Few-Shot Adversarial Domain Adaptation for Warm-Starting Policy Network

Seungjae Lee*, Wonguk Cho*, Sua Lee*, and Hyungseo Ahn*
Seoul National University

Abstract

In numerous practical scenarios of visual control problems, it is common for the target environments to differ from the environments in which the policy was trained. To overcome this challenge, recent studies have introduced methodologies to develop generalizable policies or automatically adapt learned policies to target environments. Unfortunately, these existing methods cannot be generally applied to arbitrary pretrained policies, since they require additional assumptions about the pretraining phase such as multiple pretraining environments or specific auxiliary loss in the pretraining phase. In this study, we propose a simple yet effective adaptation method that is agnostic to the pretraining process. Our approach enables warm-starting the policy in the target environment, using only the learned policy and a few episodes from the pretraining (source) environment. To do so, we adapt the learned representations of the policy to the target environment by minimizing the discrepancy between source and target environments while preserving essential information for reinforcement learning. We demonstrate that the proposed method efficiently accelerates policy learning in test environments.

1. Introduction

Recent Reinforcement Learning (RL) methods have shown promising results in various control tasks directly from high-dimensional observation [18, 19, 32]. One of the most significant hurdles in vision-based RL studies is obtaining low-dimensional latent representations from high-dimensional raw observations, which is crucial for training downstream RL agents. Thus, prior studies over the past few years, focusing on learning representations, have led to remarkable improvements by proposing to pretrain the encoder [3, 16, 24], or learn the representations alongside the RL agent with auxiliary tasks [19, 23].

However, one of the long-standing challenges in vision-based RL research still limits the applicability of the previous approaches; the difficulty of generalizing an RL agent learned in one domain to a new environment. This limitation is particularly exacerbated in real-world problems. Consider an RL agent trained in a simulation or training environment. When this agent is applied to real-world situations, even a small amount of disturbances in the environment (e.g. camera configurations or the texture of the object) can lead to severe performance degradation of the agent, and the agent requires a significant amount of target domain data to re-train or fine-tune the RL agent.

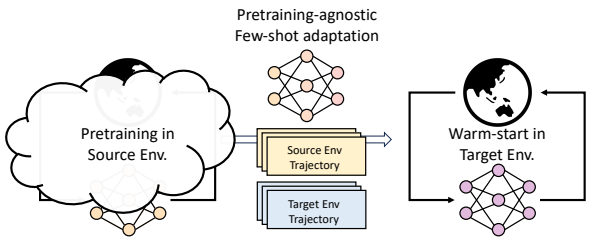


Figure 1. We propose a simple adaptation method that is independent of the pretraining phase and can effectively enhance learning in the target environment. In contrast to existing approaches that involve modifying the pretraining process, our approach can be utilized with any learned policy, supported by a small number of episodes from the pretraining (source) environment.

To address such overheads on applying the agents in the target environment, previous methods have been proposed to learn generalizable policies or adapt the learned policies to target environments. Unfortunately, the majority of previous works assume access to multiple pretraining environments to train robust policies against distracting (task-irrelevant) elements [2, 12, 15, 27]. This assumption implies that they are aware of which aspects will change in the test environment and can generate diverse environments with variations during pretraining, which is quite unrealistic in real-world problems.

While a few studies have attempted to adapt policies from a single pretraining environment to unseen target environments without relying on multiple pretraining environments, these approaches still have certain limitations. They are only capable of handling a limited range of variations [34], demonstrate their effectiveness solely in low-dimensional observations [7], or rely on the assumption that specific auxiliary losses need to be employed during the pre-

*Equal contribution.

training process, which becomes a limitation when we have access only to a pretrained model and cannot modify the pretraining process [11].

In this work, we consider practical scenarios of domain adaptation where modification of the pretraining is NOT possible, and only the learned policy with a small amount of data from the pretraining environments is available. We propose a few-shot domain adaptation approach for an RL agent, allowing the policy network to be effectively warm-started in the target environment without relying on additional assumptions during the pretraining phase. To efficiently restore the performance of the policy network in the target environment, the proposed algorithm first adapts the encoder of the visual controller by jointly optimizing the adversarial domain adaptation loss alongside RL task-aware loss with few-shot data. Then, we fine-tune the policy network in the environment to succeed in the new task quickly. We demonstrate several experiments on visual control tasks to explore the effectiveness of the proposed adaptation method.

2. Related work

Representation Learning for Visual Policies It is one of the most challenging and costly parts of vision-based RL that extracting low-dimensional latent vectors from high-dimensional raw observations for an agent to learn an optimal policy. To enhance data efficiency and address this challenge in visual control problems, previous studies have made notable improvements by adopting contrastive learning [19, 20], data-augmentation techniques [18, 32], image reconstruction [14, 16], and leveraging various task-aware supervisions [11, 33]. In this work, we also utilize the aforementioned data augmentation techniques and auxiliary tasks for quick adaptation to the target domain with few transition data.

