Visually Robust Adversarial Imitation Learning from Videos with Contrastive Learning

Vittorio Giammarino[1], James Queeney[2] and Ioannis Ch. Paschalidis[3]

***Abstract*— We propose C-LAIfO, a computationally efficient algorithm designed for imitation learning from videos in the presence of visual mismatch between agent and expert domains. We analyze the problem of imitation from expert videos with visual discrepancies, and introduce a solution for robust latent space estimation using contrastive learning and data augmentation. Provided a visually robust latent space, our algorithm performs imitation entirely within this space using off-policy adversarial imitation learning. We conduct a thorough ablation study to justify our design and test C-LAIfO on high-dimensional continuous robotic tasks. Additionally, we demonstrate how C-LAIfO can be combined with other reward signals to facilitate learning on a set of challenging hand manipulation tasks with sparse rewards. Our experiments show improved performance compared to baseline methods, highlighting the effectiveness of C-LAIfO. To ensure reproducibility, we open source our code.**

I. INTRODUCTION

In recent years, there has been a significant surge in research on imitation learning from expert videos, commonly referred to as the *Visual Imitation from Observations (V-IfO)* problem. The approach of mimicking experts from videos holds great promise for the future, as it offers a cost-effective way to teach autonomous agents new skills and behaviors. To achieve this goal, prior research has developed methods capable of concurrently addressing two primary challenges of the V-IfO framework: the partial observability of the decision-making process and the absence of expert actions [1]. Despite these advancements, end-to-end state-of-the-art algorithms still face significant barriers in real-world applications due to the assumption that both the expert and learning agent operate within the same environment [1], [2]. For instance, consider the scenario described in Fig. 1, where expert videos are collected under the conditions in Fig. 1a and an autonomous agent is deployed in Fig. 1b or Fig. 1c. Current methods are not designed to handle such variations in lighting and background, leading to failures in these contexts. Our goal, in this paper, is to enhance the imitation capabilities of autonomous agents in the presence of visual mismatches.

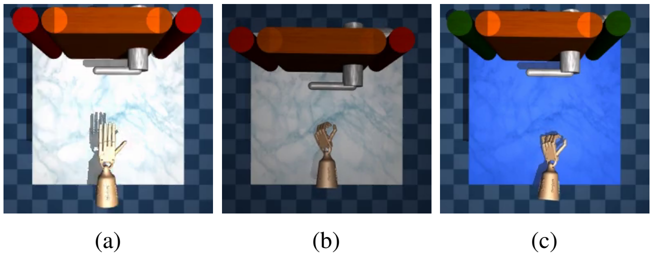


Fig. 1: Robotic manipulation task. Current end-to-end methods for imitation from expert videos assume that the expert and the agent operate in the same environment. Consequently, they are unable to handle variations in lighting or background.

We introduce a novel end-to-end pipeline for imitation from expert videos with visual mismatch. We begin by analyzing the V-IfO problem with visual mismatch and propose a novel, simple, and computationally efficient algorithm called Contrastive Latent Adversarial Imitation from Observations (C-LAIfO). Notably, C-LAIfO builds upon the recent LAIfO algorithm [1] and achieves visually robust latent state estimation through data augmentation and contrastive learning techniques [3], [4]. We justify each design choice for our algorithm, including the types of data augmentation and contrastive loss, through a comprehensive ablation study. Furthermore, we compare C-LAIfO against two V-IfO baselines—LAIfO [1] and PatchAIL [2]—as well as DisentanGAIL [5], which serves as a baseline for V-IfO with domain mismatch. Additionally, we show how the reward signal learned from expert data using C-LAIfO can be easily integrated with other signals to enhance efficiency and enable learning in robotic tasks with sparse reward functions. Therefore, we further evaluate our algorithm on the Adroit platform for dynamic dexterous manipulation [6]. These additional experiments highlight the versatility of our approach, showcasing its efficacy in handling complex robotic tasks.

II. RELATED WORK

a) Imitation from observation: Imitation Learning (IL) is a powerful approach that allows agents to mimic expert behavior by using demonstrations of a task typically in the form of state-action pairs. Our work builds on *Adversarial Imitation Learning (AIL)* [7], [8] which frames IL as a two-player game between a discriminator and an agent's policy. Here, the discriminator distinguishes whether a state-action pair is generated by the agent or the expert policy. In practice, AIL is formulated as a joint process of *Reinforcement Learning (RL)* and inverse RL [9], [10], [11], [12]. First a

VG and ICP were partially supported by the DOE under grants DE-AC02-05CH11231 and DEEE0009696, the NSF under grants CCF2200052, DMS1664644, and IIS1914792, and by the ONR under grants N0001419-12571 and N000142112844. JQ was exclusively supported by Mitsubishi Electric Research Laboratories (MERL).

[1] Vittorio Giammarino is with Division of Systems Engineering, Boston University, Boston, MA 02215, USA vgiammar@bu.edu

[2] James Queeney is with Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA 02139, USA queeney@merl.com

[3] Ioannis Ch. Paschalidis is with the Department of Electrical and Computer Engineering, Division of Systems Engineering, and Department of Biomedical Engineering, Boston University, Boston, MA 02215, USA yannisp@bu.edu

reward function is inferred from expert demonstrations and then it is used in the RL step to train agents. In scenarios with partial observability, AIL has been applied to cases with missing information [13] and to Visual IL, where agents learn from video frames as state observations [14]. Compared to standard IL, Imitation from observation [15], [16], [17] assumes that action information is not observable in the demonstrations data. This setting is more practical compared to IL but also more difficult to tackle. The combination of learning from videos in the absence of expert actions gives rise to the V-IfO problem, which is the primary focus of our work. End-to-end state-of-the-art algorithms for the V-IfO setting include PatchAIL [2], which applies AIL directly on the pixel space utilizing a PatchGAN discriminator [18], [19], and LAIfO [1], where AIL operates on a latent representation of the agent state. Notably, these approaches are built on the assumption that both the expert and the learning agent act within the same decision process, which rarely holds true in real-world scenarios.

b) Imitation from videos with environment mismatch: Our research targets imitation from visual observations in the presence of mismatches between the expert and the learner environments, a problem referred to as third-person IL [20], domain-adaptive IL [21], or cross-domain IL [22]. Solutions in the literature either decompose this problem into sequential stages or formulate end-to-end methods. Sequential approaches include: [23], where a reward function is learned by leveraging video prediction with context translation; [24], where reward functions are obtained using time-contrastive networks trained offline; [25], where cycle-consistent adversarial networks [19] are trained offline to generate instruction images in the agent domain from videos; [26], which uses inverse models and adversarial domain adaptation [27] to train navigation policies from videos, and [28], where 3D trajectory reconstruction from videos is used to obtain physically plausible trajectories. Our work differs from this literature as we formulate a fully end-to-end approach.

c) End-to-end algorithms for imitation from videos with mismatch: Previous end-to-end solutions were presented in [20], [29], [5], [30]. The studies in [20] and [5] extract domain-independent features to infer domain-independent reward functions. More specifically, in [20], the authors propose to learn domain-independent features using an adversarial approach similar to [31], while DisentanGAIL in [5] achieves a similar result by adding a mutual information constraint to the binary cross-entropy loss used for AIL. Similar to our algorithm, these studies formulate fully end-to-end model-free algorithms that avoid costly generative steps during the imitation process. Our approach adopts similar reasoning to [20], [5]; however, we leverage contrastive learning for domain-independent feature extraction and build the entire AIL pipeline (both reward inference and RL step) on this learned feature space, rather than only the reward inference as done in [20], [5]. As shown in our experiments, this leads to significant improvements in performance. Other works rely on expensive generative steps to address the mismatch problem. In [29], imitation is performed using expert observation-action pairs through a learned domain-agnostic recurrent state space model [32]. Our algorithm, on the other hand, is model-free and only requires expert observations. In [30], cycle-consistent adversarial networks [19] are trained online to generate expert videos in the agent's domain, thus reducing the problem to the standard V-IfO without mismatches. Our approach does not require such a generative step, as it learns a domain-independent feature space directly.

III. Preliminaries

We use uppercase letters (e.g., S_t) for random variables, lowercase letters (e.g., s_t) for values of random variables, script letters (e.g., $\mathcal{S}$) for sets, and bold lowercase letters (e.g., $\boldsymbol{\theta}$) for vectors. Let $[t_1 : t_2]$ be the set of integers t such that $t_1 \leq t \leq t_2$; we write S_t such that $t_1 \leq t \leq t_2$ as $S_{t_1:t_2}$. We denote with $\mathbb{E}[\cdot]$ expectation, with $\mathbb{P}(\cdot)$ probability, and with $\mathbb{D}_f(\cdot,\cdot)$ an f-divergence between two distributions of which the Jensen-Shannon divergence, $\mathbb{D}_{\text{JS}}(\cdot||\cdot)$, is a special case.

a) Partially Observable Markov Decision Process: We model the decision processes as infinite-horizon discounted Partially Observable Markov Decision Processes (POMDPs) described by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{X}, \mathcal{T}, \mathcal{U}, \mathcal{R}, \rho_0, \gamma)$, where $\mathcal{S}$ is the set of states, $\mathcal{A}$ is the set of actions, and $\mathcal{X}$ is the set of observations. $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to P(\mathcal{S})$ is the transition probability function where $P(\mathcal{S})$ denotes the space of probability distributions over $\mathcal{S}$, $\mathcal{U} : \mathcal{S} \to P(\mathcal{X})$ is the observation probability function, and $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function which maps state-action pairs to scalar rewards. Finally, $\rho_0 \in P(\mathcal{S})$ is the initial state distribution and $\gamma \in [0, 1)$ the discount factor. The true environment state $s \in \mathcal{S}$ is unobserved by the agent. Given an action $a \in \mathcal{A}$, the next state is sampled such that $s' \sim \mathcal{T}(\cdot|s, a)$, an observation is generated as $x' \sim \mathcal{U}(\cdot|s')$, and a reward $\mathcal{R}(s, a)$ is computed. Note that an MDP is a special case of a POMDP where the underlying state s is directly observed.

b) Reinforcement learning: Given an MDP and a stationary policy $\pi : \mathcal{S} \to P(\mathcal{A})$, the RL objective is to maximize the expected total discounted return $J(\pi) = \mathbb{E}_\tau[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t)]$ where $\tau = (s_0, a_0, s_1, a_1, \dots)$. A stationary policy π induces a normalized discounted state visitation distribution defined as $d_\pi(s) = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t \mathbb{P}(s_t = s|\rho_0, \pi, \mathcal{T})$, and we define the corresponding normalized discounted state-action visitation distribution as $\rho_\pi(s, a) = d_\pi(s)\pi(a|s)$. Finally, we denote the state value function of π as $V^\pi(s) = \mathbb{E}_\tau[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t)|S_0 = s]$ and the state-action value function as $Q^\pi(s, a) = \mathbb{E}_\tau[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t)|S_0 = s, A_0 = a]$. When a function is parameterized with parameters $\boldsymbol{\theta} \in \Theta \subset \mathbb{R}^k$ we write $\pi_{\boldsymbol{\theta}}$.

