Understanding What Affects the Generalization Gap in Visual Reinforcement Learning: Theory and Empirical Evidence

Jiafei Lyu LVJF20@MAILS.TSINGHUA.EDU.CN
Tsinghua Shenzhen International Graduate School
Tsinghua University, Shenzhen, China

Le Wan VINOWAN@TENCENT.COM
IEG, Tencent
Shenzhen, China

Xiu Li LI.XIU@SZ.TSINGHUA.EDU.CN
(Corresponding Author)
Tsinghua Shenzhen International Graduate School
Tsinghua University, Shenzhen, China

Zongqing Lu ZONGQING.LU@PKU.EDU.CN
School of Computer Science, Peking University
Beijing, China

Abstract

Recently, there are many efforts attempting to learn useful policies for continuous control in visual reinforcement learning (RL). In this scenario, it is important to learn a *generalizable* policy, as the testing environment may differ from the training environment, e.g., there exist distractors during deployment. Many practical algorithms are proposed to handle this problem. However, to the best of our knowledge, none of them provide a theoretical understanding of what affects the generalization gap and why their proposed methods work. In this paper, we bridge this issue by theoretically answering the key factors that contribute to the generalization gap when the testing environment has distractors. Our theories indicate that minimizing the representation distance between training and testing environments, which aligns with human intuition, is the most critical for the benefit of reducing the generalization gap. Our theoretical results are supported by the empirical evidence in the DMControl Generalization Benchmark (DMC-GB).

1. Introduction

Visual reinforcement learning (RL) (Nair et al., 2018; Yarats et al., 2022) has aroused much attention from the community due to its success in handling complex control tasks solely from visual inputs. The advances in visual RL are promising for deploying RL algorithms in real-world applications since many physical devices (e.g., robotics) receive image observations. The progress in visual RL also has the potential to expedite the emergence of the large models in RL, or big decision models, since the visual inputs can be easily aligned with pre-processing and we only need to align the action space, whereas in state-based tasks, we need to handle the alignment of both state space and action space.

As a small step towards this goal, it is critical for the policies learned by the visual RL algorithms to be able to *generalize* to unseen scenarios, like human beings. Unfortunately, it is a challenging problem at the current stage as the difference between the clean training environment and the unseen environments is not predictable. For example, we train an RL agent in simulated environments from image observations, while need to deploy it in real-world tasks, where the evaluated scene may quite differ from the trained ones (e.g., it may rain). Meanwhile, the image observations are complex and fragile to be attacked, e.g., noises may be included in the testing environment.

Existing methods remedy the mismatch between the training and testing environment by leveraging data augmentation (Yarats et al., 2021, 2022; Hansen & Wang, 2020; Raileanu et al., 2021; Hansen et al., 2021b), self-supervision methods (Agarwal et al., 2021; Zhang et al., 2021; Hansen et al., 2021a) or pre-trained image encoders (Yuan et al., 2022; Ze et al., 2022; Dittadi et al., 2021), etc. Despite their success in achieving good performance during generalization, to the best of our knowledge, *none of them actively explain why their methods work in practice from a theoretical perspective.* In this paper, we aim at bridging this gap by theoretically analyzing what affects the generalization gap. We believe the theoretical understanding of how to reduce the generalization gap is important for guiding the design of good and generalizable algorithms in the future, instead of blindly adding new modules to pray for a 'lucky' trial.

We focus on the very specific generalization setting: the algorithm is trained in a clean environment with visual input, while deployed in an unseen testing environment with distractors. We note here that the distractor indicates that the observations in testing environment contain some diversions that may affect the performance of the agent, e.g., the color of the controlled agent or the background of the agent changes (the controlled agent remains unchanged). We also allow a slight difference of dynamics between training and testing environments. Nevertheless, it is very challenging to directly analyze the generalization gap between the training environment and the testing environment, since the policy keeps evolving during the training process.

To tackle this challenge, we resort to *reparameterization trick* to decouple the randomness in the environment from the evolving policy, the transition dynamics, and the initial state distribution. As a consequence, we present the reparameterizable visual RL framework. Under some mild assumptions, we establish concrete theoretical bounds on the generalization gap when deploying the visual policies in testing environments with distractors. Our results suggest that the most crucial factor that influences the test performance is the representation deviation before and after adding the distractor. Note that we focus on *on-policy* RL (where the episodes are sampled using the current policy during training) instead of off-policy RL, since the data-collecting policy keeps evolving in the context of visual RL.

We also examine the rationality of the assumptions we made, and the theoretical conclusions we achieved by conducting experiments of different algorithms in DMControl Generalization Benchmark (DMC-GB) (Hansen & Wang, 2020). In this benchmark, the testing environments involve distractions like color changing or background changing of the controlled agent (no extra object/hindrance are included). It turns out that the empirical evidence is consistent with the theoretical insights.

2. Related Work

Visual reinforcement learning. The success of learning visual representation in computer vision (Vincent et al., 2008; Doersch et al., 2015; Kolesnikov et al., 2019b; Cole et al., 2021; Kolesnikov et al., 2019a) has inspired the development of image-based RL. Many methods (Hafner et al., 2018; Majumdar et al., 2023; Lee et al., 2019a; Srinivas et al., 2020; Schwarzer et al., 2020) have shown the benefits of auto-encoders (Finn et al., 2015; Tschannen et al., 2018) to visual RL. The potential advantages of auxiliary tasks or objectives (Jaderberg et al., 2017; Stooke et al., 2020; Lin et al., 2019; Kulhánek et al., 2019) and data augmentations (Yarats et al., 2022, 2021; Laskin et al., 2020) are also widely explored. There are also some advances on explicit representation in visual RL like object-centric RL that aim at increasing the interpretability of decision-making (Wu et al., 2023; Delfosse et al., 2023, 2024; Luo et al., 2024; Zheng et al., 2022).

Generalization in visual RL. Generalization is a central challenge in RL (Cobbe et al., 2018; Packer et al., 2018), which is strongly correlated with overfitting issue (Cobbe et al., 2018; Zhang et al., 2018, 2018; Song et al., 2020) (it is customary that one uses the same environment for both training and testing). In visual RL, it is vital for the policy learned from pixels to be able to generalize to unseen scenarios due to the variability of image observations in many real-world tasks. Domain randomization (Chebotar et al., 2018; Peng et al., 2017; Pinto et al., 2017; Dai et al., 2019; Polvara et al., 2020; Slaoui et al., 2019; Yue et al., 2019) and data augmentation (Hansen et al., 2021b; Hansen & Wang, 2020; Raileanu et al., 2021; Fan et al., 2021; Lee et al., 2019b; Mitrano & Berenson, 2022; Huang et al., 2022; Maksymets et al., 2021) have been proven to be effective in terms of generalizing to visually different scenes. Furthermore, utilizing self-supervision methods for improving generalization gains much interest (Agarwal et al., 2021; Zhang et al., 2021; Hansen & Wang, 2020; Hansen et al., 2021a; Sun et al., 2019; Wang et al., 2021). Some researchers attempt to improve the generalization capability of RL agent via symbolic rules or structures (Zambaldi et al., 2018; Jiang & Luo, 2019; Cao et al., 2022) There are also many other interesting perspectives that encourage generalization of visual policies. Notably, Yuan et al. (2022) leverage pre-trained image encoder for universal representations in the training and testing environments; Hansen et al. (2021b) uncover the pitfalls of data augmentation in deep Q-learning, and mitigate them by enforcing regularization between the augmented and unaugmented data; Bertoin et al. (2022) mask the input image pixel-by-pixel based on the calculated attribution to extract and highlight the important regions (similar to noise handling methods like (Grooten et al., 2023)), and regularize the value function to ensure the consistency between the Q-values of input and masked images. Li et al. (2023) find that the generalization performance of the agent can be improved with some proper normalization techniques.

In spite of the huge success of the aforementioned methods in generalizing to previously unseen testing environments, as far as we can tell, none of them provide explanations of why their method improves generalization performance from a theoretical perspective. In this paper, we bridge this gap via providing theoretical bounds on the generalization gap in visual RL. Prior theoretical results of generalization in RL mainly lie in bandits (Agarwal et al., 2014; Slivkins, 2011; Jaksch et al., 2008; Li et al., 2022; Agrawal & Goyal, 2012), and most of them analyze the regret, i.e., the deviation between the expected value and the optimal return. However, studies on the generalization gap in general RL settings are few. Le Lan et al. (2022) theoretically characterize how and when the state representation generalizes based on the notion of effective dimension. Bertrán et al. (2020) treat the dynamics of the training level as instances and analyze the generalization bound of the value gap between the training and testing environments based on the training instances. The most relevant to our work is (Wang et al., 2019), which unpacks the generalization error of on-policy RL under reparameterization. However, their results do not apply to the existence of an encoder in the agent and distractors during testing. Our results also clearly anatomize how to reduce the generalization gap in visual RL.

Reparameterization in RL. The reparameterization trick has been widely adopted in deep neural networks, for the benefit of optimization efficiency (Kingma et al., 2015; Jang et al., 2017; Maddison et al., 2016; Salimans & Kingma, 2016; Mostafa & Wang, 2019; Kingma & Welling, 2013). A distribution can be reparameterized (Kingma & Welling, 2013) if (1) it has a tractable inverse CDF (cumulative distribution function); (2) it is a composition of reparameterizable distributions (e.g., Gaussian distribution); (3) we combine partial reparameterization with score function estimators (Ruiz et al., 2016; Bauer & Mnih, 2021). In the context of RL, reparameterization trick is also very popular. For example, PGPE (Sehnke et al., 2008) and SAC-style maximum entropy RL algorithms (Haarnoja et al., 2018a, 2018b) perform policy reparameterization; EPG (Ciosek & Whiteson, 2020) reparameterizes both the policy and the critic; SVG (Heess et al., 2015), instead, reparameterizes the policy and the environmental transition probability. Other interesting attempts include reparameterizing the action space (Wei et al., 2018; Bahl et al., 2020; Yang et al., 2022), the weight vectors in the neural network (Salimans & Kingma, 2016), a distribution over sample space via the learned quantile function (Dabney et al., 2018), etc. In this work, we leverage the reparameterization trick for quantifying the generalization gap in visual RL. Despite that some prior work (Heess et al., 2015; Wang et al., 2019) explore the reparameterization trick in RL, as far as we can tell, we are the first that apply the reparameterization trick on the transition function in the context of visual RL (where distractors exist in the testing environment) to facilitate theoretical derivations of generalization bounds.

3. Preliminaries

Reinforcement Learning. Reinforcement learning (RL) deals with sequential decision-making problems, and it can be specified by a Markov Decision Process (MDP) $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, r, p, p_0, \gamma\}$. $\mathcal{S}$ and $\mathcal{A}$ are state space and action space, respectively, $p(s, a, s') : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto [0, 1]$ is the transition probability from s to s' after taking action a, $p_0(s) : \mathcal{S} \mapsto [0, 1]$ is the initial state distribution, $r(s, a) : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ is the scalar reward function, and $\gamma \in [0, 1)$ is the discount factor. The policy $\pi_\theta(s)$ parameterized by θ is a mapping from state space to action space. We denote the policy class as Π, then we have $\pi_\theta(s) \in \Pi$. The goal of reinforcement learning is to find a policy that can maximize the expected long-term discounted return $J(\theta) = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)]$.

Since in practice, it is usually infeasible for infinite interactions, we consider episodic MDPs with a finite horizon in this work. We denote the length of the episode (i.e., the horizon) as $T+1$ and the corresponding trajectory as τ, i.e., $\tau = \{s_0, s_1, \dots, s_T\}$. Denote the joint distribution of the trajectories in an episode $\tau = \{s_0, \dots, s_T\}$ as $\mathcal{D}_{\pi,p,p_0}$, which is jointly determined by the transition probability p, initial state distribution p_0, and the learned policy π. For simplicity, we assume both p and p_0 are fixed. Then, $\mathcal{D}_{\pi,p,p_0}$ degenerates into $\mathcal{D}_\pi$. Under finite horizon, our goal is $\max_{\pi\in\Pi}\mathbb{E}_{\tau\sim\mathcal{D}_\pi}[J(\tau;\theta)] = \mathbb{E}_{\tau\sim\mathcal{D}_\pi}[\sum_{t=0}^{T}\gamma^t r(s_t,\pi(s_t))]$.

Generalization Gap. Generalization is a crucial problem and has been widely studied in the context of supervised learning (Motiian et al., 2017; Li et al., 2018; Zhou et al., 2021; Shalev-Shwartz & Ben-David, 2014). In supervised learning, we usually have access to a dataset $\mathcal{D}$ containing i.i.d. (independent and identically distributed) samples $\{(x_i, y_i)\}_{i=1}^n$. The generalization gap in this scenario is $\|\mathbb{E}[l(k,x,y)] - \frac{1}{n}\sum_{i=1}^n l(k,x_i,y_i)\|_2^2$, where k is the learned prediction function, l is the user-defined loss function. The generalization gap here measures the deviation between the expected loss and the empirical average loss. Following a similar formulation, the generalization gap in reinforcement learning can be defined as: $\|\mathbb{E}_{\tau\sim\mathcal{D}'_{\hat{\pi}}}[J(\tau)] - \frac{1}{n}\sum_{i=1}^n J(\tau_i)\|_2^2$, where $\mathcal{D}'_{\hat{\pi}}$ is the state sequence distribution in the testing environment, $\hat{\pi} = \arg\max_{\pi\in\Pi,\tau_i\in\mathcal{D}_\pi}\frac{1}{n}\sum_{i=1}^n J(\tau_i)$, n is the number of training episodes. It is difficult to quantify the generalization gap in RL due to the fact that the underlying sample distribution in the training environment $\mathcal{D}_\pi$ changes as the policy evolves, whereas the sample distribution in the supervised learning is kept fixed. Meanwhile, $\mathcal{D}'_{\hat{\pi}}$ may differ from the sample distribution in the training environment since there may exist a distribution shift of the transition dynamics and initial state distribution between the training and testing environments.

