2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Unsupervised Domain Adaptation for Robust Imitation Learning under Visual Perturbations

Yasuhiro Kato[1,3], Thomas Westfechtel[1], Jen-Yen Chang[1,3], Naoki Morihira[2], Akinobu Hayashi[2], Tatsuya Harada[1,3], and Takayuki Osa[3]

***Abstract*— Vision-based robot manipulation systems often suffer from performance degradation under domain shifts in visual inputs. While data augmentation is commonly employed in reinforcement learning, its application in imitation learning remains relatively underexplored. Our preliminary experiments indicate that simply incorporating augmentation techniques does not yield effective improvements in imitation learning. To address this challenge, we propose a two-stage learning process. First, we develop an adversarial feature learning framework that leverages data augmentation to enhance robustness against domain shifts. Second, we introduce an unsupervised domain adaptation method that adapts models to target environments using only easily collected image data. In robotic tasks, visual domain shifts can often be detected from initial observations alone. Since collecting complete action-labeled episodes in new domains is expensive, adapting with only initial images greatly reduces data collection costs. To this end, we develop an adaptation strategy that relies solely on initial target-domain observations, eliminating the need for labeled demonstrations. Experimental results across both simulation and physical robot implementations demonstrate that our method preserves source domain performance while exhibiting enhanced resilience to visual perturbations, including varying lighting conditions, background modifications, and environmental distractors.**

I. INTRODUCTION

With the emergence of large-scale robot datasets [1], [2] and the development of platforms that enable efficient collection of human demonstration data [3], [4], [5], teaching robots actions and skills through imitation learning has become increasingly prevalent [6], [7]. To facilitate the implementation of systems that can interact in the real world, visual information is an essential component for enabling robots to perceive and understand their environment. Some approaches have demonstrated the ability to perform imitation solely based on visual inputs [8], [9]. These results highlight that visual information is particularly important in imitation learning.

Policies that rely on visual observations as input suffer from domain gaps between the environment used for collecting demonstrations and the test environment. When performing robot manipulation based on visual information, the system must contend with visual changes known as domain shifts, which can occur due to variations in lighting, background, or environmental conditions. For instance, even if demonstrations are collected in the morning, executing the policy later in the day such as in the evening when strong sunlight enters from the west, can introduce domain shifts due to changes in lighting conditions. One possible solution is to use data augmentation to increase variability in the collected data. However, unlike the reinforcement learning field [10] where data augmentation techniques and domain randomization are commonly used to address such challenges [11], [12], the application of data augmentation has not been extensively explored in the context of imitation learning, presenting a significant opportunity for advancement in the field.

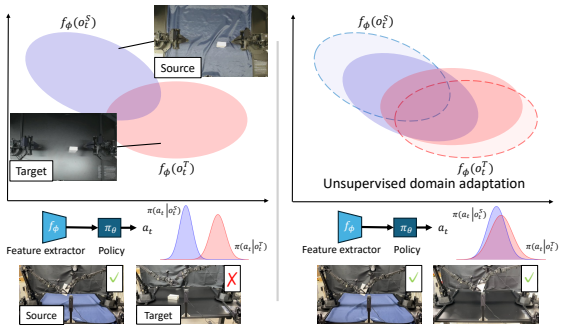


Fig. 1. Concept illustration: Feature representations $f_\phi(o_t^S)$ from the source domain and $f_\phi(o_t^T)$ from the target domain, which are separated due to domain differences (left), are aligned in a feature space through unsupervised domain adaptation (right). This significantly influences the action distribution, allowing the model to transfer its performance from the source domain to the target domain. Here, f_ϕ denotes the feature extractor with parameter ϕ and o_t denotes the observation at time t.

Multimodal imitation learning presents two fundamental challenges: learning effective representations and learning appropriate actions, where the learned representations should facilitate action acquisition [13]. Although data augmentation may help systems adapt to novel environments [14], [15], our preliminary experiments revealed that merely incorporating data augmentation into imitation learning does not always enhance the policy's robustness under domain shifts. Unlike reinforcement learning where the agent can collect additional data through exploration, imitation learning typically relies on a fixed set of demonstrations. In such scenarios, aggressive data augmentation may impede the learning process by making it more challenging to extract task-relevant features from the limited demonstration data. This finding highlights

*This work was supported in part by the Social Collaboration Program (Scalable Robot Learning) between The University of Tokyo and Honda R&D Co., Ltd., JST Moonshot R&D Grant Number JPMJPS2011, CREST Grant Number JPMJCR2015 and Basic Research Grant (Super AI) of Institute for AI and Beyond of the University of Tokyo. T. O. was supported by JSPS KAKENHI Grant Number JP25K03176.

[1]University of Tokyo, Japan, {y-kato, thomas, Chou, Harada}@mi.t.u-tokyo.ac.jp

[2]Honda R&D Co., Ltd. Innovative Research Excellence, Japan, firstname_lastname@jp.honda

[3]RIKEN Center for Advanced Intelligent Project, Japan, firstname.lastname@riken.jp

the need for a learning framework that effectively extract task-relevant information from augmented data in order for policy to adapt towards test domain.

To enable a learned policy to function effectively, domain adaptation to the target environment is as crucial as robustness. Even when adopting a policy trained on large-scale datasets or demonstrations collected in a local environment, fine-tuning is often required to ensure performance in the test environment. However, collecting demonstrations for every test environment is impractical. In contrast, leveraging only images from the test environment can substantially reduce the data collection cost. While robot learning typically relies on collecting sequences of observations with action labels, in this work we investigate an approach that adapts to the domain using only initial images/observations from the test environment.

In this study, we propose a robust feature learning framework for imitation learning, leveraging adversarial training. Our framework adopts a two-stage learning process. In the first stage, a discriminator attempts to identify different augmentation styles while a feature extractor learns to generate representations that are invariant across these augmentations. In the second stage, we fine-tune the feature extractor and policy using only initial observation image of test domains for unsupervised domain adaptation. This framework enables the extraction of task-relevant information that is robust to domain shifts.

