Domain Adaptation based Transfer Learning using Adversarial Network

Farzaneh Shoeleh
University of Tehran
Tehran, Iran
f.shoeleh@ut.ac.ir

Mohammad Mehdi Yadollahi
University of Tehran
Tehran, Iran
mm.yadollahi@ut.ac.ir

Masoud Asadpour
University of Tehran
Tehran, Iran
asadpour@ut.ac.ir

ABSTRACT

There is an implicit assumption in machine learning techniques that each new task has no relation to the tasks previously learned. Therefore, tasks are often addressed independently. However, in some domains, particularly Reinforcement Learning (RL), this assumption is often incorrect because tasks in the same or similar domain tend to be related, means even though tasks are quite different in their specifics, they may have general similarities, such as shared skills, making them related. In this paper, a novel domain adaptation based method using adversarial networks is proposed to do transfer learning in RL problems. Our proposed method incorporates skills previously learned from source task to speed up learning on a new target task by providing generalization not only within a task but also across different but related tasks. The experimental results indicate the effectiveness of our method in dealing with reinforcement learning problems.

KEYWORDS

Reinforcement Learning; Transfer Learning; Domain Adaptation; Adversarial Learning; Deep Learning

1 INTRODUCTION

The reinforcement learning (RL) paradigm is a popular way for an autonomous agent to learn from experience with minimal feedback. The required learning time and the curse of dimensionality restrict the applicability of RL on real-world problems. According to the literature [24, 32, 35], it is believed that state abstraction methods and hierarchical architectures can improve the learning curve and lessen the hampering effect of the curse of dimensionality. While significant progress has been made to improve learning in a single task, the idea of transfer learning has only recently been applied to reinforcement learning tasks [1, 8, 33, 34]. As a result, the researches expressed that *"transfer learning has recently gained popularity due to the development of algorithms that can successfully generalize information across multiple tasks"* [37]. One of the critical aspects of an intelligent agent is the ability to learn one or multiple environments and transfer previous knowledge to new environments, with similar situations to the previous one. Towards this goal, an autonomous agent must be able to learn first how to behave in a task effectively and then generalize its obtained knowledge as much as needed to transfer and apply in a new domain.

The insight behind Transfer Learning (TL) is that generalization may occur not only within tasks but also across different tasks which are from similar domains. Generally, the idea of the transfer of knowledge in order to improve the performance of machine learning algorithms stems from cognitive science research. A vast number of psychological studies show human beings can learn amazingly fast because they effectively bias the learning process towards a very limited set of solutions obtained by transferring the knowledge retained from solving similar tasks. Similarly, the idea of TL is that it is possible to improve the performance of any machine learning algorithms, like the learning algorithm of autonomous agents, by biasing their hypothesis space towards a set of good hypotheses according to the knowledge retained from solving other tasks.

The main aim of this paper is to facilitate transfer learning for an autonomous agent who can face not only homogeneous problems but also the heterogeneous ones. In general, TL problems can be divided into heterogeneous and homogeneous by considering whether the feature spaces between the source and target domains are the same or not. So, our challenging question must be answered is "How the source and target tasks are related?". To answer this question, we propose a novel domain adaptation based transfer learning approach using an adversarial network. Our method is called $DATL_{AN}$. It is able to learn a transformation which helps an autonomous RL based agent to adapt the domains of the source and target task and consequently transfer its skills acquired from source task into the target task.

Domain adaptation is a well-known technique associated with transfer learning which seeks the same goal in machine learning problems, especially pattern recognition. The goal of a domain adaptation approach is to learn and find transformations which can map both source and target domains into a common feature space. On the other hand, Generative Adversarial Networks(*GAN*) [22] are a promising approach to train a deep network and generate samples across diverse domains. In many application, these networks can also improve recognition despite the presence of domain changes or dataset bias [21, 30, 39]. a *GAN* consists of two networks named *generator* and a *discriminator*. They are against each other, means generator is trained to produce samples with the objective to confuse the discriminator. Recently, one type of domain adaptation approaches which have recently become increasingly popular is known as adversarial adaptation methods. These type of methods seek to minimize an approximate domain discrepancy distance through an adversarial objective with respect to a domain discriminator. They are so closely related to the principles of GAN based approaches. In domain adaptation, the principle of *GAN* has been employed to ensure that the network cannot distinguish between the distributions of samples coming from the source and target domain [19, 23, 30]

Our proposed method, $DATL_{AN}$, leverages adversarial domain adaptation principles to discover related skills between the source and target tasks, transfer them, and boost the learning performance of agent in the target task. $DATL_{AN}$ method has three main steps: first, learning source task and extracting abstract skills by modeling both agent experiences and environment dynamics in *connectivity graph*. Second, finding the state-action inter-task mappings implicitly by leveraging adversarial domain adaptation technique to learn a common feature space where the source and target domains can be aligned, and then efficiently transferring the previously learned skills into the target task to learn this new environment. The results from experiments demonstrate that the proposed method is able to find the relation between tasks and consequently transfer effectively skills which were learned in source task. The proposed method improves the performance of agent in target task using the transferred knowledge.

The rest of this paper is organized as follows. Section 2 presents an overview of the related work. In Section 3, the proposed skill based transfer learning via domain adaptation approach is described. Experiments and results are reported in Section 4, and Section 5 contains the conclusion and direction for future works.