Visual RL Setting. In this paper, we consider visual RL where the agent receives image observations and executes actions based on them. It is generally formulated by a Partial Observable Markov Decision Process (POMDP) $\mathcal{M} = \{\mathcal{S}, \mathcal{O}, \mathcal{A}, r, p, p_0, \gamma\}$, where $\mathcal{O}$ is the observation space. We assume direct access to $\mathcal{S}$, while our analysis also applies when involving $\mathcal{O}$. This is valid since under the setting we consider, generalizing to environments with distractors, $\mathcal{O} = \mathcal{S}$. We denote the visual input at timestep t as s_t, which can be high-dimensional. In visual RL, it is a common practice that we adopt an encoder $\phi(\cdot) : \mathcal{S} \mapsto \Phi$ for extracting knowledge from complex image inputs, Φ is the representation space. The underlying RL algorithm accepts the representations outputted by the encoder and optimizes itself accordingly. The reward function gives $r(s, \pi(\phi(s)))$ and the policy is $\pi(\phi(s))$. Therefore, the randomness in the policy is deeply influenced by the encoder. During testing, we assume there exists the distractor $f(\cdot) \in \mathcal{F}$ that transforms the vanilla image observation into a new image, where $\mathcal{F}$ is the function class that contains all possible distractors during deployment. We name $f(\cdot)$ as the *transpose function* (or *distraction function*), which can take an arbitrary form and can even have no explicit expression. However, we do not allow it to modify the physical structure of the target agent, i.e., the color, camera pose, or background of the agent can be changed. After adding the distractor, the state space is augmented, and we assume that the reward function and the policy are well-defined in this augmented state space. This can also be formulated as the following setting: we have N MDPs, $\{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_N\}$ that share the same structure, while we only have access to $\mathcal{M}_1$ during training and aim at learning a policy that can achieve good performance in other MDPs in a zero-shot manner. Without loss of generality, we present the detailed objective function in the testing and training environment below,

$$\text{Testing environment objective :} \qquad \mathbb{E}_{\tau\sim\mathcal{D}'_{\hat{\pi}}}[J(\overbrace{\phi(\underbrace{f(\tau)}_{\text{transposed trajectory}})}^{\text{representation}})],$$

$$\text{Training environment objective :} \qquad \mathbb{E}_{\tau\sim\mathcal{D}_{\pi}}[J(\overbrace{\phi(\tau)}^{\text{representation}})].$$

Regardless of whether there exist distractors during deployment, the reward is decided by the action at state s, hence the objective function in the testing environment can be expressed as

$$\mathbb{E}_{\tau\sim\mathcal{D}'_{\hat{\pi}}}[J(\phi(f(\tau)))] = \mathbb{E}_{\tau\sim\mathcal{D}'_{\hat{\pi}}}\left[\sum_{t=0}^{T}\gamma^t r(s_t, \pi(\phi(f(s_t))))\right], \tag{1}$$

and similarly, we have the objective function in the training environment,

$$\mathbb{E}_{\tau\sim\mathcal{D}_{\hat{\pi}}}[J(\phi(\tau))] = \mathbb{E}_{\tau\sim\mathcal{D}_{\hat{\pi}}}\left[\sum_{t=0}^{T}\gamma^t r(s_t, \pi(\phi(s_t)))\right]. \tag{2}$$

Throughout this paper, we denote $\|\cdot\|$ as the L_2-norm.

4. Reparameterizable Visual RL

As we discuss in Section 3, it is hard to quantify the generalization gap in RL since the policy can evolve and different trajectories can be gathered in $\mathcal{D}_\pi$, resulting in a changing sample distribution $\mathcal{D}_\pi$. **It incurs challenges when we directly apply the theoretical results from supervised learning** because the theoretical results from supervised learning require a fixed sample distribution (the loss function can change) while in RL, the reward function is fixed but the sample distribution is not. It then necessities to decouple the randomness of the policy from the expected return. We resort to the reparameterization tool to address this issue. The goal of reparameterization is to find a way to recast a statistical expression in a different way while preserving its meaning. By reparameterizing the transition dynamics, we can represent the randomness of the sample distribution with a random variable. We show in Figure 1 the comparison of how state transfers before and after reparameterization. We assume that both the transition dynamics and the state initialization process can be reparameterized, then by using the reparameterization trick (Jang et al., 2017; Kingma et al., 2015; Figurnov et al., 2018; Xu et al., 2018), we can rewrite the objective function in the training environment as follows:

$$\mathbb{E}_{\tau\sim\mathcal{D}_{\pi}}[J(\phi(\tau))] = \mathbb{E}_{\xi\sim q(\xi)}\left[J(\phi(\tau(g(\xi;\pi_\theta))))\right], \tag{3}$$

where g is a deterministic function of parameter θ, $q(\xi)$ is the distribution of the random variable ξ. Since we do not care much about the form of g, we *absorb* g into τ (as SVG (Heess et al., 2015) does), yielding $\tau(\xi;\pi_\theta)$. This is valid because g shares the parameter with the policy, and the underlying meaning of s_t does not change by doing so. Observing Equation 3, we find that now the objective function no longer depends on the sample distribution

$\mathcal{D}_\pi$ and the policy π. That is, we isolate the randomness of the policy π from the expected return, and the policy π can now only affect the reward signal $r(s_t, \pi(\phi(s_t)))$ through the representation $\phi(s_t)$. Note that the state is transitioned with the reparametrized transition dynamics $\mathcal{T}$ (if the state is transitioned with the original dynamics function, then the resulting distribution $\mathcal{D}_\pi$ still relies on the policy). The introduced random variable ξ can be sampled before the start of the episode and the distribution $q(\xi)$ does not evolve. Then, we could leverage the conventional generalization theory to derive the generalization gap.

It is interesting to note here that if we assume the states in the MDP are discrete and $p_0(s), p(s,a)$ are multinomial distributions, then we can reparameterize it with Gumbel distribution (Gumbel, 1954), which are extensively adopted and studied in (Jang et al., 2017; Maddison et al., 2016; Potapczynski et al., 2019; Huijben et al., 2021; Lorberbom et al., 2021; Joo et al., 2020). Denote $\mathcal{G}^{|\mathcal{S}|}$ as the $|\mathcal{S}|$-dimensional standard Gumbel distribution, then Gumbel random variable $\xi_0, \xi_1, \dots, \xi_T$ can be sampled from $\mathcal{G}^{|\mathcal{S}|}$ (i.e., $\xi_i \sim \exp(-\xi - \exp(-\xi))$), and the sampling procedure in the MDP can be acquired by utilizing the Gumbel-max trick (Huijben et al., 2021; Wang et al., 2019; Jang et al., 2017; Oberst & Sontag, 2019):

$$s_{t+1} = \arg\max(\xi_t + \log p(s_t, \pi(s_t))), t = 0, 1, \dots, T. \tag{4}$$

There is also an interesting connection between $\arg\max$ operator and Gumbel-softmax as illustrated in (Jang et al., 2017; Maddison et al., 2016; Gumbel, 1954), if some relaxations are adopted for the state space (e.g., one-hot).

Algorithm 1 Reparameterizable Visual RL

```
Sample $\xi_0, \xi_1, \dots, \xi_T$
Get $s_0 = \mathcal{I}(\xi_0)$
Initialize $R = 0$
Set encoder $\phi(\cdot)$, policy $\pi(\cdot)$
for $t = 0$ to $T$ do
    $R = R + \gamma^t r(s_t, \pi(\phi(s_t)))$
    $s_{t+1} = \mathcal{T}(s_t, \pi(\phi(s_t)), \xi_t)$
end for
```

We denote $\mathcal{T}(s, \pi(s)) = p(s, \pi(s), s')$ as the state transition probability of the system, and $\mathcal{I} : \Xi \mapsto \mathcal{S}$ is the initialization function, where Ξ is the space of the random variable ξs. We then formally present the pseudo code of reparameterizable visual RL in Algorithm 1, where we reparameterize the *transition dynamics* of the system to fulfill Equation 3. Notably, we do not require the dimension of ξs to be the same as state s in Algorithm 1, and the random variables $\xi_0, \xi_1, \dots, \xi_T$ can be sampled from different distributions. The initialization function takes ξ_0 as input to produce s_0, and the transition probability $\mathcal{T}(s_t, \pi(\phi(s_t)), \xi_t)$ takes the current state s_t, the action induced by the policy $\pi(\phi(s_t))$, and ξ_t to produce next state $s_{t+1}, \forall\, t$.

Note that the random variables $\xi_0, \xi_1, \dots, \xi_T$ can be drawn from some distributions before the episode starts, hence isolating the randomness of the policy. The above formulation also applies when the policy evolves during training, since the trajectory can be decided deterministically by executing reparameterized transition function $\mathcal{T}(s_t, \pi(\phi(s_t)), \xi_t)$ repeatedly. Intuitively, the randomness from the encoder is also decoupled, and its influence can

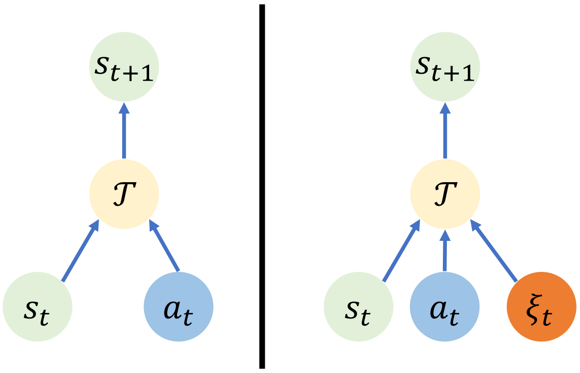


Figure 1: Comparison of state transition before (left) and after (right) reparameterization.

be included in that of the policy. Importantly, we do not require the encoder parameters to be fixed across the budget of n episodes.

5. Theoretical Analysis on the Generalization Error

In this section, we formally present our theoretical results on the generalization gap in visual RL, with the aid of the reparameterization tool. First, we impose some Lipschitz assumptions on the transition probability, the policy, and the reward function, which are fundamental for the following analysis. Most of these assumptions are reasonable and can be satisfied in practice. We discuss the rationality of these assumptions in Section 6.1.

Assumption 1. *The transition dynamics $\mathcal{T}(s,a,\xi):\mathcal{S}\times\mathcal{A}\times\Xi\mapsto\mathcal{S}$ is L_{t_1}-Lipschitz in terms of state s and L_{t_2}-Lipschitz in terms of action a, i.e., $\forall s,s',a,a',\xi$,*

$$\|\mathcal{T}(s,a,\xi)-\mathcal{T}(s',a,\xi)\|\le L_{t_1}\|s-s'\|,$$
$$\|\mathcal{T}(s,a,\xi)-\mathcal{T}(s,a',\xi)\|\le L_{t_2}\|a-a'\|.$$

Assumption 2. *The policy $\pi(\phi;\theta)$ is L_{π_1}-Lipschitz in terms of the variable ϕ and L_{π_2}-Lipschitz in terms of the parameter θ, i.e., $\forall\phi,\phi',\theta,\theta'$,*

$$\|\pi(\phi;\theta)-\pi(\phi';\theta)\|\le L_{\pi_1}\|\phi-\phi'\|$$
$$\|\pi(\phi;\theta)-\pi(\phi;\theta')\|\le L_{\pi_2}\|\theta-\theta'\|.$$

Assumption 3. *The reward function $r(s,a)$ is bounded, i.e., $\forall s,a$, $|r(s,a)|\le r_{\max}$, and is L_{r_1}-Lipschitz in terms of state s, L_{r_2}-Lipschitz in terms of action a, i.e., $\forall s,s'$ and $\forall\, a,a'$,*

$$|r(s,a)-r(s',a)|\le L_{r_1}\|s-s'\|,$$
$$|r(s,a)-r(s,a')|\le L_{r_2}\|a-a'\|$$

Based on these assumptions, we first can derive how the output of the policy changes using different inputs and parameters.

Lemma 1 (Policy deviation)**.** *Assume that Assumption 2 hold, denote $\phi(\cdot)$ as the encoder. Then under the transition $\mathcal{T}(s, a, \xi)$, at step $t \in \{0, 1, \ldots, T\}$ in an episode of length $T + 1$, we have*

$$\|\pi(\phi(s'_t); \theta') - \pi(\phi(s_t); \theta)\| \leq L_{\pi_1}\|\phi(s'_t) - \phi(s_t)\| + L_{\pi_2}\|\theta' - \theta\|, \forall s_t, s'_t \in \mathcal{S}, \forall \theta, \theta'. \tag{5}$$

Proof. This lemma is a direct product of Assumption 2.

$$\begin{aligned}
&\|\pi(\phi(s'_t); \theta') - \pi(\phi(s_t); \theta)\| \\
&= \|\pi(\phi(s'_t); \theta') - \pi(\phi(s'_t); \theta) + \pi(\phi(s'_t); \theta) - \pi(\phi(s_t); \theta)\| \\
&\leq \|\pi(\phi(s'_t); \theta') - \pi(\phi(s'_t); \theta)\| + \|\pi(\phi(s'_t); \theta) - \pi(\phi(s_t); \theta)\| \\
&\leq L_{\pi_1}\|\phi(s'_t) - \phi(s_t)\| + L_{\pi_2}\|\theta - \theta'\|.
\end{aligned}$$

□

Note that the triangle inequality is used in this proof, and it is also widely used for the following proofs. Furthermore, we can derive the relationship between the deviations of two states at timestep t and timestep $t - 1$ as depicted in the following lemma.