Adversarial Domain Adaptation Domain adaptation is a critical aspect of machine learning, which aims to improve the performance of models when faced with different source and target distributions by leveraging knowledge learned from the source domain and adapting it to the target domain. In recent studies, adversarial domain adaptation (ADA) methods [6, 21, 22, 25] have emerged as prominent deep learning architectures, yielding state-of-the-art results. Inspired by generative adversarial networks (GANs) [8], these approaches address domain shift through an adversarial min-max game involving two players: a feature extractor that acts as a generator and a domain classifier that acts as a discriminator. One of the pioneering efforts in this field, domain-adversarial training of neural networks (DANN) [5], employs a gradient reversal layer as a domain discriminator to differentiate between source and target distributions, while concurrently training a deep classification model to produce transferable representations that are indistinguishable to the domain discriminator. Moreover, adversarial discriminative domain adaptation (ADDA) [28] has been introduced, which involves using the training data from the source domain to initialize the target model. This is followed by an adversarial adaptation process, which ultimately results in a target domain-specific classifier. The impressive performance of adversarial learning in domain adaptation has prompted extensive research in various tasks beyond image classification [21, 25], encompassing semantic segmentation [29, 31], object detection [9, 13], and natural language processing (NLP) tasks [4, 30].

Adaptations for Visual Policies Visual control tasks have been developing over the last few years. Some studies have proposed methodologies for automatically adapting a learned policy to the target environment. The majority of the previous works utilized domain randomization or meta-learning methodologies, which not only have the disadvantage of needing to generate a plethora of training environments during learning, but also require the test environment to be included within the distribution of the training environments. Other attempts have been made, such as the concept of bisimulation [34] or generalization of representation learning through data augmentation during training [12, 15]. Each of these has its drawbacks: the bisimulation requires an invariant representation that should also be preserved in the test environment, and the augmentation cannot be generalized in test environments that deviate from the domain adaptation applied during training. In other words, these methodologies can handle domain shifts such as simple background changes, color shifts, or those within the range of the domain adaptation applied during training, but they cannot be applied in other unseen environments. There have been a few previous studies that proposed methodologies applicable to unseen environments [7, 11]. However, in this case of [11], it requires a specific auxiliary loss during training, and its effect diminishes when the degree of domain shift is severe. Furthermore, [7] has the limitation that it has only demonstrated performance in state-based tasks, not in visual control tasks.

Contrastive Self-supervised Learning Contrastive learning [10] and self-supervised learning have emerged as effective strategies to leverage unlabeled data in visual tasks. Contrastive learning aims to learn rich representations by contrasting similar and dissimilar instances in the representation space, thereby improving downstream task performance. Meanwhile, self-supervised learning is a learning paradigm that capitalizes on unlabeled data by formulating auxiliary tasks. It leverages the inherent structure of data to learn useful representations without the necessity of explicit supervision. Combining these two

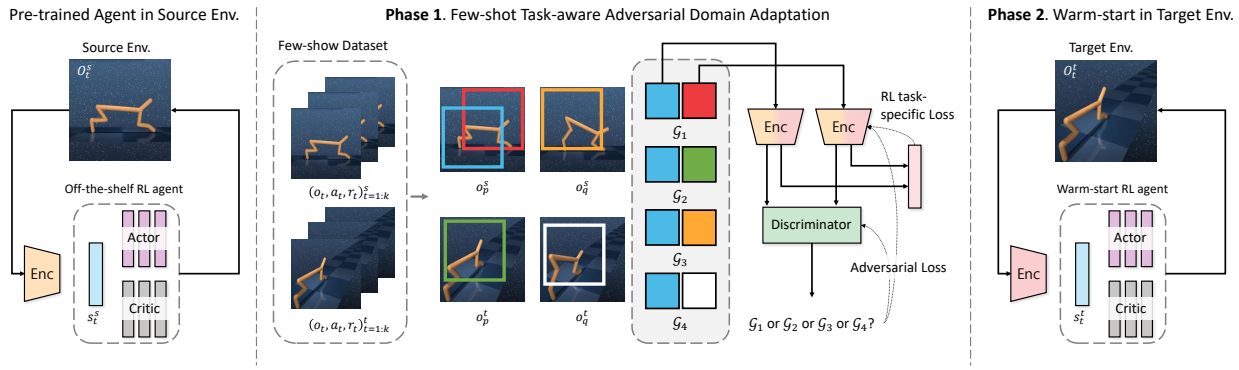


Figure 2. To adapt the visual controller to the target tasks in a new environment, the proposed method first adapts the encoder of the policy network by jointly optimizing the RL task-aware loss alongside auxiliary adversarial loss (Phase 1.). Then, ours fine-tune the policy network in the target environment to achieve the new task quickly (Phase 2.).

powerful paradigms, the SimCLR [1] has been developed. It creates positive and negative pairs from the data to teach the model about similarity. A positive pair is two differently augmented versions of the same instance, and negative pairs consist of different instances. SimCLR uses a base encoder and a projection head to map these images to a latent space, where a contrastive loss is used to maximize agreement between augmented views. SimCLR has achieved top performance in various tasks, highlighting its potential in improving learned representations.

3. Preliminaries

Problem Settings We consider a Markov Decision Process (MDP) with a tuple $(\mathcal{O}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma)$, where $o_k \in \mathcal{O}$, $a_k \in \mathcal{A}$, $\mathcal{T}(o_{k+1}|o_k, a_k)$, $\mathcal{R} : \mathcal{O} \times \mathcal{A} \to \mathbb{R}$ and γ indicates high-dimensional observation at timestep k, action at timestep k, transition function, reward, and discount factor respectively. Also, the RL agent in our framework employs the encoder $\phi : \mathcal{O} \to \mathcal{S}$, where $\mathcal{S}$ is the low-dimensional latent space. We note that the superscript s and t across this manuscript indicate the source and target respectively.

