What Effects the Generalization in Visual Reinforcement Learning: Policy Consistency with Truncated Return Prediction

Shuo Wang[1, 2, 3], **Zhihao Wu**[1, 2], **Xiaobo Hu**[1, 2], **Jinwen Wang**[1, 2], **Youfang Lin**[1, 2], **Kai Lv**[1, 2, 3] *

[1]School of Computer and Information Technology, Beijing Jiaotong University, Beijing, China
[2]Beijing Key Laboratory of Traffic Data Analysis and Mining, Beijing, China
[3]National Key Laboratory of Air-based Information Perception and Fusion, Luoyang, China
{shuo.wang, zhwu, xiaobohu, 20291281, yflin, lvkai}@bjtu.edu.cn

Abstract

In visual Reinforcement Learning (RL), the challenge of generalization to new environments is paramount. This study pioneers a theoretical analysis of visual RL generalization, establishing an upper bound on the generalization objective, encompassing policy divergence and Bellman error components. Motivated by this analysis, we propose maintaining the cross-domain consistency for each policy in the policy space, which can reduce the divergence of the learned policy during the test. In practice, we introduce the Truncated Return Prediction (TRP) task, promoting cross-domain policy consistency by predicting truncated returns of historical trajectories. Moreover, we also propose a Transformer-based predictor for this auxiliary task. Extensive experiments on DeepMind Control Suite and Robotic Manipulation tasks demonstrate that TRP achieves state-of-the-art generalization performance. We further demonstrate that TRP outperforms previous methods in terms of sample efficiency during training.

Introduction

Visual reinforcement learning has garnered significant attention recently and witnessed notable advancements (James and Davison 2022; Hafner et al. 2023; Wang et al. 2023; Hu et al. 2023). High-dimensional image observation contains lots of task-irrelevant information, which disturbs the policy. Moreover, the policy is prone to overfit the training environment, which hinders generalization to unseen environments. In this work, we focus on generalization in visual RL. Specifically, we address scenarios where observations from test environments are unobserved during training, yet the state space and underlying dynamics remain consistent.

Domain randomization and data augmentation show great potential for learning an invariant representation in computer vision (Zhao, Queralta, and Westerlund 2020; Shorten and Khoshgoftaar 2019). Many works have recently utilized various data augmentations for visual RL and achieved significant improvement. Previous research can be divided into two main categories: 1) methods that utilize advanced visual processing and 2) methods that leverage reinforcement learning structures. For visual processing, some works try to extract foreground information for policies. The others learn an invariant representation for the same state. For leveraging RL structures, various consistency are proposed based on inherent RL characteristics, *e.g.,* temporal difference equation, transition function, and intrinsic reward function. However, the above works only empirically validate effectiveness and lack theoretical support.

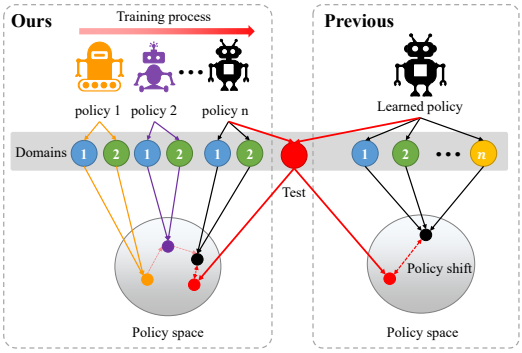

Figure 1: Sketch of our methods. Previous methods improve generalization by learning one policy that performs well across multiple domains. In contrast, we maintain cross-domain consistency across various policies. Our approach demonstrates a lower policy shift, particularly in scenarios with limited domains.

In this study, we theoretically present an upper bound for the generalization objective. The upper bound is composed of two main components: 1) *policy divergence*, representing distribution dissimilarity in state-action distribution for a policy executed across different domains, and 2) *Bellman error*, illustrating the discrepancy between the training policy and the optimal policy. To the best of our knowledge, this study represents the pioneering attempt to present a theoretical analysis for visual RL generalization.

For the first component of the upper bound, previous methods (Hansen and Wang 2021; Hansen, Su, and Wang 2021) create various augmented domains with randomized styles to learn a well-generalized policy as shown in Figure 1. However, providing many domains suitable for the task is quite challenging. Alternatively, we propose maintaining cross-domain consistency for each policy *in the policy space*. Notably, acquiring different policies is more fea-

*Corresponding author: Kai Lv (lvkai@bjtu.edu.cn).

sible than obtaining various domains. For the second component, we need to learn an accurate state-action value function. As state-action value is closely connected with policy in the context of off-policy RL, we address the Bellman error by focusing on the consistency of state-action value functions instead of policy.

Based on the above ideas, we introduce an auxiliary task named Truncated Return Prediction (TRP) to improve the generalization performance. TRP achieves cross-domain consistency of each policy by predicting truncated returns of historical trajectories. Concretely, we exploit policies at different training stages as diverse policies sampled from the policy space. Furthermore, we predict the truncated returns as the state-action values of each sampled policy before and after augmentation. Such an approach sacrifices the diversity of the policies and avoids more interactions with the environment. Notably, we only utilize the first observation in the sampled trajectory along with the following action sequence as input. In addition, we propose a variant network based on Transformer (Vaswani et al. 2017) for prediction.

We conduct experiments on five continuous control visual tasks from the DeepMind Control suit Generalization Benchmark (DMControl-GB) and two Robotic Manipulation tasks. The results demonstrate that our method promotes sample efficiency during training. Moreover, our method achieves state-of-the-art performance across most tasks.

The main contributions of our study are as follows:

- We theoretically derive an upper bound from the generalization objective of visual RL. With the support of this upper bound, we provide an explanation to understand the effectiveness of some prior works.
- We propose a practical algorithm (TRP) to maintain the cross-domain consistency of each policy in the policy space by predicting truncated returns. We also propose a variant Transformer for prediction.
- Extensive experimental results demonstrate that TRP achieves state-of-the-art generalization performance. Furthermore, we also perform an ablation study to analyze the role of the Transformer-based network.