2 RELATED WORK

As the type of transferred knowledge can be primarily characterized by its level of specificity [36], the possible knowledge transfer approaches can be classified into two main categories accordingly: low-level knowledge transfer and high-level knowledge transfer. In RL domain, low-level information can be considered as (s, a, r, s') tuples, an action-value function Q, a policy π, or a full model of the task, whereas high-level information can be considered as a subset of all actions used in some situations or partial policies, skills or options, rules, important features for learning, proto-value functions [31], shaping rewards, or subtask definition.

As claimed in [36] and [38], it makes intuitive sense that high-level knowledge may transfer better across tasks since they can be obtained more independently compared to low-level information. Low-level knowledge can all be directly leveraged to initialize a learner in the target task. On the other hand, high-level information may not directly be applicable to transfer learning algorithms to fully define an initial policy for the agent in the target task. However, such information would guide the agent during its learning in the new target environment. Moreover, transfer learning algorithms using high-level knowledge assist the agent to learn a new task more effectively than lower-level information. Please note that the proposed method tries to transfer high-level knowledge, namely a set of skills which were acquired from source task into the target task.

By transferring skills, our method tends to detect and transfer similar region among source and target task. The idea of transferring similar regions among tasks was firstly proposed in [28, 29] where the similar regions are determined using the similarity between samples in source and target, indeed using low-level knowledge. In contrast, our proposed method tries to transfer similar regions identified with high-level knowledge. Asadi et. al [6, 7] present an agent that learns options and transfers them between different tasks. The agent tries to find subgoals in the source task through identifying states that are "locally from a significantly stronger *attractor* for state space trajectories" [6]. Considering such subgoals helps the agent define options. In [6, 7] source and target tasks differ only in the reward function, while the proposed method would be applied to the source and target tasks that may differ in possible state transitions and state-action space.

As suggested by Lazaric et al. in [27], transfer learning approaches in RL problems can be categorized based on the number of involving source tasks and the difference between source and target domains: 1) Transfer from one source task to one target task with fixed domain, 2) Transfer across several tasks (including a set of source tasks) with fixed domain, 3) Transfer across several tasks with different domains. As a principle, it is stated that "the domain of a task is determined by its state-action space, while the specific structure and goal of the task are defined by the dynamics and rewards" [27]. According to this definition, the first and second categories consist of problems whose state-action spaces are the same. In contrast, source and target tasks in the third category have different domains, meaning different state-action variables. While this category is more common in real-world problems, it involves one additional challenging issue to define the mapping between the source and target state-action variables. In literature, such mappings are referred to *Inter-task Mappings*. According to the presented categories, our proposed method lies in the third case. It leverages a domain adaptation neural network to find the inter-task mapping between source and target tasks driven from different domains.

In literature, most of the researchers assume that the inter-task mappings are predefined by experts according to their experience or intuition. There are some researchers who design mechanisms to select good mappings from several predefined mappings. In [16], two algorithms were proposed to select the best mapping from multiple mappings for both model-based and model-free RL algorithms to transfer from multiple inter-task mappings. Similarly, the authors proposed a method in [17] to autonomously select mappings from the set of all possible inter-task mappings. In [14], authors proposed a many-to-one mapping for the transfer learning, named linear multi-variable mapping. It uses the linear combination of the information from different related state variables and action to initialize the target task learning. However, their approach still requires an expert to provide the parameters of the linear combination, and the optimal parameter values are not easy to be given.

There are also some methods for learning inter-task mapping automatically. In [12], authors used a Neural Network to map actions from the source domain to the target domain by observing the results of the two actions in the source domain and target domain so as to learn the weights of the network. However, the mapping between the states is predefined by an expert. On the other hand, researches in [13] proposed an artificial neural network based method to learn both action and state inter-task mapping between source task and target

task. The obtained inter-task mapping is used to transfer the knowledge learned in the source task into the target task for initialization.

The closest approaches related to our work are the approaches which are trying to find an inter-task mapping for a pair of tasks or finding the MDP similarities in order to have effective TL approach [4, 18]. Authors in [5] proposed a transfer learning framework which learns the inter-task mapping by representing the source and target data, in a form of (s, a, s'), in a high dimensional space discovered using sparse coding, projection, and Gaussian process.

In [2, 3], authors proposed a transfer learning method in the context of policy gradient RL. The multi-task learning method proposed in [2] transfers the shared knowledge between sequential decision making tasks by incorporating latent basis into policy gradient learning. In [3], the proposed system transfers the source samples into the target by discovering a high-level feature space through learning inter-task mapping via an unsupervised manifold alignment. Similarly, Bocsi et. al in [11] proposed an alignment-based transfer learning method for robot models. The primary differences with our work are that they focused on transferring models or policies/samples between different tasks, rather than high-level knowledge, i.e. skills, therefore the authors needed a similarity metric for MDPs. Deposit the invaluable research done for transfer learning in the RL realm, to the best of our knowledge, there are still open directions in this area to transfer autonomously learning without requiring any background knowledge.

