alignment step we execute a novel unsupervised MDP alignment algorithm, Generative Adversarial MDP Alignment (GAMA), to learn state and action correspondences from *unpaired, unaligned* demonstrations. In the adaptation step we leverage the correspondences to zero-shot imitate tasks across domains. To describe when DAIL is feasible via alignment and adaptation, we introduce a theory of MDP alignability. We experimentally evaluate GAMA against baselines in embodiment, viewpoint, and dynamics mismatch scenarios where aligned demonstrations don't exist and show the effectiveness of our approach.

1. Introduction

Humans possess an astonishing ability to recognize latent structural similarities between behaviors in related but distinct domains, and learn new skills from cross domain demonstrations alone. Not only are we capable of learning from third person observations that have no obvious correspondence to our internal self representations (Stadie et al., 2017; Liu et al., 2018; Sermanet et al., 2018), but we also are capable of imitating experts with different embodiments (Gupta et al., 2017; Rizzolatti & Craighero, 2004; Liu et al., 2020) in foreign environments (Liu et al., 2020) - e.g an infant is able to imitate visuomotor skills by watching adults with different biomechanics (Jones, 2009) acting in environments distinct from their playroom. Previous work in neuroscience (Marshall & Meltzoff, 2015) and robotics (Kuniyoshi & Inoue, 1993; Kuniyoshi et al., 1994) have recognized the pitfalls of exact behavioral cloning in the presence of domain discrepancies and posited that the effectiveness of the human imitation learning mechanism hinges on the ability to learn structure preserving domain correspondences. These correspondences enable the learner to internalize cross domain demonstrations and produce a reconstruction of the behavior in the self domain. Consider a young child that has learned to associate (or "align") his internal body map with the limbs of an adult. When the adult demonstrates running, the child is able internalize the demonstration, and reproduce the behavior.

Recently, separate solutions have been proposed for imitation learning across three main types of domain discrepancies: dynamics (Liu et al., 2020), embodiment (Gupta et al., 2017), and viewpoint (Liu et al., 2018; Sermanet et al., 2018) mismatch. Most works (Liu et al., 2018; Sermanet et al., 2018; Gupta et al., 2017) require *paired, time-aligned demonstrations* to obtain state correspondences and an RL step involving environment interactions. (see Figure 1a) Unfortunately, paired, aligned demonstrations are seldom obtainable and RL procedures are expensive.

In this work we formalize the Domain Adaptive Imitation Learning (DAIL) problem - a unified framework for imitation learning across domains with dynamics, embodiment, and/or viewpoint mismatch. Informally, DAIL is the *process of learning how to perform a task optimally in a self domain, given demonstrations of the task in a distinct expert domain*. We propose a two-step approach to DAIL: alignment followed by adaptation. In the alignment step we execute a novel unsupervised MDP alignment algorithm, Generative Adversarial MDP Alignment (GAMA), to learn state, action maps from *unpaired, unaligned* demonstrations. In the adaptation step we leverage the learned state,

---

[1]Department of Computer Science, Stanford University [2]Department of Computer Science, Tsinghua University. Correspondence to: Kuno Kim <khkim@cs.stanford.edu>.
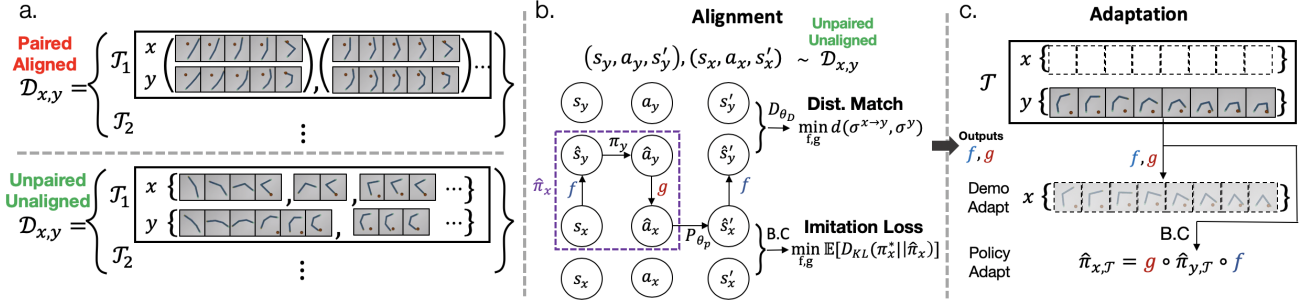
Figure 1. **DAIL Pipeline**. (a). *Inputs*: Illustration of paired, aligned vs unpaired, unaligned demonstrations in the alignment task set $\mathcal{D}_{x,y}$ containing tasks $T_1, T_2, ...$ (b). *Alignment* phase: we learn state, action maps f, g between the self (x) and expert (y) domain from unpaired, unaligned demonstrations by minimizing a distribution matching loss and an imitation loss on a composite policy $\hat{\pi}_x$ (c) *Adaptation* phase: adapt the expert domain policy $\pi_{y,\mathcal{T}}$ or demonstrations to obtain a self domain policy $\hat{\pi}_{x,\mathcal{T}}$

action maps to zero-shot imitate tasks across domains without an extra RL step. To shed light on when DAIL can be solved by alignment and adaptation, we introduce a theory of MDP alignability. We conduct experiments with a variety of domains that have dynamics, embodiment, and viewpoint mismatch and demonstrate significant gains on learning from unpaired data. The primary contributions of this work are as follows:

1. We propose an unsupervised MDP alignment algorithm that succeeds at DAIL from *unpaired, unaligned demonstrations* removing the need for costly paired, aligned data.

2. We achieve *zero-shot imitation*, thereby removing a costly RL procedure involving environment interactions.

3. We propose a *unifying theoretical framework* for imitation learning across domains with dynamics, embodiment, and/or viewpoint mismatch.

