Reinforcement Learning

Transfer Learning

Gerard Caravaca Ibáñez
gerard.caravaca.ibanez@estudiantat.upc.edu

19/6/2023

Contents

1 Introduction

Transfer learning is a well-known machine learning technique that allows models to apply knowledge obtained from one task to a different but related one. In the field of Reinforcement Learning, this technique can considerably accelerate the learning process and increase the performance of reinforcement learning (RL) agents. In consequence, there has been a surge of interest in developing effective transfer learning approaches in that field. Despite the fact that various strategies have been offered, there is no universally acknowledged standard strategy. Taking this into account, this report examines two papers from the renowned International Conference on Machine Learning (ICML) that suggest diverse techniques to investigate the state-of-the-art in this discipline.

The first paper, titled **"REPAINT: Knowledge Transfer in Deep Reinforcement Learning"** [1], introduces a novel approach for knowledge transfer in RL. The authors outline a technique that uses deep neural network layers to reuse and adapt information from a source task to a target task. This strategy allows quicker learning and improved performance in the target task by successfully updating the learnt knowledge.

The second paper, **"Reinforcement Learning with Action-Free Pre-Training from Videos"** [2], explores the concept of action-free pre-training as a means of transfer learning in RL. The authors propose a framework where RL agents are pre-trained on videos without taking explicit actions. The learned visual representations are then utilized in the RL task, enabling the agent to acquire transferable knowledge and effectively learn the target task.

By examining these papers, I aim to compare and contrast different transfer learning techniques in RL and gain insights into the advancements made in this field. Understanding the current state of transfer learning in RL is crucial for identifying the most promising approaches and driving further progress in this area.

2 REPAINT: Knowledge Transfer in Deep Reinforcement Learning

In this initial publication, the REPAINT algorithm for deep reinforcement learning knowledge transfer is proposed. Even though there is no connection between the source and target tasks, the objective is to accelerate the learning of complicated tasks. To explain in detail what this post offers, I will first provide some background information on the research's main concepts, then go into detail about the suggested algorithm, present some of the experiments, and then come to some conclusions and discuss further research.

2.1 Background

Transfer learning algorithms in RL can be characterized by the definition of transferred knowledge, which contains the parameters of the RL algorithm, the representation of the trained policy, and the instances collected from the environment. When the teacher and student tasks share the same state-action space and they are considered similar parameter transfer is the most straightforward approach, namely, one can initialize the policy or value network in the student tasks by that from teacher tasks. Parameter transfer with different state-action variables is more complex, where the crucial aspect is to find a suitable mapping from the teacher state-action space to the student state-action space. It is the latter case that is studied in more detail in this paper. Two known approaches to do that are:

- **Representation transfer:** the algorithm first learns a particular representation of the problem or answer, which it then abstracts to fit into the student task. Representation transfer tries to extract and generalize underlying features or patterns that are useful across numerous tasks rather than acquiring a separate representation for each task. Various techniques, such as deep neural networks or other feature learning methods, can be utilized to achieve representation transfer by extracting and encoding task-relevant features that may be reused or transferred across tasks.

- **Instance transfer:** refers to the process of leveraging knowledge or experience gained from a source task or instance to improve the learning or performance of a target task or instance. It involves transferring relevant information, such as policies, value functions, or representations, from a previously learned or solved task to a new, similar task. The goal of instance transfer is to enhance the learning efficiency and generalization capability of RL agents by capitalizing on similarities between tasks and leveraging prior knowledge or experience.

To address instances of task similarity in RL, the researchers in the study suggest a **representation-instance transfer** strategy that combines both approaches. The algorithm may easily be modified to work with different other RL algorithms but is ideally suited for the actor-critic architecture.

2.2 Algorithm

In the REPAINT algorithm, the critic update follows a traditional supervised regression approach, making it akin to a conventional actor-critic algorithm with Clipped PPO. However, there are two fundamental concepts underlying the actor update: on-policy representation transfer learning and off-policy instance transfer learning. With that, a novel experience selection approach is employed to choose samples with high semantic relatedness, rather than mere similarity, to the target task.