Suppose that there are two environments with different domains of observation: a *source* environment $\mathrm{Env}^{\mathrm{s}}$ and a *target* environment $\mathrm{Env}^{\mathrm{t}}$. From an interaction with $\mathrm{Env}^{\mathrm{s}}$, we observe $o^s \in \mathcal{O}$, which is a realization from a random variable O^s. From interacting with $\mathrm{Env}^{\mathrm{t}}$, we are given $o^t \in \mathcal{O}$, which is a realization from a random variable O^t. Accordingly, O^s and O^t represent a source domain and a target domain, respectively. We assume that there exists a domain shift between O^s and O^t, such that their marginal probability distributions are different (*i.e.*, $p(O^s) \neq p(O^t)$).

Our goal is to adapt an RL agent trained in the source environment $\mathrm{Env}^{\mathrm{s}}$ to the target environment $\mathrm{Env}^{\mathrm{t}}$ through few-shot interactions. For every interaction i, an agent acquires a tuple of data, (o_i, a_i, r_i) ($o \in \mathcal{O}$: observation, $a \in \mathcal{A}$: action, $r \in \mathbb{R}$: reward). We assume that the agent stores a fraction of source data $\mathcal{D}^s = \{(o_i^s, a_i, r_i)\}_{i=1}^{n^s}$ from its previous source training, and acquires target data $\mathcal{D}^t = \{(o_i^t, a_i, r_i)\}_{i=1}^{n^t}$ sampled from few-shot interactions with the target environment.

Discussion on Data Pairing The previous approach in the interim report ("**Method 1**") assumes that we have n samples from the target data, each of which can be matched sample-by-sample with the corresponding sample from the source data. That is, we assume that the tuples of $\mathcal{D}^{\mathrm{s}}$ and $\mathcal{D}^{\mathrm{t}}$ form pairs, and the only difference between them is the appearance discrepancy caused by domain shift. ($(\Omega(o_t), a_t, r_t)^s = (o_t, a_t, r_t)^t$).

However, the feedback received during the interim review highlighted that this assumption of collecting paired samples may not hold in real-world scenarios outside of simulated environments. Upon careful consideration, we have realized that relying on the ability to pair samples between the source and target data is not always feasible in practical situations. If the process of pairing samples between source and target data is easily achievable, then there is no need for training. In such cases, we can directly employ the source model in the target environment by transforming the target observations (o^t) into the format of source observations (o^s) through $\Omega^{-1}(o^t) = o^s$.

Hence, in this final report, we present an alternative approach ("**Method 2**") where samples from different domains are not necessarily correlated with one another. We define a domain shift from a broader perspective as a change in marginal probability distributions in the observation space between two environments, as described earlier in this section. Consequently, we have made modifications to previous adaptation methods, which will be introduced in the following section.

4. Methods

In this work, we aim to adapt the encoder ϕ for few-shot supervised domain adaptation, addressing shifted ob-

servations in the target environment. Specifically, our approach involves a two-step process for adaptation: (1) performing few-shot task-aware adversarial domain adaptation to bridge the gap between source and target domains by leveraging data from $\mathcal{D}^{\text{s}}$ and $\mathcal{D}^{\text{t}}$ (Phase 1 in Fig. 2), and (2) initiating a warm-start with the adapted encoder in the target environment (Phase 2 in Fig. 2). To optimize the encoder during the domain adaptation phase, we employ a loss function that simultaneously learns robust feature representations that capture the dynamics of the environment, as described by the MDP components $\mathcal{T}$ and $\mathcal{R}$, and adversarially reduces discrepancies between source and target domains.

4.1. Representation Learning for Downstream RL

To learn effective low-dimensional latent representations from high-dimensional observation images, we adapt the pretrained encoder to the target environment using the few-shot target dataset $\mathcal{D}^{\text{t}}$. By leveraging reward prediction and successor feature representation auxiliary loss, the encoder can be fine-tuned to better adapt to the target environment, focusing on relevant dynamics while maintaining informativeness for the downstream RL agent. The adaptation process minimizes the discrepancy between source and target domains while preserving essential environment dynamics, enabling the RL agent to warm-start in the target environment and perform well despite the domain shift.

Reward prediction The reward prediction focuses on learning a latent state representation that captures the environment's reward structure. The encoder ϕ is trained to extract informative features for reward prediction head r_θ which predict ground truth rewards r_{GT} from the latent states $s_t = \phi(o_t)$. This task facilitates the learning of representations that are informative about the rewards, which are critical for downstream RL agents to make optimal decisions. The loss for reward prediction is defined as the L2-norm between the predicted rewards $r_\theta(s_t)$ and the ground truth rewards r_{GT}:

$$\mathcal{L}_{\text{RE}} = \mathbb{E}[\|r_\theta(s_t) - r_{\text{GT}}\|_2] \tag{1}$$

Successor feature representation The successor feature representation, which is inspired by [17], aims to learn a latent state representation that captures the long-term dynamics of the environment. By training the encoder ϕ to satisfy the condition $\phi(s_t) = 1 + \gamma\phi(s_{t+1})$, the model learns to predict the expected future states given the current state and action, enabling the RL agent to plan ahead. The loss for successor feature representation is defined as the L2-norm between the current latent state and the expected future latent state:

$$\mathcal{L}_{\text{SFR}} = \mathbb{E}[\|\phi(s_t) - [1 + \gamma\phi(s_{t+1})]\|_2] \tag{2}$$

By optimizing the encoder ϕ with these auxiliary tasks, we ensure that it learns a latent representation that captures essential information about the environment dynamics. This representation serves as a foundation for subsequent adaptation to the target environment, enabling the RL agent to warm-start and perform well in the new domain.