c) Generative adversarial imitation learning: Assume we have a set of expert demonstrations $\tau_E = (s_{0:T}, a_{0:T})$ generated by the expert policy π_E, a set of trajectories $\tau_{\boldsymbol{\theta}}$ generated by the policy $\pi_{\boldsymbol{\theta}}$, and a discriminator network $D_{\boldsymbol{\chi}} : \mathcal{S} \times \mathcal{A} \to [0, 1]$ parameterized by $\boldsymbol{\chi}$. Generative adversarial IL [7] optimizes the min-max objective

$$\min_{\boldsymbol{\theta}} \max_{\boldsymbol{\chi}} \; \mathbb{E}_{\tau_E}[\log(D_{\boldsymbol{\chi}}(s, a))] + \mathbb{E}_{\tau_{\boldsymbol{\theta}}}[\log(1 - D_{\boldsymbol{\chi}}(s, a))]. \tag{1}$$

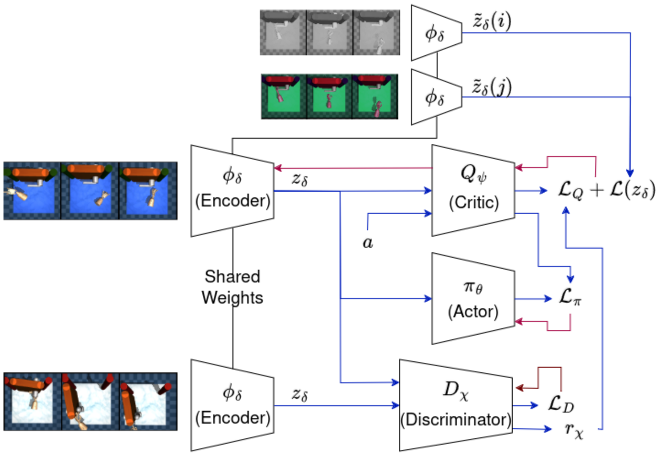


Fig. 2: Summary of C-LAIfO. The target-POMDP has a blue background while the source-POMDP has a white background. In the diagram, black lines indicate shared weights among networks, blue arrows indicate forward pass through the networks, and red arrows indicate backward pass. The losses $\mathcal{L}_D$, $\mathcal{L}_Q$ and $\mathcal{L}(z_{\boldsymbol{\delta}})$ are respectively in (2), (3), and (5). $\mathcal{L}_\pi$ indicates the deterministic actor-critic loss [34].

Maximizing (1) with respect to $\boldsymbol{\chi}$ is effectively an inverse RL step where a reward function, $r_{\boldsymbol{\chi}}(s,a) = -\log(1-D_{\boldsymbol{\chi}}(s,a))$, is inferred by leveraging τ_E and $\tau_{\boldsymbol{\theta}}$. Minimizing (1) with respect to $\boldsymbol{\theta}$ is an RL step, where the agent aims to minimize its expected cost. Optimizing (1) is equivalent to minimizing $\mathbb{D}_{\text{JS}}(\rho_{\pi_{\boldsymbol{\theta}}}(s,a)||\rho_{\pi_E}(s,a))$, so we are recovering the expert state-action visitation distribution [33].

d) Modeling the visual mismatch in POMDPs: Traditionally, the V-IfO problem assumes that both the expert and the agent operate within the same POMDP. Throughout this paper we relax this assumption and define two different decision processes: namely a *target-POMDP* for the agent and a *source-POMDP* for the expert. The target-POMDP is characterized by the tuple $(\mathcal{S},\mathcal{A},\mathcal{X},\mathcal{T},\mathcal{U}_T,\mathcal{R},\rho_0,\gamma)$, whereas the source-POMDP is characterized by the tuple $(\mathcal{S},\mathcal{A},\mathcal{X},\mathcal{T},\mathcal{U}_S,\mathcal{R},\rho_0,\gamma)$. The primary distinction between these POMDPs lies in their observation probability functions. Despite sharing identical state and action spaces, given the same state s_t, the expert's observation $x^S_t \sim \mathcal{U}_S(\cdot|s_t)$ from the source-POMDP and the agent's observation $x^T_t \sim \mathcal{U}_T(\cdot|s_t)$ from the target-POMDP may be different (i.e., we may have $x^S_t \neq x^T_t$). We refer to this as visual mismatch.

IV. Contrastive Latent Adversarial Imitation from Observations

Considering a target-POMDP and a source-POMDP as introduced in the previous section, we can identify two levels of information in the observation space $\mathcal{X}$: (i) information related to task completion, and (ii) visual distractors that do not contribute to task completion. Thus, we define $\mathcal{X}$ as $\mathcal{X} = (\bar{\mathcal{X}},\hat{\mathcal{X}})$, where $\bar{\mathcal{X}}$ represents the *goal-completion information* that is invariant between source-POMDP and target-POMDP; whereas, $\hat{\mathcal{X}}$ represents the set of *visual distractors* that do not contribute to goal completion. We express the source and target observations respectively as $x^S_t = (\bar{x}^S_t,\hat{x}^S_t)$ and $x^T_t = (\bar{x}^T_t,\hat{x}^T_t)$. Our objective is to filter out the visually-distracting information $\hat{\mathcal{X}}$ from both the source and the target observations while retaining the goal-completion information $\bar{\mathcal{X}}$ to effectively solve the V-IfO problem. This objective can be achieved by attaining *domain invariance in a feature space* $\mathcal{Z}$. As a result, our goal becomes to learn $\mathcal{Z}$ such that only goal-completion information is retained while the visually-distracting information is discarded (cf. Appendix VI-A for formal analysis).

In the following, we present the main components of our algorithm C-LAIfO, which performs imitation directly in a domain-invariant feature space $\mathcal{Z}$ (Sec. IV-A). In order to do so, we learn a domain-invariant encoder $\phi_{\boldsymbol{\delta}}$ that can successfully map $x^T_{\leq t}$ and $x^S_{\leq t}$ to z_t through two main steps. First, we train the encoder $\phi_{\boldsymbol{\delta}}$ alongside the critic networks $Q_{\boldsymbol{\psi}_k}$ where $k = \{1,2\}$ (Sec. IV-B). This step is essential for solving the imitation problem and embedding goal-completion information within the latent space $\mathcal{Z}$. We further train $\phi_{\boldsymbol{\delta}}$ to optimize an auxiliary contrastive loss and perform randomized augmentation of the observations, taking into account the type of visual mismatch between the source and the target domains (Sec. IV-C). This step is crucial to efficiently discard visually-distracting information from $\mathcal{Z}$. A schematic diagram summarizing the whole C-LAIfO pipeline is provided in Fig. 2.

A. *Adversarial imitation in latent space*

Given a domain-invariant feature space $\mathcal{Z}$, our AIL pipeline is defined as follows. We initialize two replay buffers $\mathcal{B}_E$ and $\mathcal{B}$ to respectively store the sequences of observations generated by the expert and the agent policies, from which we infer the latent state-transitions $(z_{\boldsymbol{\delta}},z'_{\boldsymbol{\delta}})$. We write $(z_{\boldsymbol{\delta}},z'_{\boldsymbol{\delta}}) \sim \mathcal{B}$ to streamline the notation. Then, given a discriminator $D_{\boldsymbol{\chi}}:\mathcal{Z}\times\mathcal{Z}\to[0,1]$, we write

$$\begin{aligned}\max_{\boldsymbol{\chi}}\ &\mathbb{E}_{(z_{\boldsymbol{\delta}},z'_{\boldsymbol{\delta}})\sim\mathcal{B}_E}[\log(D_{\boldsymbol{\chi}}(z_{\boldsymbol{\delta}},z'_{\boldsymbol{\delta}}))] \\ &+\mathbb{E}_{(z_{\boldsymbol{\delta}},z'_{\boldsymbol{\delta}})\sim\mathcal{B}}[\log(1-D_{\boldsymbol{\chi}}(z_{\boldsymbol{\delta}},z'_{\boldsymbol{\delta}}))].\end{aligned} \tag{2}$$

As mentioned, alternating (2) with an RL step using $r_{\boldsymbol{\chi}}(z_{\boldsymbol{\delta}},z'_{\boldsymbol{\delta}}) = -\log(1-D_{\boldsymbol{\chi}}(z_{\boldsymbol{\delta}},z'_{\boldsymbol{\delta}}))$ leads to the minimization of $\mathbb{D}_{\text{JS}}\big(\rho_{\pi_{\boldsymbol{\theta}}}(z_{\boldsymbol{\delta}},z'_{\boldsymbol{\delta}})||\rho_{\pi_E}(z_{\boldsymbol{\delta}},z'_{\boldsymbol{\delta}})\big)$ [35]. Therefore, we are effectively imitating the expert in the latent space $\mathcal{Z}$. Note that the presented AIL can only succeed if $\mathcal{Z}$ is domain-invariant and embeds the relevant goal-completion information necessary to solve the imitation problem. Next, we show how our algorithm C-LAIfO addresses this challenge.