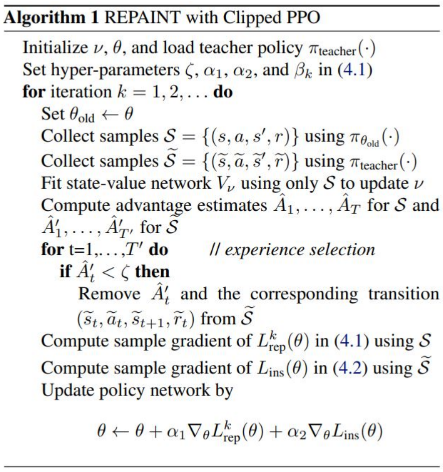
On-policy representation transfer learning serves as an initial mechanism. It involves incorporating an auxiliary loss function into the Clipped PPO objective function to encourage the student policy to closely align with the teacher policy. This auxiliary loss, weighted by the scaling factor $\beta >= 0$, is essentially the cross-entropy between the teacher and student policies. Consequently, the objective function becomes:

$$L_k(\theta) = L_{clip}(\theta) - \beta_k L_{aux}(\theta) \tag{1}$$

This kickstarting aims to replicate the behavior of teacher policy in the early training stage, so that it can improve the agent's initial performance.

On the other hand, **off-policy instance transfer learning** aims to enhance performance when the target task significantly differs from the source task, as kickstarting alone may yield limited improvements. This approach, referred to as advantage-based experience selection, involves constructing a replay buffer by collecting training samples using the teacher policy, but evaluating the rewards based on the current reward function of the target task. Furthermore, for improved sample efficiency, only samples with advantages exceeding a specified threshold are considered. The advantage can be thought of as the additional reward that could be gained by taking a specific action. Since advantages are calculated using the reward function of the target task, state-action transitions with high advantage values are considered "good" transfer samples, irrespective of the disparities between the source and target tasks.

These are the key elements that distinguish the algorithm from the standard actor-critic RL approach. However, Figure 1 provides a more detailed overview of the algorithm. It is worth noting that both the representation transfer weighting parameter and the instance transfer threshold are specific to each task. In practice, larger values of β_k incentivize the agent to closely align with the teacher policy, while larger values of ζ result in fewer samples being retained for policy updates. Consequently, this concentration on high-advantage experiences directs the current learning process towards focusing on more valuable experiences.



```
Algorithm 1 REPAINT with Clipped PPO
Initialize $\nu$, $\theta$, and load teacher policy $\pi_{\text{teacher}}(\cdot)$
Set hyper-parameters $\zeta$, $\alpha_1$, $\alpha_2$, and $\beta_k$ in (4.1)
for iteration $k = 1, 2, \ldots$ do
    Set $\theta_{\text{old}} \leftarrow \theta$
    Collect samples $\mathcal{S} = \{(s, a, s', r)\}$ using $\pi_{\theta_{\text{old}}}(\cdot)$
    Collect samples $\widetilde{\mathcal{S}} = \{(\widetilde{s}, \widetilde{a}, \widetilde{s}', \widetilde{r})\}$ using $\pi_{\text{teacher}}(\cdot)$
    Fit state-value network $V_\nu$ using only $\mathcal{S}$ to update $\nu$
    Compute advantage estimates $\hat{A}_1, \ldots, \hat{A}_T$ for $\mathcal{S}$ and $\hat{A}'_1, \ldots, \hat{A}'_{T'}$ for $\widetilde{\mathcal{S}}$
    for t=1,...,$T'$ do        // experience selection
        if $\hat{A}'_t < \zeta$ then
            Remove $\hat{A}'_t$ and the corresponding transition $(\widetilde{s}_t, \widetilde{a}_t, \widetilde{s}_{t+1}, \widetilde{r}_t)$ from $\widetilde{\mathcal{S}}$
    Compute sample gradient of $L^k_{\text{rep}}(\theta)$ in (4.1) using $\mathcal{S}$
    Compute sample gradient of $L_{\text{ins}}(\theta)$ in (4.2) using $\widetilde{\mathcal{S}}$
    Update policy network by
        $\theta \leftarrow \theta + \alpha_1 \nabla_\theta L^k_{\text{rep}}(\theta) + \alpha_2 \nabla_\theta L_{\text{ins}}(\theta)$
```

Fig. 1: REPAINT algorithm's pseudocode [1].