Related Work

Data Augmentation Methods

Data augmentation has found extensive application in the field of Computer Vision (Shorten and Khoshgoftaar 2019). (Krizhevsky, Sutskever, and Hinton 2012) introduces Crop, which extracts a patch from the original image to improve classification accuracy. (Wang et al. 2020) presents MixReg which mixes a pair of transition tuples to improve policy generalization in RL. (Zhong et al. 2020) introduces Random Erasing aimed at mitigating overfitting. Inspired by MAE (He et al. 2022), MTM (Yu et al. 2022) proposes a masking approach for stacked frames in visual RL. Random convolution (Lee et al. 2020) proposes random convolutional layers that help focus on global shaping. Random overlay (Hansen and Wang 2021) utilizes a mixture between observation and other images.

Generalization for Visual RL

Visual reinforcement learning has emerged as a hot topic due to the ubiquity of vision as a fundamental perception. (Hafner et al. 2023; Schwarzer et al. 2021; Yu et al. 2022) try to improve sample efficiency for learning from pixel-based observations. (Zhang et al. 2021; Zang, Li, and Wang 2022; Fan and Li 2022) focus on learning a robust policy from images in the presence of visual distractions. In this paper, we focus on learning a policy from a single environment and ensuring generalization to unseen environments.

We can categorize previous works into two groups. For visual processing, VAI (Wang, Lian, and Yu 2021) extracts the foreground with unsupervised key-point detection and generates a foreground mask for each step. RAD (Laskin et al. 2020) conducts extensive experiments to investigate how different data augmentation affects the generalization. CURL (Laskin, Srinivas, and Abbeel 2020) is the first study to introduce contrastive learning into visual RL for learning robust representation. SODA (Hansen and Wang 2021) employs BYOL (Grill et al. 2020), an advanced contrastive learning method, to acquire consistent latent representations of augmented and unaugmented observations. SGQN (Bertoin et al. 2022) learns a mask to exclude irrelevant pixels by saliency map and consistency regularization.

For utilizing RL structures, DrQ (Yarats, Kostrikov, and Fergus 2021) updates temporal difference loss via augmented Q and augmented target Q. SECANT (Fan et al. 2021) employs strong data augmentation to distill from the teacher policy. PAD (Hansen et al. 2021) is a sim-to-real method that uses inverse dynamics to fine-tune observation representation during the test. SIM (Wu et al. 2022) fuses self-supervised loss into reward as intrinsic motivation, encouraging the agent to explore poorly embedded states. SVEA (Hansen, Su, and Wang 2021) empirically proposes two pitfalls that cause the instability of DrQ. Thus, SVEA only augments Q during training.

Preliminaries

Reinforcement Learning

Reinforcement Learning (RL) learns a decision-making policy by interacting with the environment. (Sutton and Barto 2018) formulates the interaction as a Markov Decision Process (MDP) $\mathcal{M} =< \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma >$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \mapsto \mathcal{S}$ is the dynamic transition function, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ is the reward function, and $\gamma \in [0, 1)$ is the discount factor. The goal is to learn a policy $\pi : S \mapsto A$ that maximizes expected discounted return $U = \mathbb{E}_{\boldsymbol{s}_t, \boldsymbol{a}_t \sim \tau}[\sum_{t=0}^{T-1} \gamma^t \mathcal{R}(\boldsymbol{s}_t, \boldsymbol{a}_t)]$, where $\tau = (\boldsymbol{s}_0, \boldsymbol{a}_0, \boldsymbol{s}_1, \boldsymbol{a}_1, \cdots, \boldsymbol{s}_{t-1}, \boldsymbol{a}_{t-1})$ is the trajectories sampled by following the policy π from an initial state $\boldsymbol{s}_0 \sim p(\boldsymbol{s}_0)$ with the transition function $\mathcal{P}(\cdot|\boldsymbol{s}_{t-1}, \boldsymbol{a}_{t-1})$.

Model-free off-policy RL algorithms obtain optimal policy $\pi_\theta \approx \pi^*$ by estimating optimal state-action value function $Q^{\pi^*} : S \times A \mapsto \mathbb{R}$ as $Q_\phi(\boldsymbol{s}_t, \boldsymbol{a}_t) \approx Q^{\pi^*}(\boldsymbol{s}_t, \boldsymbol{a}_t) = \max_{\pi_\theta} \mathbb{E}[\sum_{t=0}^{T-1} \gamma^t \mathcal{R}(\boldsymbol{s}_t, \boldsymbol{a}_t)|\boldsymbol{s}_t, \boldsymbol{a}_t]$ with function approximation. In particular, we minimize Temporal Difference

(TD) error (Sutton 1988) to update Q_ϕ, formulated as:

$$\mathcal{L}_{td} = \mathbb{E}_{\boldsymbol{s}_t,\boldsymbol{a}_t,\boldsymbol{s}_{t+1}\sim\mathcal{B}}[\frac{1}{2}\|Q_\phi(\boldsymbol{s}_t,\boldsymbol{a}_t) - y_t\|_2^2], \quad (1)$$

where $y_t = \mathcal{R}(\boldsymbol{s}_t,\boldsymbol{a}_t) + \gamma\mathbb{E}_{\pi_\theta}[Q_\phi^{\text{tgt}}(\boldsymbol{s}_{t+1},\boldsymbol{a}_{t+1})]$ and $\mathcal{B}$ is a replay buffer of transitions $(\boldsymbol{s}_t,\boldsymbol{a}_t,\boldsymbol{s}_{t+1})$ collected by behavior policy (Lin 1992). In practice, the parameters of Q_ϕ^{tgt} are periodically updated with a slow-moving average of Q_ϕ (Lillicrap et al. 2015):

$$\phi_{n+1}^{\text{tgt}} = (1-\eta)\phi_n^{\text{tgt}} + \eta\phi_n, \quad (2)$$

where ϕ_n^{tgt} is the parameters of Q_ϕ^{tgt} of the n^{th} step, and $\eta \in (0,1]$ is the momentum coefficient.