4.2. Auxiliary Adversarial Domain Adaptation Loss

To adapt the encoder for the target environment, we propose two different approaches for few-shot adversarial domain adaptation. Method 1 will assume that there exist paired samples from source and target domains and present how to take advantages of these samples. On the other hand, Method 2 have no assumptions on data pairing, which allows us to use data samples collected from source and target domains independently.

Method 1 (Paired Samples) In this approach, we use paired data from $\mathcal{D}^{\text{s}}$ and $\mathcal{D}^{\text{t}}$ and optimize the encoder ϕ to minimize the domain discrepancy between the source and target domains. Given an anchor image, which is a cropped region from source s at time p, o^s_p, we define the relationships $\{G_1, G_2, G_3, G_4\}$ for the adversarial training, utilizing other random cropped regions from the paired-image set $\{o^s_p, o^t_p, o^s_q, o^t_q\}$: G_1 associates the anchor with another region within the same image o^s_p, G_2 connects the anchor with a region from the target domain at the same timestep as o^t_p, G_3 links the anchor to a region from the source domain but at a different timestep from o^s_q, and G_4 associates the anchor with a region from a target domain at a different timestep from o^t_q. Utilizing these relationships, the adversarial domain adaptation loss can be defined as follows:

$$\mathcal{L}_{ADAL} = -\mathbb{E}[y_{G_1}\log(\psi(\phi(G_2))) - y_{G_3}\log(\psi(\phi(G_4)))] \tag{3}$$

where y_{G_i} is a label of G_i, ψ is a discriminator, and ϕ is the encoder, which plays a similar role with the generator from GANs. Also, the discriminator is jointly optimized to guess the label of each group using the following loss:

$$\mathcal{L}_{ADAL-D} = -\mathbb{E}[\sum_{i=1}^{4} y_{G_i}\log(\psi(\phi(G_i)))] \tag{4}$$

Method 2 (Independent Samples) The main goal of this method is to discriminate whether given two images are from the same domain or not. The two images are sampled independently and do not need to correspond to each other. For this task, we make two groups of data; The first group G_5 consists of pairs of two randomly sampled images, both from the source domain, which is equivalent to the union of G_1 and G_3. The second groups G_6 consists of two randomly sampled images, one from the source domain and the other from the target domain, which is equivalent to

the union of G_2 and G_4. Then, for the discriminator ψ, we train it to properly discriminate whether a given image pair belongs to either G_5 or G_6, which is given by

$$\mathcal{L}_{\text{ADAL-D}} = -\mathbb{E}[\sum_{i=5}^{6} y_{G_i} \log(\psi(\phi(G_i)))], \tag{5}$$

On the other hand, we update the encoder ϕ in a way that the discriminator ψ cannot distinguish between groups 5 and 6. Thus, the adversarial loss for the encoder is defined as

$$\mathcal{L}_{\text{ADAL}} = -\mathbb{E}[y_{G_5} \log(\psi(\phi(G_6)))], \tag{6}$$

which forces the encoder ϕ to deliberately mislead the discriminator ψ by falsely claiming that a pair of images belong to the same domain, even though they do not.

4.3. Auxiliary Instance Discrimination Loss

To address the absence of time-wise comparisons in adversarial learning, we integrate InfoNCE loss based on SimCLR [1] as an auxiliary loss. This instance discrimination loss enables our model to discriminate samples from different time periods based on self-supervised learning, enhancing temporal understanding and overall performance. Specifically, we establish a contrastive prediction task for every minibatch containing B samples from the target dataset. This task involves creating pairs of augmented examples within the minibatch, resulting in a total of $2B$ data points. In this task, a positive pair consists of the same example, but with different augmentations. The other $2(B-1)$ augmented examples are considered as negative examples. Then, we define the loss function for a positive pair of examples (i, j) as

$$\mathcal{L}_{\text{SCLR}}(i,j) = -\log \frac{\exp(\text{sim}(\phi_i, \phi_j)/\tau)}{\sum_{k=1}^{2B} \mathbb{1}_{[k \neq i]} \exp(\text{sim}(\phi_i, \phi_k)/\tau)}, \tag{7}$$

where $\text{sim}(\phi_i, \phi_j) = \phi_i^\top \phi_j / \|\phi_i\| \|\phi_j\|$, τ represents a temperature parameter, and $\mathbb{1}_{[k \neq i]}$ is an indicator function, which returns 1 if $k \neq i$ and 0 otherwise. We sum up the losses of all positive pairs for each minibatch for training.

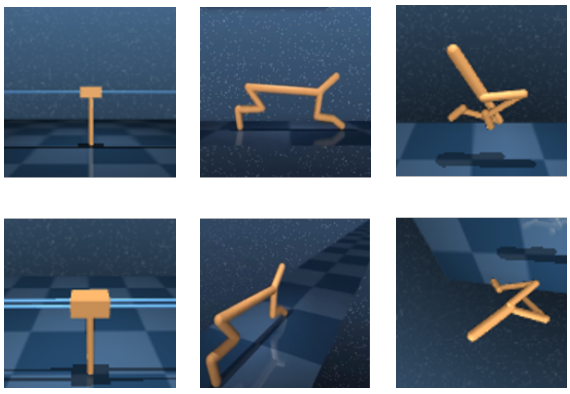
4.4. Total Loss

Finally, the total losses for the encoder ϕ and the discriminator ψ for the adversarial domain adaptation can be represented as follows:

$$\min_{\phi} \mathcal{L}_{\text{RE}} + \mathcal{L}_{\text{SFR}} + \alpha \mathcal{L}_{\text{ADAL}} + \beta \mathcal{L}_{\text{SCLR}} \tag{8}$$

$$\min_{\psi} \mathcal{L}_{\text{ADAL-D}}, \tag{9}$$

where α and β are the hyperparameters that balance the representation learning task losses and the auxiliary losses. The losses 8 and 9 are updated alternately. When 8 is optimized, we freeze the discriminator ψ. Similarly, when 9 is optimized, we freeze the encoder ϕ. This adversarial training process leads the encoder to learn features that are not only useful for reinforcement learning tasks but also more robust to differences between the source and target domains, thereby enabling warm-starting in the target environment.