2.3 Experiments

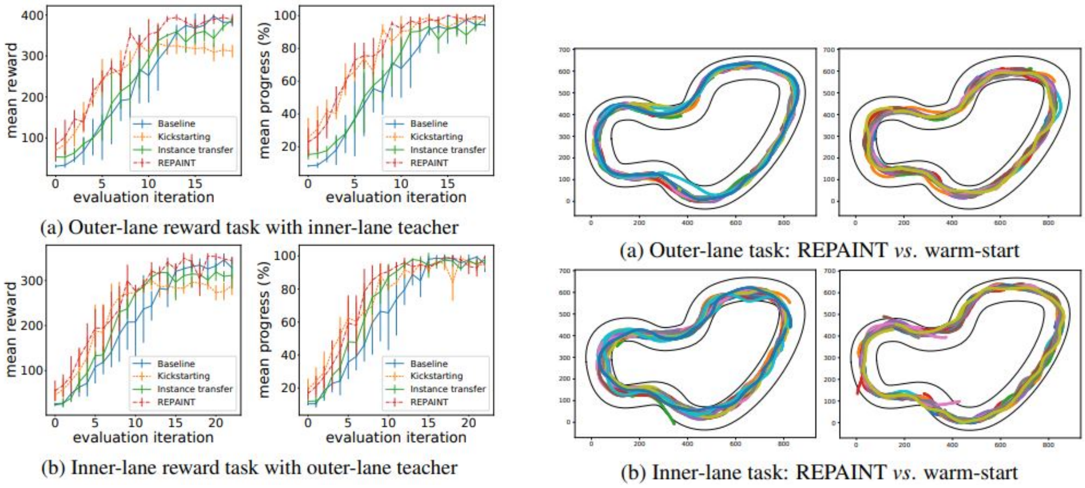
Measuring how good a transfer learning algorithm is when applied to RL is not a trivial task. For this, the paper proposes three ways to evaluate the algorithm. The evaluation metrics proposed in the paper are:

- Improvement of the agent's initial performance when learning from a pre-trained policy.
- Improvement of final performance and total accumulated reward after transfer.
- Reduction of training convergence time.

In this case, the paper focuses more on the last mentioned metric. This is because the focus of the paper is on the case where transfer learning is done with quite different tasks and therefore it is difficult to see an improvement of the results when the tasks are so different.

The paper analyses three different environments. In this report I will explain one of them in depth and give a final summary with a general comparison between all experiments. The experiment I have decided to analyse in depth is the one applied to **autonomous racing in AWS DeepRacer**.

In the experiment, two distinct reward functions are employed, each with a contrasting impact on the agent's behavior. One reward function encourages the agent when it occupies the inner lane but penalizes it when in the outer lane, while the other reward function follows the opposite pattern. When applying a specific reward function to the student task, the teacher policy utilized during training is based on the alternative reward function. With this in mind, 4 different algorithms have been tested. First the algorithm without transfer learning was used as a baseline, then the algorithm with only the representation transfer part, then the algorithm with only the instance transfer part and finally the complete REPAINT algorithm. The results obtained using these 4 techniques are shown in Figure 2.



(a) Outer-lane reward task with inner-lane teacher

(b) Inner-lane reward task with outer-lane teacher

(a) Outer-lane task: REPAINT *vs.* warm-start

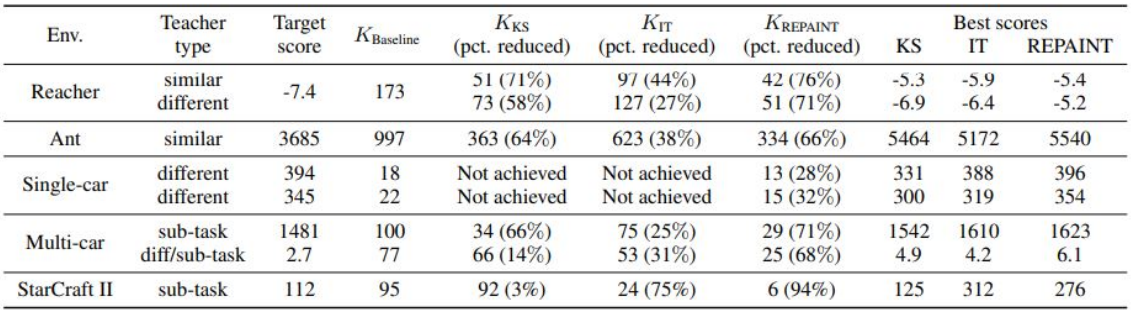
(b) Inner-lane task: REPAINT *vs.* warm-start

Fig. 2: Evaluation performance for DeepRacer single-car timetrial race (left). Trajectories of policy evaluations (right) [1].