Generalization in Visual Reinforcement Learning

In this work, we focus on learning a policy π_θ that generalizes well to unseen MDPs $\mathcal{M}_\omega =< \mathcal{S},\mathcal{A},\mathcal{P},\mathcal{R},\gamma,\mathcal{O} >$, where $\mathcal{O}$ is visual observation space. For a space of MDPs $\mathbb{M}$, $\mathcal{M}_\omega \sim \mathbb{M}$ share a common underlying structure since the transition function $\mathcal{P}(\cdot)$ is invariant but with different observation spaces $\mathcal{O}$ perturbed by the visual transform function $g_\omega(\cdot)$ from the same state space $\mathcal{S}$. For convenience, we denote the training and testing environments as $\mathcal{M}_{\text{train}}$ and $\mathcal{M}_{\text{test}}$, respectively, drawn from $\mathbb{M}$. We aim to learn a policy $\pi_\theta \in \Pi$ that maximizes the average return $\mathbb{E}_{\mathcal{M}_\omega}\mathbb{E}_{\pi_\theta}[U]$ over all possible MDPs $\mathcal{M}_\omega$. Although exactly evaluating the average return is impossible, we can estimate it with finite MDPs and formulate it as:

$$\max_\pi \frac{1}{M}\sum_{m=1}^{M}\frac{1}{K}\sum_{k=1}^{K}\sum_{t=0}^{T-1}\gamma^t\mathcal{R}(\boldsymbol{s}_t,\boldsymbol{a}_t), \quad (3)$$

where M is the number of test MDPs sampled from $\mathbb{M}$, and K indicates the number of episodes that policy executed within an $\mathcal{M}_\omega$.

Soft Actor-Critic

Soft Actor-Critic (SAC) (Haarnoja et al. 2018) is an off-policy actor-critic algorithm consisting of a state-action value function $Q_\phi(\boldsymbol{s},\boldsymbol{a})$ and a stochastic policy π_θ. $Q_\phi(\boldsymbol{s},\boldsymbol{a})$ is optimized by $L_Q(\phi)$, a modified form of Equ. 1. Concurrently, the optimization of π_θ employs the maximum-entropy objective, indicated as $L_\pi(\theta)$ (Ziebart et al. 2008). SAC adopts the same structure (double Q-network) as Double Q-learning (Hasselt 2010) to improve stability. We describe our method with the general off-policy framework in the rest of this paper. Then, we evaluate the performance of our method in experiments using SAC as the base algorithm.

Methodology

In this section, we present our method from both theoretical and empirical perspectives. First, we derive a theoretical upper bound comprising two components from the generalization objective function. Then, we propose maintaining the cross-domain consistency of each policy in the policy space by Truncated Return Prediction (TRP) to reduce the learned policy shift during testing. At last, we propose a Transformer-based network to help predict truncated return.

Theoretical Analysis

For a certain policy π, we can *i.i.d* sample $(\boldsymbol{s},\boldsymbol{a})$ from marginal distribution μ_t at the t^{th} step. We assume marginal distribution μ belongs to expert policy π^* and satisfies the following assumption.

Assumption 1 *Given a policy π, let ν be the marginal distribution over $(\boldsymbol{s},\boldsymbol{a})$ at the t^{th} step. There exists a constant C satisfies:*

$$\sup_{(\boldsymbol{s},\boldsymbol{a},t)}\frac{\mathrm{d}\nu_t}{\mathrm{d}\mu_t}(\boldsymbol{s},\boldsymbol{a}) \le C, \forall\pi\in\Pi. \quad (4)$$

In visual reinforcement learning, we are interested in the performance of learned policy π_θ deployed in test environments. Thus, we can measure generalization by sub-optimality in state-action values, *i.e.*, $d(\pi^*,\pi_\theta) = \mathbb{E}_{\pi_\theta,\pi^*}[Q^{\pi^*}(\boldsymbol{s}_0,\pi^*(\cdot|\boldsymbol{s}_0)) - Q^{\pi_\theta}(\boldsymbol{s}_0,\pi_\theta(\cdot|\boldsymbol{s}_0))]$. Since the gap is highly non-smoothly and intractable, a popular alternative approach is using the Bellman error (Bradtke and Barto 1996; Antos, Szepesvári, and Munos 2008).

Definition 1 *Under marginal distribution μ over $(\boldsymbol{s},\boldsymbol{a})$, the Bellman error can be defined as the following:*

$$\varepsilon(\theta) = \frac{1}{T}\sum_{t=0}^{T-1}\mathrm{d}\mu_t\cdot\|Q^{\pi_\theta}(\boldsymbol{s}_t,\boldsymbol{a}_t) - y_t^*\|_2^2, \quad (5)$$

where $y_t^* = \mathbb{E}_{\boldsymbol{s}_{t+1},\pi^*}[R_t + \gamma Q^{\pi^*}(\boldsymbol{s}_{t+1},\boldsymbol{a}_{t+1})]$.

Under Assumption 1, we can derive the following theorem that controls the sub-optimality by the Bellman error, *i.e.*, we derive an upper bound for RL generalization.

Theorem 1 *Under Assumption 1, the generalization error of π_θ can be bounded as:*

$$d(\pi^*,\pi_\theta) \le 2T\sqrt{C\cdot\varepsilon(\theta)}. \quad (6)$$

See Appendix A for more details (proof).

For the first component, C is the upper bound of the Radon-Nikodym derivative $\frac{\mathrm{d}\nu_t}{\mathrm{d}\mu_t}(\boldsymbol{s},\boldsymbol{a})$, which indicates the change rate of ν^{π_θ} with respect to μ^{π^*}. For the second component, $\varepsilon(\theta)$ represents the discrepancy in the accuracy of the state-action value function. DrQ, SVEA, and SGQN adopt various Q function consistency, which can be regarded as minimizing $\varepsilon(\theta)$. SODA learns an invariant embedding for the same state is equivalent to minimizing C.

Domain Consistency in Policy Space

Policy Divergence. As illustrated in the above section, we need to minimize both components of the upper bound to improve generalization. Firstly, we need to minimize the divergent ratio of the state-action distribution. Since the marginal distribution over $(\boldsymbol{s},\boldsymbol{a})$ is generated by the policy and the transition function, $\frac{\mathrm{d}\nu_t}{\mathrm{d}\mu_t}(\boldsymbol{s},\boldsymbol{a})$ can be expressed as:

$$\prod_{i=0}^{i=t-1}\frac{p(\boldsymbol{s}_0)\pi_\theta(\boldsymbol{a}_i|\boldsymbol{s}_i)\mathcal{P}(\boldsymbol{s}_{i+1}|\boldsymbol{s}_i,\boldsymbol{a}_i)}{p(\boldsymbol{s}_0)\pi^*(\boldsymbol{a}_i|\boldsymbol{s}_i)\mathcal{P}(\boldsymbol{s}_{i+1}|\boldsymbol{s}_i,\boldsymbol{a}_i)}, \quad (7)$$

where $p(\boldsymbol{s}_0)$ denotes the distribution of the initial state.