Our proposed method approximately maximizes the lower bound of the mutual information between visual features and task-relevant information, thereby creating a robust representation that maintains task performance across different visual conditions while preserving task-relevant features. As the proposed framework is agnostic with data augmentation techniques, it can be incorporated with any data augmentation techniques.

We evaluated the proposed method on bimanual manipulation tasks using both simulations and a real robotic system. In the experiments, the policies were tested under lighting conditions, table color and object arrangements different from those in the training data. The results demonstrate that the proposed method significantly enhances the policy's robustness against domain shifts and adaptation to test domains.

II. Related work

Data augmentation for policy learning: In the context of simulation-based reinforcement learning (RL), data augmentation and domain randomization techniques have been increasingly employed to improve model robustness and generalization capabilities [16]. To further facilitate generalization, researchers have developed simulation environments with diverse environmental settings [17]. In imitation learning, several successful applications of data augmentation have been reported [18]. Among these, some approaches leverage pre-trained generative models to augment visual inputs; however, such methods may introduce discrepancies between the augmented and real-world observations [15]. Another line of work explores saliency-guided augmentation to enhance domain robustness [19], though these methods often lack explicit adaptation mechanisms for the target domain.

Adversarial learning and domain adaptation: Building upon these insights from data augmentation and representation learning, domain adversarial learning offers a promising approach to combine their benefits. In the field of computer vision, domain adversarial learning [20], [21], [22], [23] is known to enable a feature extractor to learn features that focus on task-relevant information such as class discrimination, while becoming invariant to domain-specific characteristics. In the field of robotics, adversarial learning is utilized to capture invariant features between expert and agent domain [24] or sim-to-real transfer [25], [26], and robust latent space learning via contrastive learning and augmentation [27]. However, while these methods enhance robustness to domain shifts, they often lack explicit adaptation to the target domain, which is critical for maintaining policy performance under real-world domain discrepancies.

III. Method

A. Learning architecture

Proposed pipeline: Vision-based manipulation systems face significant challenges due to domain discrepancies between training and test environments, as these variations often lead to deterioration in learned policy performance. Therefore, learned policies are required to possess both robustness against various visual perturbations and adaptability to test environments. The proposed method consists of Action Chunking Transformer (ACT) [3] for behavior cloning, augmentation discriminator for learning augmentation-invariant features, and domain discriminator for capturing domain invariant features as shown in Fig. 2. The proposed approach aims to enhance policy robustness through learning features invariant to both data augmentations and domain-specific characteristics. We will describe each module in the following subsections.

B. Behavior learning

Behavioral cloning (BC) replicates desired behavior which simplifies imitation learning by framing it as a supervised learning task where observations are mapped to corresponding actions [28], [29]. We consider a policy π_θ, parameterized by a vector θ. Let $f_\phi(o_t)$ represent a feature extractor that extracts visual features from the observation o_t at time t. The policy then outputs an action based on the visual features provided by $f_\phi(o_t)$. Given a training dataset $\mathcal{D}$, the objective function of BC is expressed as follows:

$$\mathcal{L}_{\text{BC}}(\theta, \phi) = \mathbb{E}_{(o_t, a_t)\sim\mathcal{D}}\left[\ell_{\text{BC}}(\pi_\theta)\right], \tag{1}$$

where ℓ_{BC} refers to a loss function that measures the deviation between the predicted continuous-valued actions and the actual ground truth actions (such as using l_1 or l_2 norms). In this work, ACT is utilized as a BC method. Denoting a sequence of actions from time t to $t+N$ as

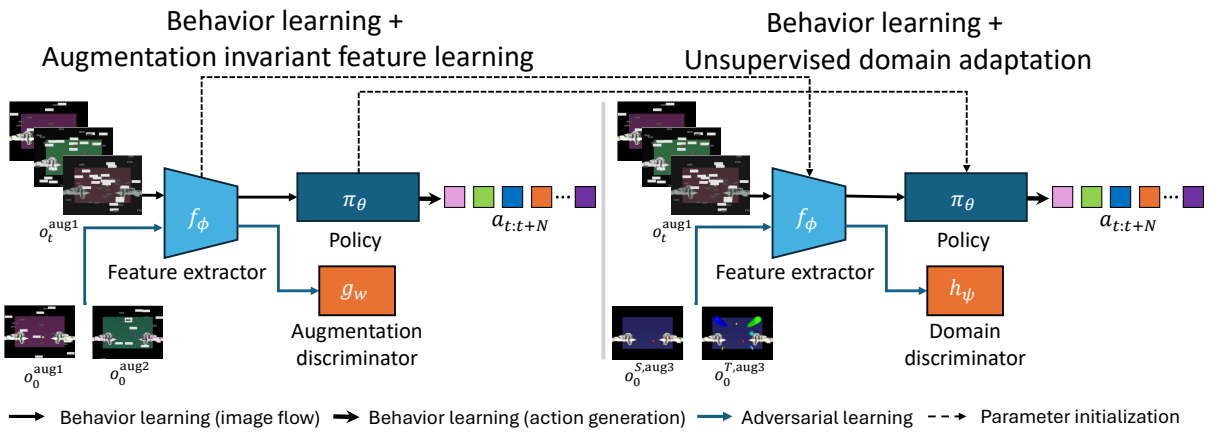


Fig. 2. Overview of the proposed learning pipeline: the black arrows depict the behavior learning pipeline, where the system learns to map augmented visual inputs o_t^{aug1} to their associated actions a_t. In the first stage, the blue arrows illustrate the adversarial feature learning process, which trains the system to extract augmentation-invariant features across various augmented image samples o_0^{aug1}, o_0^{aug2}. In the second stage, a domain discriminator is introduced for domain-invariant feature learning using the weakly augmented initial observations from the source domain $o_0^{S,\text{aug3}}$ and target domain $o_0^{T,\text{aug3}}$ as illustrated by the blue arrows. In addition, the dashed arrows indicate initialization using learned parameters.