2. Domain Adaptive Imitation Learning

An infinite horizon Markov Decision Process (MDP) $\mathcal{M} \in \Omega$ with deterministic dynamics is a tuple $(\mathcal{S}, \mathcal{A}, P, \eta, R)$ where Ω is the set of all MDPs, $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $P : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is a (deterministic) transition function, $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, and η is the initial state distribution. A domain is an MDP without the reward, i.e $(\mathcal{S}, \mathcal{A}, P, \eta)$. Intuitively, a domain fully characterizes the embodied agent and the environment dynamics, but not the desired behavior. A task $\mathcal{T}$ is a label for an MDP corresponding to the high level description of optimal behavior, such as "walking". $\mathcal{T}$ is analogous to category labels for images. An MDP with domain x for task $\mathcal{T}$ is denoted by $\mathcal{M}_{x,\mathcal{T}} = (\mathcal{S}_x, \mathcal{A}_x, P_x, \eta_x, R_{x,\mathcal{T}})$, where $R_{x,\mathcal{T}}$ is a reward function encapsulating the behavior labeled by $\mathcal{T}$. For example, different reward functions are needed to realize the "walking" behavior in two morphologically different humanoids. A (stationary) policy for $\mathcal{M}_{x,\mathcal{T}}$ is a map $\pi_{x,\mathcal{T}} : \mathcal{S}_x \to \mathcal{B}(\mathcal{A}_x)$ where $\mathcal{B}$ is the set of probability measures on $\mathcal{A}_x$ and an optimal policy $\pi^*_{x,\mathcal{T}} =$

$\arg\max_{\pi_x} J(\pi_x)$ achieves the highest policy performance $J(\pi_x) = \mathbb{E}_{\pi_x}[\sum_{t=0}^{\infty} \gamma^t R_{x,\mathcal{T}}(s_x^{(t)}, a_x^{(t)})]$ where $0 < \gamma < 1$ is a discount factor. A demonstration $\tau_{\mathcal{M}_{x,\mathcal{T}}}$ of length H for an MDP $\mathcal{M}_{x,\mathcal{T}}$ is a sequence of state, action tuples $\tau_{\mathcal{M}_{x,\mathcal{T}}} = \{(s_x^{(t)}, a_x^{(t)})\}_{t=1}^{H}$ and $\mathcal{D}_{\mathcal{M}_{x,\mathcal{T}}} = \{\tau_{\mathcal{M}_{x,\mathcal{T}}}^{(k)}\}_{k=1}^{K}$ is a set of demonstrations.

Let $\mathcal{M}_{x,\mathcal{T}}, \mathcal{M}_{y,\mathcal{T}}$ be self and expert MDPs for a target task $\mathcal{T}$. Given expert domain demonstrations $\mathcal{D}_{\mathcal{M}_{y,\mathcal{T}}}$, Domain Adaptive Imitation Learning (DAIL) aims to determine an optimal self domain policy $\pi^*_{x,\mathcal{T}}$ without access to the reward function $R_{x,\mathcal{T}}$. In this work we propose to first solve an MDP alignment problem and then leverage the alignments to zero-shot imitate expert domain demonstrations. Like prior work (Gupta et al., 2017; Liu et al., 2018; Sermanet et al., 2018), we assume the availability of an alignment task set $\mathcal{D}_{x,y} = \{(\mathcal{D}_{\mathcal{M}_{x,\mathcal{T}_i}}, \mathcal{D}_{\mathcal{M}_{y,\mathcal{T}_i}})\}_{i=1}^{N}$ containing demonstrations for N tasks $\{\mathcal{T}_i\}_{i=1}^{N}$ from both the self and expert domain. $\mathcal{D}_{x,y}$ could, for example, contain both robot (x) and human (y) demonstrations for a set primitive tasks such as walking, running, and jumping. Unlike prior work, demonstrations are *unpaired and unaligned*, i.e $(s_x^{(t)}, s_y^{(t)})$ may not be a valid state correspondence (see Figure 1(a)). Paired, time-aligned cross domain data is expensive and may not even exist when task execution rates differ or there exists systematic embodiment mismatch between the domains. For example, a child can imitate an adult running, but not achieve the same speed. Our set up emulates a natural setting in which humans compare how they perform tasks to how other agents perform the same tasks in order to find structural similarities and identify domain correspondences. We now proceed to introduce a theoretical framework that explains how and when the DAIL problem can be solved by MDP alignment followed by adaptation.

3. Alignable MDPs

Let $\Pi^*_{\mathcal{M}}$ be the set of all optimal policies for MDP $\mathcal{M}$. We define an occupancy measure (Syed et al., 2008) q_π :

$\mathcal{S} \times \mathcal{A} \to \mathbb{R}$ for policy π executed in MDP $\mathcal{M}$ as $q_\pi(s,a) = \pi(a|s) \sum_{t=0}^{\infty} \gamma^t \Pr(s^{(t)} = s; \pi, \mathcal{M})$. We further define the optimality function $O_{\mathcal{M}_x} : \mathcal{S}_x \times \mathcal{A}_x \to \{0,1\}$ for an MDP $\mathcal{M}_x$ as $O_{\mathcal{M}_x}(s_x, a_x) = 1$ if $\exists \pi_x^* \in \Pi_{\mathcal{M}_x}^*$ such that $(s_x, a_x) \in \operatorname{supp}(q_{\pi_x^*})$ and $O_{\mathcal{M}_x}(s_x, a_x) = 0$ otherwise. We are now ready to formalize MDP reductions: a class of structure preserving maps between MDPS.

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.*

In words, Eq. 1 states that only optimal state, action pairs in x can be mapped to optimal state, action pairs in y and Eq. 2 states that r must be surjective on the set of optimal state, action pairs in y. These properties imply that a policy in $\mathcal{M}_x$ should have more flexibility choosing optimal actions than one in $\mathcal{M}_y$. Eq. 3 states that a reduction must preserve (deterministic) dynamics. We use the notation $\mathcal{M}_x \geq_{\phi, \psi} \mathcal{M}_y$ to denote that (ϕ, ψ) is a reduction from $\mathcal{M}_x$ to $\mathcal{M}_y$, and the shorthand $\mathcal{M}_x \geq \mathcal{M}_y$ to denote that $\mathcal{M}_x$ reduces to $\mathcal{M}_y$. To gain an intuitive understanding of MDP reductions, picture the execution trace of an optimal policy as a directed graph with colored edges in which the nodes correspond to states visited by an optimal policy, and the colored edges correspond to actions taken. An MDP reduction from $\mathcal{M}_x$ to $\mathcal{M}_y$ homomorphs the execution graph of an optimal policy in $\mathcal{M}_x$ to a execution graph of an optimal policy in $\mathcal{M}_y$. Figure 2 shows an example of a valid reduction from $\mathcal{M}_x$ to $\mathcal{M}_y$: states $1, 2$ in $\mathcal{S}_x$ are mapped (merged) to state a in $\mathcal{S}_y$ and the blue, red actions in $\mathcal{A}_x$ are mapped to the green action in $\mathcal{A}_y$. Intuitively, if $\mathcal{M}_x \geq_{\phi, \psi} \mathcal{M}_y$, then (ϕ, ψ) compresses $\mathcal{M}_x$ by merging all optimal state, action pairs that have identical dynamics properties.

Definition 2. *Two MDPs* $\mathcal{M}_x, \mathcal{M}_y$ *are **alignable** if and only if* $\mathcal{M}_x \geq \mathcal{M}_y$ *or* $\mathcal{M}_y \geq \mathcal{M}_x$.