In visual RL, the test environments are unknown. A reasonable assumption entails that the policy learned from the

training environment represents the optimal policy we can achieve. Therefore, we can utilize actions selected by π_θ over $g_{\text{train}}(\boldsymbol{s}_t)$ as the decisions made by the optimal policy π^*. For convenience, the minimization of C can be represented as follows:

$$\min_{\theta}[\sup_{(\boldsymbol{s},\boldsymbol{a},t,g(\cdot))} \mathbb{E}_{\boldsymbol{s}_t}[\frac{\pi_\theta(\cdot|g_{\text{test}}(\boldsymbol{s}_t))}{\pi_\theta(\cdot|g_{\text{train}}(\boldsymbol{s}_t))}]]. \tag{8}$$

Although minimizing the Radon-Nikodym derivative (Equ. 8) is challenging, maintaining $\pi_\theta(\cdot|g_{\text{test}}(\boldsymbol{s}_t))$ similar to $\pi_\theta(\cdot|g_{\text{train}}(\boldsymbol{s}_t))$ is a local optimum. Previous researches primarily concentrate on domain randomization, intending to minimize policy divergence across various data augmentations. However, selecting more effective data augmentations is challenging. Since $\pi_\theta \circ g_{\text{test}} : S \mapsto A$, we can regard $\pi_\theta \circ g_{\text{test}}$ as a composite function $\pi_\vartheta \in \Pi$, representing the shift policy. As a result, Equ. 8 can be rewritten as:

$$\min_{\vartheta}[\sup_{(\boldsymbol{s},\boldsymbol{a},t)} \mathbb{E}_{\boldsymbol{s}_t}[\frac{\pi_\vartheta(\cdot|\boldsymbol{s}_t)}{\pi_\theta(\cdot|g_{\text{train}}(\boldsymbol{s}_t))}]]. \tag{9}$$

Therefore, when π_ϑ aligns consistently with its corresponding policy $\pi_{\vartheta'}$ in the training environment, this alignment contributes to decreasing the likelihood of the learned policy $\pi_\theta(\cdot|g_{\text{train}}(\boldsymbol{s}_t))$ shifting towards π_ϑ during testing. In other words, we traverse the policy space and maintain the cross-domain consistency of each policy in the policy space, which can improve the generalization performance of the learned policy. Note that sampling different policies is simpler than proposing various domains.

Bellman Error. Secondly, we also need to consider the second component of the upper bound, *i.e.*, minimizing Bellman error $\varepsilon(\theta)$. In practice, directly maintaining consistency of action distributions across different domains for each policy is challenging. Given the intimate link between policy and state-action value function within off-policy RL, it is feasible to represent the policy via state-action values. Therefore, we utilize value consistency instead of policy consistency. This approach simultaneously minimizes the Bellman error implicitly, thus contributing to learning an accurate state-action value function.

Importance of Policy Diversity. We design a straightforward experiment, as shown in Figure 2. Throughout the training process, SVEA undergoes different policies. Therefore, the consistency maintained by SVEA at different stages belongs to different policies. For comparison, we employ an expert policy to guide the learning of Q_ϕ, denoted as SVEA-Expert. In contrast to SVEA, the Q_ϕ function is always consistent with the Q^{π^*} of the expert policy throughout all stages of training. Apart from that, SVEA-Expert has the same settings as SVEA. Figure 2 shows that SVEA achieves better performance on all tasks. Experimental results demonstrate that maintaining cross-domain consistency for various policies yields better generalization. Notably, the Q network belongs to the current policy, even utilizing historical data. This limitation indicates that SVEA can only maintain the consistency of one policy in a single stage. In contrast to SVEA, we maintain policy consistency for multiple policies at each step.

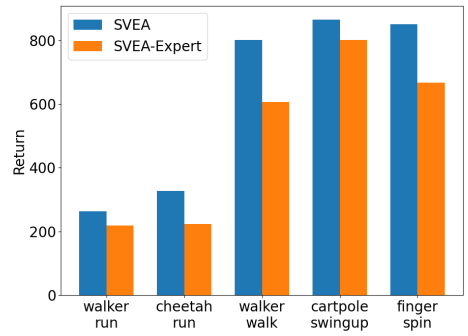

Figure 2: Comparison between SVEA and SVEA-Expert. SVEA-Expert utilizes an expert policy to guide Q-function learning. The vertical axis represents the cumulative reward, while the horizontal axis represents different tasks.

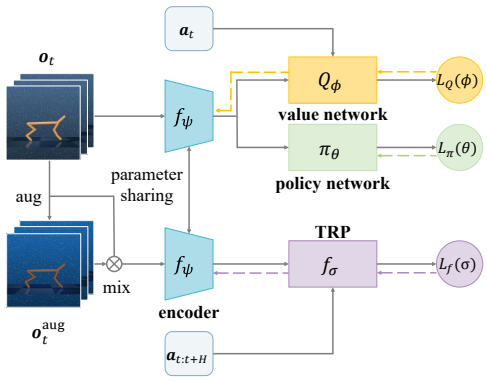

Figure 3: Overview. An observation $\boldsymbol{o}_t$ is augmented as $\boldsymbol{o}_t^{\text{aug}}$. On the one hand, only unaugmented observations are used when updating the policy network π_θ and the value network Q_ϕ. On the other hand, we feed mixed observations ($\boldsymbol{o}_t$ and $\boldsymbol{o}_t^{\text{aug}}$) and the action sequence $\boldsymbol{a}_{t:t+H}$ into the predictor f_σ to predict truncated return. Observation encoder f_ψ shared parameters in two branches. Note that observation encoder f_ψ only updates when optimize $L_Q(\phi)$ and $L_f(\sigma)$.