$a_{t:t+N} = [a_t, a_{t+1}, \ldots, a_{t+N}]$, the objective function of ACT is expressed as follows:

$$\ell_{\text{BC}}(\pi_\theta) = \|a_{t:t+N} - \pi(a_{t:t+N} \mid o_t, z_t)\|_2^2 + \beta D_{\text{KL}}\left(q(z_t|a_{t:t+N}, o_t)||\mathcal{N}(0, I)\right), \quad (2)$$

where z_t is a style variable sampled from the variational distribution $q(z_t|a_{t:t+N}, o_t)$ and β is a weight for the KL divergence D_{KL}.

C. Adversarial training for learning augmentation-invariant visual features

The proposed framework employs adversarial training to learn augmentation-invariant visual features through a competitive process between a feature extractor and an augmentation discriminator. The discriminator attempts to distinguish between feature representations of differently augmented versions of the same image, while the feature extractor aims to generate representations that prevent such discrimination. In this process, our method focuses on learning representations that remain consistent across augmented variations of the same visual input. This objective is formulated as a binary classification problem with cross-entropy loss. In our framework, we introduce an augmentation discriminator g_w, parameterized by a vector w, which distinguishes the type of data augmentation applied to each image. Our goal is to train a feature extractor f_ϕ, parameterized by a vector ϕ to extract augmentation-invariant visual features. To achieve this, we formulate the following min-max optimization problem:

$$\min_w \max_\phi \mathcal{L}_{\text{aug}}(w, \phi), \quad (3)$$

where

$$\mathcal{L}_{\text{aug}}(w, \phi) = -\mathbb{E}_{o\sim\mathcal{D}}\Big[y_{\text{aug}} \log\left(g_w(f_\phi(o))\right) + (1 - y_{\text{aug}}) \log\left(1 - g_w(f_\phi(o))\right)\Big]. \quad (4)$$

Herein, $y_{\text{aug}} \in \{0, 1\}$ is a binary label that indicates the difference of data augmentation techniques applied to the original images. This optimization problem is solved via backpropagation, utilizing Gradient Reversal Layer (GRL) as proposed in [20]. Through this process, the feature extractor f_ϕ is trained to produce features that are difficult for the augmentation discriminator g_w to distinguish.

D. Unsupervised domain adaptation

To ensure that the policy functions effectively in the test domain, we introduce a domain discriminator module into the framework for the second stage of the training. This discriminator is designed to detect differences between pairs of images sampled from different domains. By directly optimizing this objective, the feature extractor is encouraged to learn domain-invariant representations that are likely to be relevant for robotic manipulation tasks. The domain discriminator operates as a binary classifier, predicting whether a given pair of input images originates from the same or different domains. To further enhance robustness, we incorporated subtle data augmentations such as white noise into the input images. This is intended to preserve the policy's robustness gained through augmentation-invariant feature learning. This learning stage complements the policy's ability to adapt to the test domain.

Denoting the domain discriminator by h_ψ parameterized by a vector ψ, the loss function for learning domain differences is defined as the following cross-entropy loss:

$$\mathcal{L}_{\text{domain}}(\psi, \phi) = -\mathbb{E}_{o\sim\mathcal{D}}\Big[y_{\text{domain}} \log\left(h_\psi(f_\phi(o))\right) + (1 - y_{\text{domain}}) \log\left(1 - h_\psi(f_\phi(o))\right)\Big], \quad (5)$$

where $y_{\text{domain}} \in \{0, 1\}$ is a binary label indicating the domain. The domain discriminator h_ψ is trained to minimize $\mathcal{L}_{\text{domain}}(\psi, \phi)$ as defined in (5), while the feature extractor f_ϕ is optimized to maximize it, so as to learn domain-invariant features. For source domain adaptation, this stage can be interpreted as fine-tuning that learns augmentation-invariant feature extraction in the source domain.

Based on the above discussion, the following optimization problem is addressed in the first stage:

$$\min_{\theta,\phi} \max_{w} \left(\mathcal{L}_{\text{BC}}(\theta, \phi) - \lambda_1 \mathcal{L}_{\text{aug}}(w, \phi) \right), \tag{6}$$

In the second stage, the problem is formulated as

$$\min_{\theta,\phi} \max_{\psi} \left(\mathcal{L}_{\text{BC}}(\theta, \phi) - \lambda_2 \mathcal{L}_{\text{domain}}(\psi, \phi) \right), \tag{7}$$

where λ_1 and λ_2 are constants that balance the contributions of each term. The implementation of the proposed methodology is summarized in Algorithms 1 and 2.

Algorithm 1 Augment-invariant feature learning

Require: $\mathcal{D}$: Demonstration dataset, K: number of update iterations, L: batch sizes, $\text{Aug}(\cdot; \alpha)$: function for data augmentation parameterized with α

Initialize networks: feature extractor f_ϕ, augmentation discriminator g_w, policy network π_θ
for $k = 1$ to K **do**
  Sample initial observation $o_0 \sim \mathcal{D}$
  Sample data augmentation parameters α_1, α_2
  $o_0^{\text{aug1}}, o_0^{\text{aug2}} \leftarrow \text{Aug}(o_0; \alpha_1), \text{Aug}(o_0; \alpha_2)$
  Set labels for o_0^{aug1} and o_0^{aug2}: $y_{\text{aug}}^1 \leftarrow 0$, $y_{\text{aug}}^2 \leftarrow 1$
  Compute $\mathcal{L}_{\text{aug}}(w, \phi)$ in (4)
  Compute $\mathcal{L}_{\text{BC}}(\theta, \phi)$ in (1)
  Update ϕ, w, and θ by solving the optimization problem in (6)
end for