3 DOMAIN ADAPTATION BASED TRANSFER LEARNING USING ADVERSARIAL NETWORK

In our proposed method, the autonomous agent uses a domain adaptation technique to discover a mapping that can align the state-action spaces of the new environment to the one which was learned previously. This mapping is called inter-task mapping between state-action spaces of the source and target environments. Here, we utilize the concept of domain adaptation technique in order to facilitate transfer learning across domains with different state-action spaces. Our proposed agent must perform three learning phases: 1) Learning source task, 2) Learning the similarities between the source and target tasks, and 3) Learning target task.

As the first step, the agent should learn the source task properly and its experiences must be captured as high-level knowledge such as skills in order to be appropriately transferred. Many approaches have been proposed to extract skills in RL realm. Among them, we suggest using Graph-based Skill Learning method (*GSL*) which is proposed in [33]. The promising results demonstrated that *GSL* approach not only can find appropriate skills but also results in notable improvements in the learning performance of the agent. The agent's experiences are captured as a *connectivity graph* which gives information about both the agent's dynamic behavior and the environment's dynamics. The communities found from such graph divides the state-space into regions called *accessible regions* and the agent learns the problem by extracting a skill for each *accessible regions*. *GSL* accomplishes hierarchical learning by decomposing a problem into the set of skills and then benefits *Option framework* [35] to learn those skills.

After learning the source task, the next step is determining "How the two tasks with different state variables and actions are related?". A possible way to answer this question is finding a common latent space where the source and target tasks state-action spaces can be aligned. Domain adaptation is a well-known technique which seeks the same goal in pattern recognition. Considering a classification task where X is the input space and $Y = \{0, 1, ..., L-1\}$ is the set of L possible labels. Moreover, there are two different distributions over $X \times Y$, called the *source domain* D_S and the *target domain* D_T. An *unsupervised domain adaptation* learning algorithm is then provided with a *labeled source sample* S drawn *i.i.d.* from D_S, and an unlabeled target sample T drawn *i.i.d.* from D_T^X, which is the marginal distribution of D_T over X.

$S = \{(x_i; y_i)\}_{i=1}^{n} \sim (D_S)^n$; $T = \{(x_i)\}_{i=n+1}^{N} \sim (D_T^X)^{n'}$

with $N = n + n'$ being the total number of samples. The goal is to build a classifier $\nu : X \rightarrow Y$ with a low *target risk* while having no information about the labels of D_T.

In following, we detail how to develop and feed a GAN to find the inter-task mapping, align the samples of the source and target tasks to each other, and consequently transfer the skills which were learned before into the new environment. To do so, we offer to adapt the state-of-the-art approach called domain-adversarial neural network (*DANN*) which incorporating a domain adaptation component to neural networks [21]. We feed *DANN* with two following sets of samples, S and T which are collected from the source and target domain respectively:

$$
\begin{aligned}
&S = \{ (x_i^S, y_i^S) \}_{i=1}^{n} \qquad where \\
&x_i^S = < s^S, a_1^S, s'_{a_1^S}, r_{a_1^S}, Q_{a_1^S}^{s^S}, a_2^S, s'_{a_2^S}, r_{a_2^S}, Q_{a_2^S}^{s^S}, \dots \\
&\qquad \dots, a_k^S, s'_{a_k^S}, r_{a_k^S}, Q_{a_k^S}^{s^S} > \\
&y_i^S = ID \text{ of the skill that sample(state) } s^S \text{ located in.}
\end{aligned}
\tag{1}
$$

$$
\begin{aligned}
&T = \{ (x_i^T) \}_{i=n+1}^{N} \qquad where \\
&x_i^T = < s^T, a_1^T, s'_{a_1^T}, r_{a_1^T}, Q_{a_1^T}^{s^T}, a_2^T, s'_{a_2^T}, r_{a_2^T}, Q_{a_2^T}^{s^T}, \dots \\
&\qquad \dots, a_k^T, s'_{a_k^T}, r_{a_k^T}, Q_{a_k^T}^{s^T} >
\end{aligned}
\tag{2}
$$

where x_i^S is the ith sample collected from source domain. This sample represents the current state s^S, the state transition (doing action a_j^S in the source environment makes the state of agent change into $s'_{a_j^S}$, where $j \in [1, 2, .., k]$ and k is the possible number of actions can be choosen in the source environment), given reward from the source environment by choosing jth action $r_{a_j^S}$, and the q-value $Q_{a_j^S}^{s^S}$ which presents the value of selecting a_j^S in current state, s^S. y_i^S indicates the ID of the skill in which the current state s^S located. Similarly, x_i^T represents the ith instance sampled from the target domain. Figure 1 illustrates an example of sample representation in Maze environment. Please note that in our setting, for each sample in source domain y_i^S is discovered through learning the source task, but y_i^T is not defined yet in the target domain.

Therefore, we utilize *DANN* as an unsupervised domain adaptation to determine the y_i^T in the target domain by adapting the source and target domains. Its architecture is shown in Figure 2. It includes three deep neural networks: a deep feature extractor, a deep skill predictor, and a domain classifier.