B. *Critic and encoder training step*

We define the encoder as $\phi_{\boldsymbol{\delta}}:\mathcal{X}^d\to\mathcal{Z}$, a function mapping sequences of $d\in\mathbb{N}$ observations to the latent space $\mathcal{Z}$. Specifically, we write $z_{\boldsymbol{\delta}} = \phi_{\boldsymbol{\delta}}(x_{t^-:t})$ and $z'_{\boldsymbol{\delta}} = \phi_{\boldsymbol{\delta}}(x_{t^-+1:t+1})$ where $t - t^- + 1 = d$. When a data augmentation function $\text{aug}(\cdot)$ is applied to the sequence of observations, we write $\tilde{z}_{\boldsymbol{\delta}} = \phi_{\boldsymbol{\delta}}(\text{aug}(x_{t^-:t}))$ and $\tilde{z}'_{\boldsymbol{\delta}} = \phi_{\boldsymbol{\delta}}(\text{aug}(x_{t^-+1:t+1}))$. We train

$\phi_{\boldsymbol{\delta}}$ to optimize

$$\min_{\boldsymbol{\psi}_k, \boldsymbol{\delta}} \mathbb{E}_{(\tilde{z}_{\boldsymbol{\delta}}, a_t, \tilde{z}'_{\boldsymbol{\delta}}) \sim \mathcal{B}} \big[\big(Q_{\boldsymbol{\psi}_k}(\tilde{z}_{\boldsymbol{\delta}}, a_t) - \mathrm{sg}(y) \big)^2 \big] + \mathbb{E}_{\tilde{z}_{\boldsymbol{\delta}} \sim \mathcal{B}} [\mathcal{L}(\tilde{z}_{\boldsymbol{\delta}})] \tag{3}$$

$$\text{s.t. } y = r_{\boldsymbol{\chi}}(z_{\boldsymbol{\delta}}, z'_{\boldsymbol{\delta}}) + \gamma \min_{k=1,2} Q_{\bar{\boldsymbol{\psi}}_k}(\tilde{z}'_{\boldsymbol{\delta}}, a'). \tag{4}$$

The steps in (3)–(4) follow the deep Q-network optimization pipeline [36], [37], where we add a contrastive auxiliary loss $\mathbb{E}_{\tilde{z}_{\boldsymbol{\delta}} \sim \mathcal{B}}[\mathcal{L}(\tilde{z}_{\boldsymbol{\delta}})]$ in (3) defined on the encoder $\phi_{\boldsymbol{\delta}}$. In (4), the reward function $r_{\boldsymbol{\chi}}(z_{\boldsymbol{\delta}}, z'_{\boldsymbol{\delta}})$ is computed through the AIL step in (2). Note that we do not perform data augmentation when computing $r_{\boldsymbol{\chi}}$ since, during reward inference, we are deploying and not training the encoder $\phi_{\boldsymbol{\delta}}$. In practice, adding data augmentation to AIL in (2) decreases the performance. Refer to the supplementary material for empirical evidence on this claim. In (3), the encoder $\phi_{\boldsymbol{\delta}}$ is trained together with the critic networks $Q_{\boldsymbol{\psi}_k}$ ($k = \{1, 2\}$) in order to regress y in (4), where $\mathrm{sg}(\cdot)$ stands for stop gradient of the encoder parameters $\boldsymbol{\delta}$. The value of y in (4) is computed by summing $r_{\boldsymbol{\chi}}$ with the discounted target critic network at time $t + 1$. In (4), $a' = \pi_{\boldsymbol{\theta}}(\tilde{z}'_{\boldsymbol{\delta}}) + \epsilon$ where $\epsilon \sim \mathrm{clip}(\mathcal{N}(0, \sigma^2), -c, c)$ is a clipped exploration noise with c the clipping parameter and $\mathcal{N}(0, \sigma^2)$ a univariate normal distribution with zero mean and σ standard deviation. $\bar{\boldsymbol{\psi}}_1$ and $\bar{\boldsymbol{\psi}}_2$ are the slow moving weights for the target critic networks. $\mathcal{B}$ is a replay buffer initialized to store interactions (x_t^T, a_t, x_{t+1}^T) of the agent with the target environment. Note that the latent state-transitions $(\tilde{z}_{\boldsymbol{\delta}}, \tilde{z}'_{\boldsymbol{\delta}})$ are inferred from sequences of observations using $\phi_{\boldsymbol{\delta}}$ and, as above, we write $(\tilde{z}_{\boldsymbol{\delta}}, \tilde{z}'_{\boldsymbol{\delta}}) \sim \mathcal{B}$ to streamline the notation. Refer to the supplementary material for the complete pseudo-code.

By solving the optimization problem in (3)–(4), our primary goal is to train both the encoder network $\phi_{\boldsymbol{\delta}}$ and the critic networks $Q_{\boldsymbol{\psi}_k}$ to solve the RL problem with reward $r_{\boldsymbol{\chi}}$. In other words, this step focuses on retaining the goal-completion information within the latent space $\mathcal{Z}$ such that critic networks $Q_{\boldsymbol{\psi}_k}$ are successfully learned. We show in our ablation study that backpropagating the gradient from $Q_{\boldsymbol{\psi}_k}$ to $\phi_{\boldsymbol{\delta}}$ is an important step for achieving this goal and solving the imitation problem. Similarly, the types of augmentations performed on the sequences of observations and the choice of auxiliary loss play a crucial role in discarding the visually-distracting information from $\mathcal{Z}$ and dealing with the visual mismatch.

C. Contrastive loss

In the following, we introduce the data augmentation techniques and the auxiliary loss $\mathcal{L}$ in (3) for C-LAIfO. We opt for a contrastive method as this leads to good empirical results and good computational efficiency. Contrastive learning constructs low-dimensional representations of high-dimensional data by maximizing agreement between augmented views of the same data example via a contrastive loss in the latent space $\mathcal{Z}$. In our specific case, we define equivalent data as sequences of observations with the same goal-completion information. The data augmentation is randomized by considering a set of pre-determined functions. We will show in our experiments (Section V) that both the choice of contrastive loss and the set of augmentation functions play an important role in filtering out visually-distracting information.

First, a stochastic data augmentation module transforms any given sequence of observations $x_{t^-:t}$ into two views, denoted $\mathrm{aug}(x_{t^-:t})_i$ and $\mathrm{aug}(x_{t^-:t})_j$, which are denoted as the positive pairs. Note that two positive pairs must contain the same goal-completion information. Next, the encoder $\phi_{\boldsymbol{\delta}} : \mathcal{X}^d \to \mathcal{Z}$ extracts representation vectors from augmented sequences of observations. We write $\tilde{z}_{\boldsymbol{\delta}}(i) = \phi_{\boldsymbol{\delta}}(\mathrm{aug}(x_{t^-:t})_i)$ and $\tilde{z}_{\boldsymbol{\delta}}(j) = \phi_{\boldsymbol{\delta}}(\mathrm{aug}(x_{t^-:t})_j)$. Finally, we apply the contrastive loss function

$$\mathcal{L}(z_{\boldsymbol{\delta}}) = -\log \frac{\exp(\mathrm{sim}(\tilde{z}_{\boldsymbol{\delta}}(i), \tilde{z}_{\boldsymbol{\delta}}(j))/\eta)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\mathrm{sim}(\tilde{z}_{\boldsymbol{\delta}}(i), \tilde{z}_{\boldsymbol{\delta}}(k))/\eta)}, \tag{5}$$

where $\mathbb{1}_{[k \neq i]} \in \{0, 1\}$ is an indicator function equal to 1 if $k \neq i$, η denotes a temperature parameter, and $\mathrm{sim}(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u}^\top \boldsymbol{v} / \|\boldsymbol{u}\| \|\boldsymbol{v}\|$ denotes the cosine similarity. We sample a batch of N sequences of observations from the buffer $\mathcal{B}$ and define pairs of augmented sequences derived from the batch, resulting in $2N$ data points. The negative data points are not sampled explicitly. Instead, given a positive pair, we treat the other $2(N-1)$ augmented data points within the batch as negatives. Note that the loss in (5) is called the normalized temperature-scaled cross entropy loss or the Information Noise-Contrastive Estimation (InfoNCE) loss [38] and represents an upper bound of the negative mutual information between positive pairs. Therefore, by minimizing (5) as in (3) we are maximizing the mutual information between positive pairs in the latent space $\mathcal{Z}$.

V. Experiments

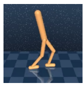
In this section, we begin with an ablation study to justify the design choices of our algorithm (Sec. V-A). Next, we demonstrate how C-LAIfO effectively handles various types of visual mismatches in the V-IfO setting (Sec. V-B). Finally, we showcase how C-LAIfO facilitates learning in challenging robotic manipulation tasks with sparse rewards and realistic visual inputs (Sec. V-C). In all the experiments, we use DDPG [39] to train experts in a fully observable setting and collect 100 episodes of expert data. The learned policies are evaluated based on average return over 10 episodes. We report the mean and standard deviation of the final return over 6 seeds and **highlight** the best performance.

A. Ablation study

In this section, we perform the following ablations:

1) **Contrastive loss function**: We demonstrate the importance of the contrastive loss function type by comparing the InfoNCE loss in (5) with BYOL in [4].
2) **Gradient backpropagation**: We highlight the necessity of backpropagating the gradient from $Q_{\boldsymbol{\psi}_k}$ to $\phi_{\boldsymbol{\delta}}$ in (3) for solving the imitation problem and embedding the goal-completion information in the latent space $\mathcal{Z}$.
3) **Data augmentation**: We emphasize the importance of selecting the appropriate augmentation for a given mismatch, showing that a mismatch-informed augmentation outperforms general augmentations or no augmentation.

TABLE I: Summary of the ablation experiments. All algorithms are trained for 10^6 steps. The experiments *C-LAIfO w/o Q backprop*, *C-LAIfO full aug*, and *C-LAIfO w/o aug* are only conducted in the easier setting due to their low performance.

Source Env: , Performance = 950		
Target Env		
C-LAIfO	**808 ± 269**	**768 ± 231**
BYOL-LAIfO	707 ± 337	142 ± 124
C-LAIfO w/o Q backprop	48.7 ± 8.7	-
C-LAIfO full aug	96.8 ± 53.4	-
C-LAIfO w/o aug	113 ± 25.3	-

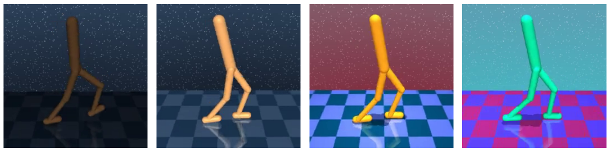


Fig. 3: Different environments used for the experiments in Table II and the PCA in Fig. 4 and 5.

TABLE II: Summary of the experiments for the mismatches in Fig. 3. The Light experiment consists in (3b) as source-POMDP and (3a) as target-POMDP. The Full experiments have (3b) as target-POMDP and (3c) as source-POMDP. We train all the algorithms in the Light mismatch for 10^6 steps and in the Full mismatch for 2×10^6 steps.

	Light	Full
C-LAIfO	**895 ± 36.6**	**509 ± 235**
LAIfO [1] w/ data aug	64.6 ± 62.9	206 ± 210
DisentanGAIL [5]	28.1 ± 8.7	30.6 ± 13.2
PatchAIL [2] w/ data aug	18.4 ± 5.6	122 ± 66.6

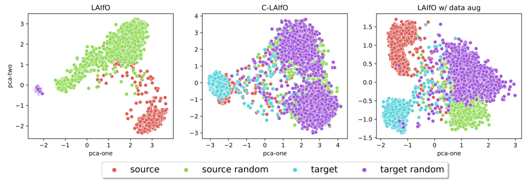


Fig. 4: PCA results for the Light experiment in Table II.

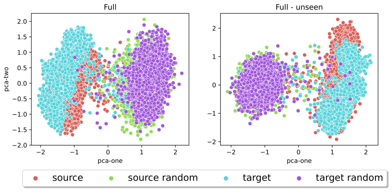


Fig. 5: PCA results on C-LAIfO for the Full experiment in Table II and the unseen environment in Fig. 3d.