Lemma 2 (State deviation)**.** *Assume that Assumption 1 and 2 hold, denote $\phi(\cdot)$ as the encoder, then under the transition $\mathcal{T}(s, a, \xi)$ and the policy $\pi(\cdot; \theta)$, at step t in an episode of length $T + 1$, we have $\forall s_t, s'_t \in \mathcal{S}, t \in \{1, \ldots, T\}$,*

$$\|s_t - s'_t\| \leq L_{t_1}\|s'_{t-1} - s_{t-1}\| + L_{t_2}L_{\pi_1}\|\phi(s'_{t-1}) - \phi(s_{t-1})\|. \tag{6}$$

Proof. This lemma is also quite straightforward. By using the reparameterized visual RL framework, we have

$$\begin{aligned}
\|s_t - s'_t\| &= \|\mathcal{T}(s_{t-1}, \pi(\phi(s_{t-1}); \theta), \xi_{t-1}) - \mathcal{T}(s'_{t-1}, \pi(\phi(s'_{t-1}); \theta), \xi_{t-1})\| \\
&= \|\mathcal{T}(s_{t-1}, \pi(\phi(s_{t-1}); \theta), \xi_{t-1}) - \mathcal{T}(s'_{t-1}, \pi(\phi(s_{t-1}); \theta), \xi_{t-1}) \\
&\qquad + \mathcal{T}(s'_{t-1}, \pi(\phi(s_{t-1}); \theta), \xi_{t-1}) - \mathcal{T}(s'_{t-1}, \pi(\phi(s'_{t-1}); \theta), \xi_{t-1})\| \\
&\leq \|\mathcal{T}(s_{t-1}, \pi(\phi(s_{t-1}); \theta), \xi_{t-1}) - \mathcal{T}(s'_{t-1}, \pi(\phi(s_{t-1}); \theta), \xi_{t-1})\| \\
&\qquad + \|\mathcal{T}(s'_{t-1}, \pi(\phi(s_{t-1}); \theta), \xi_{t-1}) - \mathcal{T}(s'_{t-1}, \pi(\phi(s'_{t-1}); \theta), \xi_{t-1})\| \\
&\leq L_{t_1}\|s_{t-1} - s'_{t-1}\| + L_{t_2}\|\pi(\phi(s_{t-1}); \theta) - \pi(\phi(s'_{t-1}); \theta)\| \\
&\leq L_{t_1}\|s_{t-1} - s'_{t-1}\| + L_{t_2}L_{\pi_1}\|\phi(s_{t-1}) - \phi(s'_{t-1})\|,
\end{aligned}$$

where we use the Lemma 1 by setting $\theta = \theta'$. □

Similarly, we can derive how reward function changes with different input states. The resulting lemma can be found below.

Lemma 3 (Reward deviation)**.** *Assume that Assumption 2, 3 hold, denote $\phi(\cdot)$ as the encoder, then under the transition $\mathcal{T}(s, a, \xi)$ and the policy $\pi(\cdot; \theta)$, at step t in an episode of length $T + 1$, we have $\forall s_t, s'_t \in \mathcal{S}$,*

$$|r(s_t, \pi(\phi(s_t))) - r(s'_t, \pi(\phi(s'_t)))| \leq L_{r_1}\|s_t - s'_t\| + L_{r_2}L_{\pi_1}\|\phi(s'_t) - \phi(s_t)\|. \tag{7}$$

Proof. It is easy to find that,

$$
\begin{aligned}
&|r(s_t, \pi(\phi(s_t))) - r(s'_t, \pi(\phi(s'_t)))| \\
&= |r(s_t, \pi(\phi(s_t))) - r(s'_t, \pi(\phi(s_t))) + r(s'_t, \pi(\phi(s_t))) - r(s'_t, \pi(\phi(s'_t)))| \\
&\leq |r(s_t, \pi(\phi(s_t))) - r(s'_t, \pi(\phi(s_t)))| + |r(s'_t, \pi(\phi(s_t))) - r(s'_t, \pi(\phi(s'_t)))| \\
&\leq L_{r_1}\|s_t - s'_t\| + L_{r_2}\|\pi(\phi(s_t)) - \pi(\phi(s'_t))\| \\
&\leq L_{r_1}\|s_t - s'_t\| + L_{r_2}L_{\pi_1}\|\phi(s_t) - \phi(s'_t)\|,
\end{aligned}
$$

where we also use the Lemma 1 by setting $\theta = \theta'$. □

Lemma 1, 2 and 3 indicate that *representation matters* in reparameterized visual RL, as state deviation, policy deviation, and reward deviation are all highly correlated with the representation deviation. For example, if s_t is close to s'_t, their representation distance also ought to be small such that the agent executes similar actions upon them. These lemmas pave the way for Theorem 1, where we show that given a (fixed) policy π, and the trajectories $\tau(\xi)$, the performance difference between adding the distractor (i.e., the transpose function) or without the distractor is determined by the difference between the transposed state $f(s_t)$ and the original state $s_t, \forall\, t \in [0, T]$.

Theorem 1 (Fixed policy shift error). *Assume that Assumption 1, 2, 3 hold and the encoder $\phi(\cdot)$ is L_ϕ-Lipschitz, i.e., $\forall s, s', \|\phi(s) - \phi(s')\| \leq L_\phi\|s - s'\|$, then we have*

$$
\|\mathbb{E}_\xi[J(\phi(f(\tau(\xi;\pi,\mathcal{T},\mathcal{I}))))] - \mathbb{E}_\xi[J(\phi(\tau(\xi;\pi,\mathcal{T},\mathcal{I})))]\| \leq L_{r_2}L_{\pi_1}L_\phi \sum_{t=0}^{T} \gamma^t \mathbb{E}_\xi\left[\|f(s_t) - s_t\|\right]. \tag{8}
$$

Proof. Since the transition dynamics, initialization function, and policy are kept unchanged, we have

$$
\begin{aligned}
&\|\mathbb{E}_\xi[J(\phi(f(\tau(\xi;\pi,\mathcal{T},\mathcal{I}))))] - \mathbb{E}_\xi[J(\phi(\tau(\xi;\pi,\mathcal{T},\mathcal{I})))]\| \\
&\qquad = \left\| \int_\xi [J(\phi(f(\tau(\xi;\pi,\mathcal{T},\mathcal{I}))))]d\xi - \int_\xi [J(\phi(\tau(\xi;\pi,\mathcal{T},\mathcal{I})))]d\xi \right\| \\
&\qquad \leq \int_\xi \|[J(\phi(f(\tau(\xi;\pi,\mathcal{T},\mathcal{I}))))] - [J(\phi(\tau(\xi;\pi,\mathcal{T},\mathcal{I})))]\|\, d\xi.
\end{aligned}
$$

Note that the only difference between the two trajectories lies in whether it involves the transpose function. Therefore, we have

$$\begin{aligned}
&\|[J(\phi(f(\tau(\xi;\pi,\mathcal{T},\mathcal{I}))))] - [J(\phi(\tau(\xi;\pi,\mathcal{T},\mathcal{I})))]\| \\
&= \left\| \sum_{t=0}^{T} \gamma^t r(s_t, \pi(\phi(f(s_t)))) - \sum_{t=0}^{T} \gamma^t r(s_t, \pi(\phi(s_t))) \right\| \\
&\leq \sum_{t=0}^{T} \gamma^t \left\| r(s_t, \pi(\phi(f(s_t)))) - r(s_t, \pi(\phi(s_t))) \right\| \\
&\leq \sum_{t=0}^{T} \gamma^t L_{r_2} \|\pi(\phi(f(s_t))) - \pi(\phi(s_t))\| \\
&\leq \sum_{t=0}^{T} \gamma^t L_{r_2} L_{\pi_1} \|\phi(f(s_t)) - \phi(s_t)\| \\
&\leq L_{r_2} L_{\pi_1} L_\phi \sum_{t=0}^{T} \gamma^t \|f(s_t) - s_t\|.
\end{aligned}$$

Then, it is natural to derive that

$$\begin{aligned}
&\|\mathbb{E}_\xi[J(\phi(f(\tau(\xi;\pi,\mathcal{T},\mathcal{I}))))] - \mathbb{E}_\xi[J(\phi(\tau(\xi;\pi,\mathcal{T},\mathcal{I})))]\| \\
&\qquad \leq \int_\xi \|[J(\phi(f(\tau(\xi;\pi,\mathcal{T},\mathcal{I}))))] - [J(\phi(\tau(\xi;\pi,\mathcal{T},\mathcal{I})))]\| \, d\xi \\
&\qquad \leq L_{r_2} L_{\pi_1} L_\phi \int_\xi \sum_{t=0}^{T} \gamma^t \|f(s_t) - s_t\| d\xi = L_{r_2} L_{\pi_1} L_\phi \sum_{t=0}^{T} \gamma^t \mathbb{E}_\xi \left[\|f(s_t) - s_t\| \right].
\end{aligned}$$

□

Furthermore, if the transpose function $f(s)$ represents a linear noise distractor, the above bound can be simplified, as illustrated in the following corollary. Note that it is common that the noise is involved in a linear way upon the visual observations in real-world applications (Stone et al., 2021).

Corollary 1. *If the transpose function satisfies $f(s_t) = s_t + \epsilon_t$ where ϵ_t is time-dependent bounded transpose term, i.e., $\forall t, \|\epsilon_t\| \leq \eta < \infty$, then based on the assumptions in Theorem 1, we have:*

$$\|\mathbb{E}_\xi[J(\phi(f(\tau(\xi;\pi,\mathcal{T},\mathcal{I}))))] - \mathbb{E}_\xi[J(\phi(\tau(\xi;\pi,\mathcal{T},\mathcal{I})))]\| \leq L_{r_2} L_{\pi_1} L_\phi \eta \frac{1-\gamma^{T+1}}{1-\gamma}. \tag{9}$$

Proof. The proof is quite straightforward. With the conclusion in Theorem 1, we have

$$\|\mathbb{E}_\xi[J(\phi(f(\tau(\xi;\pi,\mathcal{T},\mathcal{I}))))] - \mathbb{E}_\xi[J(\phi(\tau(\xi;\pi,\mathcal{T},\mathcal{I})))]\| \leq L_{r_2} L_{\pi_1} L_\phi \sum_{t=0}^{T} \gamma^t \mathbb{E}_\xi \left[\|f(s_t) - s_t\| \right].$$

Then by using that $f(s_t) = s_t + \epsilon_t$, and $\|\epsilon_t\| \leq \eta, \forall t$, we have

$$\begin{aligned}\|\mathbb{E}_\xi[J(\phi(f(\tau(\xi;\pi,\mathcal{T},\mathcal{I}))))] - \mathbb{E}_\xi[J(\phi(\tau(\xi;\pi,\mathcal{T},\mathcal{I})))]\| &\leq L_{r_2}L_{\pi_1}L_\phi \sum_{t=0}^{T} \gamma^t \mathbb{E}_\xi\left[\|f(s_t) - s_t\|\right] \\ &= L_{r_2}L_{\pi_1}L_\phi \sum_{t=0}^{T} \gamma^t \mathbb{E}_\xi\left[\|\epsilon_t\|\right] \\ &\leq L_{r_2}L_{\pi_1}L_\phi \sum_{t=0}^{T} \gamma^t \mathbb{E}_\xi\left[\eta\right] \\ &= L_{r_2}L_{\pi_1}L_\phi \eta \frac{1-\gamma^{T+1}}{1-\gamma}.\end{aligned}$$

This concludes the proof. □

The above corollary requires that the noises are linear, which might be restrictive in practice. We then loose this requirement and even allow that the transpose function does not necessarily to be deterministic. Instead, the transposed state is sampled from an unknown distribution. We do not make any assumption on this distribution. For the benefit of theoretical derivation, we assume that the average effect of the distraction (i.e., transpose function f) is bounded, and the variance of $f(s)$ gives σ^2. Then, we can still derive a similar bound as Corollary 1 and show that the policy performance shift after adding $f(\cdot)$ can be large even on the identical trajectory.

Corollary 2. *If the transpose function $f(s_t)$ is sampled from a distribution and satisfies $\|\mathbb{E}[f(s_t)] - s_t\| \leq \eta < \infty$, i.e., the average effect of the distraction is bounded, and suppose that the variance of $f(s)$ is σ^2 which is fixed, then based on the assumptions in Theorem 1, we have with probability at least $1-\delta$:*

$$\|\mathbb{E}_\xi[J(\phi(f(\tau(\xi;\pi,\mathcal{T},\mathcal{I}))))] - \mathbb{E}_\xi[J(\phi(\tau(\xi;\pi,\mathcal{T},\mathcal{I})))]\| \leq L_{r_2}L_{\pi_1}L_\phi \frac{1-\gamma^{T+1}}{1-\gamma}\left(\eta + \sqrt{\frac{\sigma^2}{\delta}}\right). \tag{10}$$

Proof. Based on the conclusion in Theorem 1, we have

$$\begin{aligned}&\|\mathbb{E}_\xi[J(\phi(f(\tau(\xi;\pi,\mathcal{T},\mathcal{I}))))] - \mathbb{E}_\xi[J(\phi(\tau(\xi;\pi,\mathcal{T},\mathcal{I})))]\| \\ &\qquad\leq L_{r_2}L_{\pi_1}L_\phi \sum_{t=0}^{T} \gamma^t \mathbb{E}_\xi\left[\|f(s_t) - s_t\|\right] \\ &\qquad\leq L_{r_2}L_{\pi_1}L_\phi \sum_{t=0}^{T} \gamma^t \mathbb{E}_\xi\left[\|f(s_t) - \mathbb{E}[f(s_t)]\| + \|\mathbb{E}[f(s_t)] - s_t\|\right] \\ &\qquad\leq L_{r_2}L_{\pi_1}L_\phi \sum_{t=0}^{T} \gamma^t \mathbb{E}_\xi\left[\eta + \|f(s_t) - \mathbb{E}[f(s_t)]\|\right].\end{aligned}$$

Then by Chebyshev's inequality, we have with probability at least $1-\delta$, $\|f(s_t)-\mathbb{E}[f(s_t)]\| \leq \sqrt{\frac{\sigma^2}{\delta}}$. It is then easy to find that,

$$\begin{aligned}
&\|\mathbb{E}_\xi[J(\phi(f(\tau(\xi;\pi,\mathcal{T},\mathcal{I}))))] - \mathbb{E}_\xi[J(\phi(\tau(\xi;\pi,\mathcal{T},\mathcal{I})))]\| \\
&\qquad \leq L_{r_2} L_{\pi_1} L_\phi \sum_{t=0}^{T} \gamma^t \mathbb{E}_\xi \left[\eta + \|f(s_t) - \mathbb{E}[f(s_t)]\|\right] \\
&\qquad \leq L_{r_2} L_{\pi_1} L_\phi \sum_{t=0}^{T} \gamma^t \mathbb{E}_\xi \left[\eta + \sqrt{\frac{\sigma^2}{\delta}}\right] \\
&\qquad \leq L_{r_2} L_{\pi_1} L_\phi \frac{1-\gamma^{T+1}}{1-\gamma} \left(\eta + \sqrt{\frac{\sigma^2}{\delta}}\right).
\end{aligned}$$

This concludes the proof. □

Theorem 1 demonstrate that given a fixed policy, the return deviation before and after adding the distractor in the trajectory is determined by the deviation between the state and the transposed state. Corollary 1 and Corollary 2 indicate that the policy performance shift after involving the transpose function $f(\cdot)$ can be large (e.g., η is large) and the error accumulates with larger T, i.e., the upper bound becomes larger if T increases. In practice, the testing environment may differ from the training environment, e.g., the initial state distribution, the transition probability. We allow a slight mismatch between the training and testing environments, as shown below.