In summary, the left plot demonstrates that all models can complete a lap successfully once they have converged. However, the REPAINT algorithm, combined with kickstarting, significantly improves initial performance. Nevertheless, when the teacher task differs significantly from the target task, kickstarting alone does not enhance the final performance through transfer learning. On the other hand, instance transfer reduces training convergence time and leads to a better final performance compared to kickstarting, with a small improvement margin in this specific example.

The REPAINT algorithm is contrasted with the warm-start parameter transfer technique in the right plot. When the target task and the instructor task are comparable, warm-start works effectively. REPAINT performs better than warm-start in the DeepRacer single-car experiment with drastically different workloads. During assessments, the agent's track trajectories are displayed. Each model receives two hours of training before being put through 20 episodes of evaluation. In spite of the different behaviors conveyed by the reward functions in both scenarios, REPAINT successfully concentrates on the current task while drawing on knowledge from the teacher policy. This demonstrates the effectiveness of advantage-based experience selection in instance transfer. In contrast, warm-start fails to eliminate unexpected behavior at convergence, as it can get stuck in local optimal. Therefore, warm-start's impact on the final performance is significant only when the two tasks are highly similar.

In reference to the other experiments, Figure 3 shows a summary of these. The teacher type in this context indicates whether the teacher task is a sub-task of or similar to the target task, determined by cosine similarity. The target score represents the best performance achievable by the baseline model. We also provide the number of training iterations required by each model to reach the target score, including training with baseline, kickstarting (KS), instance transfer (IT), and REPAINT. Although the primary focus is on the training time reduction achieved through transfer learning, it is included the best scores obtained by each knowledge transfer model in the table. The kickstarting model performs well when the tasks are similar or the target task is straightforward. However, it fails to significantly enhance initial performance through transfer learning. On the other hand, the instance transfer model is capable of transferring relevant samples but does not provide a substantial boost to initial performance. In contrast, REPAINT consistently demonstrates superior performance regardless of task similarity.



Env.	Teacher type	Target score	K_{Baseline}	K_{KS} (pct. reduced)	K_{IT} (pct. reduced)	K_{REPAINT} (pct. reduced)	Best scores KS	IT	REPAINT
Reacher	similar	-7.4	173	51 (71%)	97 (44%)	42 (76%)	-5.3	-5.9	-5.4
	different			73 (58%)	127 (27%)	51 (71%)	-6.9	-6.4	-5.2
Ant	similar	3685	997	363 (64%)	623 (38%)	334 (66%)	5464	5172	5540
Single-car	different	394	18	Not achieved	Not achieved	13 (28%)	331	388	396
	different	345	22	Not achieved	Not achieved	15 (32%)	300	319	354
Multi-car	sub-task	1481	100	34 (66%)	75 (25%)	29 (71%)	1542	1610	1623
	diff/sub-task	2.7	77	66 (14%)	53 (31%)	25 (68%)	4.9	4.2	6.1
StarCraft II	sub-task	112	95	92 (3%)	24 (75%)	6 (94%)	125	312	276

Fig. 3: Summary of paper's experiments [1].

In conclusion, the algorithm significantly reduces training time for each target task and generally improves the final return scores across most tasks.

2.4 Future research

In future work, the researchers propose to investigate how REPAINT can autonomously learn task similarity and dynamically determine the optimal values of β_k and ζ during training based on the similarity. Preliminary results suggest that larger values of β_k may negatively impact the asymptotic performance of the agent when the task similarity is low. Additionally, the researchers are interested in exploring the influence of neural network architectures on transfer performance.