These results are summarized in Table I, which includes results for C-LAIfO and its various configurations. All the learning curves are provided in the supplementary material. In Table I, *C-LAIfO* is implemented as described in Section IV, with the data augmentation function aug($\cdot$) defined as a brightness transformation. In *BYOL-LAIfO*, we retain the design identical to C-LAIfO except for replacing the InfoNCE loss in (5) with BYOL [4]. In *C-LAIfO w/o Q backprop*, we disable gradient backpropagation from $Q_{\psi k}$ to ϕ_δ in (3). Lastly, in *C-LAIfO full aug* and *C-LAIfO w/o aug*, we respectively modify the data augmentation function aug($\cdot$) to include full augmentation (brightness, color, and geometric transformations, as detailed in the supplementary material) and no augmentation. Notably, without backpropagating the gradient from Q_{ψ_k} to ϕ_δ, C-LAIfO fails to solve the imitation task even in the simplest mismatch scenario. This result demonstrates the importance of this step for embedding goal-completion information in $\mathcal{Z}$. Similar considerations can be done for the design of aug($\cdot$) where C-LAIfO struggles to efficiently solve the task both when the augmentation is too generic and when it is absent. These results highlight the critical role of properly defining aug($\cdot$) for efficient visual generalization in $\mathcal{Z}$. Finally, the superior performance of the InfoNCE loss in (5) compared to BYOL is evident in handling the hardest mismatch in Table I.

B. Visual Imitation from Observations with mismatch

In this section, we test C-LAIfO in the V-IfO setting with different types of mismatches (cf. Fig. 3) and compare it with three baselines: LAIfO [1] and PatchAIL [2], both equipped with the same aug($\cdot$) used for C-LAIfO, and DisentanGAIL [5]. The results, summarized in Table II, demonstrate that C-LAIfO successfully addresses the V-IfO with mismatch problem, achieving superior performance compared to all the baselines across the proposed mismatches. All the learning curves are provided in the supplementary material. For the *Light* experiment, aug($\cdot$) is defined as a brightness transformation; while for the others it is defined as a color transformation. Details on aug($\cdot$) are provided in the supplementary material. Furthermore, to assess whether C-LAIfO achieves *domain invariance in the feature space* $\mathcal{Z}$, we perform PCA on the latent space $\mathcal{Z}$ learned by different algorithms during training. In Fig. 4 we compare the latent space $\mathcal{Z}$ learned by LAIfO, C-LAIfO, and LAIfO with data augmentation in the Light setting from Table II. Specifically, we define *source* and *target* as the observations generated by an optimal policy in the source and target POMDPs, respectively. Similarly, *source random* and *target random* are observations generated by random policies. These observations are processed with the encoder $\phi_\delta : \mathcal{X}^d \to \mathcal{Z}$ trained using the respective algorithms. We perform PCA on this set of latent variables $z_{1:T}$ and plot the first two principal components. The results show that C-LAIfO is the only algorithm capable of filtering out visual distractors and clustering together data points with the same goal-completion information. In Fig. 5, we focus on C-LAIfO and test for

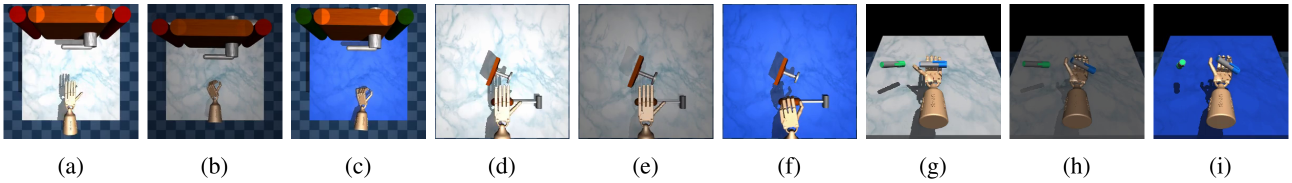


Fig. 6: Adroit environments used for the experiments in Table III.

TABLE III: Summary of the experiments in Fig. 6. The Door-Light and Door-Color experiments consider (6a) as the source-POMDP and (6b) and (6c) as the respective target-POMDPs. Similarly, the Hammer-Light and Hammer-Color experiments consider (6d) as source-POMDP and (6e) and (6f) as the respective target-POMDPs, while the Pen-Light and Pen-Color experiments consider (6g) as source-POMDP and (6h) and (6i) as the respective target-POMDPs. We use the VRL3 pipeline in [40] to train expert policies and collect 100 episodes of expert data. For these experiments, all the algorithms are trained for 4×10^6 steps and we report mean and standard deviation over 10 random seeds.

	Door		Hammer		Pen	
	Light	Color	Light	Color	Light	Color
Expert	170		184		73	
RL+LAIfO [1]	106 ± 75	96 ± 80	181 ± 4.0	103 ± 86	59 ± 12	59 ± 4.8
RL+C-LAIfO	**165 ± 5.8**	**160 ± 12**	**183 ± 1.4**	**178 ± 11**	**64 ± 5.6**	**62 ± 6.9**

generalization to the unseen environment in Fig. 3d. We train ϕ_δ on the Full setting from Table II and perform PCA as described. The results in the unseen setting match those in the Full experiment, indicating that ϕ_δ trained on (3c) can successfully generalize to (3d). Refer to the supplementary material also for the t-SNE visualization and additional details.

C. C-LAIfO for dexterous manipulation

In the following section, we evaluate our algorithm on a series of challenging robotic manipulation tasks from the Adroit platform for dynamic dexterous manipulation [6]. These experiments demonstrate how the reward r_χ, learned by C-LAIfO from expert videos, can be effectively combined with a sparse reward $\mathcal{R}$, collected by the agent through interaction with the environment, to enhance learning efficiency. The RL problem, therefore, aims to maximize the total reward $\mathcal{R}_{\text{tot}} = \mathcal{R}(s_t, a_t) + r_\chi(z_t, z_{t+1})$, where r_χ is learned through the AIL step in (2). This approach is particularly relevant for robotic tasks, where sparse rewards are often the most feasible option in real-world settings. However, relying solely on sparse rewards can make learning highly challenging and inefficient [41]. In this context, leveraging expert videos can significantly enhance efficiency. We compare C-LAIfO with the standard LAIfO algorithm [1], which does not explicitly address the visual mismatch between source and target POMDPs. Both C-LAIfO and LAIfO utilize an encoder to process pixel observations and extract embeddings in $\mathcal{Z}$, which are then concatenated with robot sensory observations. Notably, the expert's sensory observations are not used in the imitation process, as we only assume access to expert videos. Our approach, denoted as RL+C-LAIfO (or RL+LAIfO), seeks to maximize $\mathcal{R}_{\text{tot}}$, rather than just r_χ, as in the standard imitation learning problem. The results, summarized in Table III, show that C-LAIfO more effectively leverages expert videos with visual mismatches to facilitate learning when compared to LAIfO. This demonstrates the potential of our approach to enable learning in challenging robotic tasks by utilizing a minimal form of supervision, relying solely on expert videos. All learning curves are provided in the supplementary material.

VI. Conclusion

In this work, we analyze the V-IfO problem with visual mismatches and propose a novel algorithm named C-LAIfO as an effective solution. Through comprehensive ablation studies, we provide insights into our design and demonstrate the superior performance of our approach compared to a range of baselines in imitation from videos under various mismatch scenarios. Furthermore, we illustrate how C-LAIfO effectively utilizes expert videos with visual mismatches to facilitate learning in challenging hand manipulation tasks characterized by sparse rewards and realistic visual inputs.

A main limitation of the current approach is given by the reliance of C-LAIfO on a well-designed, possibly mismatch-informed, data augmentation function. As illustrated in our ablations in Section V, general augmentation can lead to poor performance or can remarkably reduce the algorithmic sample efficiency. Furthermore, it can be challenging to design effective augmentations for certain types of mismatches. To address this problem, exploring generative models for automatic data augmentation represents an interesting research direction. Generative models could produce diverse, mismatch-informed augmentations, potentially overcoming the limitations of manually designed strategies. Alternatively, investigating different auxiliary losses that are less reliant on augmentation techniques represents another interesting direction. Finally, future work will be devoted to go beyond simulated environments and test our algorithms on hardware in real-world scenarios.

References

[1] V. Giammarino, J. Queeney, and I. Paschalidis, "Adversarial imitation learning from visual observations using latent information," *Transactions on Machine Learning Research*, 2024. [Online]. Available: https://openreview.net/forum?id=ydPHjgf6h0

[2] M. Liu, T. He, W. Zhang, Y. Shuicheng, and Z. Xu, "Visual imitation learning with patch rewards," in *International Conference on Learning Representations*, 2022.

[3] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International Conference on Machine Learning*. PMLR, 2020, pp. 1597–1607.

[4] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar *et al.*, "Bootstrap your own latent-a new approach to self-supervised learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 271–21 284, 2020.

[5] E. Cetin and O. Celiktutan, "Domain-robust visual imitation learning with mutual information constraints," in *International Conference on Learning Representations*, 2020.

[6] V. Kumar, "Manipulators and manipulation in high dimensional spaces," Ph.D. dissertation, University of Washington, Seattle, 2016. [Online]. Available: https://digital.lib.washington.edu/researchworks/handle/1773/38104

[7] J. Ho and S. Ermon, "Generative adversarial imitation learning," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[8] J. Fu, K. Luo, and S. Levine, "Learning robust rewards with adversarial inverse reinforcement learning," *arXiv preprint arXiv:1710.11248*, 2017.

[9] S. Russell, "Learning agents for uncertain environments," in *Proceedings of the Annual Conference on Computational Learning Theory*, 1998, pp. 101–103.

[10] A. Y. Ng, S. Russell *et al.*, "Algorithms for inverse reinforcement learning." in *International Conference on Machine Learning*, vol. 1, 2000, p. 2.

[11] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *International Conference on Machine Learning*, 2004, p. 1.

[12] B. D. Ziebart, A. L. Maas, J. A. Bagnell, A. K. Dey *et al.*, "Maximum entropy inverse reinforcement learning." in *AAAI Conference on Artificial Intelligence*, vol. 8. Chicago, IL, USA, 2008, pp. 1433–1438.

[13] T. Gangwani, J. Lehman, Q. Liu, and J. Peng, "Learning belief representations for imitation learning in POMDPs," in *Uncertainty in Artificial Intelligence*. PMLR, 2020, pp. 1061–1071.

[14] R. Rafailov, T. Yu, A. Rajeswaran, and C. Finn, "Visual adversarial imitation learning using variational models," *Advances in Neural Information Processing Systems*, vol. 34, pp. 3016–3028, 2021.