Assumption 4. *The transition $\mathcal{T}$, initialization function $\mathcal{I}$ in the training environment and $\mathcal{T}',\mathcal{I}'$ in the testing environment satisfy: $\forall s,a,\xi, \|(\mathcal{T}-\mathcal{T}')(s,a,\xi)\| \leq \zeta$, $\|(\mathcal{I}-\mathcal{I}')(\xi)\| \leq \epsilon$.*

We then investigate the theoretical bound when involving the above assumption. It is challenging because the transition dynamics and the initialization function between the training and testing environments can be different. We need to derive how state under the initialization $\mathcal{I}$ and state under $\mathcal{I}'$ deviates each other at timestep t because It is necessary to get the bound for returns. We show the corresponding lemma below.

Lemma 4. *Assume that Assumptions 1, 2, 3 hold, the encoder $\phi(\cdot)$ is L_ϕ-Lipschitz, the initialization function $\mathcal{I}'$ in the testing environment satisfies $\forall\xi, \|(\mathcal{I}-\mathcal{I}')(\xi)\| \leq \epsilon$, and the transition dynamics is the same for the training and testing environments. We further assume that the testing environment has no distractors, then under policy $\pi(\cdot;\theta)$, we have*

$$\|s_t - s_t'\| \leq (L_{t_1} + L_{t_2} L_{\pi_1} L_\phi)^t \epsilon := \nu^t \epsilon, \forall s_t, s_t'. \tag{11}$$

Proof. With the framework of reparameterized visual RL, we have

$$\begin{aligned}\|s_t - s'_t\| &= \|\mathcal{T}(s_{t-1}, \pi(\phi(s_{t-1}); \theta), \xi_{t-1}) - \mathcal{T}(s'_{t-1}, \pi(\phi(s'_{t-1}); \theta), \xi_{t-1})\| \\ &\leq \|\mathcal{T}(s_{t-1}, \pi(\phi(s_{t-1}); \theta), \xi_{t-1}) - \mathcal{T}(s_{t-1}, \pi(\phi(s'_{t-1}); \theta), \xi_{t-1})\| \\ &\quad + \|\mathcal{T}(s_{t-1}, \pi(\phi(s'_{t-1}); \theta), \xi_{t-1}) - \mathcal{T}(s'_{t-1}, \pi(\phi(s'_{t-1}); \theta), \xi_{t-1})\| \\ &\leq L_{t_1}\|s_{t-1} - s'_{t-1}\| + L_{t_2}\|\pi(\phi(s_{t-1}); \theta) - \pi(\phi(s'_{t-1}); \theta)\| \\ &\leq L_{t_1}\|s_{t-1} - s'_{t-1}\| + L_{t_2}L_{\pi_1}\|\phi(s_{t-1}) - \phi(s'_{t-1})\| \\ &\leq L_{t_1}\|s_{t-1} - s'_{t-1}\| + L_{t_2}L_{\pi_1}L_{\phi}\|s_{t-1} - s'_{t-1}\| \\ &= (L_{t_1} + L_{t_2}L_{\pi_1}L_{\phi})\|s_{t-1} - s'_{t-1}\|.\end{aligned}$$

By denoting $\nu = L_{t_1} + L_{t_2}L_{\pi_1}L_{\phi}$, and doing iteration, we have

$$\|s_t - s'_t\| \leq \nu^t\|s_0 - s'_0\| = \nu^t\epsilon,$$

where the last equality is due to the fact that $\|s_0 - s'_0\| = \|(\mathcal{I} - \mathcal{I}')(\xi)\| \leq \epsilon$. □

We also need the relationship between s_t under training environment dynamics $\mathcal{T}$ and s'_t testing environment dynamics $\mathcal{T}'$. The resulting bound is shown below.

Lemma 5. *Assume that Assumptions 1, 2, 3 hold, the encoder $\phi(\cdot)$ is L_ϕ-Lipschitz, the transition dynamics $\mathcal{T}'$ in the testing environment satisfies $\forall s, a, \xi, \|(\mathcal{T} - \mathcal{T}')(\xi)\| \leq \zeta$, and the initialization function is the same for the training and testing environments. We further assume that the testing environment has no distractors. Suppose s comes from the training environment, and s' from the testing environment, then under policy $\pi(\cdot; \theta)$, we have*

$$\|s_t - s'_t\| \leq \zeta\frac{1 - \nu^t}{1 - \nu}, \tag{12}$$

where $\nu = L_{t_1} + L_{t_2}L_{\pi_1}L_{\phi}$.

Proof. Since s_t is from the training environment, and s'_t is from the testing environment, it is easy to find that

$$\begin{aligned}\|s_t - s'_t\| &= \|\mathcal{T}(s_{t-1}, \pi(\phi(s_{t-1}); \theta), \xi_{t-1}) - \mathcal{T}'(s'_{t-1}, \pi(\phi(s'_{t-1}); \theta), \xi_{t-1})\| \\ &\leq \|\mathcal{T}(s_{t-1}, \pi(\phi(s_{t-1}); \theta), \xi_{t-1}) - \mathcal{T}(s'_{t-1}, \pi(\phi(s'_{t-1}); \theta), \xi_{t-1})\| \\ &\quad \|\mathcal{T}(s'_{t-1}, \pi(\phi(s'_{t-1}); \theta), \xi_{t-1}) - \mathcal{T}'(s'_{t-1}, \pi(\phi(s'_{t-1}); \theta), \xi_{t-1})\| \\ &\leq \zeta + \|\mathcal{T}(s_{t-1}, \pi(\phi(s_{t-1}); \theta), \xi_{t-1}) - \mathcal{T}(s'_{t-1}, \pi(\phi(s'_{t-1}); \theta), \xi_{t-1})\|.\end{aligned}$$

By following the same procedure in the proof of Lemma 4, we have

$$\|\mathcal{T}(s_{t-1}, \pi(\phi(s_{t-1}); \theta), \xi_{t-1}) - \mathcal{T}(s'_{t-1}, \pi(\phi(s'_{t-1}); \theta), \xi_{t-1})\| \leq \nu\|s_{t-1} - s'_{t-1}\|.$$

Combining these results, we have

$$\|s_t - s'_t\| \leq \zeta + \nu\|s_{t-1} - s'_{t-1}\|.$$

By recursion, we have

$$\|s_t - s'_t\| \leq \zeta(1 + \nu + \ldots + \nu^{t-1}) + \nu^t\|s_0 - s'_0\| = \zeta\frac{1 - \nu^t}{1 - \nu}.$$

The last equality is due to the fact that the initialization function is identical for the training and testing environments, and hence $s_0 = s'_0$. □

Now, we are ready to bound the returns in the training environment (without distractor) and testing environment (with distractors). The theorem below tells us that the performance difference in the training and testing environments are jointly influenced by (1) the initialization difference ϵ, (2) transition dynamics difference ζ, and (3) the representation difference ϱ.

Theorem 2. *Assume that Assumptions 1, 2, 3, 4 hold. We further assume that (1) the encoder $\phi(\cdot)$ is L_ϕ-Lipschitz, i.e., $\forall s, s', \|\phi(s) - \phi(s')\| \leq L_\phi \|s - s'\|$, and (2) $\|\phi(f(s)) - \phi(s)\| \leq \varrho, \forall s$. $f(\cdot)$ is the transpose function in the testing environment. Denote $\nu = L_{t_1} + L_{t_2} L_{\pi_1} L_\phi, \lambda = L_{r_1} + L_{r_2} L_{\pi_1} L_\phi$, then we have,*

$$\| \underbrace{\mathbb{E}_\xi [J(\phi(f(\tau(\xi;\pi,\mathcal{T}',\mathcal{I}'))))]}_{\text{performance in the testing env}} - \underbrace{\mathbb{E}_\xi [J(\phi(\tau(\xi;\pi,\mathcal{T},\mathcal{I})))]}_{\text{performance in the training env}} \| \\ \leq \lambda\zeta \sum_{t=0}^{T} \gamma^t \frac{\nu^t - 1}{\nu - 1} + \lambda\epsilon \sum_{t=0}^{T} \gamma^t \nu^t + \frac{L_{r_2} L_{\pi_1} \varrho}{1-\gamma}(1 - \gamma^{T+1}).$$

Proof. We decompose the performance difference in the training and testing environments as follows:

$$\begin{aligned}
&\|\mathbb{E}_\xi [J(\phi(f(\tau(\xi;\pi,\mathcal{T}',\mathcal{I}'))))] - \mathbb{E}_\xi [J(\phi(\tau(\xi;\pi,\mathcal{T},\mathcal{I})))]\| \\
&\qquad \leq \underbrace{\|\mathbb{E}_\xi [J(\phi(f(\tau(\xi;\pi,\mathcal{T}',\mathcal{I}'))))] - \mathbb{E}_\xi [J(\phi(\tau(\xi;\pi,\mathcal{T}',\mathcal{I}')))]\|}_{\text{(I)}} \\
&\qquad + \underbrace{\|\mathbb{E}_\xi [J(\phi(\tau(\xi;\pi,\mathcal{T}',\mathcal{I}')))] - \mathbb{E}_\xi [J(\phi(\tau(\xi;\pi,\mathcal{T}',\mathcal{I})))]\|}_{\text{(II)}} \\
&\qquad + \underbrace{\|\mathbb{E}_\xi [J(\phi(\tau(\xi;\pi,\mathcal{T}',\mathcal{I})))] - \mathbb{E}_\xi [J(\phi(\tau(\xi;\pi,\mathcal{T},\mathcal{I})))]\|}_{\text{(III)}}.
\end{aligned}$$

We then bound each term separately. For term (I), we have

$$\begin{aligned}
\text{(I)} &= \|\mathbb{E}_\xi [J(\phi(f(\tau(\xi;\pi,\mathcal{T}',\mathcal{I}'))))] - \mathbb{E}_\xi [J(\phi(\tau(\xi;\pi,\mathcal{T}',\mathcal{I}')))]\| \\
&\leq \int_\xi \left\| \sum_{t=0}^{T} \gamma^t \left(r(s_t, \pi(\phi(f(s_t)))) - r(s_t, \pi(\phi(s_t))) \right) \right\| d\xi \\
&\leq \int_\xi \sum_{t=0}^{T} \gamma^t \left\| (r(s_t, \pi(\phi(f(s_t)))) - r(s_t, \pi(\phi(s_t)))) \right\| d\xi \\
&\leq \int_\xi \sum_{t=0}^{T} \gamma^t L_{r_2} \left\| \pi(\phi(f(s_t))) - \pi(\phi(s_t)) \right\| d\xi \\
&\leq \int_\xi \sum_{t=0}^{T} \gamma^t L_{r_2} L_{\pi_1} \left\| \phi(f(s_t)) - \phi(s_t) \right\| d\xi \\
&\leq \int_\xi \sum_{t=0}^{T} \gamma^t L_{r_2} L_{\pi_1} \varrho d\xi = L_{r_2} L_{\pi_1} \varrho \frac{1-\gamma^{T+1}}{1-\gamma}.
\end{aligned}$$

For term (II), we have

$$\begin{aligned}
\text{(II)} &= \|\mathbb{E}_\xi[J(\phi(\tau(\xi;\pi,\mathcal{T}',\mathcal{I}')))] - \mathbb{E}_\xi[J(\phi(\tau(\xi;\pi,\mathcal{T}',\mathcal{I})))]\| \\
&\le \int_\xi \left\| \sum_{t=0}^{T} \gamma^t \left(r(s_t, \pi(\phi(s_t);\theta)) - r(s_t', \pi(\phi(s_t');\theta)) \right) \right\| d\xi \\
&\le \int_\xi \sum_{t=0}^{T} \gamma^t \| r(s_t, \pi(\phi(s_t);\theta)) - r(s_t', \pi(\phi(s_t');\theta)) \| d\xi.
\end{aligned}$$

By using Lemma 3, and that the encoder is L_ϕ-Lipschitz, we have

$$\begin{aligned}
\text{(II)} &\le \int_\xi \sum_{t=0}^{T} \gamma^t \left(L_{r_1} \|s_t - s_t'\| + L_{r_2} L_{\pi_1} \|\phi(s_t) - \phi(s_t')\| \right) d\xi \\
&\le \int_\xi \sum_{t=0}^{T} \gamma^t \underbrace{(L_{r_1} + L_{r_2} L_{\pi_1} L_\phi)}_{:=\lambda} \|s_t - s_t'\| d\xi
\end{aligned}$$

Term (II) has the identical transition dynamics but different initialization functions, then by using Lemma 4, we have

$$\text{(II)} \le \lambda \int_\xi \sum_{t=0}^{T} \gamma^t \nu^t \epsilon d\xi = \lambda \epsilon \sum_{t=0}^{T} \gamma^t \nu^t.$$

For term (III), we follow the same way of bounding term (II) and have

$$\text{(III)} = \|\mathbb{E}_\xi[J(\phi(\tau(\xi;\pi,\mathcal{T}',\mathcal{I})))] - \mathbb{E}_\xi[J(\phi(\tau(\xi;\pi,\mathcal{T},\mathcal{I})))]\| \le \int_\xi \sum_{t=0}^{T} \gamma^t \lambda \|s_t - s_t'\| d\xi.$$

Term (III) has the same initialization function, but different transition dynamics. By using Lemma 5, we have

$$\text{(III)} \le \int_\xi \sum_{t=0}^{T} \gamma^t \lambda \|s_t - s_t'\| d\xi \le \int_\xi \sum_{t=0}^{T} \gamma^t \lambda \zeta \frac{1-\nu^t}{1-\nu} d\xi = \lambda \zeta \sum_{t=0}^{T} \gamma^t \frac{1-\nu^t}{1-\nu}.$$

Finally, Theorem 2 holds by combining the bounds of term (I), (II), and (III). □

Remark 1: By assuming $\|\phi(f(s)) - \phi(s)\| \le \varrho, \forall\, s$, we actually are enforcing a 'local' regularization on the encoder, i.e., the representation distance upon the captured state between the training environment s and the testing environment $f(s)$ ought to be bounded. This naturally leads to yet another requirement on the Lipschitz constant L_ϕ, i.e., it has to satisfy $L_\phi \le \frac{\varrho}{\max_{s\in\mathcal{S}} \|f(s)-s\|}$.