Algorithm 2 Unsupervised domain adaptation

Require: $\mathcal{D}$: Demonstration dataset, $\mathcal{D}_{\text{target}}$: Observation data of target domain, K: number of update iterations, L: batch sizes, $\text{Aug}(\cdot; \alpha)$: function for data augmentation parameterized with α

Initialize networks: domain discriminator h_ψ, and feature extractor f_ϕ, policy network π_θ with pretrained weights
for $k = 1$ to K **do**
  Sample initial observation $o_0^S \sim \mathcal{D}$ and $o_0^T \sim \mathcal{D}_{\text{target}}$
  Sample data augmentation parameters α_3
  $o_0^{S,\text{aug3}}, o_0^{T,\text{aug3}} \leftarrow \text{Aug}(o_0^S; \alpha_3), \text{Aug}(o_0^T; \alpha_3)$
  Set labels for $o_0^{S,\text{aug3}}$ and $o_0^{T,\text{aug3}}$: $y_{\text{domain}}^1 \leftarrow 0$, $y_{\text{domain}}^2 \leftarrow 1$
  Compute $\mathcal{L}_{\text{domain}}(\psi, \phi)$ in (5)
  Compute $\mathcal{L}_{\text{BC}}(\theta, \phi)$ in (1)
  Update ϕ, ψ, and θ by solving the optimization problem in (7)
end for

E. Connection to mutual information maximization

The proposed method builds on mutual information theory, which quantifies statistical dependence by measuring the reduction in uncertainty of one variable given another. We optimize task-relevant information in indicators conditioned on features from a visual feature extractor. Our framework combines adversarial learning with augmentation and domain discriminators: the augmentation discriminator is trained adversarially to minimizes mutual information between features and augmentation indicators to induce augmentation-invariant representations, while the domain discriminator is trained adversarially to minimizes dependence on domain indicators. Together, these mechanisms maximize action-related information by eliminating feature dependence on augmentation and domain.

The mutual information [30], [31] between two variables, y and x, is defined as:

$$\begin{aligned} I(y; x) &= H(y) - H(y \mid x) && (8) \\ &= \mathbb{E}_{(y,x)\sim p(y,x)} \left[\log p(y \mid x) \right] + H(y), && (9) \end{aligned}$$

where $H(y)$ denotes the entropy of y, and $H(y \mid x)$ represents the conditional entropy of y given x. The equation above can be further rewritten as:

$$\begin{aligned} I(y; x) \geq\ & \mathbb{E}_{x\sim p(x)} D_{\text{KL}} \big[p(y \mid x) \parallel q_\eta(y \mid x) \big] \\ & + \mathbb{E}_{(y,x)\sim p(y,x)} \big[\log q_\eta(y \mid x) \big] + H(y). \end{aligned} \tag{10}$$

Given that $\mathbb{E}_{x\sim p(x)} D_{\text{KL}} \big[p(y \mid x) \parallel q_\eta(y \mid x) \big] > 0$, where $q_\eta(y \mid x)$ parameterized by a vector η approximates the true conditional distribution $p(y \mid x)$, the lower bound of the mutual information can be expressed as:

$$I(y; x) \geq \mathbb{E}_{(y,x)\sim p(y,x)} \left[\log q_\eta(y|x) \right] + H(y). \tag{11}$$

In our framework, we consider the mutual information between $f_\phi(o)$ and y, where $f_\phi(o)$ represents the image features extracted by the feature extractor, and y denotes the label indicating information about the data augmentation style and domain. The above equation shows that the lower bound of the mutual information can be maximized by maximizing the expected log-likelihood, as given in (11). Training the feature extractor to generate augmentation-invariant features using the augmentation discriminator, which identifies data augmentation styles, corresponds to minimizing the lower bound of the mutual information between the image features $f_\phi(o)$ and the label y_{aug} for the augmentation discriminator.

$$I(y_{\text{aug}}; f_\phi(o)) \geq \mathbb{E}_{(y_{\text{aug}},o)\sim p(y_{\text{aug}},o)} [\log q_w(y_{\text{aug}}|f_\phi(o))] + H(y_{\text{aug}}). \tag{12}$$

Learning feature extractor of visual information using the domain discriminator, which identifies domain-dependent features, is equivalent to minimizing the lower bound of mutual information between the image feature $f_\phi(o)$ and the output y_{domain} from domain discriminator.

$$\begin{aligned} I(y_{\text{domain}}; f_\phi(o)) \geq\ & \mathbb{E}_{(y_{\text{domain}},o)\sim p(y_{\text{domain}},o)} \left[\log q_\psi(y_{\text{domain}}|f_\phi(o)) \right] \\ & + H(y_{\text{domain}}). \end{aligned} \tag{13}$$

Therefore, the proposed method can be interpreted as approximately maximizing the following information bottleneck objective in the first stage:

$$I(y_{\text{action}}; f_\phi(o)) - aI(y_{\text{aug}}; f_\phi(o)), \tag{14}$$

In the second stage,

$$I(y_{\text{action}}; f_\phi(o)) - bI(y_{\text{domain}}; f_\phi(o)), \tag{15}$$

where y_{action} denotes the expert action associated with o. The parameters a and b are constants that balance the contributions of the corresponding terms. The above equation

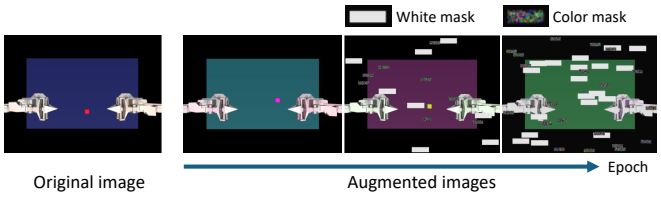


Fig. 3. Data augmentation: The white masks simulate visibility defects caused by light reflections, while the color masks simulate the placement of objects irrelevant to the task. The intensity of the color changes and the number of masks are configured to increase with each successive epoch.

indicates that the proposed method encourages the feature extractor to encode action-related information while discouraging the encoding of features altered by data augmentation and domains.