[15] F. Torabi, G. Warnell, and P. Stone, "Generative adversarial imitation from observation," *arXiv preprint arXiv:1807.06158*, 2018.

[16] C. Yang, X. Ma, W. Huang, F. Sun, H. Liu, J. Huang, and C. Gan, "Imitation learning from observations by minimizing inverse dynamics disagreement," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[17] Z. Cheng, L. Liu, A. Liu, H. Sun, M. Fang, and D. Tao, "On the guaranteed almost equivalence between imitation learning from observation and demonstration," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[18] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1125–1134.

[19] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2223–2232.

[20] B. C. Stadie, P. Abbeel, and I. Sutskever, "Third-person imitation learning," *arXiv preprint arXiv:1703.01703*, 2017.

[21] K. Kim, Y. Gu, J. Song, S. Zhao, and S. Ermon, "Domain adaptive imitation learning," in *International Conference on Machine Learning*. PMLR, 2020, pp. 5286–5295.

[22] D. S. Raychaudhuri, S. Paul, J. Vanbaar, and A. K. Roy-Chowdhury, "Cross-domain imitation from observations," in *International Conference on Machine Learning*. PMLR, 2021, pp. 8902–8912.

[23] Y. Liu, A. Gupta, P. Abbeel, and S. Levine, "Imitation from observation: Learning to imitate behaviors from raw video via context translation," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1118–1125.

[24] P. Sermanet, C. Lynch, Y. Chebotar, J. Hsu, E. Jang, S. Schaal, S. Levine, and G. Brain, "Time-contrastive networks: Self-supervised learning from video," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1134–1141.

[25] L. Smith, N. Dhawan, M. Zhang, P. Abbeel, and S. Levine, "Avid: Learning multi-stage tasks via pixel-level translation of human videos," *Robotics: Science and Systems XVI*, 2020.

[26] V. Giammarino, J. Queeney, L. C. Carstensen, M. E. Hasselmo, and I. C. Paschalidis, "Opportunities and challenges from using animal videos in reinforcement learning for navigation," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 9056–9061, 2023.

[27] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7167–7176.

[28] J. Z. Zhang, S. Yang, G. Yang, A. L. Bishop, S. Gurumurthy, D. Ramanan, and Z. Manchester, "Slomo: A general system for legged robot motion imitation from casual videos," *IEEE Robotics and Automation Letters*, 2023.

[29] R. Okumura, M. Okada, and T. Taniguchi, "Domain-adversarial and-conditional state space model for imitation learning," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 5179–5186.

[30] S. Choi, S. Han, W. Kim, J. Chae, W. Jung, and Y. Sung, "Domain adaptive imitation learning with visual observation," in *Advances in Neural Information Processing Systems*, 2023.

[31] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *International Conference on Machine Learning*. PMLR, 2015, pp. 1180–1189.

[32] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson, "Learning latent dynamics for planning from pixels," in *International Conference on Machine Learning*. PMLR, 2019, pp. 2555–2565.

[33] S. K. S. Ghasemipour, R. Zemel, and S. Gu, "A divergence minimization perspective on imitation learning methods," in *Proceedings of the Conference on Robot Learning*. PMLR, 2020, pp. 1259–1277.

[34] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *International Conference on Machine Learning*. Pmlr, 2014, pp. 387–395.

[35] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[36] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[37] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[38] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[39] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[40] C. Wang, X. Luo, K. Ross, and D. Li, "Vrl3: A data-driven framework for visual deep reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 32 974–32 988, 2022.

[41] V. Giammarino, A. Giammarino, and M. Pearce, "A reinforcement learning approach for robotic unloading from visual observations," *arXiv preprint arXiv:2309.06621*, 2023.

Supplementary material

A. Analysis

In the following, we show how *domain invariance in the feature space $\mathcal{Z}$* leads to the same V-IfO guarantees as in [1], even in the presence of visual mismatches.

Proposition 1: Consider source and target POMDPs respectively defined by the tuples $(\mathcal{S}, \mathcal{A}, \mathcal{X}, \mathcal{T}, \mathcal{U}_T, \mathcal{R}, \rho_0, \gamma)$ and $(\mathcal{S}, \mathcal{A}, \mathcal{X}, \mathcal{T}, \mathcal{U}_S, \mathcal{R}, \rho_0, \gamma)$. Let $\mathcal{X} = (\bar{\mathcal{X}}, \hat{\mathcal{X}})$, where $\bar{\mathcal{X}}$ is an observations set invariant between source and target POMDP, and $\hat{\mathcal{X}}$ is a set of visual distractors. We write $x_t = (\bar{x}_t, \hat{x}_t)$. Assume $z_t = \phi(x_{\leq t}) = \phi(\bar{x}_{\leq t})$ such that $\mathbb{P}(s_t|z_t, a_t) = \mathbb{P}(s_t|z_t) = \mathbb{P}(s_t|x_{\leq t}, a_{<t}) = \mathbb{P}(s_t|\bar{x}_{\leq t}, a_{<t})$. Then, the filtering posterior distributions $\mathbb{P}(s_t|z_t)$ and $\mathbb{P}(s_{t+1}, s_t|z_{t+1}, z_t)$ do not depend on the policy π and are invariant between source and target POMDPs.

Proof: Considering the definition of the latent variable z_t, which only depends on $\bar{x}_{\leq t}$ and not the visually-distracting information $\hat{x}_{\leq t}$, we can write

$$\mathbb{P}(s_t|z_t) = \mathbb{P}(s_t|x_t, a_{t-1}, z_{t-1}) = \mathbb{P}(s_t|\bar{x}_t, a_{t-1}, z_{t-1}).$$

Then, by leveraging Bayes rule we have that

$$\begin{aligned}\mathbb{P}(s_t|z_t) &= \mathbb{P}(s_t|\bar{x}_t, a_{t-1}, z_{t-1}) \\ &= \frac{\mathbb{P}(\bar{x}_t|s_t, a_{t-1}, z_{t-1})\mathbb{P}(s_t|a_{t-1}, z_{t-1})}{\mathbb{P}(\bar{x}_t|a_{t-1}, z_{t-1})} \\ &= \frac{\mathbb{P}(\bar{x}_t|s_t)\int_{\mathcal{S}}\mathcal{T}(s_t|s_{t-1}, a_{t-1})\mathbb{P}(s_{t-1}|z_{t-1})ds_{t-1}}{\int_{\mathcal{S}}\int_{\mathcal{S}}\mathbb{P}(\bar{x}_t|s_t)\mathcal{T}(s_t|s_{t-1}, a_{t-1})\mathbb{P}(s_{t-1}|z_{t-1})ds_t ds_{t-1}},\end{aligned}$$

where the denominator can be seen as a normalizing factor. Note that $\mathbb{P}(\bar{x}_t|s_t)$ is the same for both the source and target POMDPs by the definition of $\bar{\mathcal{X}}$ above. Therefore, $\mathbb{P}(s_t|z_t)$ has no dependence on the policy π and is invariant between source and target POMDP.

Similarly, for $\mathbb{P}(s_{t+1}, s_t|z_t, z_{t+1})$ we have that

$$\begin{aligned}\mathbb{P}(s_{t+1}, s_t|z_t, z_{t+1}) &= \mathbb{P}(s_t, s_{t+1}|\bar{x}_{t+1}, a_t, z_t) \\ &= \frac{\mathbb{P}(\bar{x}_{t+1}|s_t, s_{t+1}, a_t, z_t)\mathbb{P}(s_t, s_{t+1}|a_t, z_t)}{\mathbb{P}(\bar{x}_{t+1}|a_t, z_t)} \\ &= \frac{\mathbb{P}(\bar{x}_{t+1}|s_{t+1})\mathcal{T}(s_{t+1}|s_t, a_t)\mathbb{P}(s_t|z_t)}{\int_{\mathcal{S}}\int_{\mathcal{S}}\mathbb{P}(\bar{x}_{t+1}|s_{t+1})\mathcal{T}(s_{t+1}|s_t, a_t)\mathbb{P}(s_t|z_t)ds_{t+1}ds_t}.\end{aligned}$$

Because $\mathbb{P}(s_t|z_t)$ does not depend on π and is the same for both source and target POMDP, the result also holds for $\mathbb{P}(s_{t+1}, s_t|z_t, z_{t+1})$. ■

Proposition 2: Given $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, for the scenarios described in Proposition 1 the following inequality holds:

$$\left|J(\pi_E) - J(\pi_{\boldsymbol{\theta}})\right| \leq \frac{2R_{\max}}{1-\gamma}\mathbb{D}_{\text{TV}}\big(\rho_{\pi_{\boldsymbol{\theta}}}(z, z'), \rho_{\pi_E}(z, z')\big) + C,$$

where $R_{\max} = \max_{(s,a)\in\mathcal{S}\times\mathcal{A}}|\mathcal{R}(s, a)|$ and

$$C = \frac{2R_{\max}}{1-\gamma}\mathbb{E}_{\rho_{\pi_{\boldsymbol{\theta}}}(z,z')}\big[\mathbb{D}_{\text{TV}}\big(\mathbb{P}_{\pi_{\boldsymbol{\theta}}}(a|z, z'), \mathbb{P}_{\pi_E}(a|z, z')\big)\big]. \tag{6}$$

If $\mathcal{R} : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$, then we have that

$$\left|J(\pi_E) - J(\pi_{\boldsymbol{\theta}})\right| \leq \frac{2R_{\max}}{1-\gamma}\mathbb{D}_{\text{TV}}\big(\rho_{\pi_{\boldsymbol{\theta}}}(z, z'), \rho_{\pi_E}(z, z')\big),$$

where $R_{\max} = \max_{(s,s')\in\mathcal{S}\times\mathcal{S}}|\mathcal{R}(s, s')|$.