Remark 2: The above theorem does not involve the process of policy improvement (which is why we emphasize a fixed policy), and is simply the performance of *any* policy in the training and testing environments. However, the result is still meaningful as it meets the most common way of evaluating RL policies, i.e., evaluate the agent with the fixed policy after certain environmental steps.

We note that among the all critical factors, the initialization difference ϵ and the transition dynamics difference ζ are uncontrollable, and are strongly correlated with the physical systems. *It turns out that the most critical way of reducing the performance shift is to regularize the representation difference ϱ.* Before moving to our most critical conclusion, we need to present the formal generalization error of visual RL when the testing environment has distractors. Recall that the generalization error has the form of $\|\mathbb{E}_{\tau\sim\mathcal{D}'_{\hat{\pi}}}[J(\tau)] - \frac{1}{n}\sum_{i=1}^{n} J(\tau_i)\|_2^2$ (as illustrated in Section 3). To that end, we first present the following lemma, which is a direct product of classical learning theory (i.e., Theorem 3.3 in (Mohri et al., 2018)).

Lemma 6. *For reparameterized visual RL, if the rewards are bounded, $|r(s,a)| \le r_{\max}$, $\forall\, s, a, r_{\max} > 0$. Assume that the sampled peripheral random variables ξs are i.i.d. for each episode. Then with probability at least $1-\delta$, for any policy $\pi \in \Pi$, the following holds,*

$$\left\|\mathbb{E}_{\xi}\left[J\left(\phi\left(\tau\left(\xi;\pi,\mathcal{T},\mathcal{I}\right)\right)\right)\right] - \frac{1}{n}\sum_{i=1}^{n} J(\phi(\tau(\xi_i;\pi,\mathcal{T},\mathcal{I})))\right\| \le 2Rad(J_{\pi,\mathcal{T},\mathcal{I}}) + \mathcal{O}\left(r_{\max}\sqrt{\frac{\log(1/\delta)}{n}}\right),$$

where $Rad(J_{\pi,\mathcal{T},\mathcal{I}}) = \mathbb{E}_{\xi}\mathbb{E}_{\sigma}[\sup_{\pi}\frac{1}{n}\sum_{i=1}^{n}\sigma_i J(\phi(\tau(\xi_i;\pi,\mathcal{T},\mathcal{I})))]$ is the Rademacher complexity, σ_i is the Rademacher variable, n is the number of training episodes.

Proof. Please refer to the proof of Theorem 3.3 in (Mohri et al., 2018). □

The above bound holds uniformly for all policies $\pi \in \Pi$. Then, we formally present the generalization gap bound of the policy in the testing environment with distractors and the clean training environment.

Theorem 3 (Generalization error)**.** *Assume that the assumptions made in Theorem 2 hold and ξ_i are i.i.d.. Denote $f(\cdot)$ as the transpose function in the testing environment. Then with probability at least $1-\delta$, the generalization gap between the training environment and testing environment satisfies:*

$$\begin{aligned}&\left\|\mathbb{E}_{\xi}\left[J\left(\phi\left(f\left(\tau\left(\xi;\pi,\mathcal{T}',\mathcal{I}'\right)\right)\right)\right)\right] - \frac{1}{n}\sum_{i=1}^{n} J(\phi(\tau(\xi_i;\pi,\mathcal{T},\mathcal{I})))\right\| \\ &\le 2Rad(J_{\pi,\mathcal{T},\mathcal{I}}) + \lambda\zeta\sum_{t=0}^{T}\gamma^t\frac{\nu^t-1}{\nu-1} + \lambda\epsilon\sum_{t=0}^{T}\gamma^t\nu^t + \frac{L_{r_2}L_{\pi_1}\varrho(1-\gamma^{T+1})}{1-\gamma} + \mathcal{O}\left(r_{\max}\sqrt{\frac{\log(1/\delta)}{n}}\right),\end{aligned}$$

where $Rad(J_{\pi,\mathcal{T},\mathcal{I}}) = \mathbb{E}_{\xi}\mathbb{E}_{\sigma}[\sup_{\pi}\frac{1}{n}\sum_{i=1}^{n}\sigma_i J(\phi(\tau(\xi_i;\pi,\mathcal{T},\mathcal{I})))]$ is the Rademacher complexity, $\nu = L_{t_1} + L_{t_2}L_{\pi_1}L_{\phi}, \lambda = L_{r_1} + L_{r_2}L_{\pi_1}L_{\phi}$, and n is the number of training episodes.

Proof. We decompose the target as follows:

$$\begin{aligned}&\left\|\mathbb{E}_{\xi}\left[J\left(\phi\left(f\left(\tau\left(\xi;\pi,\mathcal{T}',\mathcal{I}'\right)\right)\right)\right)\right] - \frac{1}{n}\sum_{i=1}^{n} J(\phi(\tau(\xi_i;\pi,\mathcal{T},\mathcal{I})))\right\| \\ &\le \underbrace{\left\|\mathbb{E}_{\xi}\left[J\left(\phi\left(f\left(\tau\left(\xi;\pi,\mathcal{T}',\mathcal{I}'\right)\right)\right)\right)\right] - \mathbb{E}_{\xi}\left[J\left(\phi\left(\tau\left(\xi;\pi,\mathcal{T},\mathcal{I}\right)\right)\right)\right]\right\|}_{\text{(I)}} \\ &\quad + \underbrace{\left\|\mathbb{E}_{\xi}\left[J\left(\phi\left(\tau\left(\xi;\pi,\mathcal{T},\mathcal{I}\right)\right)\right)\right] - \frac{1}{n}\sum_{i=1}^{n} J(\phi(\tau(\xi_i;\pi,\mathcal{T},\mathcal{I})))\right\|}_{\text{(II)}}.\end{aligned}$$

Then, the conclusion holds by bounding term (I) with Theorem 2, and term (II) with Lemma 6. □

The remaining procedure is to bound the Rademacher complexity. To fulfill end, we first introduce the following lemma which shows that the empirical return J also satisfies the Lipschitz condition.

Lemma 7. *Assume that Assumptions 1, 2, 3 hold, and the encoder is L_ϕ-Lipschitz, then the empirical return $J(\phi(\tau(\cdot;\theta)))$, as a function of θ, is L_J-Lipschitz, i.e.,*

$$\begin{aligned}&\|J(\phi(\tau(\cdot;\theta))) - J(\phi(\tau(\cdot;\theta')))\| \leq L_J\|\theta-\theta'\|,\\ &\text{where } L_J = \sum_{t=0}^{T}\gamma^t\left(\lambda L_{t_2}L_{\pi_2}\frac{\nu^t-1}{\nu-1} + L_{r_2}L_{\pi_2}\right).\end{aligned} \tag{13}$$

Proof. Under policies $\pi(\cdot;\theta), \pi(\cdot;\theta')$ with parameters θ, θ', different trajectories will be collected. However, the initialization process is identical (i.e., $s_0 = s_0'$) as it does not depend on θ. Suppose $s \sim \tau(\cdot;\theta)$ and $s' \sim \tau(\cdot;\theta')$, then we have

$$\begin{aligned}\|J(\phi(\tau(\cdot;\theta))) - J(\phi(\tau(\cdot;\theta')))\| &= \left\|\sum_{t=0}^{T}\gamma^t r(s_t, \pi(\phi(s_t);\theta)) - \sum_{t=0}^{T}\gamma^t r(s_t', \pi(\phi(s_t');\theta'))\right\|\\ &\leq \sum_{t=0}^{T}\gamma^t|r(s_t,\pi(\phi(s_t);\theta)) - r(s_t', \pi(\phi(s_t');\theta'))|.\end{aligned}$$

By following a similar procedure in proving Lemma 3, we have

$$|r(s_t,\pi(\phi(s_t);\theta)) - r(s_t',\pi(\phi(s_t');\theta'))| \leq L_{r_1}\|s_t - s_t'\| + L_{r_2}\|\pi(\phi(s_t);\theta) - \pi(\phi(s_t');\theta')\|.$$

Furthermore, by using Lemma 1, we have

$$\begin{aligned}&|r(s_t,\pi(\phi(s_t);\theta)) - r(s_t',\pi(\phi(s_t');\theta'))|\\ &\leq L_{r_1}\|s_t-s_t'\| + L_{r_2}(L_{\pi_1}\|\phi(s_t') - \phi(s_t)\| + L_{\pi_2}\|\theta-\theta'\|)\\ &\leq L_{r_1}\|s_t-s_t'\| + L_{r_2}(L_{\pi_1}L_\phi\|s_t'-s_t\| + L_{\pi_2}\|\theta-\theta'\|)\\ &= \underbrace{(L_{r_1} + L_{r_2}L_{\pi_1}L_\phi)}_{:=\lambda}\|s_t-s_t'\| + L_{r_2}L_{\pi_2}\|\theta-\theta'\|.\end{aligned}$$

We now investigate $\|s_t - s_t'\|$, by following a similar procedure we use in proving Lemma 2, we have

$$\begin{aligned}\|s_t - s_t'\| &= \|\mathcal{T}(s_{t-1},\pi(\phi(s_{t-1});\theta),\xi_{t-1}) - \mathcal{T}(s_{t-1}',\pi(\phi(s_{t-1}');\theta'),\xi_{t-1})\|\\ &\leq \|\mathcal{T}(s_{t-1},\pi(\phi(s_{t-1});\theta),\xi_{t-1}) - \mathcal{T}(s_{t-1},\pi(\phi(s_{t-1}');\theta'),\xi_{t-1})\|\\ &\quad + \|\mathcal{T}(s_{t-1},\pi(\phi(s_{t-1}');\theta'),\xi_{t-1}) - \mathcal{T}(s_{t-1}',\pi(\phi(s_{t-1}');\theta'),\xi_{t-1})\|\\ &\leq L_{t_1}\|s_{t-1}-s_{t-1}'\| + L_{t_2}\|\pi(\phi(s_{t-1});\theta) - \pi(\phi(s_{t-1}');\theta')\|.\end{aligned}$$

Again, we use Lemma 1, and can derive

$$\begin{aligned}\|s_t - s'_t\| &\leq L_{t_1}\|s_{t-1} - s'_{t-1}\| + L_{t_2}(L_{\pi_1}\|\phi(s'_{t-1}) - \phi(s_{t-1})\| + L_{\pi_2}\|\theta - \theta'\|)\\ &\leq L_{t_1}\|s_{t-1} - s'_{t-1}\| + L_{t_2}(L_{\pi_1}L_\phi\|s'_{t-1} - s_{t-1}\| + L_{\pi_2}\|\theta - \theta'\|)\\ &= (L_{t_1} + L_{t_2}L_{\pi_1}L_\phi)\|s_{t-1} - s'_{t-1}\| + L_{t_2}L_{\pi_2}\|\theta - \theta'\|\\ &= \nu\|s_{t-1} - s'_{t-1}\| + L_{t_2}L_{\pi_2}\|\theta - \theta'\|.\end{aligned}$$

By recursion, we can further have

$$\|s_t - s'_t\| \leq \nu^t\|s_0 - s'_0\| + L_{t_2}L_{\pi_2}\frac{1-\nu^t}{1-\nu}\|\theta - \theta'\| = L_{t_2}L_{\pi_2}\frac{1-\nu^t}{1-\nu}\|\theta - \theta'\|.$$

The last equality is because $s_0 = s'_0$. Combining the above results, and we have

$$\begin{aligned}\|J(\phi(\tau(\cdot;\theta))) - J(\phi(\tau(\cdot;\theta')))\| &= \left\|\sum_{t=0}^{T}\gamma^t r(s_t, \pi(\phi(s_t);\theta)) - \sum_{t=0}^{T}\gamma^t r(s'_t, \pi(\phi(s'_t);\theta'))\right\|\\ &\leq \sum_{t=0}^{T}\gamma^t|r(s_t, \pi(\phi(s_t);\theta)) - r(s'_t, \pi(\phi(s'_t);\theta'))|\\ &\leq \sum_{t=0}^{T}\gamma^t(\lambda\|s_t - s'_t\| + L_{r_2}L_{\pi_2}\|\theta - \theta'\|)\\ &\leq \sum_{t=0}^{T}\gamma^t\left(\lambda L_{t_2}L_{\pi_2}\frac{1-\nu^t}{1-\nu}\|\theta - \theta'\| + L_{r_2}L_{\pi_2}\|\theta - \theta'\|\right)\\ &= \underbrace{\sum_{t=0}^{T}\gamma^t\left(\lambda L_{t_2}L_{\pi_2}\frac{1-\nu^t}{1-\nu} + L_{r_2}L_{\pi_2}\right)}_{:=L_J}\|\theta - \theta'\|.\end{aligned}$$

We thus can conclude the proof. □

Furthermore, if the number of parameters m in the policy $\pi(\cdot;\theta)$ is bounded, the Rademacher complexity can be controlled, as desired.