F. Data augmentation

In vision-based manipulation, lighting conditions, background variations, and disturbances caused by irrelevant objects are frequently problematic as sources of domain variation. To cope with these domain changes, our method applies data augmentation to the initial observation data to include these variables in the training data. We applied data augmentation using ColorJitter and RandomErasing, both widely used techniques in the PyTorch library [32]; ColorJitter is used to simulate lighting conditions and background changes, while RandomErasing is used to simulate reflected white light and the presence of objects unrelated to the task, respectively. For RandomErasing, a white mask is prepared to simulate light reflection, and a random color mask is used to imitate the placement of objects unrelated to the task as illustrated in Fig. 3. During the first stage, we employed ColorJitter and RandomErasing augmentations, whereas in the second stage, we applied subtle ColorJitter transformations along with white noise $\epsilon \sim \mathcal{N}(0, \sigma^2)$. The intensity of data augmentation is gradually increased throughout the learning process as it has shown effectiveness in policy learning [33]. The intensity of data augmentation is adjusted by applying progressively larger scaling factor γ to the parameters as follows:

$$\gamma = 2 \cdot \frac{1}{1 + e^{-10 \cdot r}} - 1. \tag{16}$$

where r is a variable calculated in relation to the current epoch and batch.

IV. Experimental Setup

A. Simulation

1) Dataset: Two simulation environments were developed using MuJoCo [34], as illustrated in Fig. 4. For each of these

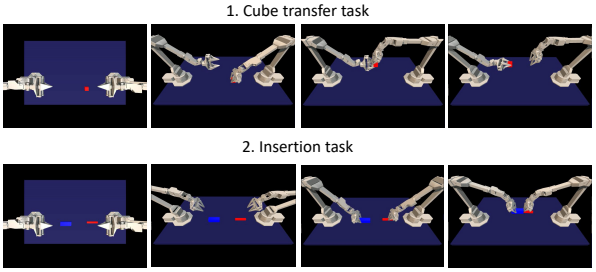


Fig. 4. Snapshots of tasks in simulation environment

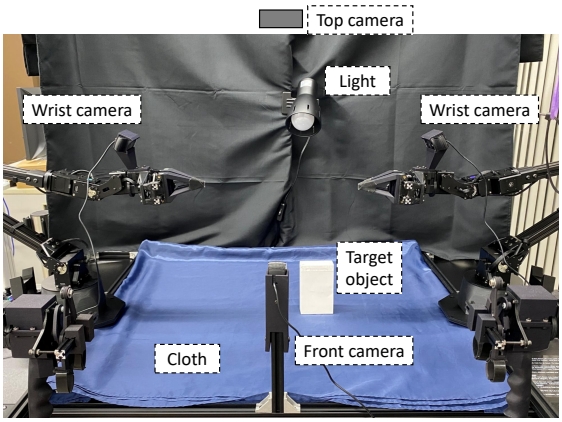


Fig. 5. Experimental setup with the real robot

simulation tasks, 50 demonstrations were collected using scripted policies, with random perturbations introduced to the initial object configurations. It is known that robots operating based on visual information often fail task execution due to changes in visual information, even when sunlight enters the workspace. In this simulation task, such challenging conditions were simulated. The experimental simulation environment emulated mild sunlight entering the workspace by setting lighting parameters.

2) Task description: Two simulation experiments were conducted. In the cube transfer task, the right arm picks up a red cube rolling on the table and passes it to the left arm's gripper. The clearance between the cube and gripper was set to 1 mm, and the cube's position was randomized within 200 mm along both horizontal axes. This setup can lead to grasp failures, dropping during transfer, or collisions between the gripper and cube. In the bimanual insertion task, the left arm holds a socket while the right arm inserts a peg into the "pin" inside the socket while maintaining mid-air contact. The clearance between socket and peg was set to 5 mm. This task requires not only grasping and transport but also contact-based insertion, making it highly challenging. The socket and peg positions were randomized within 100 mm (x-axis) and 200 mm (y-axis).

B. Real robot

1) Robot setup: We have experimented with a series of real robotics tasks using ALOHA, a low-cost, open-source, bimanual hardware setup proposed by Zhao et al. [3] as shown in Fig. 5. ALOHA consists of two WidowX ("leaders") and two ViperX ("followers"). In addition, four Logitech C922x webcams were installed, two cameras were attached to the wrists of each follower robot, and the other two were placed in front of and above the base on which the robots were installed. The camera mounted on the robot's wrist provides a detailed view of the object during manipulation, while the camera mounted on the base provides a wider field of view. The setup includes a 14-dimensional action space corresponding to the target joint positions (6DoF + gripper) of the bimanual robot and an observation space consisting of RGB images (480 x 640) from four cameras. These robot joint positions and images are collected at a

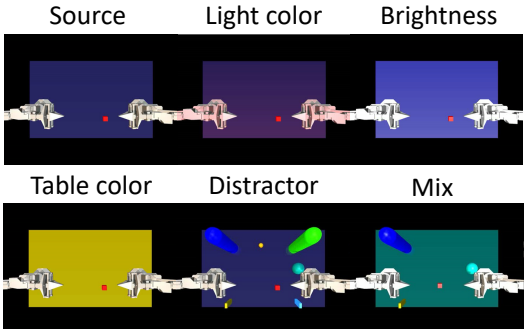


Fig. 6. Test domains in simulation: The workplace area was altered by changes in lighting conditions, table color, and the installation of distracting objects.

sampling rate of 50 Hz.