Truncated Return Prediction. However, acquiring more policies requires exploration, leading to sample inefficiency. In addition, maintaining these policies also results in increased memory consumption. As policies vary across different training stages, we sample truncated trajectories from the replay buffer to simulate trajectories generated by different policies with Monte Carlo sampling. In practice, we propose an auxiliary task named Truncated Return Prediction (TRP) to maintain the consistency of different domains. As shown in Figure 3, we illustrated the overview of TRP denoted as f_σ. Specifically, we predict the truncated return before and after augmentations to maintain the cross-domain consistency of state-action value function, *i.e.*, policy cross-domain consistency. We utilize $\boldsymbol{a}_{t:t+H}$ to represent the ac-

Algorithm 1: Generic TRP off-policy algorithm

$\theta, \phi, \sigma, \phi$: randomly initialized network parameters
η, ξ, α, β: learning rate and constant coefficients
$\mathcal{B}$: empty replay buffer

for $t = 1 \cdots T$ **do**
  sample action $\boldsymbol{a}_t \sim \pi_\theta(\cdot|\boldsymbol{s}_t)$
  interact with the environment $\boldsymbol{s}_{t+1} \sim \mathcal{P}(\cdot|\boldsymbol{s}_t, \boldsymbol{a}_t)$
  store transition $\mathcal{B} \leftarrow \mathcal{B} \cup (\boldsymbol{s}_t, \boldsymbol{a}_t, R_t(\boldsymbol{s}_t, \boldsymbol{a}_t), \boldsymbol{s}_{t+1})$
  sample mini-batch trajectories $\{\tau_i\} \sim \mathcal{B}$
  update value network $\phi \leftarrow \phi - \eta \nabla_\phi \mathcal{L}_Q(\phi)$
  update policy network $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_\pi(\theta)$
  update prediction network $\sigma \leftarrow \sigma - \xi \nabla_\sigma \mathcal{L}_f(\sigma)$
end for

tion sequence and the objective can be formulated as:

$$L_f(\sigma) = \mathbb{E}_{\tau\sim\mathcal{B}}[\alpha \| f_\sigma(\boldsymbol{o}_t, \boldsymbol{a}_{t:t+H}) - \sum_{h=0}^{H} \gamma^{t+h} R_i \|_2^2 + \beta \| f_\sigma(\boldsymbol{o}_t^{\text{aug}}, \boldsymbol{a}_{t:t+H}) - \sum_{h=0}^{H} \gamma^{t+h} R_i \|_2^2], \tag{10}$$

where α and β are constant coefficients; $\boldsymbol{o}_t$ and $\boldsymbol{o}_t^{\text{aug}}$ represent unaugmented and augmented observation, respectively. We set $\alpha = \beta = 0.5$, which is adopted in most experiments. We use the same encoder f_ψ to obtain observation embedding before feeding it into π_θ, Q_ϕ, and f_σ. It is worth noting that only the gradient from $\mathcal{L}_Q(\phi)$ and $\mathcal{L}_f(\sigma)$ updates the observation encoder. We summarize our method to a generic off-policy algorithm in Algorithm 1 and omit the details of updating SAC for clarity.

Transformer-based Prediction Network

In this section, we introduce a Transformer-based network architecture, illustrated in Figure 4. Unlike (Hansen, Su, and Wang 2021) uses Transformer for observation representation, we focus on capturing the relationship between the observation $\boldsymbol{o}_t$ and the action sequence $\boldsymbol{a}_{t:t+H}$. Given $(\boldsymbol{o}_t, \boldsymbol{a}_{t:t+H})$ to our Transformer-based prediction network, we calculate Mean Squared Error (MSE) between the predicted value and the actual truncated return. Specifically, we share the parameters of the observation encoder to embed the raw pixels as observation tokens. Meanwhile, actions $\boldsymbol{a}_{t:t+H}$ are fed into an action encoder (a linear layer) to obtain action tokens. Additionally, we randomly initialize a learnable vector as a prediction token. After that, all tokens (prediction token, observation token, and action token) are added with positional embedding, indicating the order of the input sequence. Then, we feed the added embedding into a two-layer Transformer Encoder, sharing a unit structure similar to ViT (Dosovitskiy et al. 2021). At last, we utilize the first output of the Transformer Encoder to predict the truncated return with a predictor head.

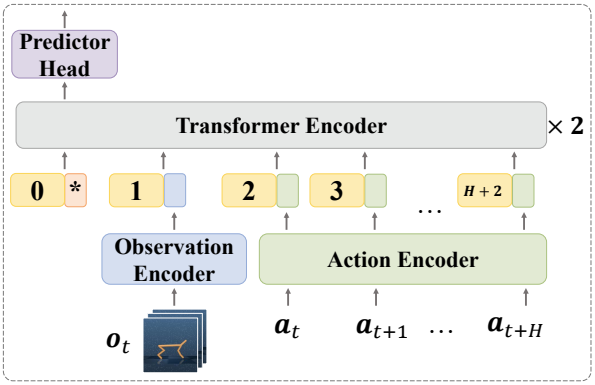

Figure 4: Transformer-based network for prediction.

Experiments

This section evaluates generalization and sample efficiency compared with a set of baselines on DeepMind Control Suite (DMControl) tasks (Tassa et al. 2018) and Robotic Manipulation tasks (Jangir et al. 2022). We evaluate the generalization performance of our method by testing the learned policy in complex visual distribution environments. DMControl Generalization Benchmark (DMControl-GB), as presented in (Hansen and Wang 2021), is a variant of DMControl where the backgrounds are replaced by random colors or natural videos. Robotic Manipulation aims to control a robotic arm to accomplish some manipulation tasks. Similarly to DMControl-GB, the testing environment introduces diverse disturbances in texture and color to the background. For a detailed description of the experimental setup, please refer to Appendix B and C. The code has been released at: `https://github.com/Rebel-Uranus/TRP`

Baselines. 1) **SAC** (Haarnoja et al. 2018), a standard off-policy reinforcement learning without any modifications for visual reinforcement learning; 2) **CURL** (Laskin, Srinivas, and Abbeel 2020), a contrastive learning method combined with RL; 3) **RAD** (Laskin et al. 2020), empirically studying the effect of different data augmentations; 4) **PAD** (Hansen et al. 2021), a sim-to-real method that adjusts observation embedding during testing; 5) **DrQ** (Yarats, Kostrikov, and Fergus 2021), an approach that leverages augmentations for both Q-values and target Q-values during the TD update; 6) **SODA** (Hansen and Wang 2021), adopting BYOL (Grill et al. 2020) for learning invariant representation; 7) **SVEA** (Hansen, Su, and Wang 2021), stabilizing Q learning with o_{t+1} unaugmented; 8) **SGQN** (Bertoin et al. 2022), leveraging a saliency-guided map to emphasize crucial pixels.