Proof: Because Proposition 1 holds, we can directly follow the proofs of Theorem 1 and Theorem 2 in [1] for the setting of no visual mismatch. ■

B. Pseudo-code and hyperparameters

Algorithm 1: C-LAIfO

```
Inputs:
Expert observations: $(x_n^S)_{0:N} \in \mathcal{B}_E$.
$\pi_{\boldsymbol{\theta}}$, $D_{\boldsymbol{\chi}}$, $Q_{\boldsymbol{\psi}_1}$, $Q_{\boldsymbol{\psi}_2}$, $\phi_{\boldsymbol{\delta}}$: networks for policy, discriminator,
  Q functions and encoder.
$T_{\text{train}}$, $\sigma(t)$, $d$, aug, $c$, $\tau$, $B$, $\alpha$, $\alpha_D$, $\gamma$, $\eta$: training steps,
  scheduled standard deviation, frames stack dimension,
  stochastic data augmentation, clip value, target update rate,
  batch size, learning rate, discriminator learning rate,
  discount factor and temperature parameter.
for $t = 1, \dots, T_{\text{train}}$ do
    $\sigma_t \leftarrow \sigma(t)$
    if $t \geq d - 1$ then
        $z_t \leftarrow \phi_{\boldsymbol{\delta}}(x^T_{t-d+1:t})$
    else
        $z_t \leftarrow \phi_{\boldsymbol{\delta}}(x^T_{0:t})$
    $a_t \leftarrow \pi_{\boldsymbol{\theta}}(z_t) + \epsilon$ and $\epsilon \sim \mathcal{N}(0, \sigma_t^2)$
    $s_{t+1} \sim \mathcal{T}(\cdot|s_t, a_t)$ and $x_{t+1} \sim \mathcal{U}_T(\cdot|s_{t+1})$
    $\mathcal{B} \leftarrow \mathcal{B} \cup (x_t^T, a_t, x_{t+1}^T)$
    UpdateEncoder($\mathcal{B}$)
    UpdateDiscriminator($\mathcal{B}$, $\mathcal{B}_E$)
    UpdateCritic($\mathcal{B}$)
    UpdateActor($\mathcal{B}$)
begin UpdateEncoder
    $\{(x^T_{t-d+1:t}, a_t, x^T_{t-d+2:t+1})\} \sim \mathcal{B}$ (sample $B$ transitions)
    $\tilde{z}_{\boldsymbol{\delta}}(i) \leftarrow \phi_{\boldsymbol{\delta}}(\text{aug}(x^T_{t-d+1:t})_i)$ and
      $\tilde{z}_{\boldsymbol{\delta}}(j) \leftarrow \phi_{\boldsymbol{\delta}}(\text{aug}(x^T_{t-d+1:t})_j)$
    Update $\phi_{\boldsymbol{\delta}}$ to minimize (5) with learning rate $\alpha$
begin UpdateDiscriminator
    $\{(x^S_{t-d+1:t}, x^S_{t-d+2:t+1})\} \sim \mathcal{B}_E$ and
      $\{(x^T_{t-d+1:t}, x^T_{t-d+2:t+1})\} \sim \mathcal{B}$ (sample $B$ transitions)
    $z_{\boldsymbol{\delta}} \leftarrow \phi_{\boldsymbol{\delta}}(x_{t-d+1:t})$ and $z'_{\boldsymbol{\delta}} \leftarrow \phi_{\boldsymbol{\delta}}(x_{t-d+2:t+1})$ for both
      agent and expert
    Update $D_{\boldsymbol{\chi}}$ to minimize BCE loss with learning rate $\alpha_D$
begin UpdateCritic
    $\{(x^T_{t-d+1:t}, a_t, x^T_{t-d+2:t+1})\} \sim \mathcal{B}$ (sample $B$ transitions)
    $\tilde{z}_{\boldsymbol{\delta}} \leftarrow \phi_{\boldsymbol{\delta}}(\text{aug}(x^T_{t-d+1:t}))$ and $\tilde{z}'_{\boldsymbol{\delta}} \leftarrow \phi_{\boldsymbol{\delta}}(\text{aug}(x^T_{t-d+2:t+1}))$
    $a_{t+1} \leftarrow \pi_{\boldsymbol{\theta}}(\tilde{z}'_{\boldsymbol{\delta}}) + \epsilon$ and $\epsilon \sim \text{clip}(\mathcal{N}(0, \sigma_t^2), -c, c)$
    Update $Q_{\boldsymbol{\psi}_1}$, $Q_{\boldsymbol{\psi}_2}$ and $\phi_{\boldsymbol{\delta}}$ to minimize (3) with
      $\bar{r}_{\boldsymbol{\chi}}(z_{\boldsymbol{\delta}}, z'_{\boldsymbol{\delta}})$ and learning rate $\alpha$
    $\bar{\boldsymbol{\psi}}_k \leftarrow (1 - \tau)\bar{\boldsymbol{\psi}}_k + \tau\boldsymbol{\psi}_k \quad \forall k \in \{1, 2\}$
begin UpdateActor
    $\{x^T_{t-d+1:t}\} \sim \mathcal{B}$ (sample $B$ observations)
    $\tilde{z}_{\boldsymbol{\delta}} \leftarrow \phi_{\boldsymbol{\delta}}(\text{aug}(x^T_{t-d+1:t}))$
    $a_t \leftarrow \pi_{\boldsymbol{\theta}}(\tilde{z}_{\boldsymbol{\delta}}) + \epsilon$ and $\epsilon \sim \text{clip}(\mathcal{N}(0, \sigma_t^2), -c, c)$
    Update $\pi_{\boldsymbol{\theta}}$ using DDPG [39] with learning rate $\alpha$
```

TABLE IV: Hyperparameter values for C-LAIfO experiments.

Hyperparameter Name	Value
Frames stack (d)	3
Discount factor (γ)	0.99
Image size	64×64
Batch size (B)	256
Optimizer	Adam
Learning rate (α)	10^{-4}
Discriminator learning rate (α_D)	4×10^{-4}
Target update rate (τ)	0.01
Clip value (c)	0.3
Temperature parameter (η)	1.0

C. Data augmentation

The operations used in our data augmentation functions for the experiments in Section V are summarized in Table V. Fig. 7 shows an example of color augmentation as implemented for the experiments in Table II, where we randomly perform all the operations in the color transformations row in Table V. On the other hand, Fig. 8 shows only a brightness transformation as implemented in the ablation study in Table. I. Full augmentation in Table I performs all the operations in Table V. For additional details refer to our code[1].

TABLE V: Operations used in our data augmentation function.

	Operations
Color transformations	Brightness Contrast Saturation Hue Grayscale Gaussian blur Invert
Affine transformations	Horizontal flip Vertical flip Resized crop

D. Learning curves

All the experiments are run using Nvidia-A40 GPUs on an internal cluster. For each algorithm, we run two experiments in parallel on the same GPU and each experiment takes 1 to 4 days depending on the simulated environment and the considered algorithm. For all the implementation details refer to our code.

Fig. 9, Fig. 11, and Fig. 12 show the learning curves for the results in Table I, Table II, and Table III, respectively. Fig. 10 shows the effect of randomized data augmentation when used in the AIL step in (2).

E. PCA and t-SNE analysis

In Fig. 13 we compare the latent space $\mathcal{Z}$ learned by LAIfO, C-LAIfO, and LAIfO with data augmentation in the Light setting from Table II and Fig. 9. Specifically, we define *source* and *target* as the observations generated by an optimal policy in the source and target POMDPs, respectively. Similarly, *source random* and *target random* are observations generated by random policies. These observations are processed with the encoder $\phi_{\boldsymbol{\delta}} : \mathcal{X}^d \to \mathcal{Z}$ trained using the respective algorithms. We perform PCA and t-SNE on this set of latent variables $z_{1:T}$ and plot the first two principal components. The results show that C-LAIfO is the only algorithm capable of filtering out visual distractors and clustering together data points with the same goal-completion information. In Fig. 14, we focus on C-LAIfO and test for generalization to the unseen environment in Fig. 14c. We train $\phi_{\boldsymbol{\delta}}$ on the Full setting from Table II and Fig. 11 and perform PCA and t-SNE as described. In Fig. 14, the Full figures have (14a)-(14b) as source-POMDP and (14e)-(14f) as target-POMDP. Similarly the Full-unseen figures have (14c)-(14d) as source-POMDP and (14e)-(14f) as target-POMDP. The results in the Full-unseen setting match those in the Full experiment, indicating that $\phi_{\boldsymbol{\delta}}$ trained on (14a) can successfully generalize to (14c).

[1] https://github.com/VittorioGiammarino/C-LAIfO

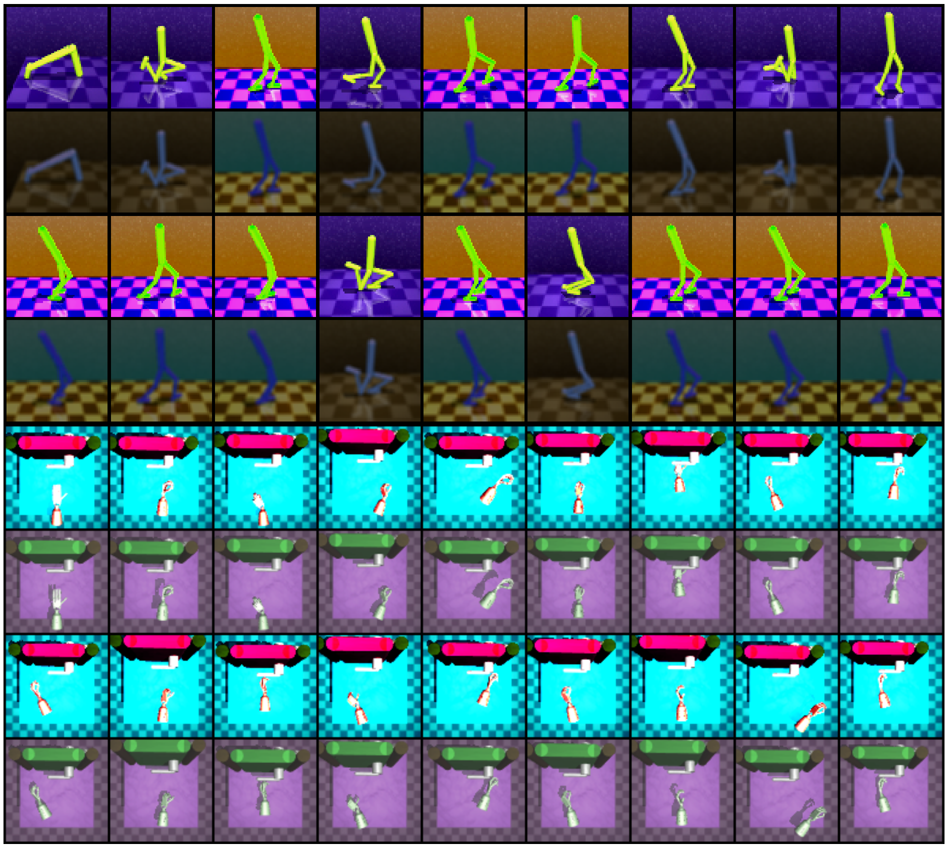


Fig. 7: Examples of augmentation as color transformation.