3 Reinforcement Learning with Action-Free Pre-Training from Videos

Another promising avenue in transfer learning for RL is Reinforcement Learning Pretraining from Videos (RLPV), which combines the strengths of unsupervised representation learning from video data and reinforcement learning. RLPV takes advantage of the abundant video data available today. The idea is to pretrain an RL agent on a large collection of videos to learn general-purpose visual representations. In consequence, by capturing the temporal dynamics and spatial relationships in the videos, the agent can acquire a rich understanding of the visual world, similar to how humans learn from observing the environment. This is exactly the approach taken in this paper.

In general, the paper presents **APV: Action-Free Pre-training from Videos**, a novel framework that performs generative pre-training on videos for improving the sample-efficiency of vision-based RL. However, since the goal is to learn a representation that can be transferred to various tasks, the framework do not require the videos to be collected in the same domain of the target task and also it is not assumed the datasets contain action information.

In the following sections, I aim to provide a comprehensive overview of the key concepts underpinning the proposed framework. To ensure clarity and coherence, I will adopt a structured approach similar to that employed in the previous paper. First I will explain some background concepts necessary to understand the method used. Then I will explain the method itself. Then I will show some of the experiments run. Finally, I will draw some conclusions and future work.

3.1 Background

All the concepts that sustain the method of the paper are based on three main background aspects which are: unsupervised representation learning for CV and NLP, unsupervised representation learning for RL and behaviour learning with videos.

Unsupervised representation learning has been investigated in the context of RL to improve the sample-efficiency of RL algorithms. Notably, Jaderberg et al. (2017) showed how improving auxiliary unsupervised losses might improve RL agents' performance. The strategy taken in the work differs from that of previous research in that it emphasizes pre-training of representations, in contrast to studies that have focused on optimizing auxiliary unsupervised objectives during RL training. As a result, a novel pre-training framework is suggested that makes use of action-free movies derived from various domains with unique visual qualities and manifestations. In contrast to other methods, the focus here is on using the richness of data seen in unlabeled videos to build rich visual representations before RL training begins.

On the other hand, video datasets have also been utilized for **behavior learning** in various ways. For example, Aytar et al. (2018) addressed challenging exploration tasks in the Atari benchmark by devising an imitation reward mechanism derived from YouTube videos. Similarly, Peng et al. (2018) introduced a method to learn physical skills by extracting reference motions from human demonstration videos and training an RL agent to imitate these motions. Therefore, while previous studies focused on leveraging videos to extract behavior-specific information or imitate specific motions, here the emphasis lies in utilizing the visual information contained in videos to pre-train representations.

3.2 Method

To implement all the concepts explained in the previous section, the paper presents a method called **APV: Action-Free Pre-trainig from Videos built on DreamerV2** (Hafner et al., 2021). **DreamerV2** is an advanced reinforcement learning (RL) algorithm that builds upon the original Dreamer algorithm, designed to solve complex sequential decision-making problems. The key idea behind DreamerV2 is to leverage a learned world model, which is a predictive model that captures the dynamics of the environment, to generate imaginary rollouts or simulated trajectories. These rollouts help the agent to explore and plan in the environment without the need for costly real-world interactions.

With this in mind, I will focus on explaining how the APV model has been constructed. This framework is based on the following three main concepts:

- **Action-free pre-training from videos:** the idea is to learn a latent video prediction model, which is an action-free variant of a latent dynamics model proposed in DreamerV2. This should help to capture rich information from diverse videos without action information.
- **Stacked latent prediction model:** once the previous model is pre-trained, the next step is to tune it into an action-conditional prediction model, which can be used for solving various visual control tasks.
- **Video based intrinsic bonus:** technique used to improve exploration by encouraging agents to learn diverse behaviours.

The functioning of these three elements can be summarized by the picture shown in Figure 4.

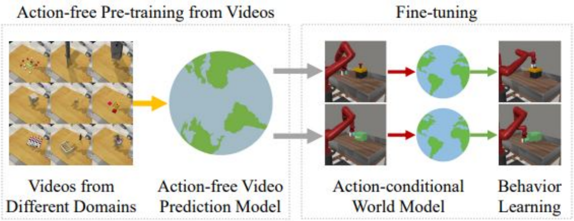

Action-free Pre-training from Videos
Fine-tuning
Videos from Different Domains
Action-free Video Prediction Model
Action-conditional World Model
Behavior Learning

Fig. 4: Summary of method's process [2].