DANN is motivated and supported by the theory on domain adaptation presented in [9, 10], "*a good representation for cross-domain transfer is one for which an algorithm cannot learn to identify the domain of origin of the input observation.* So, the unsupervised domain adaptation architecture (Figure 2) focuses on extracting and learning a feature set which combines both discriminativeness (discriminative for learning the skill of a given state in source environment) and domain-invariance (invariant to the change of domains). It jointly optimizes the underlying feature set as well as two discriminative classifiers operating on this feature set: 1) *Skill predictor* predicting *ID* of the skill that a given state located in, and 2) *Domain classifier* discriminating between the source and the target domains. As proposed in [21], *DANN* uses standard layers and loss functions. It trains using standard backpropagation algorithms based on stochastic gradient descent or its modifications (e.g., *SGD* with momentum). The domain classifier is connected to the obtained feature set via a gradient reversal layer resulting in the domain-invariant features, means the feature set distribution over the two domains are made similar and as indistinguishable as possible to classify the domain. The optimization of training *DANN* is as follow:

$$E(\theta_f, \theta_y, \theta_d) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}_y^i(\theta_f, \theta_y) - \lambda(\frac{1}{n} \sum_{i=1}^{n} \mathcal{L}_d^i(\theta_f, \theta_y) + \frac{1}{n'} \sum_{i=n+1}^{N} \mathcal{L}_d^i(\theta_f, \theta_y)) \tag{3}$$

As suggested in [21], the saddle point optimizing above equation can be found as a stationary point of the following gradient updates:

$$\theta_f \leftarrow \theta_f - \mu(\frac{\partial \mathcal{L}_y^i}{\partial \theta_f} - \lambda \frac{\partial \mathcal{L}_d^i}{\partial \theta_f}) \tag{4}$$

$$\theta_y \leftarrow \theta_y - \mu(\frac{\partial \mathcal{L}_y^i}{\partial \theta_y}) \tag{5}$$

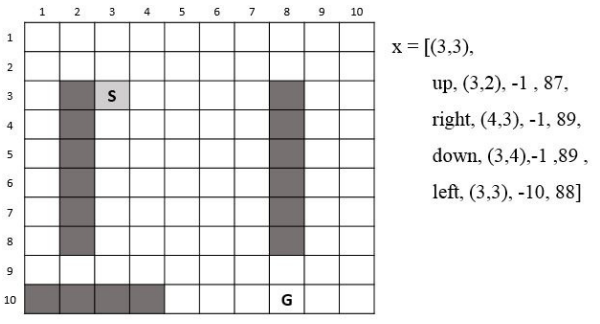


Figure 1: An example of sample representation in Maze environment.

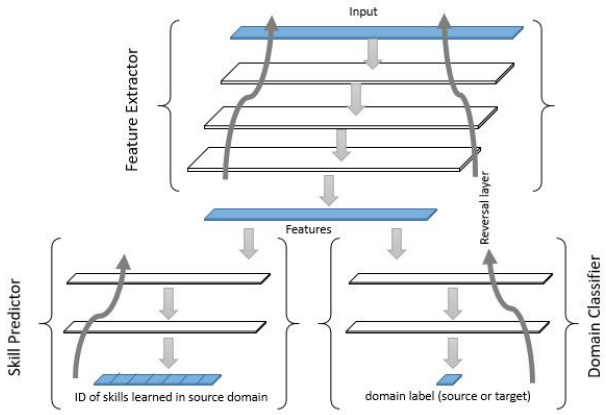


Figure 2: The unsupervised domain adaptation architecture includes a deep feature extractor, a deep skill predictor, and a domain classifier connected to the feature extractor via a gradient reversal layer. Gradient reversal ensures that the feature distributions over the two domains are made similar (as indistinguishable as possible for the domain classifier), thus resulting in the domain-invariant features.

$$\theta_d \leftarrow \theta_d - \mu\lambda(\frac{\partial \mathcal{L}_d^i}{\partial \theta_d}) \tag{6}$$

where μ is the learning rate. Stochastic estimates of these gradients are used by sampling examples from the data set.

In our proposed approach, the output of first learning phase is a set of skills extracted and learned from the source task and the output of second learning phase is a learned deep neural network which identifies the state-action space mappings. Since in RL literature skills can be formulated using Option framework [35] as a well-defined temporal abstraction frameworks extending RL algorithms from primitive actions to time extended activities, our agent uses this framework not only to learn skills in source task and construct its own high level skill hierarchy, but also to transfer them into target task. So, for each skill, an option O is created and its properties, namely the termination condition O_T, reward function O_R, initiation set O_I, and its internal function approximator should be defined or learned. Following skill *id* assignment to each sample of the target using DANN, the termination condition O_T, initiation set O_I, and reward function O_R of each option can be defined as suggested in [33]. But, the function approximator of a skill cannot be transferred directly, because its parameters were learned in the source domain, where both state and action spaces are different from the target. To successfully transfer the function approximators, the agent learns skills' function approximators offline by utilizing the output of *GANN* and the *q-value* of samples in source task. To negative transferring, we suggest a *KNN* based mechanism to eliminate samples whose k nearest neighbors do not have the same skill *id*. These samples are considered as negative samples, indeed they may be noisy or border samples and in both cases, it is better to eliminate them.

In spite of the previous researches that have initiated new option policies using the past experiences, it is indicated in [25] that these extra updates may be experimentally confounding. Therefore, we would not directly add the transferred skills to the agent's action repertoire. These skills are firstly considered as gestating skills which are allowed to have a gestating period (e.g., 10 episodes), where they cannot be selected for execution but their policies are updated using off-policy learning. Each gestating skill finishing its gestating period would be added to the agent's action set as a learned skill and assign appropriate initial values as its value. The initial value of a new transferred skill is considered as the maximum of Q values of its border states estimated during the gestating period.