Data Augmentations. In this paper, the primary augmentation method employed for most experiments is random overlay (Hansen and Wang 2021). Meanwhile, we report some competitive results conducted with random convolution (Lee et al. 2020). Random overlay is an augmentation method that linearly interpolates an observation and another image sampled from Places (Zhou et al. 2018), a repository of 10 million scene photographs. Random convolution incorporates stochastically parameterized convolutional layers

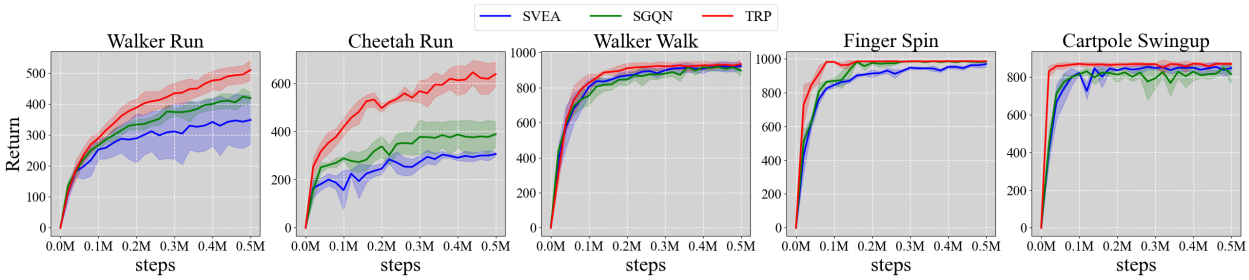

Figure 5: Sample efficiency in the training environment. We compare SVEA, SGQN, and TRP across 5 environments. Mean and standard deviation are calculated based on 5 runs.

DMControl-GB (color-hard)	CURL	RAD	DrQ	PAD	SODA	SVEA	SGQN	TRP
walker, run	193 ± 22	160 ± 18	192 ± 10	232 ± 43	230 ± 35	274 ± 72	311 ± 14	$\mathbf{372 \pm 25}$
cheetah, run	153 ± 18	171 ± 11	100 ± 27	159 ± 28	294 ± 34	273 ± 23	302 ± 35	$\mathbf{476 \pm 21}$
walker, walk	445 ± 99	400 ± 61	520 ± 91	468 ± 47	697 ± 66	760 ± 145	780 ± 58	$\mathbf{823 \pm 28}$
cartpole, swingup	454 ± 110	590 ± 53	586 ± 52	630 ± 63	831 ± 21	$\mathbf{837 \pm 23}$	777 ± 25	799 ± 40
finger, spin	691 ± 12	667 ± 154	776 ± 143	803 ± 72	901 ± 51	$\mathbf{977 \pm 5}$	874 ± 31	918 ± 34
DMControl-GB (video-easy)	CURL	RAD	DrQ	PAD	SODA	SVEA	SGQN	TRP
walker, run	256 ± 21	185 ± 16	229 ± 17	219 ± 18	272 ± 15	249 ± 63	346 ± 21	$\mathbf{426 \pm 44}$
cheetah, run	153 ± 24	185 ± 26	102 ± 30	206 ± 34	229 ± 29	292 ± 32	285 ± 46	$\mathbf{443 \pm 25}$
walker, walk	556 ± 133	606 ± 63	682 ± 89	717 ± 79	768 ± 38	819 ± 71	865 ± 13	$\mathbf{871 \pm 18}$
cartpole, swingup	404 ± 67	373 ± 72	485 ± 105	521 ± 76	758 ± 62	$\mathbf{782 \pm 27}$	761 ± 28	733 ± 62
finger, spin	502 ± 19	400 ± 64	533 ± 119	691 ± 80	695 ± 97	808 ± 33	$\mathbf{956 \pm 26}$	816 ± 26

Table 1: Comparison with the state-of-the-art methods on DMControl-GB. Our results reported include the average and standard deviation of return. Each result is calculated across 5 random seeds. We denote optimal performance in bold. We use underscores to indicate methods augmented by random convolution, while others are augmented by random overlay.

into the observations, preserving the overall shape while introducing blurring effects to texture features.

Evaluation on DMControl

We evaluate generalization performance and sample efficiency of TRP in comparison to other methods using *color-hard* and *video-easy* settings from DMControl-GB across 5 tasks, *i.e.*, *walker run, cheetah run, walker walk, cartpole swingup*, and *finger spin*.

Generalization Performance. We evaluate the generalization performance compared with the other six methods mentioned above. We run each task with 5 seeds to calculate its average and standard deviation of return. Since SODA and SVEA are significantly affected by different data augmentations, we utilize underscores to represent methods augmented by random convolution. Throughout this paper, we choose random overlay to augment observations without additional instructions. We bold the results with the best generalization performance in the same environment. As indicated in Table 1, TRP outperforms prior state-of-the-art methods in 6 out of 10 environments. Moreover, among the 4 remaining environments, 2 display suboptimal performance. TRP achieves significant improvements on more difficult tasks, such as *walker run* and *cheetah run*.

Sample Efficiency. We compare sample efficiency between SVEA, SGQN, and TRP in the training domain on five environments from DMControl-GB with 5 seeds. The solid lines represent the mean return at each step, and the shaded parts represent one standard deviation. As shown in Fig. 5, TRP outperforms other methods in terms of both asymptotic performance and sample efficiency, especially on *walker run* and *cheetah run*. Note that TRP requires a few steps on *finger spin* and *cartpole swingup* to learn the convergent policy. In other words, most of the transitions in the replay buffer come from similar policies. This limitation prevents TRP from mitigating policy shifts without adequate constraints. This may be one reason for the inferiority of TRP in generalization performance on the two tasks.

Evaluation on Robotic Manipulation

To demonstrate the generality of our method, we conduct experiments on Robotic Manipulation tasks introduced by (Jangir et al. 2022). We consider two goal-reaching tasks: 1) *Reach*, where the robot is required to reach for a red mask on the table; 2) *PegBox*, where the robot attempts to insert the peg into a box. Test environments only disturb the texture or color of the background and preserve crucial information. We adopt four test environments for each task. Following the setup of (Bertoin et al. 2022), we apply random convolution on SODA, SVEA, and SGQN. TRP adopts random overlay,