(a) Cartpole swingup (b) Cheetah run (c) Walker walk

Figure 3. The images of each experiment and domain shift. From left to right, each experiment is cart-pole swingup, cheetah run, and walker walk. The upper setting is original source setting and the lower setting is target setting which changed the camera angle

5. Experiment details

We evaluate the proposed framework on a visual-control task on the DeepMind Control Suite (DMC) [26]. We selected three RL tasks from it, named cart-pole swing-up, cheetah run, and walker walk. Each example image used for training on each RL task is shown in figure 3. The above images are the setting from original source settings, and the below images are domain-shifted images with camera settings changed. Here are a brief explanations of each task.

Cartpole swingup has two components, a cart, and a pole. The cart is used as a base and the pole is connected to the cart. Also, the cart can only move on one dimension. By applying force on the cart, we balance the pole to maintain the pole pointing up. Also, it is a swing-up task, thus pole is initially pointing downwards.

Cheetah run is a task for making a cheetah-shaped structure go forward. The dimension of action of the cheetah is 6. The reward is given by the speed of going forward, with a maximum value limit.

Walker walk is similar to the cheetah run task. There is a planar shape with a body and feet with an action dimension of 6. By applying force, the walker should walk forward and its velocity is used for the reward.

On evaluation, we executed 35k steps on the cart pole swing-up task, and 100k steps on other tasks. For stability, we tested three random seeds of each experiment and

averaged them.

5.1. Baseline

As a base structure of RL, we used the CURL [19]. We used two baseline settings, including "from scratch" and "source encoder".

From scratch is a naive approach that used a randomly initialized encoder and trained it from scratch. We skipped the pretraining step and directly started training on the target environment. We trained with a learning rate of $lr = 2 \times 10^{-4}$ on the cheetah run task and used a learning rate of $lr = 1 \times 10^{-3}$ on other tasks.

Source encoder setting means we used the pretrained encoder trained on the source domain. Then without domain adaptation, the pretrained encoder is fine-tuned on the target environment. On fine-tuning, we used $lr = 2 \times 10^{-5}$ for the cheetah run task and $lr = 1 \times 10^{-4}$ for the remaining tasks as a learning rate which is 1/10 value of the "from scratch" setting.

5.2. Domain Adaptation

Our Domain Adaptation Method In the domain adaptation training step (Phase 1 of figure 2), we used 2000 sets of images from the target environment for training. The discriminator and encoder (generator) trained alternately in an adversarial fashion with a ratio of 1:1 and the total iteration was 250 epochs. We set the learning rate of the discriminator, encoder, and reward head as $1 \times 10^{-3}, 1 \times 10^{-4}, 1 \times 10^{-3}$, respectively. The coefficient of auxiliary adversarial loss and SimCLR loss was set with $\alpha = 0.1, \beta = 0.3$ on equation 8. Moreover, the temperature of SimCLR loss was $\tau = 0.07$ on equation 7 and other hyperparameter on equation 2 was set to $\gamma = 0.99$

We used 2000 steps of additional adaptation on the target environment before evaluating the result as a warm start(Phase 2 of figure 2). Thus we subtracted these 2000 steps on evaluation. Same as the source encoder setting, we used $lr = 2 \times 10^{-5}$ for the cheetah run task and $lr = 1 \times 10^{-4}$ for other tasks.

Domain Adaptation Baseline For comparing the domain adaptation ability with other methods, we used Policy Adaptation during Deployment method (PAD) [11] as a domain adaptation baseline. However, there is a slight difference in the problem setting between our algorithm and PAD; PAD does not utilize a reward signal and RL fine-tuning, but instead employs an auxiliary loss (inverse dynamics prediction) in an online adaptation fashion. For fair comparisons, we adjusted our algorithm to align with the setup of PAD. First, we removed the RL objectives and auxiliary losses related to reward signals in adaptation. Also, following PAD, we adapt the encoder in an online adaptation fashion. The main difference between PAD and ours is the use of adversarial domain adaptation loss and instance discrimination loss (SimCLR), alongside the original PAD auxiliary loss (inverse dynamics prediction). We evaluated these adaptation methods for two settings each changing the number of adapted episodes. We tried using 1 and 10 episodes for adapting and denoting after the adaptation with -(For example, PAD-10 means PAD method with 10 adapted episodes). Especially on comparing with PAD, we experimented with two domain shifts, including color shift and viewpoint shift.

6. Results

6.1. Domain Adaptation Evaluation

We first evaluated the proposed framework with baseline settings("from scratch" and "source encoder"). Corresponding steps-return graphs are displayed in figure 4. The experiment settings are distinguished by the colors and shading indicates a standard deviation across 3 seeds.