Lemma 8. *Given Assumptions 1, 2, 3, and suppose the parameters of the policy $\theta \in \mathbb{R}^m$ is bounded such that $\|\theta\| \leq K$, $K \in \mathbb{R}^+$, then the Rademacher complexity $Rad(J_{\pi,\mathcal{T},\mathcal{I}})$ is bounded, i.e.,*

$$Rad(J_{\pi,\mathcal{T},\mathcal{I}}) = \mathcal{O}\left(L_J K\sqrt{\frac{m}{n}}\right). \tag{14}$$

Proof. Note that we have a budget of n episodes. Denote $A = (\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_N)$ is a finite set of vectors, with each element containing $(J(\phi(\tau(\xi_1))), J(\phi(\tau(\xi_2))), \ldots, J(\phi(\tau(\xi_n))))$. For simplicity, we write $J_i = J(\phi(\tau(\xi_i)))$. One can see that $Rad(A) = Rad(J_{\pi,\mathcal{T},\mathcal{I}})$. Denote σ as a n-dimensional Rademacher variable. Denote $\langle f, g\rangle$ as the inner product of two vectors f, g. We define $\bar{\mathbf{a}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{a}_i$. Let $\lambda > 0$ and $A' = (\lambda a_1, \lambda a_2, \ldots, \lambda a_n)$. The proof is quite similar to that of the well-known Massart Lemma. We have

$$nRad(A') = \mathbb{E}_\xi\mathbb{E}_\sigma\left[\max_{\mathbf{a}\in A}\langle\sigma, \lambda\mathbf{a}\rangle\right] = \mathbb{E}_\xi\mathbb{E}_\sigma\left[\log\left(\max_{\mathbf{a}\in A}e^{\lambda\langle\sigma,\mathbf{a}\rangle}\right)\right] \leq \mathbb{E}_\xi\mathbb{E}_\sigma\left[\log\left(\sum_{\mathbf{a}\in A}e^{\lambda\langle\sigma,\mathbf{a}\rangle}\right)\right].$$

By using the Jensen's inequality, we have

$$nRad(A') \leq \mathbb{E}_\xi \mathbb{E}_\sigma \left[\log \left(\sum_{\mathbf{a} \in A} e^{\lambda \langle \sigma, \mathbf{a} \rangle} \right) \right] \leq \log \left(\mathbb{E}_\xi \mathbb{E}_\sigma \left[\sum_{\mathbf{a} \in A} e^{\lambda \langle \sigma, \mathbf{a} \rangle} \right] \right) = \log \left(\sum_{\mathbf{a} \in A} \mathbb{E}_\xi \mathbb{E}_\sigma \left[e^{\lambda \langle \sigma, \mathbf{a} \rangle} \right] \right).$$

Expanding the above formula, and we have

$$nRad(A') \leq \log \left(\sum_{\mathbf{a} \in A} \mathbb{E}_\xi \mathbb{E}_\sigma \left[e^{\lambda \langle \sigma, \mathbf{a} \rangle} \right] \right) = \log \left(\sum_{\mathbf{a} \in A} \prod_{i=1}^{n} \mathbb{E}_\xi \mathbb{E}_\sigma \left[e^{\lambda \sigma_i J_i} \right] \right).$$

We now claim that $\dfrac{e^a + e^{-a}}{2} \leq e^{\frac{a^2}{2}}, \forall a \in \mathbb{R}$. Notice that $e^a = \sum_{n=0}^{\infty} \dfrac{a^n}{n!}$, hence $\dfrac{e^a + e^{-a}}{2} = \sum_{n=0}^{\infty} \dfrac{a^{2n}}{(2n)!}$. Furthermore, we have $e^{\frac{a^2}{2}} = \dfrac{a^{2n}}{n! 2^n}$. The claim holds since $(2n)! \geq 2^n n!$.

Therefore, $\mathbb{E}_{\sigma_i}[e^{\sigma_i J_i}] = \dfrac{e^{J_i} + e^{-J_i}}{2} \leq e^{J_i^2/2}$, and we have

$$\begin{aligned} nRad(A') = n\lambda Rad(J_{\pi,\mathcal{T},\mathcal{I}}) &\leq \log \left(\sum_{\mathbf{a} \in A} \prod_{i=1}^{n} \mathbb{E}_\xi \mathbb{E}_\sigma \left[e^{\lambda \sigma_i J_i} \right] \right) \\ &\leq \log \left(\sum_{\mathbf{a} \in A} \prod_{i=1}^{n} e^{\lambda^2 J_i^2 / 2} \right) \\ &= \log \left(\sum_{\mathbf{a} \in A} e^{\lambda^2 \|\mathbf{a}\|_2^2 / 2} \right) \\ &\leq \log \left(|A| \max_{\mathbf{a} \in A} e^{\lambda^2 \|\mathbf{a}\|_2^2 / 2} \right) \\ &= \log(|A|) + \max_{\mathbf{a} \in A} \frac{\lambda^2 \|\mathbf{a}\|_2^2}{2}. \end{aligned}$$

Hence, we have

$$R(J_{\pi,\mathcal{T},\mathcal{I}}) \leq \frac{\log(|A|) + \max_{\mathbf{a} \in A} \frac{\lambda^2 \|\mathbf{a}\|_2^2}{2}}{\lambda n}.$$

Then, by setting $\lambda = \sqrt{\frac{2 \log(|A|)}{\|\mathbf{a}\|_2^2}}$, we have

$$Rad(J_{\pi,\mathcal{T},\mathcal{I}}) \leq \|\mathbf{a}\| \frac{\sqrt{2 \log(|A|)}}{n}.$$

Based on Lemma 26.6 in (Shalev-Shwartz & Ben-David, 2014), the Rademacher complexity satisfies $Rad(\{\mathbf{a} + \mathbf{a}_0 : \mathbf{a} \sim A\}) = Rad(A)$. We can hence rewrite the above inequality into $Rad(J_{\pi,\mathcal{T},\mathcal{I}}) \leq \|\mathbf{a} - \bar{\mathbf{a}}\| \dfrac{\sqrt{2 \log(|A|)}}{n}$. Note that by definition, $\mathbf{a} = (J_1, \ldots, J_n)$. Therefore by using the L_J-Lipschitz property of the return, we have $\|\mathbf{a} - \bar{\mathbf{a}}\| \leq \sqrt{n} L_J \|\theta - \theta'\| \leq \sqrt{n} L_J (\|\theta\| + \|\theta'\|) \leq 2K\sqrt{n} L_J$. We then have

$$Rad(J_{\pi,\mathcal{T},\mathcal{I}}) \leq 2K\sqrt{n} L_J \frac{\sqrt{2 \log(|A|)}}{n}.$$

It remains to bound $\log(|A|)$. Since the policy parameter $\theta \in \mathbb{R}^m$, and is bounded, then $\log(|A|)$ generally scales with $\mathcal{O}(d^m)$ (Shalev-Shwartz & Ben-David, 2014) where d is some bounded constant number. Finally, we have

$$Rad(J_{\pi,\mathcal{T},\mathcal{I}}) \leq \mathcal{O}\left(KL_J\sqrt{\frac{m}{n}}\right).$$

□

Finally, by combining previous results, we get the following generalization bound.

Theorem 4. *Suppose that the assumptions made in Theorem 3 and Lemma 8 hold. Then we have with probability at least $1-\delta$, the generalization error gives,*

$$\begin{aligned}&\left\|\mathbb{E}_\xi\left[J\left(\phi\left(f\left(\tau\left(\xi;\pi,\mathcal{T}',\mathcal{I}'\right)\right)\right)\right)\right] - \frac{1}{n}\sum_{i=1}^{n} J(\phi(\tau(\xi_i;\pi,\mathcal{T},\mathcal{I})))\right\| \\ &\leq \lambda\zeta\sum_{t=0}^{T}\gamma^t\frac{\nu^t-1}{\nu-1} + \lambda\epsilon\sum_{t=0}^{T}\gamma^t\nu^t + \frac{L_{r_2}L_{\pi_1}\varrho}{1-\gamma}(1-\gamma^{T+1}) + \mathcal{O}\left(L_JK\sqrt{\frac{m}{n}}\right) + \mathcal{O}\left(r_{\max}\sqrt{\frac{\log(1/\delta)}{n}}\right).\end{aligned}$$

Proof. It follows by combining Theorem 3 and Lemma 8. □

Remark: We summarize a key insight based on the above bound, *the generalization gap can only be small if the representation distance between the training and testing environments is small*, due to the fact that ϱ is the only factor that one can control in the bound. This observation is somewhat consistent with a human's intuition: the representations before and after involving distractors are similar and hence the policy can retrieve good behaviors it learned in the training environment.

Note that the Assumption 1 requires that the environment dynamics is *smooth*, i.e., no sudden change occurs. Our theory still holds if the Lipschitz continuity in terms of transition dynamics is slightly violated, e.g., the transition dynamics satisfies $\|\mathcal{T}(s,a,\xi)-\mathcal{T}(s',a,\xi)\| \leq L_{t_1}\|s-s'\|+B$, where $B \in \mathbb{R}, B < \infty$ is a bounded constant number. Then extra constant terms will be included in the bounds. Our core conclusion also holds since B is a constant, i.e., under this condition, minimizing the representation deviation is still the key way of reducing the generalization error. However, we have to admit that if B is uncontrollable, or no similar constraints like $\|\mathcal{T}(s,a,\xi)-\mathcal{T}(s',a,\xi)\| \leq L_{t_1}\|s-s'\|+B$ can be satisfied, our theories will fail and cannot be used to explain the generalization capability then. We stress that many real-world (or simulation) dynamics do not often encounter abrupt changes (if abrupt changes occur, those are often referred to as emergent conditions, e.g., the robot dog fall). It is hard to present a general theoretical framework that fits all of the conditions. We do not view this as a drawback of our theory.

6. Experimental Support

In this section, we resolve the following concerns: (1) are the assumptions we made in Section 5 reasonable? (Section 6.1) (2) do existing algorithms agree with our theoretical insight? (Section 6.2).

6.1 The rationality of the assumptions

The main assumptions we made for reaching the theoretical bound are the *smoothness* on the reward, policy, and transition dynamics. Note that such smoothness assumptions are widely adopted in prior work (Liu et al., 2022; Miyato et al., 2018; Scaman & Virmaux, 2018; Lecarpentier et al., 2020). Since the reward functions are mostly designed manually, it is easy to satisfy the Lipschitz condition. Assuming the Lipschitz continuity in the environmental dynamics is also valid and it is satisfied if the state space is bounded and no sudden change in this physical system occurs. The Lipschitz assumption for the policy π can be easily satisfied for many policy classes in practice, e.g., neural networks (Fazlyab et al., 2019; D'Oro & Jaśkowski, 2020). We empirically examine whether such property still holds with distractors. We compare the policy learned by DrQ (Yarats et al., 2021), which uses plain data augmentation, and PIE-G (Yuan et al., 2022), which leverages the pre-trained (on ImageNet) image encoder (He et al., 2015), on `walker-walk` task from DMControl Generalization Benchmark (DMC-GB) (Hansen & Wang, 2020). To that end, we first gather 10 trajectories τ in the clean training environment with the learned DrQ agent, and draw the scatter plot of $\|\pi(\phi(s))-\pi(\phi(s'))\|_2^2$ against $\|\phi(s)-\phi(s')\|_2^2$ via sampling $s, s' \sim \tau$ in a bootstrapping way for 10^4 times. We then add distractors in τ by replacing the background of the agent with playing videos, and plot the corresponding scatter plot. The results are shown in Figure 2, and one can find that the Lipschitz condition for the policy network is generally satisfied with or without distractors.

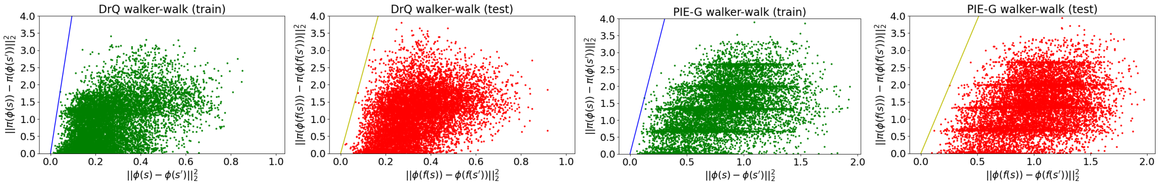


Figure 2: **Evidence that the Lipschitz condition of the policy holds before and after adding distractors.** We present the scatter plot of the policy deviation against the representation deviation of DrQ and PIE-G on walker-walk *video-easy* task with and without distractors (i.e., the transpose function $f(\cdot)$). The solid line denotes the maximum slope in the batch, i.e., $y = kx$ where $k = \max \frac{\|\pi(\phi(s))-\pi(\phi(s'))\|}{\|\phi(s)-\phi(s')\|}$ for the training environment, and $k = \max \frac{\|\pi(\phi(f(s)))-\pi(\phi(f(s')))\|}{\|\phi(f(s))-\phi(f(s'))\|}$ for the testing environment. Since there always exist a k such that the policy deviation of all samples can be bounded, the Lipschitz condition for the policy holds naturally.