Task	Environment	SAC	SODA	SVEA	SGQN	TRP
Reach	Test1	-20.9 ± 16	-30.9 ± 43	-17.6 ± 10	14.4 ± 14	$\mathbf{18.0 \pm 7}$
	Test2	-21.9 ± 14	-20.2 ± 29	-2.1 ± 39	$\mathbf{31.0 \pm 3}$	-25.4 ± 19
	Test3	-43.2 ± 6	-68.4 ± 30	1.4 ± 29	29.2 ± 7	$\mathbf{29.6 \pm 2}$
	Test4	-36.7 ± 3	-37.9 ± 26	-22.2 ± 24	-5.6 ± 20	$\mathbf{24.1 \pm 5}$
PegBox	Test1	-59.6 ± 26	16.9 ± 44	-21.3 ± 10	-72.1 ± 14	$\mathbf{63.5 \pm 17}$
	Test2	-60.5 ± 12	0.7 ± 30	96.8 ± 41	$\mathbf{110.7 \pm 3}$	55.6 ± 31
	Test3	-48.8 ± 17	73.6 ± 31	40.5 ± 28	154.6 ± 7	$\mathbf{162.5 \pm 23}$
	Test4	-89.6 ± 31	-16 ± 83	33.2 ± 43	-70.9 ± 41	$\mathbf{117.4 \pm 47}$

Table 2: Comparison with the state-of-the-art methods on robotic manipulation tasks. Our results reported include the average and standard deviation of return. Each result is calculated across 5 random seeds. We denote the optimal result in bold.

DMControl-GB (color-hard)	Base	RNN	TF
walker, run	313 ± 57	321 ± 24	$\mathbf{372 \pm 25}$
cheetah, run	420 ± 12	415 ± 30	$\mathbf{476 \pm 21}$
walker, walk	777 ± 35	799 ± 27	$\mathbf{823 \pm 28}$
cartpole, swingup	787 ± 32	761 ± 56	$\mathbf{799 \pm 40}$
finger, spin	896 ± 22	914 ± 18	$\mathbf{918 \pm 34}$
DMControl-GB (video-easy)	**Base**	**RNN**	**TF**
walker, run	355 ± 29	358 ± 13	$\mathbf{426 \pm 44}$
cheetah, run	402 ± 32	400 ± 31	$\mathbf{443 \pm 25}$
walker, walk	867 ± 31	846 ± 10	$\mathbf{871 \pm 18}$
cartpole, swingup	730 ± 61	671 ± 89	$\mathbf{733 \pm 62}$
finger, spin	$\mathbf{868 \pm 99}$	815 ± 21	816 ± 26

Table 3: Ablation study of different architectures. Results include the average and standard deviation of the return across 5 random seeds. We denote the optimal one in bold.

which is empirically suitable for our methods.

The results are reported in Table 2. TRP achieves the state-of-the-art in 6 out of 8 test environments. In the *Reach* task, TRP outperforms other methods in Test1, Test3, and Test4 environments. Specifically, TRP has been significantly improved on Test1 and Test4 by 25% and 530%, respectively. Meanwhile, TRP achieves the best result on environment Test3. In the *PegBox* task, TRP surpasses the second-best method (SODA) by 275% on Test1 and achieves a 253% performance improvement on Test4. In addition, our method outperforms SGQN 5% on Test3. Intriguingly, our method underperforms on Test2 for both tasks, warranting further investigation.

The above experiments demonstrate the advancement of our method. Moreover, TRP is applicable to different visual reinforcement learning tasks without any additional assumptions. We also provide additional experiments in Appendix D to support the above conclusions. These experiments include autonomous driving tasks on Carla, hyperparameter sensitivity experiments, and visualization of attention maps.

Ablation Study

To illustrate the effectiveness of a Transformer-based predictor, denoted as *TF*, we compare our method with two classical structures: 1) *Base*, a fully connected neural network; 2) *RNN*, a two-layer GRU network using observation as initial hidden. While *Base* does not explicitly leverage the sequential order information of input sequences, *RNN* naturally incorporates the sequence order. However, *RNN* is prone to forgetting long-term historical information. *TF* combines the advantages of the fully connected and recurrent neural network, *i.e.*, it uses sequence order information and avoids historical information forgetting. Results are shown in Table 3. Overall, there is no obvious performance gap between the three structures. In other words, the superiority of TRP over other previous approaches arises from policy consistency rather than structural design. Nevertheless, *TF* exhibits a significant advantage and attains superior results in most environments.

In most environments, *RNN* achieves the lowest standard deviation and average return. One possible explanation is that sequence order information is crucial for predicting truncated return, as the agent interacts with the environment sequentially. Compared to *RNN*, *Base* achieves a higher average return on most environments but is also more susceptible to a higher standard deviation. This phenomenon indicates that *RNN* easily suffers from forgetting historical information. Particularly, it is fatal to forget observation o_t. In summary, the Transformer-based network that combines the advantages of the preceding architectures is more suitable for truncated return prediction.

Conclusion

We theoretically derive an upper bound for the generalization objective in visual reinforcement learning. Based on the upper bound, we provide an explanation for the effectiveness of some previous works. Inspired by theoretical analysis, we introduce a practical auxiliary task, *i.e.*, predicting the truncated return of trajectories sampled by any policy. In addition, we design a variant Transformer as the prediction network. Finally, experimental results show that TRP achieves the best generalization performance in most environments.

This paper is the first to theoretically analyze the factors affecting the generalization of visual reinforcement learning. We hope that this work can be beneficial to the community to advance the research on the generalization of visual reinforcement learning. Moreover, exploring the extension of our theoretical analysis to gradient-based reinforcement learning algorithms is an exciting perspective.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 62206013) and the Aeronautical Science Foundation of China (Grant No. 202300010M5001).

References

Antos, A.; Szepesvári, C.; and Munos, R. 2008. Learning near-optimal policies with Bellman-residual minimization based fitted policy iteration and a single sample path. *Mach. Learn.*, 71(1): 89–129.

Bertoin, D.; Zouitine, A.; Zouitine, M.; and Rachelson, E. 2022. Look where you look! Saliency-guided Q-networks for generalization in visual Reinforcement Learning. In *NeurIPS*.

Bradtke, S. J.; and Barto, A. G. 1996. Linear Least-Squares Algorithms for Temporal Difference Learning. *Mach. Learn.*, 22(1-3): 33–57.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Fan, J.; and Li, W. 2022. DRIBO: Robust Deep Reinforcement Learning via Multi-View Information Bottleneck. In Chaudhuri, K.; Jegelka, S.; Song, L.; Szepesvári, C.; Niu, G.; and Sabato, S., eds., *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, 6074–6102. PMLR.

Fan, L.; Wang, G.; Huang, D.; Yu, Z.; Fei-Fei, L.; Zhu, Y.; and Anandkumar, A. 2021. SECANT: Self-Expert Cloning for Zero-Shot Generalization of Visual Policies. In Meila, M.; and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, 3088–3099. PMLR.