2) Data collection: Data for real robot tasks was collected via teleoperated demonstrations using ALOHA. Each episode, lasting 10–16 seconds depending on the task (corresponding to 500–800 time steps at 50 Hz), was executed by a single human operator. A total of 50 demonstrations were recorded on a real robot, amounting to about 8–13 minutes of task time and roughly 50 minutes of wall-clock time, as reported in [3]. The operator was familiar with the robotic system. Object positions were randomized within a 100 mm range on the x–y plane, and operation locations varied, resulting in demonstrations with inherent stochasticity despite a single demonstrator. A navy blue tablecloth was used on the platform to enable easy changes to table color during policy evaluation. All room lights were turned on to maintain consistent lighting conditions.

V. Evaluation

In our experimental study, we conducted a comparative analysis of several methodologies, using ACT as the baseline. We examined three additional methods: ACT combined with data augmentation (DA), an integration of data augmentation with adversarial feature learning (DA+AFL), and method that combines DA+AFL with unsupervised domain adaptation (DA+AFL+UDA). For visual feature extraction, we employed a ResNet-18 architecture as in [3] that was fine-tuned in unsupervised manner using demonstration data. Parameters related to ACT training were in accordance with [3] The loss weights were fixed as $\lambda_1 = 0.1$ in Eq. 6 and $\lambda_2 = 0.1$ in Eq. 7. Statistical significance was evaluated using Bonferroni-corrected paired t-tests after confirming normality with the Shapiro-Wilk test.

A. Simulation

In our simulation experiments, we report the average task success rates across two distinct task types across several different domains illustrated in Fig. 6. The results were obtained by conducting 20 trials for each of 5 different seeds, and the final metrics represent the mean performance across all these trials.

1) Cube transfer task: The result of the cube transfer task is shown in Fig. 7. The baseline ACT method achieved an 86% success rate in the source environment. It maintained

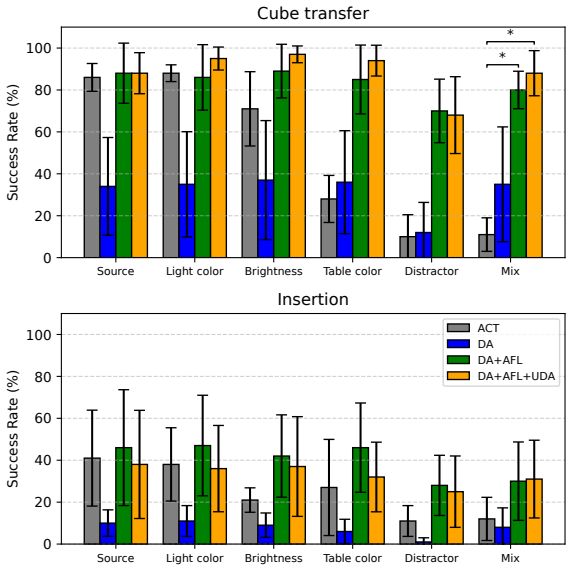
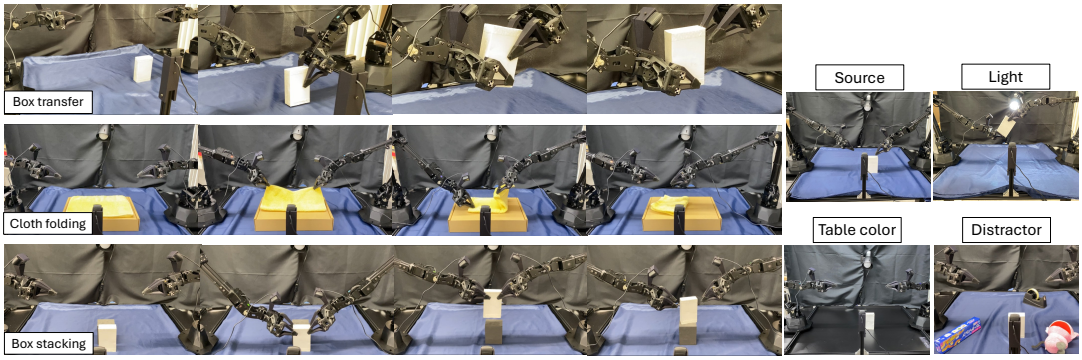


Fig. 7. Success rates of simulation tasks. The experiment was conducted using 5 different seeds. Asterisks denote statistical significance: $^*p < .05$

performance under light color changes (88%) but showed clear degradation with other variations: 71% for light intensity, 61% for table color, 10% with distractors, and 11% in the mixed condition. These results suggest that while ACT is robust to certain environmental changes such as light color changes, it remains vulnerable to other environmental variations.

When ACT was combined with data augmentation (DA), the task success rate dropped to about 35% even in the source environment, compared to ACT alone. This suggests that augmentation, while intended to handle domain variations, actually hinders task learning. In modified domains with varied lighting and distractors, the success rates were 35% (light color), 37% (brightness), 36% (table color), 12% (distractor), and 35% (mix), showing no performance improvement. These results highlight a trade-off: although augmentation may enhance domain generalization, it can interfere with task learning, underscoring the need for a more balanced approach. DA+AFL and DA+AFL+UDA methods outperformed ACT, both achieving 88% in the source domain. For DA+AFL, success rates were 86% (light color), 89% (brightness), 85% (table color), 70% (distractor), and 80% (mix). Adding UDA yielded modest gains, reaching 95%, 97%, 94%, 68%, and 88% under the same conditions.