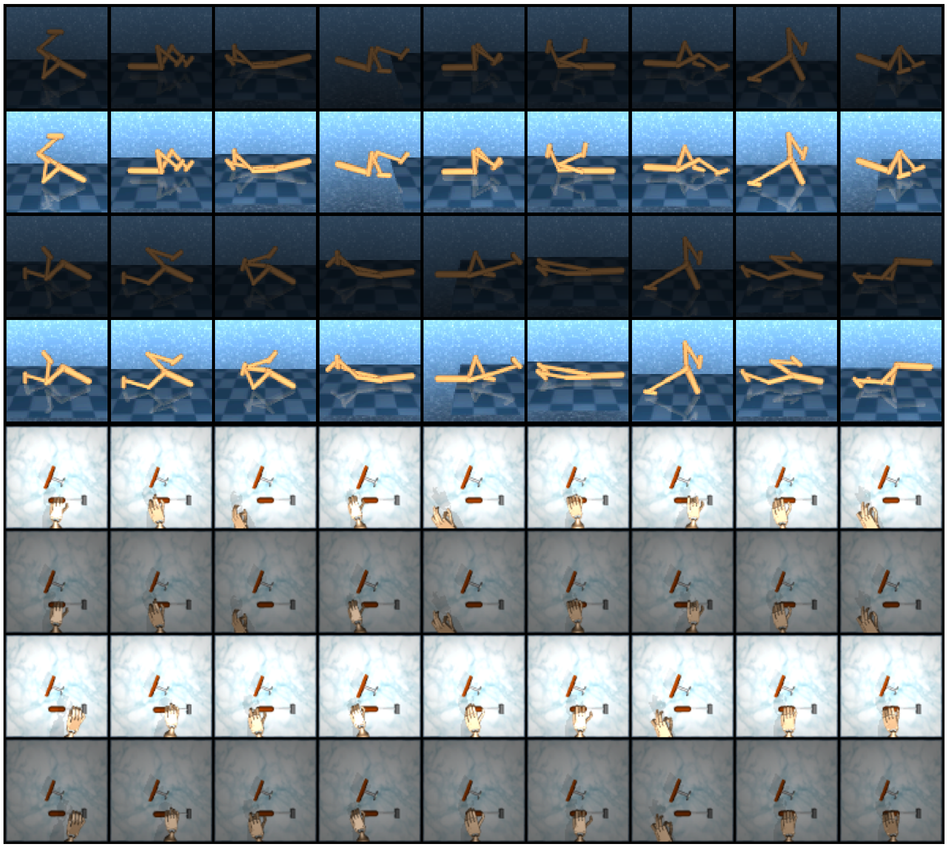


Fig. 8: Examples of augmentation as brightness transformation.

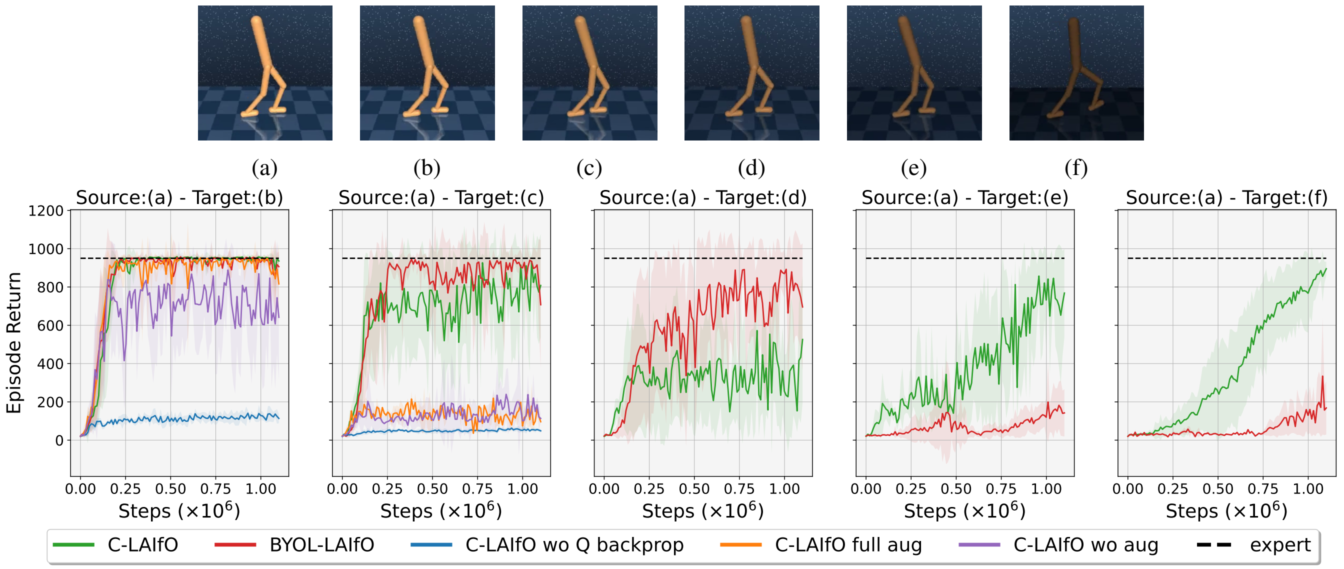


Fig. 9: Learning curves for the results in Table I. Plots show the average return per episode as a function of training steps. The environment in (9a) represents the source-POMDP used to collect expert data, while (9b)–(9f) are different target-POMDPs. In these experiments, visual mismatch is introduced by varying the light intensity, with the degree of mismatch increasing linearly from (9b) to (9f).

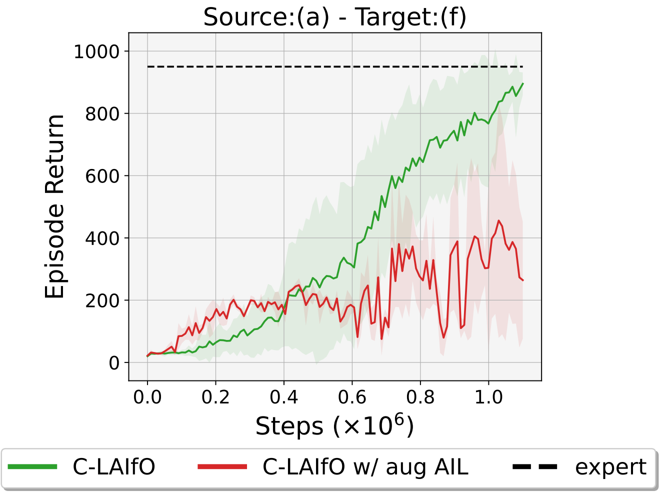


Fig. 10: Experiments on the use of data augmentation in AIL in (2). Plots show the average return per episode as a function of training steps. The environment in (9a) represents the source-POMDP used to collect expert data, while (9f) the target-POMDPs. These experiments show how randomized data augmentation used during AIL triggers instability.

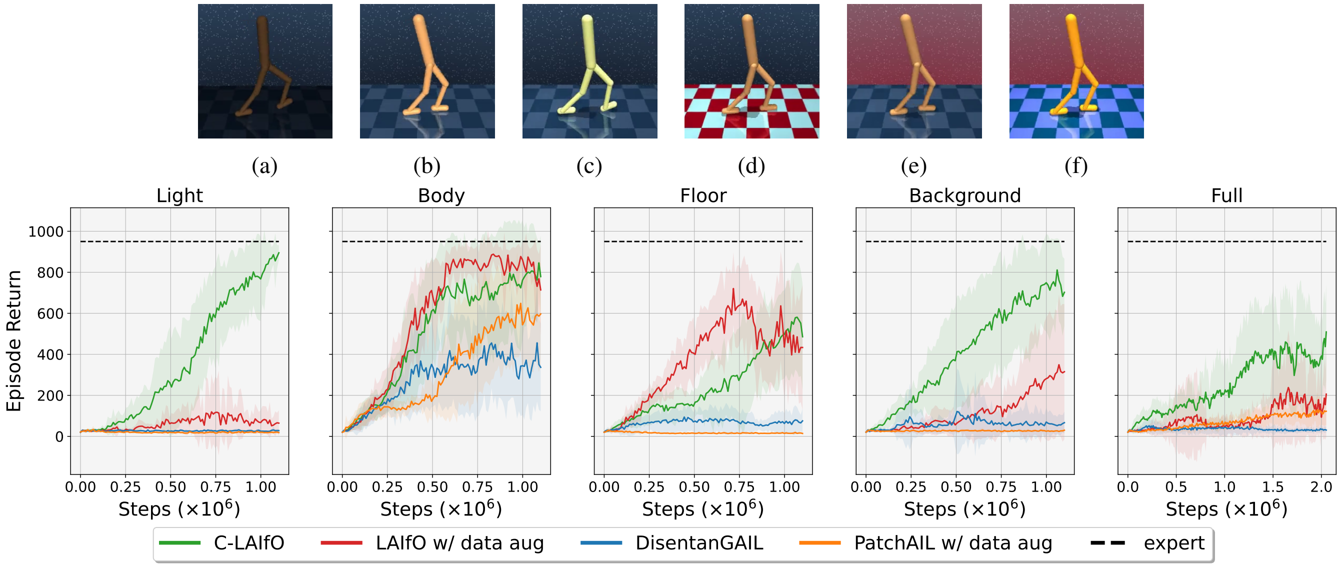


Fig. 11: Learning curves for the results in Table II. Plots show the average return per episode as a function of training steps. The Light experiment consists in (11b) as source-POMDP and (11a) as target-POMDP. The Body, Floor, Background, and Full experiments have (11b) as target-POMDP and respectively (11c), (11d), (11e), and (11f) as source-POMDP.