Grill, J.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P. H.; Buchatskaya, E.; Doersch, C.; Pires, B. Á.; Guo, Z.; Azar, M. G.; Piot, B.; Kavukcuoglu, K.; Munos, R.; and Valko, M. 2020. Bootstrap Your Own Latent - A New Approach to Self-Supervised Learning. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, 1861–1870. PMLR.

Hafner, D.; Pasukonis, J.; Ba, J.; and Lillicrap, T. P. 2023. Mastering Diverse Domains through World Models. *CoRR*, abs/2301.04104.

Hansen, N.; Jangir, R.; Sun, Y.; Alenyà, G.; Abbeel, P.; Efros, A. A.; Pinto, L.; and Wang, X. 2021. Self-Supervised Policy Adaptation during Deployment. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Hansen, N.; Su, H.; and Wang, X. 2021. Stabilizing Deep Q-Learning with ConvNets and Vision Transformers under Data Augmentation. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y. N.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, 3680–3693.

Hansen, N.; and Wang, X. 2021. Generalization in Reinforcement Learning by Soft Data Augmentation. In *IEEE International Conference on Robotics and Automation, ICRA 2021, Xi'an, China, May 30 - June 5, 2021*, 13611–13617. IEEE.

Hasselt, H. 2010. Double Q-learning. *Advances in neural information processing systems*, 23.

He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; and Girshick, R. B. 2022. Masked Autoencoders Are Scalable Vision Learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, 15979–15988. IEEE.

Hu, X.; Lin, Y.; Wang, S.; Wu, Z.; and Lv, K. 2023. Agent-centric relation graph for object visual navigation. *IEEE Transactions on Circuits and Systems for Video Technology*.

James, S.; and Davison, A. J. 2022. Q-Attention: Enabling Efficient Learning for Vision-Based Robotic Manipulation. *IEEE Robotics Autom. Lett.*, 7(2): 1612–1619.

Jangir, R.; Hansen, N.; Ghosal, S.; Jain, M.; and Wang, X. 2022. Look Closer: Bridging Egocentric and Third-Person Views With Transformers for Robotic Manipulation. *IEEE Robotics Autom. Lett.*, 7(2): 3046–3053.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In Bartlett, P. L.; Pereira, F. C. N.; Burges, C. J. C.; Bottou, L.; and Weinberger, K. Q., eds., *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, 1106–1114.

Laskin, M.; Lee, K.; Stooke, A.; Pinto, L.; Abbeel, P.; and Srinivas, A. 2020. Reinforcement Learning with Augmented Data. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Laskin, M.; Srinivas, A.; and Abbeel, P. 2020. CURL: Contrastive Unsupervised Representations for Reinforcement Learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, 5639–5650. PMLR.

Lee, K.; Lee, K.; Shin, J.; and Lee, H. 2020. Network Randomization: A Simple Technique for Generalization in Deep Reinforcement Learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

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

Lin, L.-J. 1992. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine learning*, 8: 293–321.

Schwarzer, M.; Anand, A.; Goel, R.; Hjelm, R. D.; Courville, A. C.; and Bachman, P. 2021. Data-Efficient Reinforcement Learning with Self-Predictive Representations. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Shorten, C.; and Khoshgoftaar, T. M. 2019. A survey on Image Data Augmentation for Deep Learning. *J. Big Data*, 6: 60.

Sutton, R. S. 1988. Learning to predict by the methods of temporal differences. *Machine learning*, 3: 9–44.

Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement learning: An introduction*. MIT press.

Tassa, Y.; Doron, Y.; Muldal, A.; Erez, T.; Li, Y.; de Las Casas, D.; Budden, D.; Abdolmaleki, A.; Merel, J.; Lefrancq, A.; Lillicrap, T. P.; and Riedmiller, M. A. 2018. DeepMind Control Suite. *CoRR*, abs/1801.00690.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention Is All You Need. arXiv:1706.03762.

Wang, K.; Kang, B.; Shao, J.; and Feng, J. 2020. Improving Generalization in Reinforcement Learning with Mixture Regularization. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Wang, S.; Wu, Z.; Hu, X.; Lin, Y.; and Lv, K. 2023. Skill-based Hierarchical Reinforcement Learning for Target Visual Navigation. *IEEE Transactions on Multimedia*.

Wang, X.; Lian, L.; and Yu, S. X. 2021. Unsupervised Visual Attention and Invariance for Reinforcement Learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, 6677–6687. Computer Vision Foundation / IEEE.

Wu, K.; Wu, M.; Chen, Z.; Xu, Y.; and Li, X. 2022. Generalizing Reinforcement Learning through Fusing Self-Supervised Learning into Intrinsic Motivation. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, 8683–8690. AAAI Press.

Yarats, D.; Kostrikov, I.; and Fergus, R. 2021. Image Augmentation Is All You Need: Regularizing Deep Reinforcement Learning from Pixels. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Yu, T.; Zhang, Z.; Lan, C.; Lu, Y.; and Chen, Z. 2022. Mask-based Latent Reconstruction for Reinforcement Learning. In *NeurIPS*.

Zang, H.; Li, X.; and Wang, M. 2022. SimSR: Simple Distance-Based State Representations for Deep Reinforcement Learning. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, 8997–9005. AAAI Press.

Zhang, A.; McAllister, R. T.; Calandra, R.; Gal, Y.; and Levine, S. 2021. Learning Invariant Representations for Reinforcement Learning without Reconstruction. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Zhao, W.; Queralta, J. P.; and Westerlund, T. 2020. Sim-to-Real Transfer in Deep Reinforcement Learning for Robotics: a Survey. In *2020 IEEE Symposium Series on Computational Intelligence, SSCI 2020, Canberra, Australia, December 1-4, 2020*, 737–744. IEEE.

Zhong, Z.; Zheng, L.; Kang, G.; Li, S.; and Yang, Y. 2020. Random Erasing Data Augmentation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, 13001–13008. AAAI Press.

Zhou, B.; Lapedriza, À.; Khosla, A.; Oliva, A.; and Torralba, A. 2018. Places: A 10 Million Image Database for Scene Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(6): 1452–1464.

Ziebart, B. D.; Maas, A. L.; Bagnell, J. A.; Dey, A. K.; et al. 2008. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, 1433–1438. Chicago, IL, USA.