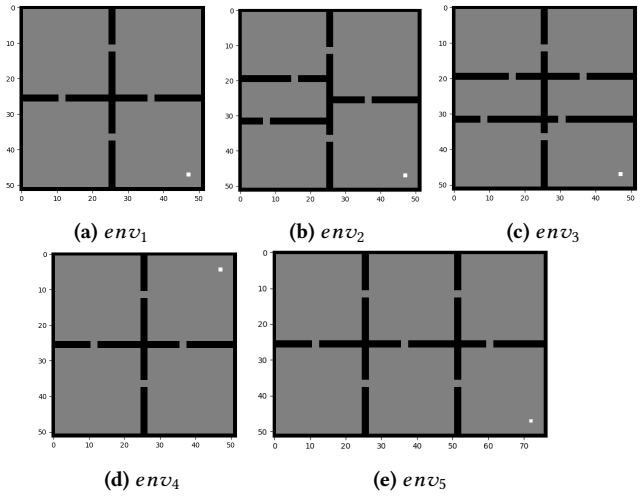
In addition, our proposed method tends to improve the performance agent by transferring the previously learned skills into the new domain. So, It is necessary to find which learned skills, mapped from source to target task through domain adaptation, are admissible for transfer. To answer this question, we calculate a matching-based fitness for all mapped and learned skills in target task. The fitness of each skill is defined as a ratio of its region size in target task to its region size in source task. If the fitness of a skill is greater than a threshold θ, it is considered as admissible for transferring, and agent expands its action-value function to include this skill.

4 EXPERIMENTAL RESULTS

We evaluate the performance of our proposed method, namely $DATL_{AN}$, through several experiments. In the following, we first introduce the test domain and then present the experiments and evaluations on our approach for transfer learning.

We examine our proposed approach using the well-known grid world domain, named *Four Room* as illustrated in Figure 3. The *Four Room* problem space is four neighbor rooms which are connected to each other with four doors. The agent's discrete state space is shown with grids. In each state four primitive actions are available: moving up or down, turning left or right. If doing each action leads to a wall hit, there is no change in the agent's state and it stays in its previous state. Each episode starts from a start state which is chosen randomly in the start of each episode and finishes when the agent reaches a goal state, which is fixed in all episodes and is shown in green in Figure 3. The agent receives a reward -1 for performing each action, -10 for hitting the wall, and +100 for reaching the goal state. Here, To examine our proposed approach, we use a set of different *Four Room* problems which are different in the locations of obstacles and environment's state space as illustrated in Figure 3.

In order to evaluate the effectiveness of our approach, we compare three learners: 1) a standard agent applying $SARSA(\lambda)$ with linear function approximation using Fourier basis [26] as a standard RL method, 2) an agent with the ability of option learning introduced in [33], named Graph-based Skill Learning (GSL) agent without using transfer learning, and 3) an agent using the proposed method, named $DATL_{AN}$. These agents are configured as 1000 learning episodes, learner Fourier order and option Fourier order are 5 and 3, respectively, λ is 0.9 and α decreases adaptively [15]. It is worth mentioning that like



(a) env_1 **(b)** env_2 **(c)** env_3

(d) env_4 **(e)** env_5

Figure 3: *Four Room* environments with different locations of obstacles and environment's state space.

the standard agent, both GSL and $DATL_{AN}$ agents use linear function approximation with Fourier basis. Since each option covers a subspace of the whole problem space, the Fourier order of option's function approximator is smaller than the agent's function approximator. Note that the first 10 episodes of the learning phase of GSL and $DATL_{AN}$ agents are devoted to gathering experiences with random policy (ϵ-greedy with $\epsilon = 1$) in order to construct the *connectivity graph* and collect a set of samples used in domain adaptation, respectively. Note that we use the released source code for the Gradient Reversal layer as an extension to Caffe [20].[1]

To examine and appraise our approach $DATL_{AN}$, we consider four transfer learning scenarios to transfer learned skills from an environment as source task to a new environment as target task:

(1) from env_1 to env_2 : homogeneous TL (adding one room)
(2) from env_1 to env_3 : homogeneous TL (adding two room)
(3) from env_1 to env_4 : heterogeneous TL (rotating -90°)
(4) from env_1 to env_5 : heterogeneous TL (expanding state space with two additional room)

Figure 4 illustrates the performance of five agents the scenarios mentioned above and the results highlight the competitiveness of our methods in terms of the obtained accumulated rewards during the time to other learners. Besides, Table 1 shows the comparison of three agents in terms of obtained *Return*, the average number of steps to reach the goal state and the average number of interaction needed between agent and environments to learn the target task. Decreasing the steps needed to reach the goal state means decreasing interactions of the agent with the environment and increasing learning speed. As expected, the agents, which uses skills, (GSL and $DATL_{AN}$) learns optimal policies with fewer experiences than the standard agent. In addition, $DATL_{AN}$ agent needs the fewer interaction with the environment since it benefits transfer learning. For example, GSL and $DATL_{AN}$ agents can respectively learn