2) Insertion task: As shown in Fig. 7, ACT achieved success rates of 41% (source domain), 38% (light color), 21% (brightness), 43% (table color), 11% (distractor), and 12% (mix condition). Using only data augmentation showed poor performance with success rates of 10% or lower across all domains, showing no improvement over baseline. DA+AFL and DA+AFL+UDA achieved comparable performance to ACT in the source domain (46% and 38% respectively). In different domains, DA+AFL showed success rates of 47% (light color), 42% (brightness), 46% (table color), 28% (distractor), and 30% (mix condition). DA+AFL+UDA achieved 36%, 37%, 32%, 25%, and 31% respectively. While adding



(a) Snapshots of real robot tasks (b) Tested domains

Fig. 8. Real robot tasks and tested domains

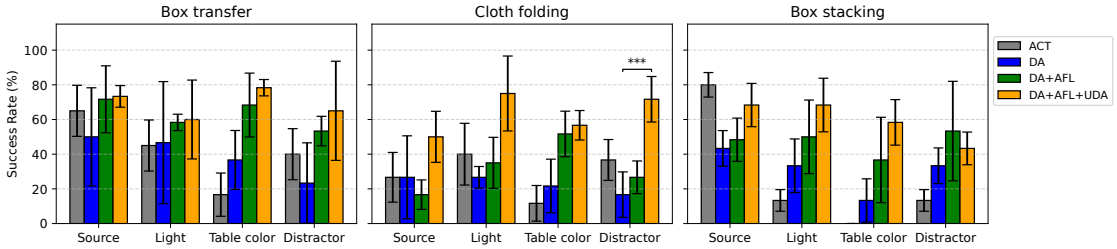


Fig. 9. Success rates of real robot tasks. Experiments were conducted using three different seeds. Asterisks denote statistical significance: $^{***}p < .001$

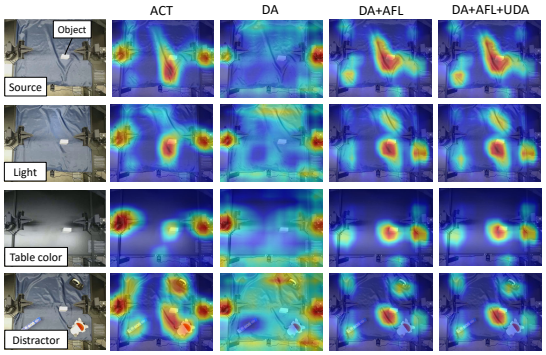


Fig. 10. Grad-CAM visualizations applied to the feature extractor representations. These results highlight the regions of the input frames that different methods focus on when extracting features.

UDA did not show significant improvements compared to cube transfer, both methods outperformed ACT and simple data augmentation in domain shift scenarios.

B. Real robot

We report the success rates of 3 real robot tasks across 4 different domains as shown in Fig. 8. We compared four methodologies: ACT, DA, DA+AFL, and DA+AFL+UDA, mirroring our simulation experiments. The results were obtained by conducting 20 trials for each of 3 different seeds, and the final metrics represent the mean performance across all these trials.

1) Box transfer: One arm picks up the box and hands it over to the other arm, enabling cross-arm object transfer. The result displayed in Fig. 9 shows ACT demonstrated success rates of 65%, 45%, 17% and 40% for source, light, table color, and distractor conditions respectively. DA method achieved success rates of 50%, 47%, 37% and 28%, while DA+AFL showed success rates of 72%, 58%, 68% and 53%. Moreover, DA+AFL+UDA method achieved the most consistent performance with success rates of 73%, 60%, 78% and 65% for the same conditions.

2) Cloth folding: Coordinated use of both arms is necessary such as one arm holds or repositions the cloth while the other folds it. Fig. 9 shows ACT demonstrated success rates of 27%, 40%, 12% and 37% for source, light, table color, and distractor conditions respectively. DA method achieved success rates of 27%, 27%, 22% and 17%, while DA+AFL showed success rates of 17%, 35%, 52% and 27%. DA+AFL+UDA method achieved generally strong and consistent performance, with success rates of 50%, 75%, 57% and 72% under the same conditions.

3) Box stacking: Both arms work together to lift the cube and place it precisely onto a block. In Fig. 9, ACT demonstrated success rates of 80%, 13%, 0% and 13% for source, light, table color, and distractor conditions respectively. The DA method achieved success rates of 43%, 33%, 13% and 33%, while DA + AFL showed success rates of 48%, 50%, 37% and 53%. DA+AFL+UDA method achieved high success rates of 68%, 68%, 58% and 43%, outperforming others in most cases under identical conditions.

We further analyzed the feature extractor's representations using Grad-CAM [35] as in Fig. 10. For ACT trained solely with imitation learning, the feature extractor attended to the manipulated object in the source domain; however, this attention was easily disturbed under domain shifts. When DA was applied to imitation learning, the attention to the manipulated object was diminished, and the focus became dispersed over the entire scene. In contrast, DA+AFL and DA+AFL+UDA, maintained strong attention to the manipulated object across domains, demonstrating their robustness to visual domain variations. By integrating the results of the downstream task, we further confirmed that applying UDA improved the policy

performance, leading to higher task success rates.

VI. CONCLUSIONS

In this work, we propose a robust feature learning framework for imitation learning, leveraging adversarial training. Our framework adopts a two-stage learning process. In the first stage, we apply adversarial feature learning to extract invariant features across augmentations, aiming to achieve robustness to domain shifts. In the second stage, we perform unsupervised domain adaptation using only the initial images from the test environment, which allows adaptation to target domains while keeping data collection costs low. The limitation of our approach lies in its reliance on the range of data augmentations applied; incorporating a broader variety of augmentations may further improve the robustness of feature extraction [15], [19]. Despite these limitations, our method represents a meaningful advancement in applying adversarial feature learning and unsupervised domain adaptation to imitation learning. We believe that our findings provide valuable insights for developing more robust robotic learning systems under diverse environmental conditions.

REFERENCES

[1] A. Khazatsky *et al.*, "DROID: A Large-Scale In-The-Wild Robot Manipulation Dataset," in *Proc. Robotics: Science and Systems (RSS)*, 2024.

[2] A. O'Neill *et al.*, "Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2024, pp. 6892–6903.