Definition 2 states that MDPs are alignable if reductions exists between them, meaning that they share structure. We use $\Gamma(\mathcal{M}_x, \mathcal{M}_y) = \{(\phi, \psi) | \mathcal{M}_x \geq_{\phi, \psi} \mathcal{M}_y\}$ to denote the
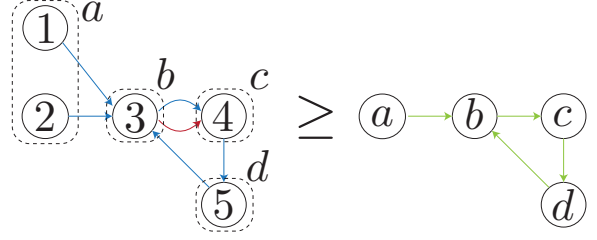

Figure 2. **MDP Reduction Example** between execution traces in $\mathcal{M}_x$ (Left) and $\mathcal{M}_y$ (Right), where $\mathcal{M}_x \geq \mathcal{M}_y$. States correspond to nodes and actions to colors. The shown reduction merges nodes in dotted boxes to their corner label, e.g $\phi(1) = \phi(2) = a$, and both blue, red actions in $\mathcal{M}_x$ to the green action in $\mathcal{M}_y$.

set of all valid reductions from $\mathcal{M}_x$ to $\mathcal{M}_y$. Reductions have a particularly useful property which is that they adapt policies across alignable MDPs. Consider a state map $f : \mathcal{S}_x \to \mathcal{S}_y$, an inverse action map $g : \mathcal{A}_y \to \mathcal{A}_x$, and a composite policy $\hat{\pi}_x = g \circ \pi_y \circ f$ (see Figure 1(b)). In words, $\hat{\pi}_x$ maps a self state to an expert state via f, simulates the expert's action choice for the mapped state via π_y, then chooses a self action that corresponds to the simulated expert action with g. The following lemma holds for $\hat{\pi}_x$.

Theorem 1. *Let* $\mathcal{M}_x, \mathcal{M}_y$ *be MDPs satisfying Assumption 1 (see Appendix F),* $\mathcal{M}_x \geq_{\phi, \psi} \mathcal{M}_y$, *and* π_y *be optimal in* $\mathcal{M}_y$. *Then,* $\forall g : \mathcal{A}_y \to \mathcal{A}_x$ *s.t* $\psi \circ g(a_y) = a_y \; \forall a_y \in \{a_y | \exists s_y \in \mathcal{S}_y \; s.t \; O_{\mathcal{M}_y}(s_y, a_y) = 1\}$, *it holds that* $\hat{\pi}_x = g \circ \pi_y \circ \phi$ *is optimal in* $\mathcal{M}_x$.

Theorem 1 states that the state, action maps (f, g^{-1}) chosen to be a reduction can adapt optimal policies between alignable MDPs. Here onwards we interchangeably refer to (f, g) as "alignments". We now show how the DAIL problem can be solved by first solving an MDP alignment problem followed by an adaptation step.

Definition 3. *Let* $(\mathcal{M}_x, \mathcal{M}_y), (\mathcal{M}_x', \mathcal{M}_y') \in \Omega^2$ *be two MDP pairs. Then,* $(\mathcal{M}_x, \mathcal{M}_y) \sim (\mathcal{M}_x', \mathcal{M}_y')$, *i.e they are **joint alignable**, if* $\Gamma(\mathcal{M}_x, \mathcal{M}_y) \cap \Gamma(\mathcal{M}_x', \mathcal{M}_y') \neq \emptyset$.

In words, two MDP pairs are joint alignable if there exists a shared reduction. We define an equivalence class $[(\mathcal{M}_x, \mathcal{M}_y)]_\sim = \{(\mathcal{M}_x', \mathcal{M}_y') \mid (\mathcal{M}_x', \mathcal{M}_y') \sim (\mathcal{M}_x, \mathcal{M}_y)\}$ of MDP pairs that share reductions. Overloading notation, $\Gamma(\{(\mathcal{M}_x^i, \mathcal{M}_y^i)\}_{i=1}^N) = \{(\phi, \psi) \mid (\phi, \psi) \in \Gamma(\mathcal{M}_x^1, \mathcal{M}_x^1) \cap \cdots \cap \Gamma(\mathcal{M}_x^N, \mathcal{M}_x^N)\}$. We now formally state the MDP alignment problem: Let $(\mathcal{M}_{x,\mathcal{T}}, \mathcal{M}_{y,\mathcal{T}})$ be an MDP pair for a target task $\mathcal{T}$. Given an alignment task set $\mathcal{D}_{x,y} = \{(\mathcal{D}_{\mathcal{M}_{x,\mathcal{T}_i}}, \mathcal{D}_{\mathcal{M}_{y,\mathcal{T}_i}})\}_{i=1}^N$ comprising unpaired, unaligned demonstrations for MDP pairs $\{(\mathcal{M}_{x,\mathcal{T}_i}, \mathcal{M}_{y,\mathcal{T}_i})\}_{i=1}^N \subseteq [(\mathcal{M}_{x,\mathcal{T}}, \mathcal{M}_{y,\mathcal{T}})]_\sim$, determine $(\phi, \psi) \in \Gamma(\{(\mathcal{M}_{x,\mathcal{T}_i}, \mathcal{M}_{y,\mathcal{T}_i})\}_{i=1}^N)$ such that $(\phi, \psi) \in \Gamma(\mathcal{M}_{x,\mathcal{T}}, \mathcal{M}_{y,\mathcal{T}})$. As shown in Figure 3, with more MDP pairs, there are likely a smaller the number of joint alignments $|\Gamma(\{(\mathcal{M}_{x,\mathcal{T}_i}, \mathcal{M}_{y,\mathcal{T}_i})\}_{i=1}^N)|$ and, as a result, $(\phi, \psi) \in \Gamma(\{(\mathcal{M}_{x,\mathcal{T}_i}, \mathcal{M}_{y,\mathcal{T}_i})\}_{i=1}^N)$ is more likely to "generalize" to
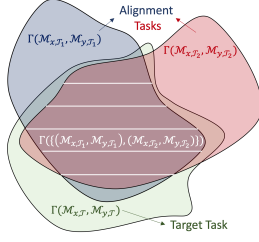
Figure 3. **MDP Alignment Problem**. Blue/red, and green regions denote sets of MDP alignments for two alignment tasks and the target task, respectively. White hatches cover the solution set to the MDP alignment problem which is the intersection of all sets

an MDP pair for a new target task $(\mathcal{M}_{x,\mathcal{T}}, \mathcal{M}_{y,\mathcal{T}})$ in the equivalence class. Analogously, in a standard supervised learning problem, more training data is likely to shrink the set of models performing optimally on the training set but poorly on the test set. We can then use (ϕ, ψ) for DAIL: given cross domain demonstrations $\mathcal{D}_{\mathcal{M}_{y,\mathcal{T}}}$ for the target task $\mathcal{T}$, learn an expert domain policy $\pi_{y,\mathcal{T}}$, and adapt it into the self domain using (ϕ, ψ) according to Theorem 1.