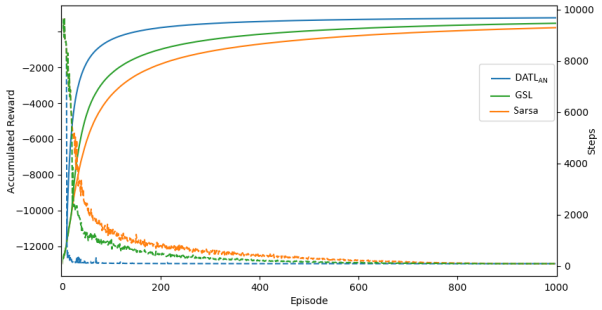
[1] $https://github.com/ddtm/caffe/tree/grl$

Table 1: Comparison of $DATL_{AN}$, *GSL* and *SARSA* agents in terms of obtained *Return*, average number of *Steps* to reach goal state and average number of *Interaction* between agent and the environment.

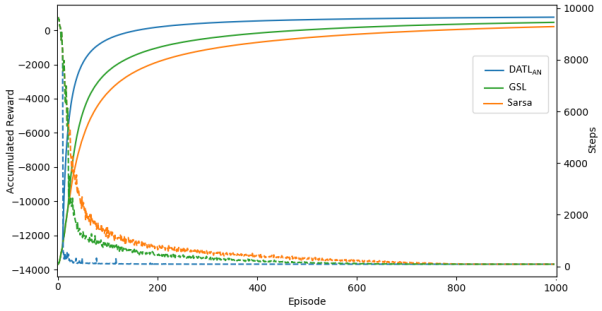
		$DATL_{AN}$	*GSL*	*SARSA*
S_1	Return	916.72	916.6	916.11
	# Steps	83.28	83.4	83.89
	# Interactions	141095	448255	653243
S_2	Return	906.8	906.41	906.3
	# Steps	93.2	93.59	93.7
	# Interactions	206715	509037	758677
S_3	Return	916.7	916.64	916.15
	# Steps	83.3	83.36	83.85
	# Interactions	210309	449078	653615
S_4	Return	890.3	862.4	836.9
	# Steps	109.7	137.6	163.1
	# Interactions	482422	761848	857893

env_2 in the first scenario with nearly 69% and 22% of the number of interactions which *SARSA* agents needs. Similarly, in the fourth scenario, agents *GSL* and $DATL_{AN}$ needs nearly 89% and 56% the number of interactions *SARSA* agents needs to learn env_5, respectively. Since in the fourth scenario the agent faces heterogeneous transfer learning problem, the ration of the agentâĂŹs interactions decreasing is less than to the one in the first scenario where there is a homogeneous transfer learning problem. The results presented in Figure 4 and Table 1 indicate that $DATL_{AN}$ agent outperforms *GSL* one, an agent without utilizing a transfer learning technique, in terms of mentioned metrics.

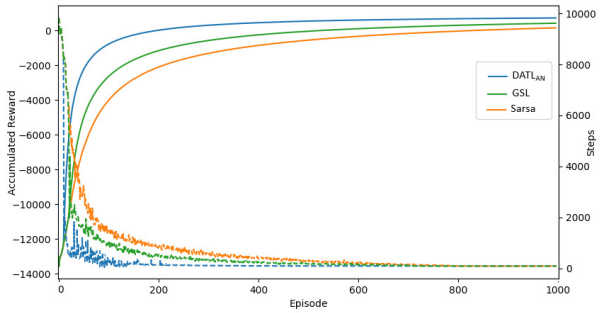
In this paper, we use four metrics introduced in [36] to measure the benefits of transfer: 1) *Jumpstart*, the improvement of an agent at the initial performance in a target task, 2) *Asymptotic Performance*, the final performance of a learned agent in a target task, 3) *Transfer Ratio*, the ratio of the total accumulated reward by the agent benefiting transfer learning to the total accumulated reward by the agent without transfer learning, 4) *Time to threshold*: the difference of learning time in terms of episodes needed by the agent to achieve a pre-specified performance level in both source and target tasks. As authors claimed [36], each metrics has drawbacks and none are sufficient to fully describe the benefits of any transfer methods. Although these metrics seems implicitly evident in Figure 4 and Table 1, they are explicitly outlined in Table 2. Note that in calculation of *Time to threshold* metric, the *asymptotic performance* of *GSL* agent is defined as threshold. According to *Jumpstart metric*, using transfer learning makes $DATL_{AN}$ agent reach *GSL* agent's performance before 500 and 640 episodes respectively in homogeneous and heterogeneous problems, while *GSL* agent achieves this after 1000 episodes. The results indicate that $DATL_{AN}$ agent outperforms *GSL* one, an agent without utilizing a transfer learning technique, in terms of these metrics. The results obtained in Table 2 demonstrates



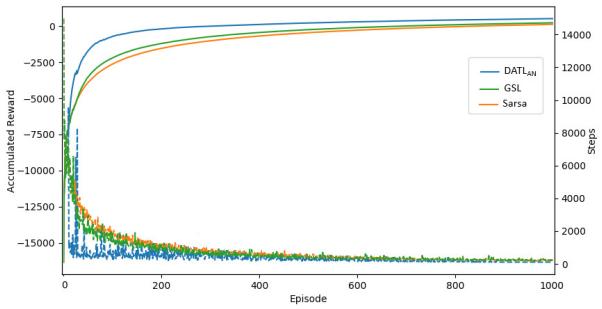
(a) S_1: Transferring from env_1 to env_2



(b) S_2: Transferring from env_1 to env_3



(c) S_3: Transferring from env_1 to env_4



(d) S_4: Transferring from env_1 to env_5

Figure 4: Comparison of learning performance of three agents: 1) agent with flat policy (*Sarsa*), 2) *GSL* agent with the best configuration as suggested in [33], and 5) agent using domain adaptation based transfer learning using an adversarial network ($DATL_{AN}$) in the predefined four scenarios.