On cheetah run and walker walk tasks, our method shows similar or slightly lower returns on early steps (before 20k steps) and gives considerably better returns after 20k steps. The returns increase over 20 and 60 on each "from scratch" and "source encoder" setting of the cheetah run task. Moreover, on the walker walk task, returns improved by over 60 on both baselines. However, on the cart-pole swing up task, our method shows lower returns on most of the steps and finally becomes similar performance on convergence. It can be seen as a slow convergence.

Discussion As discussed earlier, we observed that performance improvement is not immediate during the early steps of adaptation. We speculate that this is due to the RL policy network which requires time to learn the specifics of the target domain. Especially in the initial steps of fine-tuning in the target environment, the RL policy network learns the actor and critic network from the new features extracted by the updated encoder. After the initial steps for RL fine-tuning, we observed that the encoder's capabilities contribute to performance enhancement and these capabilities lead to higher performance in the target environment.

Moreover, our framework did not demonstrate improvement in the cartpole swingup task. We believe this is due to the minimal domain shift present in the cartpole task. This means that utilizing only RL objectives rather than the joint objectives for domain adaptation might be sufficient: the inclusion of auxiliary objectives can have a negative impact on RL performance in tasks with minimal domain shift, such as cartpole.

As a result, we conclude that our method possesses domain adaptation capabilities and can enhance the performance of the RL model when adequately trained on the target domain over a sufficient number of steps.

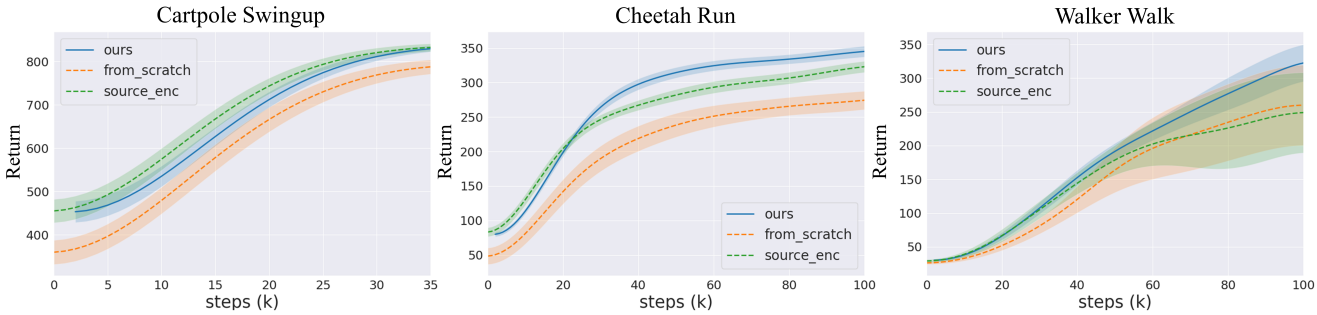


Figure 4. Steps-return graphs of our framework and baseline without domain adaptation on three different RL tasks.

6.2. Comparing with Other Method

As mentioned in section 5.2 we compared our framework to the existing domain adaptation method, PAD, and the result is displayed in table 1. First, on the color shift task, which the PAD had targeted, Ours-10 shows the best result on the walker walk, and PAD-10 gives the best result on the cheetah run. On cart-pole swing up, the model without adaptation shows the best result. Moreover, on viewpoint shift, the Ours-10 showed better results than PAD on every task. These results show our method has considerable domain adaptation ability compared to the existing method.

6.3. Ablation study

We conducted three ablation studies to analyze the impact of each design in our framework, and the corresponding return graphs are presented in Figure 5.

Auxilary loss Removing each auxiliary loss term resulted in a decrease in performance on both the cheetah and walker tasks. The effect was particularly significant on the walker task, with noticeable reward gaps. These findings indicate that each auxiliary loss term contributes to performance improvement.

Pretrained RL network Surprisingly, we observed that employing a pretrained RL network from the source environment did not contribute to learning in the target environment. In certain environments like Walker, the use of pre-trained RL networks even led to a decline in adaptation performance. We speculate that although the distribution of encoded features in the target observations resembles that of the source observations, extensive training of the RL agent is still required due to substantial domain shifts, and thus, utilizing a re-initialized RL agent could yield better performance.

Adversarial Objective Considering that Method 1 relies on the strong assumption of paired samples in the target and source environments, but offers only marginal benefits compared to Method 2, we argue that utilizing paired data for Method 1 does not provide significant advantages over Method 2 in our settings.

Table 1. Comparison on return with domain adaptation baseline

Color Shift	w/o adapt	PAD-1	PAD-10	Ours-1	Ours-10
cartpole	**626**	638	316	533	528
walker	280	464	373	380	**503**
cheetah	138	221	**263**	161	226
Viewpoint	w/o adapt	PAD-1	PAD-10	Ours-1	Ours-10
cartpole	**494**	321	96	345	336
walker	44	29	32	51	**54**
cheetah	3	2	0	17	**31**

7. Conclusion

In this paper, we propose a simple yet effective pretraining-agnostic adaptation approach for RL agents, with only the learned policy and a few episodes of source and target environments. This method overcomes the limitations of existing approaches, particularly the over-reliance on the pre-training phase and the assumption that the source and target environments will be similar. It effectively minimizes the discrepancy between source and target environments while preserving vital information essential for reinforcement learning, by integrating adversarial domain adaptation and auxiliary instance discrimination techniques.