6.2 Do existing methods align with the theoretical results?

We then examine whether our theory applies to existing algorithms and explains why they work in practice. Since our contribution mainly lies in the theoretical side, we do not include extensive studies on all recent algorithms and can only conduct experiments with some of them. We examine the following six algorithms: DrQ (Yarats et al., 2021), PAD (Hansen et al., 2021a), CURL (Srinivas et al., 2020) SODA (Hansen & Wang, 2020), SVEA (Hansen et al., 2021b), and PIE-G (Yuan et al., 2022). These methods are chosen because of the fact that they typically represent different categories of methods that handling the generalization problems in visual RL, and that their reported results can be reproduced. DrQ explores the data augmentation trick to enhance visual RL training; PAD utilizes a self-supervised objective to adapt to the testing environment during deployment; CURL leverages the contrastive learning objective to capture better representations; SODA reformulates the problem of generalization as a representational consistency learning problem and enables the encoder to map different views of the same state to similar representations; SVEA uses strong data augmentation (e.g., augmentation with random convolution networks) and regularizes the way Q value updates; PIE-G is built upon DrQ-v2 (Yarats et al., 2022) while replaces its encoder with the pre-trained image encoder.

It is known that DrQ often fails when generalizing to unseen test environments, while PIE-G and SVEA are among the strongest generalization visual RL algorithms. In order to show that our theory holds in practice, we expect that the representation deviation $\|\phi(f(s)) - \phi(s)\|$ (as well as the policy deviation $\|\pi(\phi(f(s))) - \pi(\phi(s))\|$) of a stronger generalization algorithm ought to be smaller than weak ones. We verify this by conducting experiments on two environments from DMC-GB, `walker-walk` and `finger-spin` (results on wider environments can be found in Appendix B). We note that the policy deviation counts when comparing different generalization visual RL algorithms because different algorithms train their policies in different ways, leading to different L_{π_1} that may affect the term $\frac{L_{r_2} L_{\pi_1} \varrho}{1-\gamma}(1-\gamma^{T+1})$. The policy deviation is an important metric that directly indicate *how well the agent can recover its behavior learned in the clean train environment when deployed in the test environment with distractors*. We run these algorithms under their default hyperparameters on the clean training environment first and then replace the background with playing videos or change the color of the agent (e.g., *video-easy* setting in DMC-GB contains 10 different video backgrounds). We adopt three generalization scenarios for empirical evaluation: color-hard, video-easy, and video-hard. Our experimental setting is, the trajectory remains the same, and only backgrounds or the colors are changed. This generally meets our formulation, i.e., we have s in the training environment while also needing $f(s)$ in the testing environment, upon the identical state s. We then evaluate the representation deviation using the learned encoder and the policy deviation with the policy network of each algorithm on the clean training trajectory and the testing trajectories with distractors for 100 episodes and 5 different random seeds.

All algorithms are run for 500K environmental steps. The evaluated trajectories are made up of 1000 transitions. We measure the average representation deviation $\mathbb{E}_{s\sim\tau}[\|\phi(f(s)) - \phi(s)\|]$ and the policy deviation $\mathbb{E}_{s\sim\tau}[\|\pi(\phi(f(s))) - \pi(\phi(s))\|]$ along the trajectory. We summarize the results of walker-walk task in Figure 3 and the results of finger-spin task in Figure 4.

It turns out that DrQ generally has the largest policy deviation and representation deviation among all of the methods (and its generalization performance is generally the worst). The representation deviation and policy deviation of CURL and PAD are generally smaller than DrQ (but the deviations are larger than DrQ on a limited number of tasks). Based on (Hansen et al., 2021a; Yuan et al., 2022), CURL often does not exhibit satisfying generalization performance and we observe that its policy deviation and representation

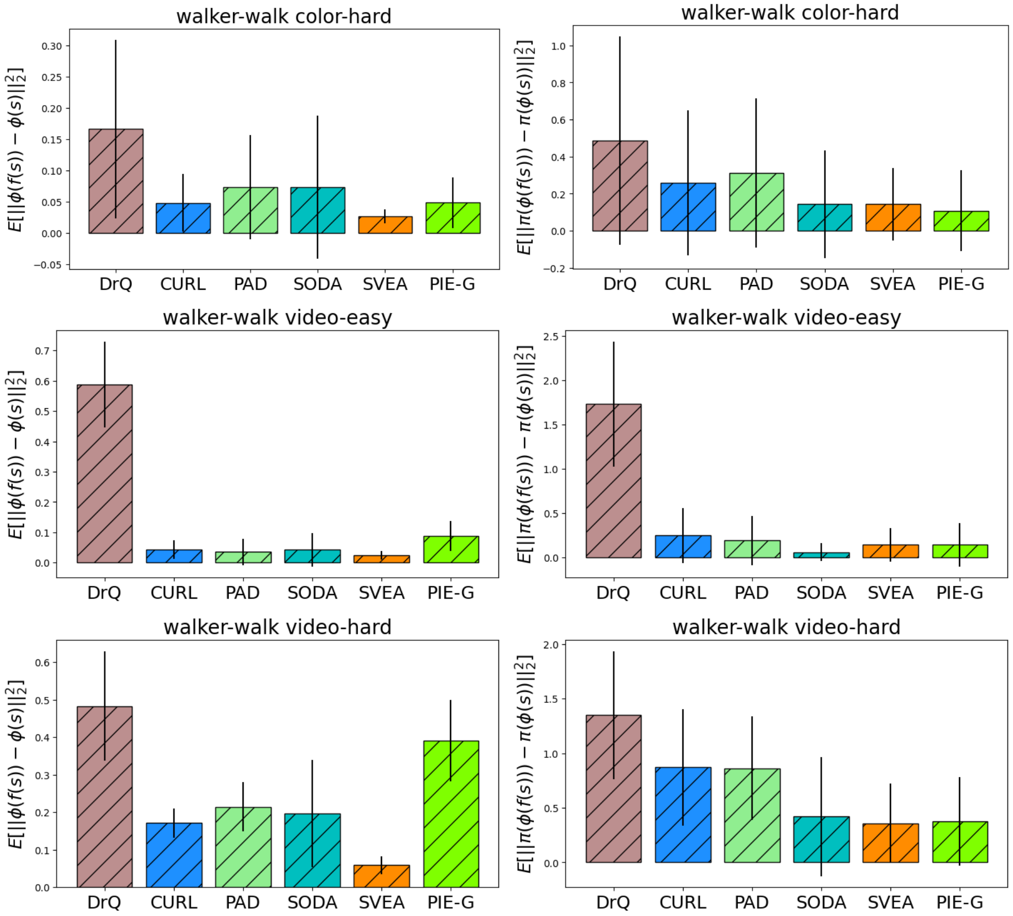


Figure 3: **Evidence that our theoretical results can explain empirical algorithms.** We present comparison of average representation deviation ($\mathbb{E}[\|\phi(s) - \phi(s')\|_2^2]$, left column) and average policy deviation ($\mathbb{E}[\|\pi(\phi(s)) - \pi(\phi(s'))\|_2^2]$, right column) of 6 typical methods on color-hard, video-easy, and video-hard settings of walker-walk task from DMC-GB. The results are averaged over the trajectory and across 5 varied random seeds. The error bar denotes the average standard deviation along the trajectory and 5 seeds.

deviation are much larger than methods like SODA and SVEA. PAD also underperforms methods like SVEA and PIE-G in many generalization tasks. It is understandable because many studies (Jaderberg et al., 2017; Hansen et al., 2021a; Lin et al., 2019; Lyle et al., 2021) have shown that the best choice of auxiliary tasks in self-supervision methods highly depends on the specific RL task, and negative impacts can incur if we choose sub-optimal auxiliary tasks. The strong generalization visual RL algorithms like SODA, SVEA and PIE-G have small policy deviation and representation deviation on most of the tasks. Notably, the standard deviation of DrQ is also the largest, while algorithms like SODA, SVEA and PIE-G enjoy much smaller standard deviations, indicating that they are more stable than

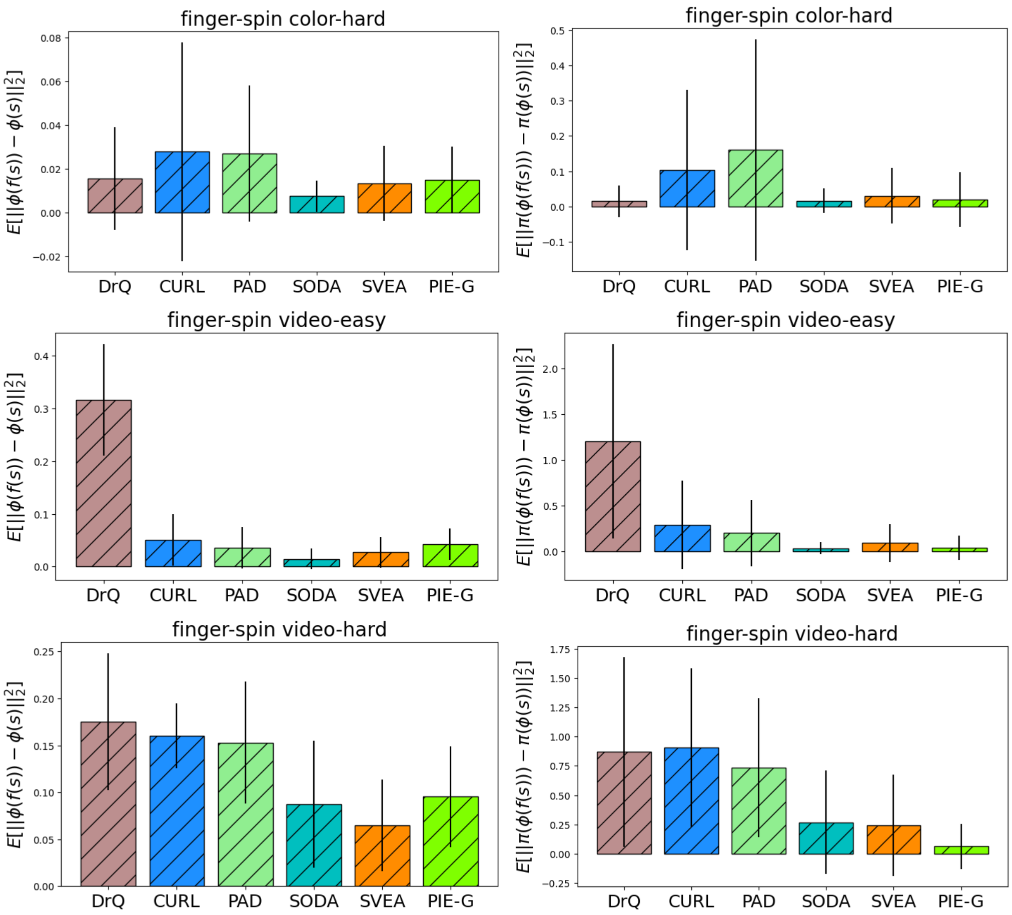


Figure 4: **Evidence that our theoretical results can explain empirical algorithms.** We demonstrate comparison of average representation deviation ($\mathbb{E}[\|\phi(s) - \phi(s')\|_2^2]$, left column) and average policy deviation ($\mathbb{E}[\|\pi(\phi(s)) - \pi(\phi(s'))\|_2^2]$, right column) of 6 typical methods on color-hard, video-easy, and video-hard settings of finger-spin task from DMC-GB. The results are averaged over the trajectory and 5 different random seeds. The error bar represents the average standard deviation along the trajectory and 5 seeds.

DrQ. The empirical results are unanimously in line with our expectations. Hence, we believe our theory explains in part why these algorithms work in practice.

For readers of interest, we include the results of average policy deviation and representation deviation along the trajectory (that is, the plots of average policy deviation and representation deviation against the steps) on some tasks in Appendix C, where we find that methods like DrQ has large representation deviation and policy deviation on almost every transition in the trajectory while strong generalization methods like SVEA and PIE-G enjoy much smaller deviations. We believe these can further validate the rationality of our theoretical analysis.

7. Discussions

In this section, we discuss some issues that readers of interest may wonder, e.g., guidance on improving test performance. Furthermore, we discuss some possibly promising ways for developing future advanced generalization visual RL algorithms and the theoretical bounds when different reward functions are considered in the training and testing environments.

How reparameterization can be applied in RL? Our core contribution in this paper is the construction of theoretical bounds on the generalization gap in visual RL, via the reparameterization trick. As we highlight in Section 2, the community has already witnessed many trials in terms of leveraging reparameterization in RL, with some of them being quite popular and widely used (e.g., the SAC algorithm). Apart from the aforementioned applications of reparameterization, one can employ it in linear quadratic regulator (LQR) (Bradtke, 1992; Cao & Ren, 2010; Simchowitz & Foster, 2020; Recht, 2019). For instance, Jost et al. (2017) reparameterize the input variables with an LQR controller to accelerate linear model predictive control. In practice, the reparameterization can also be applied in the (real-world) physical systems where the dynamics are formulated by stochastic partial differential equations that contain reparameterizable parameters or components over continuous state-action space (Heess et al., 2015; Perkins & Barto, 2001), as we do in Algorithm 1.

How to improve test performance? Our theoretical result in Theorem 4 indicates that if one wants good performance in the testing environment, then one should (1) minimize the representation deviation between training and testing environments, such that the generalization gap is small; (2) improve the performance in the training environment, since only if the agent achieves good performance during training (e.g., collected n trajectories are all of the high return) can it possibly acquire satisfying performance in the testing environment. The second part is closely correlated to the *sample efficiency* in visual RL. Though many works try to attain some progress, including model-based (Hansen et al., 2022; Ye et al., 2021) and model-free (Yarats et al., 2022; Hansen et al., 2022; Lyu et al., 2024) methods, it still remains a central challenge.

Possible directions of developing RL algorithms for generalization with image input. We believe the most critical issue for enhancing generalization in visual RL is how to get robust and generalizable representations. Studying how and when the image representations can generalize to unseen scenarios both empirically and theoretically may be vital, and (Le Lan et al., 2022) serves as a good primary work. Extracting important regions in the visual input to dismiss the influence of distractors from the testing environment is also promising. Bertoin et al. (2022) realize it by building an attribution map. Moreover, this can be done with the aid of foundation models, e.g., SAM (Kirillov et al., 2023). It is straightforward that the generalization performance can be improved if the influence of distractors in the testing environments is cleared. It is equivalent to directly minimizing $\|f(s) - s\|$ (our theories still apply here). Meanwhile, it is interesting to see whether large language models like ChatGPT can aid generalization in visual RL, e.g., ask the ChatGPT to summarize the image and tell us what action we should take, such that we use the text embedding as an extra input in the policy. By asking large language models (LLM), one may extract some key features that are not affected by the distractors. For example, the LLM may summarize that "there is a running robot in the figure", and this may be vital for the agent to execute suitable actions. By incorporating such information as an additional input, we may minimize the representation deviation between the training and testing environments. From another perspective, enhancing the robustness of the policy is promising and encouraging for improving the generalization ability, i.e., controlling L_{π_1}, L_{π_2}, and we can borrow some ideas from the field of robust RL (Morimoto & Doya, 2001; Tessler et al., 2019; Zhang et al., 2021; Derman et al., 2020).