We can now assess when domains with embodiment and viewpoint mismatch have meaningful state correspondences, i.e MDP reductions, thus allowing for domain adaptive imitation. The states of a human expert with more degrees of freedom than a robot imitator can be merged into the robot states if the task only requires the robot's degrees of freedom and the execution traces share structure, e.g traces are both cycles. However, if the task requires all degrees of freedom possessed only by the human, the robot cannot find meaningful correspondences, and also cannot imitate the task. Two MDPs for different viewpoints of an agent performing a task are MDP permutations since there is a one-to-one correspondence between state, actions at same timestep in the execution trace of an optimal policy.

4. Learning MDP Reductions

We now derive objectives that can be optimized to learn MDP reductions. We propose distribution matching and policy performance maximization. We first define the distributions to be matched.

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:*

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

The target distribution $\sigma_{\pi_y}^y$ is over transitions uniformly sampled from execution traces of π_y in $\mathcal{M}_y$ and the proxy distribution $\sigma_{\hat{\pi}_x}^{x \to y}$ is over cross domain transitions uniformly

sampled from realizations of $\mathcal{P}_{\hat{\pi}_x}$, i.e running $\hat{\pi}_x$ in $\mathcal{M}_x$.

$$\sigma_{\pi_y}^y(s_y, a_y, s_y') :=$$
$$\lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T} \Pr(s_y^{(t)} = s_y, a_y^{(t)} = a_y, s_y^{(t+1)} = s_y'; \pi_y) \quad (5)$$
$$\sigma_{\hat{\pi}_x}^{x \to y}(s_y, a_y, s_y') :=$$
$$\lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T} \Pr(\hat{s}_y^{(t)} = s_y, \hat{a}_y^{(t)} = a_y, \hat{s}_y^{(t+1)} = s_y'; \mathcal{P}_{\hat{\pi}_x}) \quad (6)$$

We now propose two concrete objectives to optimize for: 1. optimality of $\hat{\pi}_x$, 2. $\sigma_{\hat{\pi}_x}^{x \to y} = \sigma_{\pi_y}^y$. In other words, we seek to learn f, g that matches distributions over transition tuples in domain y while maximizing policy performance in domain x. The former captures the dynamics preservation property from Eq. 3 and the latter captures the optimal policy preservation property from Eq. 1, 2. The following theorem uncovers the connection between our objectives and MDP reductions.

Theorem 2. *Let $\mathcal{M}_x, \mathcal{M}_y$ be MDPs satisfying Assumption 1 (see Supp Materials). If $\mathcal{M}_x \geq \mathcal{M}_y$, then $\exists f : \mathcal{S}_x \to \mathcal{S}_y, g : \mathcal{A}_y \to \mathcal{A}_x$, and an optimal covering policy π_y (see Appendix F) that satisfy objectives 1 and 2. Conversely, if $\exists f : \mathcal{S}_x \to \mathcal{S}_y$, an injective map $g : \mathcal{A}_y \to \mathcal{A}_x$ and an optimal covering policy π_y satisfying objectives 1 and 2, then $\mathcal{M}_x \geq \mathcal{M}_y$ and $\exists(\phi, \psi) \in \Gamma(\mathcal{M}_x, \mathcal{M}_y)$ s.t $f = \phi$ and $\psi \circ g(a_y) = a_y, \forall a_y \in \mathcal{A}_y$.*

Theorem 2 states that if two MDP are alignable, then objectives 1 and 2 can be satisfied. Conversely, if 1 and 2 can be satisfied for two MDPs with state map f and an injective action map g, then the MDPs must be alignable and all solutions (f, g) are reductions. While Theorem 2 requires that MDPs be alignable to guarantee identifiability of solutions obtained via optimizing for objectives 1 and 2, our experiments will also run on MDPs that are "weakly" alignable, i.e. Eq. 1, 2, 3 do not hold exactly, but intuitively share structure. In the next section, we derive a simple algorithm to learn MDP reductions.

5. Generative Adversarial MDP Alignment

Building on Theorem 2, we propose the following training objective for aligning MDPs:

$$\min_{f,g} -J(\hat{\pi}_x) + \lambda d(\sigma_{\hat{\pi}_x}^{x \to y}, \sigma_{\pi_y}^y) \quad (7)$$

where $J(\hat{\pi}_x)$ is the performance of $\hat{\pi}_x$, d is a distance metric between distributions, and $\lambda > 0$ is a Lagrange multiplier. In practice, we found that injectivity of g is unnecessary to enforce in continuous domains. We now present an instantiation of this framework: Generative Adversarial MDP Alignment (GAMA). Recall that we are given an alignment task set $\mathcal{D}_{x,y} = \{(\mathcal{D}_{\mathcal{M}_{x,\mathcal{T}_i}}, \mathcal{D}_{\mathcal{M}_{y,\mathcal{T}_i}})\}_{i=1}^{N}$. In the alignment step, we learn $\pi_{y,\mathcal{T}_i}^*, \forall \mathcal{T}_i$ and parameterized

state, action maps $f_{\theta_f} : \mathcal{S}_x \to \mathcal{S}_y, g_{\theta_g} : \mathcal{A}_y \to \mathcal{A}_x$ that compose $\hat{\pi}_{x,\mathcal{T}_i} = g_{\theta_g} \circ \pi^*_{y,\mathcal{T}_i} \circ f_{\theta_f}$. To match $\sigma^{x \to y}_{\hat{\pi}_x}, \sigma^y_{\pi_y}$, we employ adversarial training (Goodfellow et al., 2014) in which separate discriminators $D_{\theta^i_D}$ per task are trained to distinguish between "real" transitions $(s_y, a_y, s'_y) \sim \pi^*_{y,\mathcal{T}_i}$ and "fake" transitions $(\hat{s}_y, \hat{a}_y, \hat{s}'_y) \sim \hat{\pi}_{x,\mathcal{T}_i}$, where $\hat{s}_y = f_{\theta_f}(s_x), \hat{a}_y = \pi_y(\hat{s}_y), \hat{s}'_y = f_{\theta_f}(P^x_{\theta_P}(s_x, g(\hat{a}_y)))$, and $P^x_{\theta_P}$ is a fitted model of the x domain dynamics. (see Figure 1(b)) The generator, consisting of $f_{\theta_f}, g_{\theta_g}$, is trained to fool the discriminator while maximizing policy performance. The distribution matching gradients are back propagated through the learned dynamics, $\pi^*_{y,\mathcal{T}_i}$ is learned by Imitation Learning (IL) on $\mathcal{D}_{\mathcal{M}_{y,\mathcal{T}_i}}$, and the policy performance objective on $\hat{\pi}_{x,\mathcal{T}_i}$ is achieved by IL on $\mathcal{D}_{\mathcal{M}_{x,\mathcal{T}_i}}$. In this work, we use behavioral cloning (Pomerleau, 1991) for IL. We aim to find a saddle point $\{f, g\} \cup \{D_{\theta^i_D}\}^N_{i=1}$ of the following objective:

$$\min_{f,g} \max_{\{D_{\theta^i_D}\}} \sum^N_{i=1} \big(\mathbb{E}_{s_x \sim \pi^*_{x,\mathcal{T}_i}}[D_{KL}(\pi^*_{x,\mathcal{T}_i}(\cdot|s_x)||\hat{\pi}_{x,\mathcal{T}_i}(\cdot|s_x))]$$
$$+ \lambda(\mathbb{E}_{\pi^*_{y,\mathcal{T}_i}}[\log D_{\theta^i_D}(s_y, a_y, s'_y)]$$
$$+ \mathbb{E}_{\pi^*_{x,\mathcal{T}_i}}[\log(1 - D_{\theta^i_D}(\hat{s}_y, \hat{a}_y, \hat{s}'_y))]]) \quad (8)$$

where D_{KL} is the KL-divergence. We provide the execution flow of GAMA in Algorithm 1. In the adaptation step, we are given expert demonstrations $\mathcal{D}_{\mathcal{M}_{y,\mathcal{T}}}$ of a new target task $\mathcal{T}$, from which we fit an expert domain policy $\pi_{y,\mathcal{T}}$ which are composed with the learned alignments to construct an adapted self policy $\hat{\pi}_{x,\mathcal{T}} = g_{\theta_g} \circ \pi_{y,\mathcal{T}} \circ f_{\theta_f}$. We also experiment with a demonstration adaptation method which additionally trains an inverse state map $f^{-1} : \mathcal{S}_y \to \mathcal{S}_x$, adapts demonstrations $\mathcal{D}_{\mathcal{M}_{y,\mathcal{T}}}$ into the self domain via f^{-1}, g, and applies behavioral cloning on the adapted demonstrations. (see Figure 1(c)) Notably, our entire procedure does not require paired, aligned demonstrations nor an RL step.

Related Works: Closely related to DAIL, the field of cross domain transfer learning in the context of RL has explored approaches to use state maps to exploit cross domain demonstrations in a pretraining procedure for a new target task for which self domain reward function is available. Canonical Correlation Analysis (CCA) (Hotelling, 1936) finds invertible projections into a basis in which data from different domains are maximally correlated. These projections can then be composed to obtain a direct correspondence map between states. (Ammar et al., 2015; Joshi & Chowdhary, 2018) have utilized an unsupervised manifold alignment (UMA) algorithm which finds a linear map between states with similar local geometric properties. UMA assumes the existence of hand crafted features along with a distance metric between them. This family of work commonly uses a linear statemap to define a time-step wise transfer reward and executes an RL step on the new task. Similar to our work, these works use an alignment task set of unpaired, unaligned trajectories to compute the state map. Unlike these works, we learn maps that preserve MDP structure, use deep neural network state, action maps, and achieve zero-shot transfer to the new task without an RL step. More recent work in transfer learning across embodiment (Gupta et al., 2017) and viewpoint (Liu et al., 2018; Sermanet et al., 2018) mismatch obtain state correspondences from an alignment task set comprising paired, time-aligned demonstrations and use them to learn a state map or a state encoder to a domain invariant feature space. In contrast to this family of prior work, our approach learns both state, action maps from *unpaired, unaligned* demonstrations. Also, we remove the need for additional environment interactions and an expensive RL procedure on the target task by leveraging the action map for zero-shot imitation. (Stadie et al., 2017) have shown promise in using domain confusion loss and generative adversarial imitation learning (Ho & Ermon, 2016) for learning across small viewpoint mismatch without an alignment task set, but fails in dealing with large viewpoint differences. Unlike (Stadie et al., 2017), we leverage the alignment task set to succeed in imitating across larger view-

---

Algorithm 1 Generative Adversarial MDP Alignment (GAMA)

input: Alignment task set $\mathcal{D}_{x,y} = \{(\mathcal{D}_{\mathcal{M}_{x,\mathcal{T}_i}}, \mathcal{D}_{\mathcal{M}_{y,\mathcal{T}_i}})\}^N_{i=1}$ 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 $\mathcal{B}^i_x, \mathcal{B}^i_y$

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

Sample mini-batch j from $\mathcal{B}^i_x, \mathcal{B}^i_y$

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: $\hat{\mathbb{E}}_{\pi^*_{y,\mathcal{T}_i}}[\nabla_{\theta^i_D} \log D_{\theta^i_D}(s_y, a_y, s'_y)] + \hat{\mathbb{E}}_{\pi^*_{x,\mathcal{T}_i}}[\nabla_{\theta^i_D} \log\big(1 - D_{\theta^i_D}(\hat{s}_y, \hat{a}_y, \hat{s}'_y)\big)]$