Table 2: The average and standard deviation of metrics (introduced in [36]) for proposed method over 30 independent runs.

	S_1	S_2	S_3	S_4
Jumpstart	1286 (±208)	2976 (±231)	679 (±357)	1264 (±392)
Asymptotic performance	817.55 (±11)	697.38 (±15)	808.39 (±19)	483.37 (±21)
Transfer ratio	13.98% (±0.01)	16.95% (±0.02)	13.86% (±0.02)	15.37% (±0.01)
Time to threshold	232 (±19)	441 (±31)	528 (±56)	631 (±71)

that $DATL_{AN}$ agent benefits transfer learning to learn target task better in terms of mentioned metrics.

5 CONCLUSION

In this paper, we proposed a novel domain adaptation based transfer learning method using Adversarial Networks, named $DATL_{AN}$. A $DATL_{AN}$ based agent learns source task by extracting learned skills as high-level knowledge to be leveraged in new target task. To do so, it firstly utilizes *GSL* framework, which was proposed in [33], to discover abstract skills as high-level knowledge by constructing *connectivity graph* as a model to capture agent's experiences and the environment's dynamics. After learning the source task, $DATL_{AN}$ deploys a well-known domain-adversarial learning named $DANN$, to find a feature space where the transition samples collected from both source and target tasks are related, and consequently find a given target sample would be located to which learned skill and assign it the right skill *id*. Having these mappings helps the agent to be able to apply skills that are learned previously in source task into a new but related heterogeneous task. We examined our method in four scenarios containing a well-known four-room test domain in RL. The defined scenarios contain either homogeneous or heterogeneous problems. The promising results indicate that transferring skills as a high-level knowledge from the source task to target task by using domain adaptation technique is lucrative.

In the future, we plan to extend our transfer learning framework to be applicable in continuous domains where the standard RL methods are restricted by the required learning time and the curse of dimensionality. Besides, our current transfer learning approach only considers one source. It can be extended to utilize high-level knowledge from multiple source domains. One potential solution is to assign different weights to the learned skills obtained from different tasks based on their fitness.

REFERENCES

[1] David Abel, Yuu Jinnai, Sophie Yue Guo, George Konidaris, and Michael Littman. 2018. Policy and Value Transfer in Lifelong Reinforcement Learning. In *International Conference on Machine Learning*. 20–29.

[2] Haitham Bou Ammar, Eric Eaton, Paul Ruvolo, and Matthew Taylor. 2014. Online multi-task learning for policy gradient methods. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*. 1206–1214.

[3] Haitham Bou Ammar, Eric Eaton, Paul Ruvolo, and Matthew E Taylor. 2015. Unsupervised cross-domain transfer in policy gradient reinforcement learning via manifold alignment. In *Proc. of AAAI*.

[4] Haitham Bou Ammar, Eric Eaton, Matthew E Taylor, Decebal Constantin Mocanu, Kurt Driessens, Gerhard Weiss, and Karl Tuyls. 2014. An automated measure of MDP similarity for transfer in reinforcement learning. In *Workshops at the Twenty-Eighth AAAI Conference on Artificial Intelligence*.

[5] Haitham B Ammar, Karl Tuyls, Matthew E Taylor, Kurt Driessens, and Gerhard Weiss. 2012. Reinforcement learning transfer via sparse coding. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*. International Foundation for Autonomous Agents and Multiagent Systems, 383–390.

[6] Mehran Asadi and Manfred Huber. 2007. Effective Control Knowledge Transfer Through Learning Skill and representation hierarchies. In *20th International Joint Conference on Artificial Intelligence*. 2054–2059.

[7] Mehran Asadi and Manfred Huber. 2015. A Dynamic Hierarchical Task Transfer in Multiple Robot Explorations. In *Proceedings on the International Conference on Artificial Intelligence (ICAI)*, Vol. 8. 22–27.

[8] André Barreto, Will Dabney, Rémi Munos, Jonathan J Hunt, Tom Schaul, Hado P van Hasselt, and David Silver. 2017. Successor features for transfer in reinforcement learning. In *Advances in neural information processing systems*. 4055–4065.

[9] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Machine learning* 79, 1-2 (2010), 151–175.

[10] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. 2007. Analysis of representations for domain adaptation. In *Advances in neural information processing systems*. 137–144.

[11] Botond Bocsi, Lehel Csató, and Jochen Peters. 2013. Alignment-based transfer learning for robot models. In *Neural Networks (IJCNN), The 2013 International Joint Conference on*. IEEE, 1–7.

[12] Luiz A Celiberto Jr, Jackson P Matsuura, Ramon Lopez De Mantaras, and Reinaldo AC Bianchi. 2011. Using cases as heuristics in reinforcement learning: a transfer learning application. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, Vol. 22. 1211.