We evaluated on the DeepMind Control Suite (DMC) and demonstrated the advantages in terms of adaptability and efficiency of our approach. It effectively facilitates the process of warm-starting especially when the domain shift is large, which plays an instrumental role in expediting the policy learning process within the target environment. In comparison with existing methods, ours require only a few episodes of target environments to adapt the learned policy, which drastically reduces the dependency on vast data collection. Notably, our approach is practically agnostic to the pretraining phase, thus making it especially fitting the real-world scenarios where environmental conditions are likely to significantly deviate from training settings.

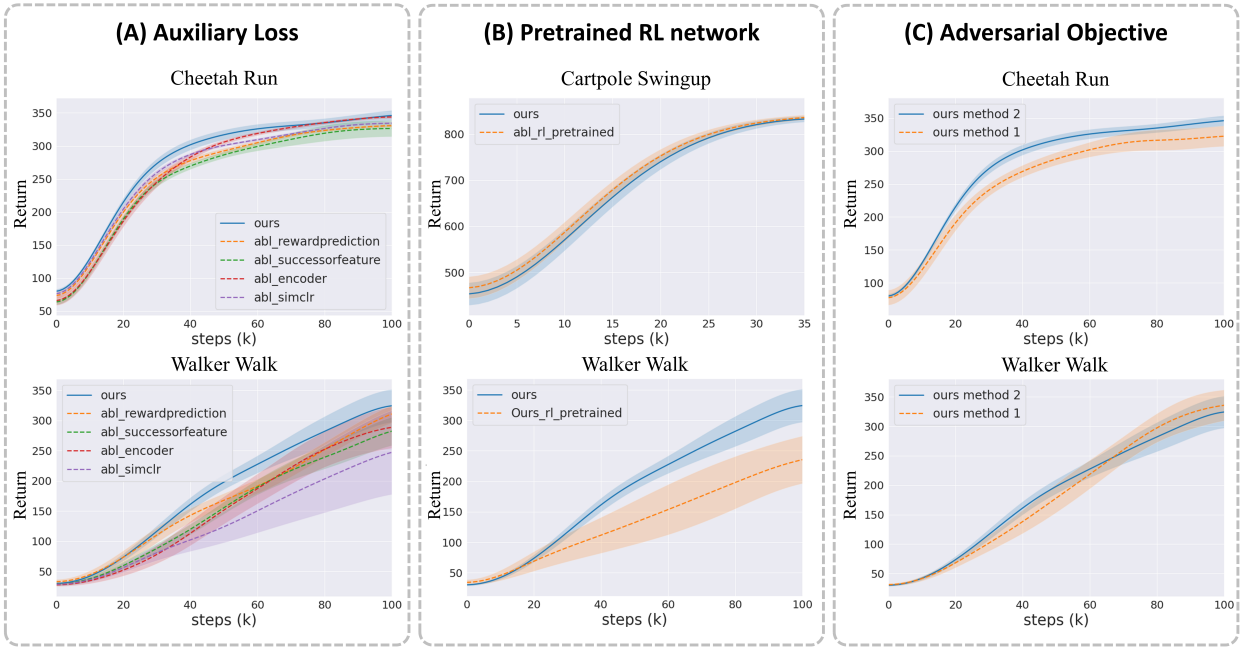


Figure 5. The steps-return graphs of each ablation study. (A) The ablation study of each auxiliary loss term. (B) The ablation study using pretrained RL network. (C) The experiment of changing data on adversarial adaptation training. (Methods mentioned on section 4.2)

Limitations and Future Works. While the proposed method shows potential in addressing the challenges of policy adaptation to the target environment with limited information about the pretraining phase, it still has some limitations. Firstly, our method still requires a considerable number of interactions with the target environment, as the policy network needs to be fine-tuned in the target environment. Additionally, in cases with small domain shifts, such as color shifts(Table 1), the advantages of our method over the baseline (PAD) are not significant.

Recognizing these limitations, we acknowledge that further improvements can be made to our method. One interesting avenue for future exploration is the integration of the adaptation phase and fine-tuning phase. Currently, these phases are treated separately, but combining them could allow us to leverage the auxiliary losses during the fine-tuning phase in the target environment, probably leading to improved data efficiency.

Another interesting future direction involves finding alternatives to the fine-tuning phase in the target environment. While our method successfully adapts the pre-trained encoder to the target environment, utilizing the pre-trained actor/critic network does not yield better performance compared to re-training the actor/critic from scratch in our settings. In this regard, we believe that there might be a way to adapt the pre-trained actor/critic network to the target environment as well as the encoder, without requiring a significant number of interactions in the target environment. We briefly share a snippet of our idea on how this could be achieved:

- Learn the quantized representations on the source environment using VQ-VAE, and train policy networks that utilize these quantized representations as input.
- Adapt the encoder to the target environment.
- Instead of retraining the policy from scratch, solely find the mapping from learned embeddings to new embeddings.

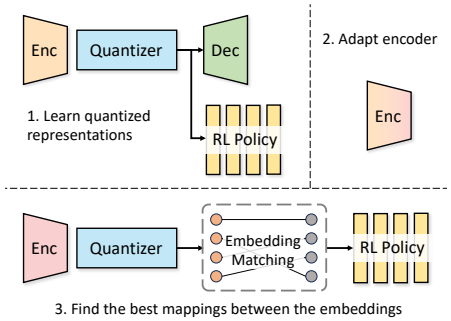


Figure 6. A snippet of the idea for utilizing the pre-trained actor and critic network for better data efficiency.