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

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

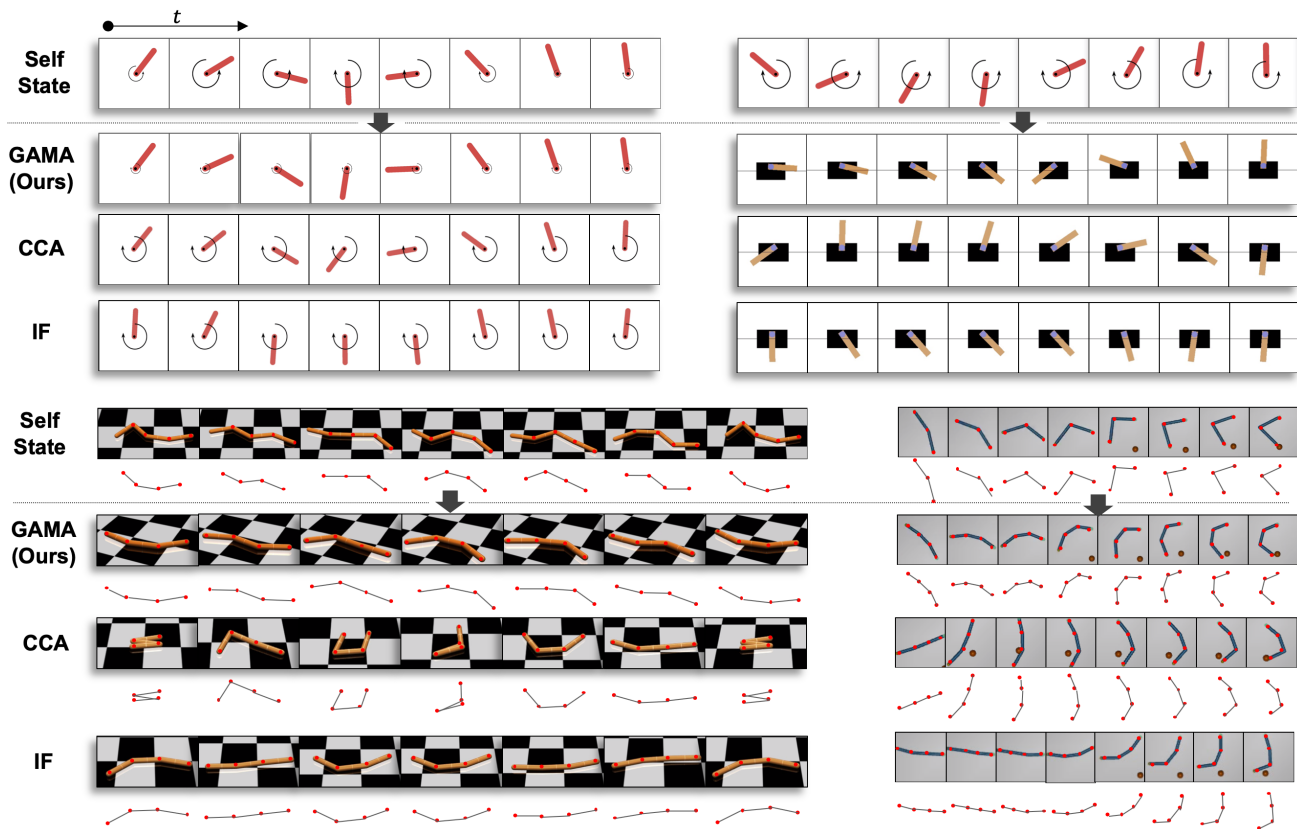*Figure 4.* **MDP Alignment Visualization**. The state maps learned by GAMA and two representative baselines - CCA and IF - are shown for pen↔pen (Top Left), pen↔cart (Top Right), snake4↔snake3 (Bottom Left), reach2↔reach3 (Bottom Right). See Appendix E to see more baselines. GAMA is able to recover MDP permutations for alignable pairs pen↔pen, pen↔cart and find meaningful correspondences between "weakly alignable" pairs snake4↔snake3, reach2↔reach3. See https://youtu.be/l0tc1JCN_1M for videos

Table 1. **MDP Alignment Performance**. Mean ℓ_2 loss between the learned state map predictions and the ground truth permutation. On average, GAMA has $17.3\times$ lower loss than the best baseline. Results are averaged across 5 seeds.

	GAMA (OURS)	CCA	UMA	IF	IFO	RANDOM
PEN ↔ PEN	**0.057 ± 0.017**	0.72 ± 0.25	>100	2.50 ± 1.08	2.24 ± 0.82	>100
PEN ↔ CART	**0.178 ± 0.051**	3.92 ± 3.77	>100	1.62 ± 0.52	3.31 ± 1.2	>100
REACH2↔REACH2-TP	**0.092 ± 0.043**	10.14 ± 5.31	>100	12.41 ± 3.12	5.12 ± 2.41	>100

point mismatch and do not require an RL procedure. Some recent works (Liu et al., 2020) have proposed matching only state occupancy measures for imitation across dynamics mismatch. We compare our method to an appropriate distillation of such methods. MDP homomorphisms (Ravindran & Barto, 2002) have been explored with the aim of compressing state, action spaces to facilitate planning. In similar vein, related works have proposed MDP similarity metrics based on bisimulation methods (Ferns et al., 2004) and Boltzmann machine reconstruction error (Ammar et al., 2014). While conceptually related to our MDP alignability theory, these works have not proposed scalable procedures to discover the homomorphisms and have not drawn connections to domain adaptive learning.

6. Experiments

Our experiments were designed to answer the following questions: (1). Can GAMA uncover MDP reductions? (2). Can the learned alignments $(f_{\theta_f}, g_{\theta_g})$ be leveraged to succeed at DAIL? We propose two metrics to measure DAIL performance. First, *alignment complexity* which is the number of MDP pairs, i.e number of tasks, in the alignment task set needed to learn alignments that enable zero-shot imitation, given ample cross domain demonstrations for the target tasks. Second, *adaptation complexity* which is the amount of cross domain demonstrations for the target tasks needed to successfully imitate tasks in the self domain without querying the target task reward function, given a sufficiently large alignment task set. Note that we include experiments with MDP pairs that are *not perfectly alignable*,
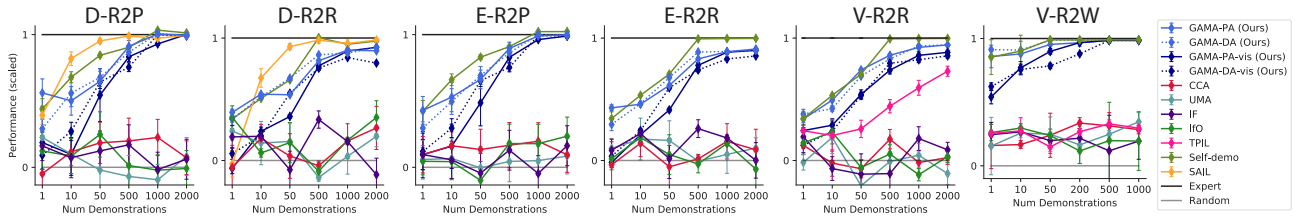
Figure 5. **Adaptation Complexity**. Notably, adaptation complexity of GAMA is close to that of the Self-demo baseline. Baselines fail at DAIL, mostly due to failing the alignment step. Results are averaged across 5 runs.

yet intuitively share structure, to show general applicability of GAMA for DAIL. We experiment with environments which are extensions of OpenAI Gym (Brockman et al., 2016): pen, cart, reacher2, reacher3, reach2-tp, snake3, and snake4 denotes the pendulum, cartpole, 2-link reacher, 3-link reacher, third person 2-link reacher, 3-link snake, and 4-link snake environments, respectively. (self domain) $\leftrightarrow$ (expert domain) specify an MDP pair in the alignment task set. Model architectures and environment details are further described Appendix B, C, D. We study two ablations of GAMA and compare against the following baselines:

GAMA - Policy Adapt (GAMA-PA): learns alignments by Algorithm 1, fits an expert policy $\pi_{y,\mathcal{T}}$ to $\mathcal{D}_{\mathcal{M}_{y,\mathcal{T}}}$ for a new target task $\mathcal{T}$ and zero-shot adapts $\pi_{y,\mathcal{T}}$ to the self domain via $\hat{\pi}_{x,\mathcal{T}} = g_{\theta_g} \circ \pi_{y,\mathcal{T}} \circ f_{\theta_f}$.