[13] Qiao Cheng, Xiangke Wang, and Lincheng Shen. 2017. An autonomous inter-task mapping learning method via artificial neural network for transfer learning. In *Robotics and Biomimetics (ROBIO), 2017 IEEE International Conference on*. IEEE, 768–773.

[14] Qiao Cheng, Xiangke Wang, and Lincheng Shen. 2017. Transfer learning via linear multi-variable mapping under reinforcement learning framework. In *Control Conference (CCC), 2017 36th Chinese*. IEEE, 8795–8799.

[15] William Dabney and Andrew G Barto. 2012. Adaptive Step-Size for Online Temporal Difference Learning. (2012).

[16] Anestis Fachantidis, Ioannis Partalas, Matthew E Taylor, and Ioannis Vlahavas. 2011. Transfer learning via multiple inter-task mappings. In *European Workshop on Reinforcement Learning*. Springer, 225–236.

[17] Anestis Fachantidis, Ioannis Partalas, Matthew E Taylor, and Ioannis Vlahavas. 2015. Transfer learning with probabilistic mapping selection. *Adaptive Behavior* 23, 1 (2015), 3–19.

[18] Norm Ferns, Prakash Panangaden, and Doina Precup. 2011. Bisimulation metrics for continuous Markov decision processes. *SIAM J. Comput.* 40, 6 (2011), 1662–1714.

[19] Yaroslav Ganin and Victor Lempitsky. 2014. Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495* (2014).

[20] Yaroslav Ganin and Victor S. Lempitsky. 2015. Unsupervised Domain Adaptation by Backpropagation. In *ICML*.

[21] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research* 17, 1 (2016), 2096–2030.

[22] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. 2672–2680.

[23] Judy Hoffman, Eric Tzeng, Trevor Darrell, and Kate Saenko. 2017. Simultaneous deep transfer across domains and tasks. In *Domain Adaptation in Computer Vision Applications*. Springer, 173–187.

[24] George Konidaris and Andrew G Barto. 2009. Skill discovery in continuous reinforcement learning domains using skill chaining. In *Advances in neural information processing systems*. 1015–1023.

[25] George Konidaris, Scott Kuindersma, Roderic Grupen, and Andrew Barto. 2011. CST : Constructing Skill Trees by Demonstration. In *Proceedings of the ICML Workshop on New Developments in Imitation Learning*.

[26] George Konidaris, Philip Thomas, Sarah Osentoski, and Philip Thomas. 2011. Value Function Approximation in Reinforcement Learning using the Fourier Basis. *Proceedings of the Twenty-Fifth Conference on Artificial*

Intelligence (2011), 380–385.

[27] Alessandro Lazaric. 2012. Transfer in Reinforcement Learning : a Framework and a Survey. *Reinforcement Learning* 12 (2012), 143–173.

[28] Alessandro Lazaric and Marcello Restelli. 2011. Transfer from Multiple MDPs. In *Advances in Neural Information Processing Systems*. 1746—-1754.

[29] Alessandro Lazaric, Marcello Restelli, and Andrea Bonarini. 2008. Transfer of samples in batch reinforcement learning. In *Proceedings of the 25th international conference on Machine learning - ICML '08*. ACM Press, New York, New York, USA, 544–551.

[30] Ming-Yu Liu and Oncel Tuzel. 2016. Coupled generative adversarial networks. In *Advances in neural information processing systems*. 469–477.

[31] Sridhar Mahadevan and Mauro Maggioni. 2007. Proto-value Functions: A Laplacian Framework for Learning Representation and Control in Markov Decision Processes. *Journal of Machine Learning Research* 8, 2169-2231 (2007), 16.

[32] Parham Moradi, Mohammad Ebrahim Shiri, Ali Ajdari Rad, Alireza Khadivi, and Martin Hasler. 2012. Automatic skill acquisition in reinforcement learning using graph centrality measures. *Intelligent Data Analysis* 16 (2012), 113–135.

[33] Farzaneh Shoeleh and Masoud Asadpour. 2017. Graph based skill acquisition and transfer Learning for continuous reinforcement learning domains. *Pattern Recognition Letters* 87 (2017), 104–116.

[34] Benjamin Spector and Serge Belongie. 2018. Sample-Efficient Reinforcement Learning through Transfer and Architectural Priors. *arXiv preprint arXiv:1801.02268* (2018).

[35] Richard S S. Sutton, Doina Precup, and Satinder Singh. 1999. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence* 112, 1-2 (aug 1999), 181–211.

[36] Matthew E Taylor and Peter Stone. 2009. Transfer Learning for Reinforcement Learning Domains : A Survey. *Journal of Machine Learning Research* 10 (2009), 1633–1685.

[37] Matthew E Taylor and Peter Stone. 2011. An introduction to intertask transfer for reinforcement learning. *Ai Magazine* 32, 1 (2011), 15.

[38] Matthew E. Taylor and Peter Stone. 2011. An Introduction to Intertask Transfer for Reinforcement Learning. *AI Magazine* 32, 1 (2011), 15.

[39] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. 2017. Adversarial discriminative domain adaptation. In *Computer Vision and Pattern Recognition (CVPR)*, Vol. 1. 4.