3.2.1 Action-free pre-training from videos

The model proposed to perform this task consists of three main components. First, a representation model encodes observations to a model state with Markovian transitions. Following, a transition model predicts future model states without access to the observations. Finally, an image decoder reconstructs image observations. This model is trained in for two main tasks, first, to reconstruct image observations but then to make the prediction from the representation model and transition model as close as possible. This is done by minimizing the negative variational lower bound (ELBO).

Since the transition model does not condition on observations, it allows the model to efficiently predict future states in latent space without needing to predict future images using the image decoder at inference time.

3.2.2 Stacked latent prediction model

To effectively utilize the pre-trained representations, a new architecture is introduced. This new architecture stacks an action-conditional prediction model on top of the action-free model. At this point the architecture of the complete model can be visualized as in Figure 5.

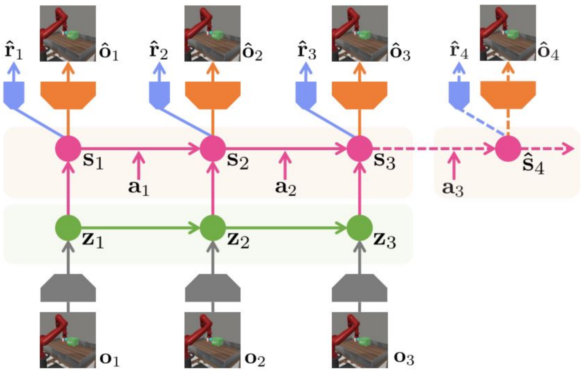


Fig. 5: Illustration of stacked latent prediction model's architecture [2].

3.2.3 Video-based Intrinsic Bonus

The researchers applied average pooling across the sequence dimension to obtain a trajectory representation, denoted as y_t, from a sequence of model states derived from the action-free prediction model, $z_t : t + \tau$. They then utilized the distance between y_t and its k-nearest neighbor in samples from a replay buffer as a metric for assessing the diversity of trajectories.

3.3 Experiments

The method explained above has been tested in two different ways based on one hand, on vision-based manipulation tasks from **Meta-world**, and on the other hand, on robotic locomotion tasks from **DeepMind Control Suite**. In this report I will show the experiments first mentioned, as they should be enough to see the performance of the method.

In order to assess the efficacy of APV in learning valuable representations across diverse domains, the researchers employed videos obtained from robotic manipulation tasks in the RL-Bench dataset as pre-training data. A total of 4950 videos were collected, consisting of 10 demonstrations for each of the 99 tasks in RLBench, with each demonstration captured from 5 different camera views. The action-free video prediction model was then trained over 600,000 gradient steps. For subsequent tasks, the model underwent fine-tuning over 250,000 environment steps, equivalent to 500 episodes.

The results of this process are compared with the DreamerV2 proposal in Figure 6. In this plot can be observed that in all the tasks examined, APV consistently demonstrated superior sample-efficiency compared to DreamerV2. Notably, the framework achieved a success rate exceeding 60% in the Lever Pull task, while DreamerV2 was unable to solve the task. These findings highlight the effectiveness of APV in leveraging action-free videos to acquire valuable representations that enhance the sample-efficiency of vision-based RL methods.