GAMA - Demonstration Adapt (GAMA-DA): trains f^{-1} in addition to Algorithm 1, adapts $\mathcal{D}_{\mathcal{M}_{y,\mathcal{T}}}$ into the self domain via (f^{-1}, g), and fits a self domain policy on the adapted demonstrations.

Self Demonstrations (Self-Demo): We behavioral clone on self domain demonstrations of the target task. This baseline sets an "upper bound" for the adaptation complexity.

Canonical Correlation Analysis (CCA) (Hotelling, 1936): finds invertible linear transformations to a space where domain data are maximally correlated when given unpaired, unaligned demonstrations.

Unsupervised Manifold Alignment (UMA) (Ammar et al., 2015): finds a map between states that have similar local geometries from unpaired, unaligned demonstrations.

Invariant Features (IF) (Gupta et al., 2017): finds invertible maps onto a feature space given state pairings. Dynamic Time Warping (Muller, 2007) is used to obtain the pairings.

Imitation from Observation (IfO) (Liu et al., 2018): learns a statemap conditioned on a cross domain observation given state pairings. Dynamic Time Warping (Muller, 2007) is used to obtain the pairings.

Third Person Imitation Learning (TPIL) (Stadie et al., 2017): simultaneously learns a domain agnostic feature space while matching distributions in the feature space.

State-Alignment Imitation Learning (Liu et al., 2020):

Distribution matching imitation learning with a state occupancy matching objective.

6.1. MDP Alignment Evaluation

Figure 4 visualizes the learned state map f_{θ_f} for several MDP pairs. The pen $\leftrightarrow$ pen alignment task (Figure 4, Top Left) and reach$\leftrightarrow$reach-tp (Table 1) task exemplify MDP pairs that are permutations. Similarly, the pen $\leftrightarrow$ cart alignment task (Figure 4, Top Right) has a reduction that maps the pendulum's angle and angular velocity to those of the pole, as the cart's position and velocity are redundant state dimensions once an optimal policy has been learned. Refer to Figure 7 in the Appendix for an explanation of poor UMA performance. Table 1 presents quantitative evaluations of these simple alignment maps. For pen$\leftrightarrow$pen and reach2$\leftrightarrow$reach2-tp we record the average ℓ_2 loss between the learned statemap prediction and the ground truth permutation. As for pen$\leftrightarrow$cart, we do the same on the dimensions for the pole's angle and angular velocity. Both Figure 4 and Table 1 shows that GAMA is able to learn simple reductions while baselines mostly fail to do so. The key reason behind this performance gap is that many baselines (Gupta et al., 2017; Liu et al., 2018) obtain state maps from time-aligned demonstration data using Dynamic Time Warping (DTW). However, the considered alignment tasks contains unaligned demonstrations with diverse starting states, up to 2x differences in demonstration lengths, and varying task execution rates. We see that GAMA also outperforms baselines that learn from unaligned demonstrations (Hotelling, 1936; Ammar et al., 2015) by learning maps that preserve MDP structure with more flexible neural network function approximators. For snake4 $\leftrightarrow$ snake3 and reach2 $\leftrightarrow$ reach3, the MDPs may not be perfectly alignable, yet they intuitively share structure. From Figure 4 (Bottom Left) we see that GAMA identically matches two adjacent joint angles of snake4 to the two joint angles of snake3 and the periodicity of the snake's wiggle is preserved. On reacher2$\leftrightarrow$reacher3, GAMA learns a state map that matches the first joint angles and states that have similar extents of contraction.

6.2. DAIL Performance

We evaluate DAIL performance on six problems that span embodiment, viewpoint, and dynamics mismatch scenarios.
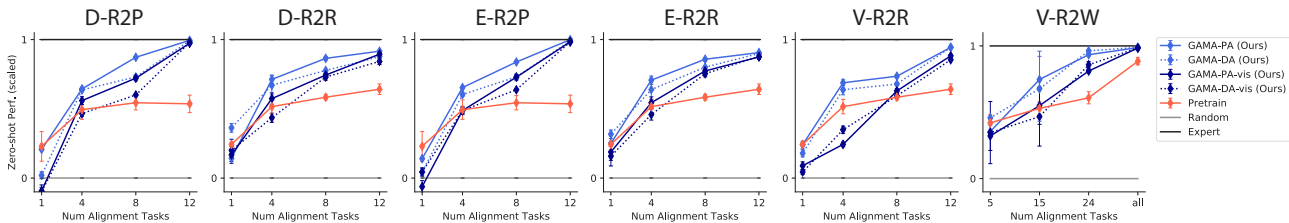
Figure 6. **Alignment Complexity**. Baselines cannot perform zero-shot imitation. Pretrain baseline shows the zero-shot performance of a policy directly pretrained on the self domain alignment tasks, when possible. Results are averaged across 5 runs.

See Appendix D for further details on each problem.

Dynamics-Reach2Reach (D-R2R): Self domain is reach2 and expert domain is reach2 with isotropic gaussian noise injected into the dynamics. We use the robot's joint level state-action space. The N alignment tasks are reaching for N goals and the target tasks are reaching for 12 new goals, placed maximally far away from the alignment task goals.

Dynamics-Reach2Push (D-R2P): Same as D-R2R except the target task is pushing a block to a goal location.

Embodiment-Reach2Reach (E-R2R): Self domain is reach2 and expert is reach3. Rest is the same as D-R2R.

Embodiment-Reach2Push (E-R2P): Self domain is reach2 and expert is reach3. Rest is the same as D-R2P.

Viewpoint-Reach2Reach (V-R2R): Self domain is reach2 and expert domain is reach2-tp1 that has the same "third person" view state space as that in (Stadie et al., 2017) with a $30°$ planar offset. We use the robot's joint level state-action space. The alignment/target tasks are the same as D-R2R.

Viewpoint-Reach2Write (V-R2W): Self domain is reach2 and expert domain is reach2-tp2 that has a different "third person" view state space with a $180°$ axial offset. We use the robot's joint level state space. The N alignment tasks are reaching for N goals and the target task is tracing letters as fast as possible. The transfer task differs from the alignment tasks in two key aspects: the end effector must draw a straight line from a letter's vertex to vertex and not slow down at the vertices in order to trace.