[3] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware," in *Proc. Robotics: Science and Systems (RSS)*, 2023.

[4] Z. Fu, T. Z. Zhao, and C. Finn, "Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation," in *Conference on Robot Learning (CoRL)*, 2024.

[5] S. Dass, K. Pertsch, H. Zhang, Y. Lee, J. J. Lim, and S. Nikolaidis, "PATO: Policy Assisted TeleOperation for Scalable Robot Data Collection," in *Proc. Robotics: Science and Systems (RSS)*, 2023.

[6] A. Brohan *et al.*, "RT-1: Robotics Transformer for Real-World Control at Scale," in *Proc. Robotics: Science and Systems (RSS)*, 2023.

[7] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid, *et al.*, "RT-2: Vision-language-action models transfer web knowledge to robotic control," in *Conference on Robot Learning (CoRL)*, 2023, pp. 2165–2183.

[8] J. Li, Y. Zhu, Y. Xie, Z. Jiang, M. Seo, G. Pavlakos, and Y. Zhu, "Okami: Teaching humanoid robots manipulation skills through single video imitation," in *Conference on Robot Learning (CoRL)*, 2024.

[9] J. Li, T. Lu, X. Cao, Y. Cai, and S. Wang, "Meta-imitation learning by watching video demonstrations," in *International Conference on Learning Representations (ICLR)*, 2021.

[10] T. Osa and T. Harada, "Discovering multiple solutions from a single task in offline reinforcement learning," in *International Conference on Machine Learning (ICML)*, vol. 235, 2024, pp. 38 864–38 884.

[11] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2017, pp. 23–30.

[12] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2018, pp. 3803–3810.

[13] J. Pari, N. Shafiullah, S. Arunachalam, and L. Pinto, "The Surprising Effectiveness of Representation Learning for Visual Imitation," in *Proc. Robotics: Science and Systems (RSS)*, 2022.

[14] P. Mitrano and D. Berenson, "Data Augmentation for Manipulation," in *Proc. Robotics: Science and Systems (RSS)*, 2022.

[15] Q. Chen, S. C. Kiami, A. Gupta, and V. Kumar, "GenAug: Retargeting behaviors to unseen situations via Generative Augmentation," in *Proc. Robotics: Science and Systems (RSS)*, 2023.

[16] Z. Yuan, T. Wei, S. Cheng, G. Zhang, Y. Chen, and H. Xu, "Learning to manipulate anywhere: A visual generalizable framework for reinforcement learning," in *Conference on Robot Learning (CoRL)*, 2024.

[17] S. Nasiriany, A. Maddukuri, L. Zhang, A. Parikh, A. Lo, A. Joshi, A. Mandlekar, and Y. Zhu, "Robocasa: Large-scale simulation of everyday tasks for generalist robots," in *Proc. Robotics: Science and Systems (RSS)*, 2024.

[18] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín, "What matters in learning from offline human demonstrations for robot manipulation," in *Conference on Robot Learning (CoRL)*, vol. 164, 2022, pp. 1678–1690.

[19] Z. Zhuang, R. Wang, N. Ingelhag, V. Kyrki, and D. Kragic, "Enhancing visual domain robustness in behaviour cloning via saliency-guided augmentation," in *Conference on Robot Learning (CoRL)*, 2024.

[20] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. March, and V. Lempitsky, "Domain-adversarial training of neural networks," *Journal of Machine Learning Research*, vol. 17, no. 59, pp. 1–35, 2016.

[21] T. Westfechtel, H.-W. Yeh, Q. Meng, Y. Mukuta, and T. Harada, "Backprop induced feature weighting for adversarial domain adaptation with iterative label distribution alignment," in *Proc. IEEE/CVF Winter Conf. on Applications of Computer Vision (WACV)*, 2023, pp. 392–401.

[22] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International Conference on Machine Learning (ICML)*, 2017, pp. 214–223.

[23] J. Shen, Y. Qu, W. Zhang, and Y. Yu, "Wasserstein distance guided representation learning for domain adaptation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

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

[25] A. Tanwani, "DIRL: Domain-invariant representation learning for sim-to-real transfer," in *Conference on Robot Learning (CoRL)*, 2021, pp. 1558–1571.

[26] K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, K. Konolige, *et al.*, "Using simulation and domain adaptation to improve efficiency of deep robotic grasping," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2018, pp. 4243–4250.

[27] V. Giammarino, J. Queeney, and I. C. Paschalidis, "Visually Robust Adversarial Imitation Learning from Videos with Contrastive Learning," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2025.

[28] T. Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel, J. Peters, *et al.*, "An algorithmic perspective on imitation learning," *Foundations and Trends® in Robotics*, vol. 7, no. 1-2, pp. 1–179, 2018.

[29] T. Tsuji, Y. Kato, G. Solak, H. Zhang, T. Petrič, F. Nori, and A. Ajoudani, "A survey on imitation learning for contact-rich tasks in robotics," *arXiv preprint arXiv:2506.13498*, 2025.

[30] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[31] T. Osa, V. Tangkaratt, and M. Sugiyama, "Discovering diverse solutions in deep reinforcement learning by maximizing state–action-based mutual information," *Neural Networks*, vol. 152, pp. 90–104, 2022.

[32] S. Imambi, K. B. Prakash, and G. Kanagachidambaresan, "Pytorch," *Programming with TensorFlow: solution for edge computing applications*, pp. 87–104, 2021.

[33] L. Fan, G. Wang, D.-A. Huang, Z. Yu, L. Fei-Fei, Y. Zhu, and A. Anandkumar, "Secant: Self-expert cloning for zero-shot generalization of visual policies," in *Proc. 38th International Conference on Machine Learning*, vol. 139, 2021, pp. 3088–3099.

[34] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2012, pp. 5026–5033.

[35] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, 2017, pp. 618–626.