References

[1] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 3, 5

[2] Linxi Fan, Guanzhi Wang, De-An Huang, Zhiding Yu, Li Fei-Fei, Yuke Zhu, and Anima Anandkumar. Secant: Self-expert cloning for zero-shot generalization of visual policies. *arXiv preprint arXiv:2106.09678*, 2021. 1

[3] Chelsea Finn, Xin Yu Tan, Yan Duan, Trevor Darrell, Sergey Levine, and Pieter Abbeel. Deep spatial autoencoders for visuomotor learning. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 512–519. IEEE, 2016. 1

[4] Lisheng Fu, Thien Huu Nguyen, Bonan Min, and Ralph Grishman. Domain adaptation for relation extraction with domain adversarial neural network. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 425–429, 2017. 2

[5] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016. 2

[6] Zhiqiang Gao, Shufei Zhang, Kaizhu Huang, Qiufeng Wang, and Chaoliang Zhong. Gradient distribution alignment certificates better adversarial domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8937–8946, 2021. 2

[7] Jean-Baptiste Gaya, Laure Soulier, and Ludovic Denoyer. Learning a subspace of policies for online adaptation in reinforcement learning. *arXiv preprint arXiv:2110.05169*, 2021. 1, 2

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 2

[9] Dayan Guan, Jiaxing Huang, Aoran Xiao, Shijian Lu, and Yanpeng Cao. Uncertainty-aware unsupervised domain adaptation in object detection. *IEEE Transactions on Multimedia*, 24:2502–2514, 2021. 2

[10] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006. 2

[11] Nicklas Hansen, Rishabh Jangir, Yu Sun, Guillem Alenyà, Pieter Abbeel, Alexei A Efros, Lerrel Pinto, and Xiaolong Wang. Self-supervised policy adaptation during deployment. *arXiv preprint arXiv:2007.04309*, 2020. 2, 6

[12] Nicklas Hansen and Xiaolong Wang. Generalization in reinforcement learning by soft data augmentation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13611–13617. IEEE, 2021. 1, 2

[13] Han-Kai Hsu, Chun-Han Yao, Yi-Hsuan Tsai, Wei-Chih Hung, Hung-Yu Tseng, Maneesh Singh, and Ming-Hsuan Yang. Progressive domain adaptation for object detection. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 749–757, 2020. 2

[14] Riashat Islam, Hongyu Zang, Anirudh Goyal, Alex Lamb, Kenji Kawaguchi, Xin Li, Romain Laroche, Yoshua Bengio, and Remi Tachet Des Combes. Discrete factorial representations as an abstraction for goal conditioned reinforcement learning. *arXiv preprint arXiv:2211.00247*, 2022. 2

[15] Kyungsoo Kim, Jeongsoo Ha, and Yusung Kim. Self-predictive dynamics for generalization of vision-based reinforcement learning. In *IJCAI International Joint Conference on Artificial Intelligence*, pages 3150–3156. International Joint Conferences on Artificial Intelligence, 2022. 1, 2

[16] Tejas D Kulkarni, Ankush Gupta, Catalin Ionescu, Sebastian Borgeaud, Malcolm Reynolds, Andrew Zisserman, and Volodymyr Mnih. Unsupervised learning of object keypoints for perception and control. *Advances in neural information processing systems*, 32, 2019. 1, 2

[17] Tejas D Kulkarni, Ardavan Saeedi, Simanta Gautam, and Samuel J Gershman. Deep successor reinforcement learning. *arXiv preprint arXiv:1606.02396*, 2016. 4

[18] Misha Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. Reinforcement learning with augmented data. *Advances in neural information processing systems*, 33:19884–19895, 2020. 1, 2

[19] Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. In *International Conference on Machine Learning*, pages 5639–5650. PMLR, 2020. 1, 2, 6

[20] Hao Liu and Pieter Abbeel. Behavior from the void: Unsupervised active pre-training. *Advances in Neural Information Processing Systems*, 34:18459–18473, 2021. 2

[21] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. *Advances in neural information processing systems*, 31, 2018. 2

[22] Zhongyi Pei, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Multi-adversarial domain adaptation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018. 2

[23] Max Schwarzer, Ankesh Anand, Rishab Goel, R Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. *arXiv preprint arXiv:2007.05929*, 2020. 1

[24] Dongseok Shim, Seungjae Lee, and H Jin Kim. Snerl: Semantic-aware neural radiance fields for reinforcement learning. *arXiv preprint arXiv:2301.11520*, 2023. 1

[25] Hui Tang and Kui Jia. Discriminative adversarial domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5940–5947, 2020. 2

[26] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018. 5

[27] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization

for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE, 2017. 1

[28] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017. 2

[29] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2517–2526, 2019. 2

[30] Huazheng Wang, Zhe Gan, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, and Hongning Wang. Adversarial domain adaptation for machine reading comprehension. *arXiv preprint arXiv:1908.09209*, 2019. 2

[31] Qi Wang, Junyu Gao, and Xuelong Li. Weakly supervised adversarial domain adaptation for semantic segmentation in urban scenes. *IEEE Transactions on Image Processing*, 28(9):4376–4386, 2019. 2

[32] Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning. *arXiv preprint arXiv:2107.09645*, 2021. 1, 2

[33] Bang You, Oleg Arenz, Youping Chen, and Jan Peters. Integrating contrastive learning with dynamic models for reinforcement learning from images. *Neurocomputing*, 476:102–114, 2022. 2

[34] Amy Zhang, Rowan McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. *arXiv preprint arXiv:2006.10742*, 2020. 1, 2