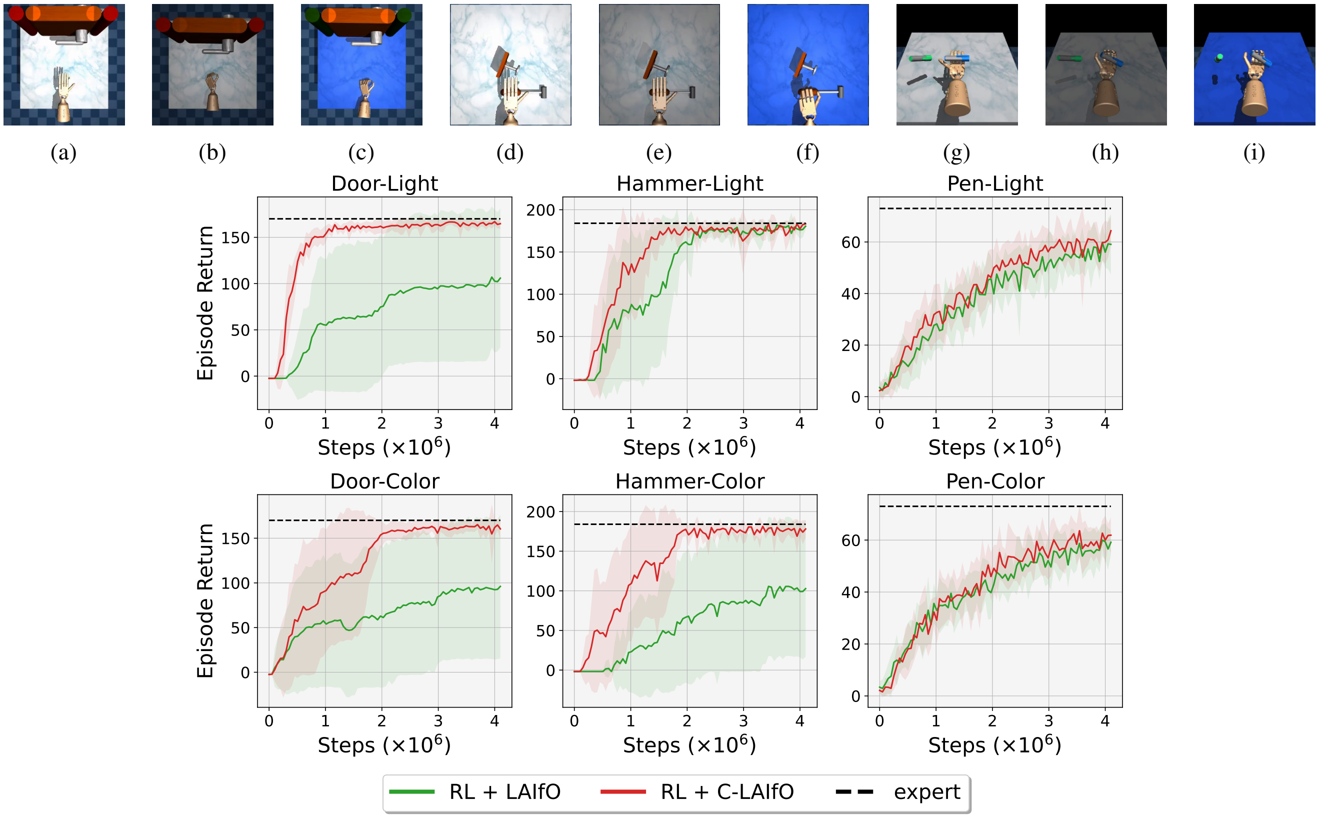
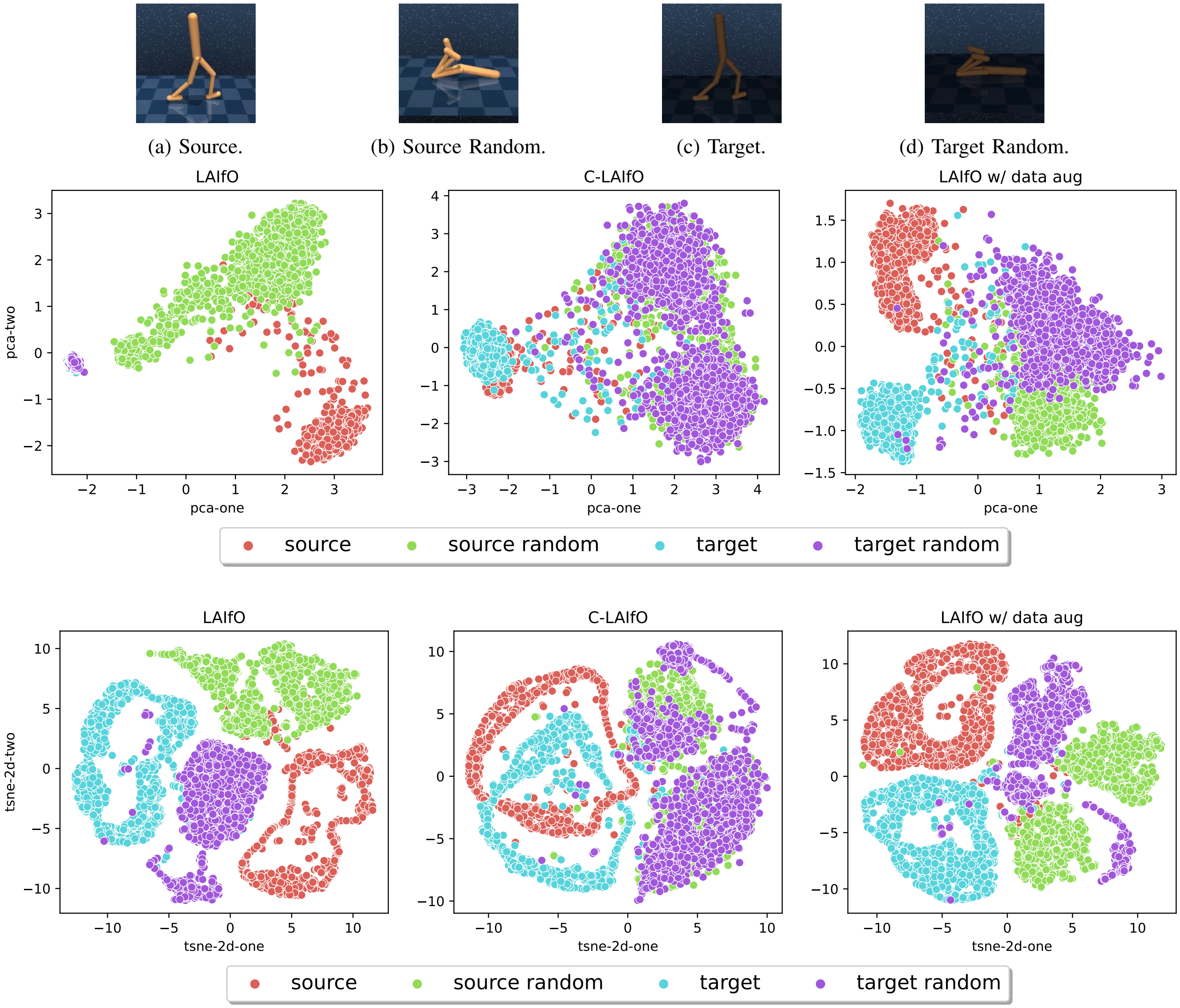
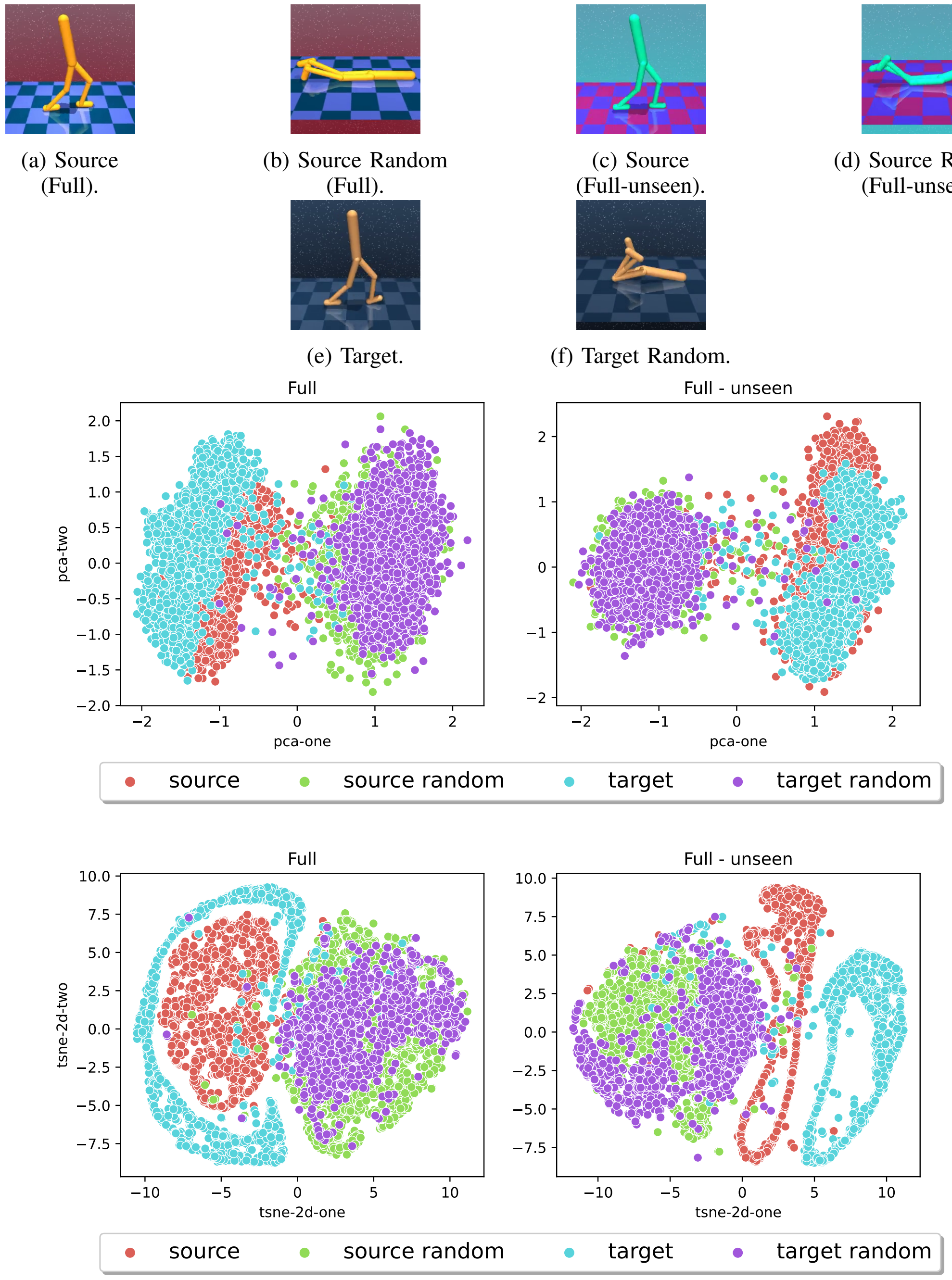


Fig. 12: Learning curves for the results in Table III. Plots show the average return per episode as a function of training steps. The Door-Light and Door-Color experiments consider (12a) as the source-POMDP and (12b) and (12c) as the respective target-POMDPs. Similarly, the Hammer-Light and Hammer-Color experiments consider (12d) as source-POMDP and (12e) and (12f) as the respective target-POMDPs, while the Pen-Light and Pen-Color experiments consider (12g) as source-POMDP and (12h) and (12i) as the respective target-POMDPs.



(a) Source. (b) Source Random. (c) Target. (d) Target Random.

Fig. 13: PCA and t-SNE visualizations for the Light experiment in Table II.



(a) Source (Full).

(b) Source Random (Full).

(c) Source (Full-unseen).

(d) Source Random (Full-unseen).

(e) Target.

(f) Target Random.

Fig. 14: PCA and t-SNE for the Full experiment in Table II. Fig. 14a and Fig. 14b denote the source environment for the Full experiment (left figures in Fig. 14). Fig. 14c and Fig. 14d denote the source environment for the Full-unseen experiment (right figures in Fig. 14).