Alignment complexity on the six problems is shown in Figure 6. GAMA (light blue) is able to learn alignments that enable zero-shot imitation on the target task, showing clear gains over a simple pretraining procedure (orange) on the self domain MDPs in the alignment task set. Other baselines require an additional expensive RL step and thus cannot zero-shot imitate. Figure 5 shows the adaptation complexity. Notably, GAMA (light blue) produces adapted demonstrations of similar usefulness as self demonstrations (olive green). Most baselines fail to learn alignments from unpaired, unaligned demonstrations and as a result fails at DAIL. TPIL succeeds at V-R2R, but fails at V-R2W which has a significantly larger viewpoint mismatch than V-R2R.

SAIL outperforms GAMA and even the self-Demo baseline, but it's important to note that SAIL uses the self domain environment simulator unlike GAMA and Self-Demo.

6.3. DAIL with Visual Inputs

The non-visual environment experiments in the previous section demonstrate the limitations of the time-alignment assumptions made in prior work without confounding variables such as the difficulty of optimization in high-dimensional spaces. In this section, we introduce two more variants of our method, GAMA-PA-vis and GAMA-DA-vis, which demonstrate that *GAMA scales to higher dimensional, visual environments* with $64 \times 64 \times 3$ image states. Specifically, we train a deep spatial autoencoder on the alignment task set to learn an encoder with the architecture from (Levine et al., 2016), then apply GAMA on the (learned) latent space. Results are shown in Figure 6, 5. We see that the alignment and adaptation complexity of GAMA-PA-vis (dark-blue, solid), GAMA-DA-vis (dark-blue, dotted) are both similar to that of GAMA-DA (light blue, solid), GAMA-PA (light blue, dotted) and better than baselines trained with the robot's joint-level representation.

7. Discussion and future work

We've formalized Domain Adaptive Imitation Learning which encompasses prior work in transfer learning across embodiment (Gupta et al., 2017) and viewpoint differences (Stadie et al., 2017; Liu et al., 2018) along with a practical algorithm that can be applied to both scenarios. We now point out directions future work. Our MDP alignability theory is a first step towards formalizing possible shared structures that enable cross domain imitation. While we've shown that GAMA empirically works well even when MDPs are not perfectly alignable, upcoming works may explore relaxing the conditions for MDP alignability to develop a theory that covers an even wider range of real world MDPs. Future works may also try applying GAMA in the imitation from observations scenario, i.e actions are not available, by aligning observations with GAMA and applying methods from (Sermanet et al., 2018; Liu et al., 2018). Finally, we hope to see future works develop principled ways design a minimal alignment task set, which is analogous to designing a

minimal training set for supervised learning.

Acknowledgements

References

Ammar, H. B., Eaton, E., Ruvolo, P., and Taylor, M. E. An automated measure of mdp similarity for transfer in reinforcement learning. 2014.

Ammar, H. B., Eaton, E., Ruvolo, P., and Taylor, M. E. Unsupervised cross-domain transfer in policy gradient reinforcement learning via manifold alignment. 2015.

Billingsley, P. Convergence of probability measures. 1968.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Wojciech, Z. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Ferns, N., Panangaden, P., and Precup, D. Metrics for finite markov decision processes. In *UAI*, 2004.

Finn, C., Tan, X. Y., Duan, Y., Darrell, T., Levine, S., and Abbeel, P. Deep spatial autoencoders for visuomotor learning. *International Conference on Robotics and Automation*, 2015.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Gupta, A., Devin, C., Liu, Y. X., Abbeel, P., and Levine, S. Learning invariant feature spaces to transfer skills with reinforcement learning. *International Conference on Learning Representations*, 2017.

Ho, J. and Ermon, S. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*, pp. 4565–4573, 2016.

Hotelling, H. Relations between two sets of variates. *Biometrika*, 28, 1936.

Jones, S. S. The development of imitation in infancy. *Philos Trans R Soc Lond B Biol Sci.*, 364:2325–2335, 2009.

Joshi, G. and Chowdhary, G. Cross-domain transfer in reinforcement learning using target apprentice. 2018.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, December 2014.

Kuniyoshi, Y. and Inoue, H. Qualitative recognition of ongoing human action sequences. *International Joint Conference on Artificial Intelligence*, 1993.

Kuniyoshi, Y., Inaba, M., and Inoue, H. Learning by watching: Extracting reusable task knowledge from visual observation of human performance. *IEEE Trans. Robot. Autom.*, 10:799–822, 1994.

Levine, S., Finn, C., Darrell, T., and Abbeel, P. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509. 02971*, 2015.

Liu, F., Ling, Z., Mu, T., and Su, H. State alignment-based imitation learning. *International Conference on Learning Representations*, 2020.

Liu, Y., Gupta, A., Abbeel, P., and Levine, S. Imitation from observation: Learning to imitate behaviors from raw video via context translation. *arXiv preprint arXiv:1707.03374*, 2018.

Marshall, P. J. and Meltzoff, A. N. Body maps in the infant brain. *Trends Cogn Sci.*, 19:499–505, 2015.

Muller, M. Dynamic time warping. *Information retrieval for music and motion*, pp. 69–84, 2007.

Ortner, R. Combinations and mixtures of optimal policies in unichain markov decision processes are optimal. *arXiv preprint arXiv:0508319*, 2005.

Pomerleau, D. A. Efficient training of artificial neural networks for autonomous navigation. *Neural computation*, 3(1):88–97, 1991. ISSN 0899-7667.

Ravindran, B. and Barto, A. G. Model minimization in hierarchical reinforcement learning. In *SARA*, 2002.

Rizzolatti, G. and Craighero, L. The mirror neuron system. *Annual Review of Neuroscience*, 27:169–192, 2004.

Sermanet, P., Lynch, C., Chebotar, Y., Hsu, J., Jang, E., Schaal, S., and Levine, S. Time-contrastive networks: Self-supervised learning from video. *arXiv preprint arXiv:1704.06888*, 2018.

Stadie, B., Abbeel, P., and Sutskever, I. Third person imitation learning. In *ICLR*, 2017.

Syed, U., Bowling, M., and Schapire, R. E. Apprenticeship learning using linear programming. In *Proceedings of the 25th international conference on Machine learning*, pp. 1032–1039. ACM, July 2008. ISBN 9781605582054. doi: 10.1145/1390156.1390286.

Wang, T., Liao, R., Ba, J., and Fidler, S. NerveNet: Learning structured policy with graph neural networks. *International Conference on Learning Representations*, 2018, 2018.