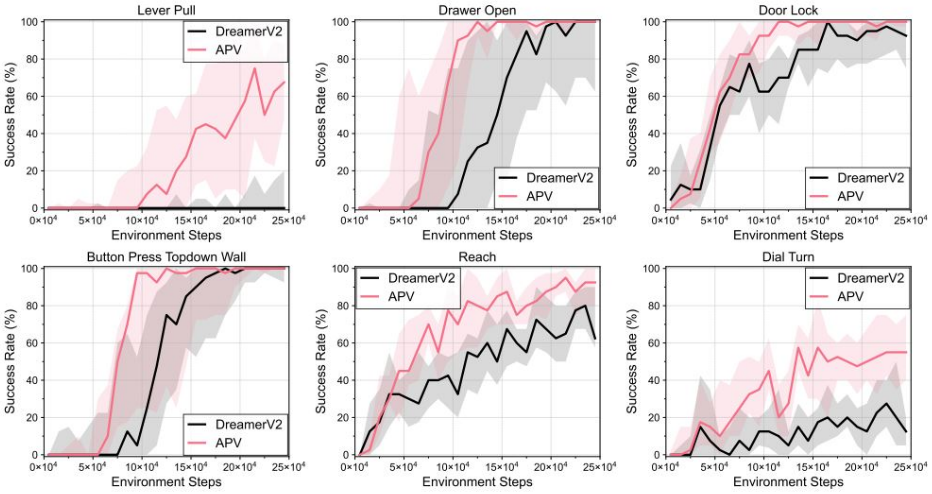


Fig. 6: Results of Meta-world experiments [2].

3.4 Future work

The study's experimental results show that APV has the ability to improve both sample-efficiency and ultimate performance of vision-based RL in various manipulation and locomotion tasks. This enhancement is accomplished by effectively transfering previously unexplored domains' pre-trained representations. However, due to the underfitting issue discovered in the research, the work has a constraint in that the pre-training is only done on simulated robotic footage.

To solve this issue, one promising future research avenue would be to evaluate the influence of prediction quality on performance by scaling up the design or employing recently established video prediction architectures. Another area worth investigating is the use of generalization approaches in RL to address the domain disparity between pre-training and fine-tuning. Furthermore, while current research focuses on representation learning via generative pre-training, it would be beneficial to study the efficacy of alternative representation learning schemes such as masked prediction.

4 Conclusions

In this paper, I have investigated the state of the art of transfer learning applied to Reinforcement Learning, based on two papers from recent years. This publications present two different approaches to address knowledge transfer and pre-training in the field of deep reinforcement learning. Although the objective is very similar, there are quite a few differences between the two methods presented.

First and foremost, the method is clearly distinct. REPAINT uses a teacher-student structure in which the teacher model generates demonstrations or gives reward signals to guide the student model's learning. This knowledge transfer assists the student model in applying previously learned skills to new activities or contexts. Reinforcement Learning with Action-Free

Pre-Training from Videos, on the other hand, uses a large dataset of unlabeled videos to pre-train the model. It learns visual representations from the videos, which improves generalization and performance in downstream reinforcement learning tasks.

Conversely, the data needs and transfer mechanisms of previous explaoined techniques vary. Reinforcement Learning with Action-Free Pre-Training from videos merely needs a large dataset of unlabeled videos, whereas REPAINT requires access to a pre-trained instructor model plus demonstrations or reward models from the teacher. REPAINT speeds learning and enhances performance by utilizing the expertise of the teacher model, whilst the video pre-training in the later technique pursues a similar goal by aiding the acquisition of helpful visual representations for boosting generalization.

In conclusion, REPAINT and Reinforcement Learning with Action-Free Pre-Training from Videos offer distinct methodologies for addressing transfer learning in deep reinforcement learning. These approaches provide valuable insights into the evolving field, but certain limitations and areas for improvement should be acknowledged.

While both methods attempt to cover a range of tasks, the experiments conducted in these papers tend to focus on specific tasks. For instance, the second method is only applicable in environments where the agent perceives the environment through artificial vision. Although both papers suggest that the agents can generalize to tasks beyond their pre-training, achieving the level of generalization seen in other fields like NLP or VC is still a significant challenge. Further research is needed to enhance the agents' generalization capabilities.

Additionally, it is worth noting that the methods described in the papers are explained using specific architectures. For example, the first method exclusively tests actor-critic algorithms. To advance the field, there is an opportunity to develop methods that can be applied to any reinforcement learning algorithm. As we have learned in the course, different algorithms are better suited for specific types of problems, and generalizing methods across various algorithms would be valuable.

References

[1] Yunzhe Tao et al. “REPAINT: Knowledge Transfer in Deep Reinforcement Learning”. In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 18–24 Jul 2021, pp. 10141–10152. URL: https://proceedings.mlr.press/v139/tao21a.html.

[2] Younggyo Seo et al. “Reinforcement Learning with Action-Free Pre-Training from Videos”. In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri et al. Vol. 162. Proceedings of Machine Learning Research. PMLR, 17–23 Jul 2022, pp. 19561–19579. URL: https://proceedings.mlr.press/v162/seo22a.html.