More general theoretical bounds when the reward functions are different. In our theoretical analysis, we assume that the training environment and the testing environment can have slight different transition dynamics and initialization function, but with the same reward function. It is then interesting to investigate the generalization gap of visual RL agent when different reward functions are adopted in training and testing environments. We denote the reward function in the training environment as $r(s,a)$ and the reward function in the testing environment as $r'(s,a)$. We denote the return that leverages the testing rewards r' as J', i.e., $\mathbb{E}_\tau[J'(\phi(f(\tau)))] = \mathbb{E}_\tau\left[\sum_{t=0}^T \gamma^t r'(s_t, \pi(\phi(f(s_t))))\right]$. Then with similar assumptions and some additional assumption on the rewards in the training and testing environments, we can derive a similar generalization bound as Theorem 4, where our core insight in the previous analysis still holds, i.e., the core factor that impact the generalization error is the representation deviation of the state before and after adding the distractors.

Theorem 5. *Suppose that the assumptions made in Theorem 3 and Lemma 8 hold. We suppose the testing environment has a different reward function $r'(s,a)$ from the training environment. We assume $|r(s,a) - r'(s,a)| \le \epsilon_r < \infty, \epsilon_r \in \mathbb{R}, \forall s, a$, i.e., given any state and action, the rewards in the training environment and the testing environments are bounded. Then we have with probability at least $1-\delta$, the generalization error gives,*

$$\left\| \mathbb{E}_\xi\left[J'\left(\phi\left(f\left(\tau\left(\xi;\pi,\mathcal{T}',\mathcal{I}'\right)\right)\right)\right)\right] - \frac{1}{n}\sum_{i=1}^n J(\phi(\tau(\xi_i;\pi,\mathcal{T},\mathcal{I})))\right\| \le \lambda\zeta\sum_{t=0}^T \gamma^t \frac{\nu^t-1}{\nu-1} + \lambda\epsilon\sum_{t=0}^T \gamma^t\nu^t$$
$$+ \frac{1-\gamma^{T+1}}{1-\gamma}(L_{r_2}L_{\pi_1}\varrho + \epsilon_r) + \mathcal{O}\left(L_J K\sqrt{\frac{m}{n}}\right) + \mathcal{O}\left(r_{\max}\sqrt{\frac{\log(1/\delta)}{n}}\right).$$

Proof. We decompose the target as follows:

$$\begin{aligned}&\left\| \mathbb{E}_\xi\left[J'\left(\phi\left(f\left(\tau\left(\xi;\pi,\mathcal{T}',\mathcal{I}'\right)\right)\right)\right)\right] - \frac{1}{n}\sum_{i=1}^n J(\phi(\tau(\xi_i;\pi,\mathcal{T},\mathcal{I})))\right\| \\ &\le \underbrace{\left\|\mathbb{E}_\xi\left[J'\left(\phi\left(f\left(\tau\left(\xi;\pi,\mathcal{T}',\mathcal{I}'\right)\right)\right)\right)\right] - \mathbb{E}_\xi\left[J\left(\phi\left(f\left(\tau\left(\xi;\pi,\mathcal{T}',\mathcal{I}'\right)\right)\right)\right)\right]\right\|}_{\text{(I)}} \\ &\quad + \underbrace{\left\|\mathbb{E}_\xi\left[J\left(\phi\left(f\left(\tau\left(\xi;\pi,\mathcal{T}',\mathcal{I}'\right)\right)\right)\right)\right] - \frac{1}{n}\sum_{i=1}^n J(\phi(\tau(\xi_i;\pi,\mathcal{T},\mathcal{I})))\right\|}_{\text{(II)}}.\end{aligned}$$

For term (I), we have

$$\begin{aligned}
\text{(I)} &= \left\|\mathbb{E}_\xi\left[J'\left(\phi\left(f\left(\tau\left(\xi;\pi,\mathcal{T}',\mathcal{I}'\right)\right)\right)\right)\right] - \mathbb{E}_\xi\left[J\left(\phi\left(f\left(\tau\left(\xi;\pi,\mathcal{T}',\mathcal{I}'\right)\right)\right)\right)\right]\right\| \\
&\le \int_\xi \left\| \sum_{t=0}^{T} \gamma^t \left(r'(s_t,\pi(\phi(f(s_t)))) - r(s_t,\pi(\phi(s_t)))\right)\right\| d\xi \\
&\le \int_\xi \sum_{t=0}^{T} \gamma^t \epsilon_r = \frac{1-\gamma^{T+1}}{1-\gamma}\epsilon_r.
\end{aligned}$$

For term (II), we directly bound it with Theorem 4, and then the conclusion holds accordingly. □

8. Conclusions and Limitations

Despite the emergence of many practical and promising algorithms for enhancing the generalization capability of visual RL policies, a clear and instructive theoretical analysis on the generalization gap, and how to minimize the generalization gap are absent. The main purpose of this work is to provide a theoretical bound on the generalization gap in visual RL when there exist distractors in the testing environment, and try to explain why previous methods work based on the derived bounds. However, directly analyzing the generalization gap is difficult since the policy keeps evolving. We isolate the randomness from the policy by resorting to the reparameterization trick. Our bound indicates that the key to reducing the generalization gap is to minimize the representation deviation between the training and testing environments. We further provide empirical evidence on the validity of the assumptions and conclusion, which we find is consistent with the theoretical results. An interesting future work is to study the generalization gap of off-policy visual RL and characterize what matters under this setting, e.g., derive the generalization gap bound of *offline* visual RL algorithms.

We believe the limitations of this work mainly lie in the following aspects: (1) the reparameterization trick cannot be experimentally applied in the simulated environments, e.g., DMC-GB, and some real-world scenarios, because they often do not meet the reparameterzable conditions; (2) some of the assumptions we made (e.g., the smoothness in dynamics) may not necessarily hold in general cases.

Acknowledgments

Work done while Jiafei Lyu was an intern at Tencent IEG. This work was supported by the STI 2030-Major Projects under Grant 2021ZD0201404. The authors would like to thank the anonymous reviewers for their valuable comments and suggestions.

Appendix A. Missing Example Trajectories on DMC-GB

In this section, we provide the example trajectories of the visual RL agent in the training environment, and the corresponding trajectories of DrQ, SVEA, and PIE-G in the testing environment. We choose walker-walk video-easy task as the testbed.

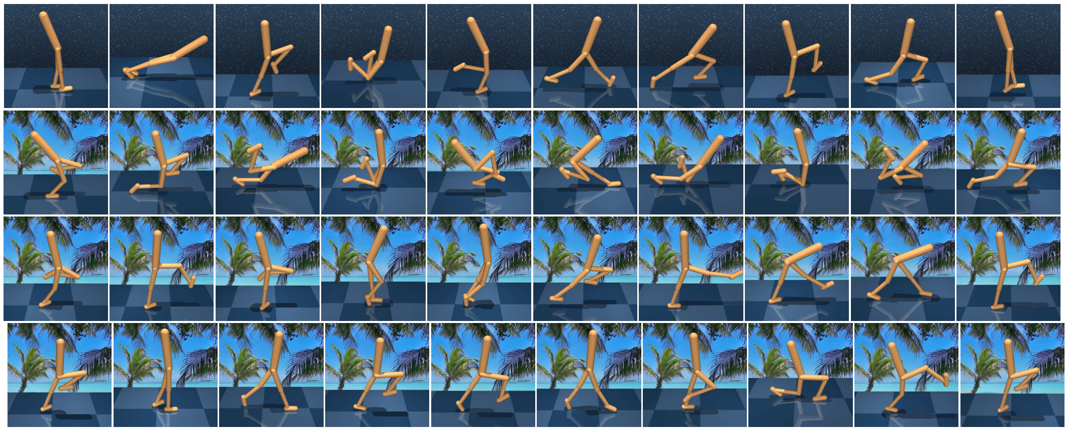
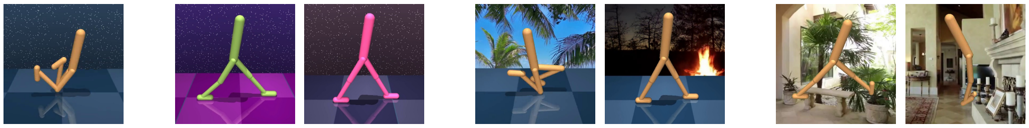


Figure 5: Example trajectories of the training environment (first row), and DrQ (second row), SVEA (third row), and PIE-G (fourth row) deployed in the walker-walk *video-easy* task with distractors. Results are illustrated by using the models of each algorithm after training 500K environmental steps.

Appendix B. Broader Experiments

In this section, we provide wider empirical results to further validate our theoretical results. Our experiments are conducted on DMC-GB, and we consider three settings: *color-hard*, *video-easy*, and *video-hard*. The color-hard tasks require the agent to generalize to color-jittered observations. Video-easy and video-hard tasks, instead, replace the background of the agent with the playing videos. Among them, video-hard tasks are the most challenging, since they contain more complex and fast-switching videos (up to 100 videos), and the reference plane of the ground is also removed. We detail the examples of the training and testing environments in DMC-GB below in Figure 6.



(a) training (b) color-hard (c) video-easy (d) video-hard

Figure 6: Examples of the clean training environment and testing environments with randomly jittered color and moving video backgrounds in DMC-GB.

We adopt the implementation of DrQ and SVEA from the DMC-GB [1] under MIT License, and adopt the implementation of PIE-G from the open-sourced codebase from the authors [2]. We use the default hyperparameters of these algorithms. We evaluate each algorithm across 100 episodes and 5 different random seeds (1-5) for all of the generalization

[1] https://github.com/nicklashansen/dmcontrol-generalization-benchmark

[2] https://anonymous.4open.science/r/PIE-G-EF75/README.md

settings. Note that we use SVEA(conv), which leverages a random convolution network for data augmentation, instead of SVEA(overlay), as SVEA(overlay) adopts an outer source of dataset for training and data augmentation, which we think somewhat has an overlap with PIE-G. In practice, SVEA(conv) exhibits similar performance compared with that of SVEA(overlay). The experiments are conducted with PyTorch 1.8 version.

In the main text, we mainly showcase the policy deviation and representation deviation of DrQ, CURL, PAD, SODA, SVEA, and PIE-G on the color-hard, video-easy, and video-hard settings of `walker-walk` and `finger-spin` tasks. We further present the results on two additional tasks here, `cartpole-swingup` and `walker-stand`. It is worth noting that our comparison is valid and reasonable since the sample efficiency and the final performance of these algorithms in the training environments are quite similar. Following the same procedure in the main text, we collect a trajectory in the training environment with the learned policy, and then add distractors (e.g., the background is replaced with moving videos) on it. We then evaluate the policy deviation and representation deviation across 100 episodes (the distractors are different in different episodes). This is repeated with 5 different random seeds. The comparison results on `cartpole-swingup` is depicted in Figure 7, and one can find that the results also match our conclusion in the main text, i.e., both the representation deviation and policy deviation of DrQ, PAD and CURL are larger than those of SODA, SVEA and PIE-G.

The comparison results `walker-stand` is shown in Figure 8, where we only compare DrQ against SVEA and PIE-G. We observe that in `walker-stand` environment, it seems the representation deviation of DrQ is smaller. We would like to clarify here that the reasons lie in the fact that DrQ itself actually already achieves quite good performance on color-hard, video-easy evaluation modes of walker-stand (comparable to SVEA and PIE-G, please refer to (Hansen et al., 2021b; Yuan et al., 2022)), indicating that this task is simple and simple data augmentation techniques utilized in DrQ can aid the capture of important knowledge from the testing environments. This does not indicate that our theory fails. It can be found in Theorem 4 that the generalization gap bound not only depends on the representation deviation ϱ, but also correlates with the Lipschitz constant of the policy, i.e., L_{π_1}. One can see that the policy deviations of DrQ on walker-stand color-hard and video-hard tasks are larger than SVEA and PIE-G, despite that the representation deviations of DrQ are smaller. It is not difficult to conclude that the Lipschitz constant is the largest in DrQ. Similar phenomenon can also be seen in Figure 2 where both $\max \frac{\|\pi(\phi(s))-\pi(\phi(s'))\|_2^2}{\|\phi(s)-\phi(s')\|_2^2}$ and $\max \frac{\|\pi(\phi(f(s)))-\pi(\phi(f(s')))\|_2^2}{\|\phi(f(s))-\phi(f(s'))\|_2^2}$ of DrQ are larger than those of PIE-G. For example, on walker-walk test trajectories, the $\max \frac{\|\pi(\phi(f(s)))-\pi(\phi(f(s')))\|_2^2}{\|\phi(f(s))-\phi(f(s'))\|_2^2}$ gives 24, while PIE-G gives **7.8**. These further reveal that the policy learned by the DrQ agent is fragile and unstable, while SVEA and PIE-G exhibit better tolerance to distractors during testing on walker-stand color-hard and video-hard tasks. Hence, with a joint influence of the L_{π_1} and the representation deviation, the generalization gap of DrQ is larger.

Interestingly, we notice that on the video-easy setting of walker-stand task, DrQ is better than SVEA in terms of both representation deviation and policy deviation. This is because, on this setting, the average test performance of DrQ gives **873**, while SVEA underperforms it with an average performance of 795. It is also worth noting that one may wonder why on some tasks like walker-walk video-hard, cartpole-swingup video-easy, the representation

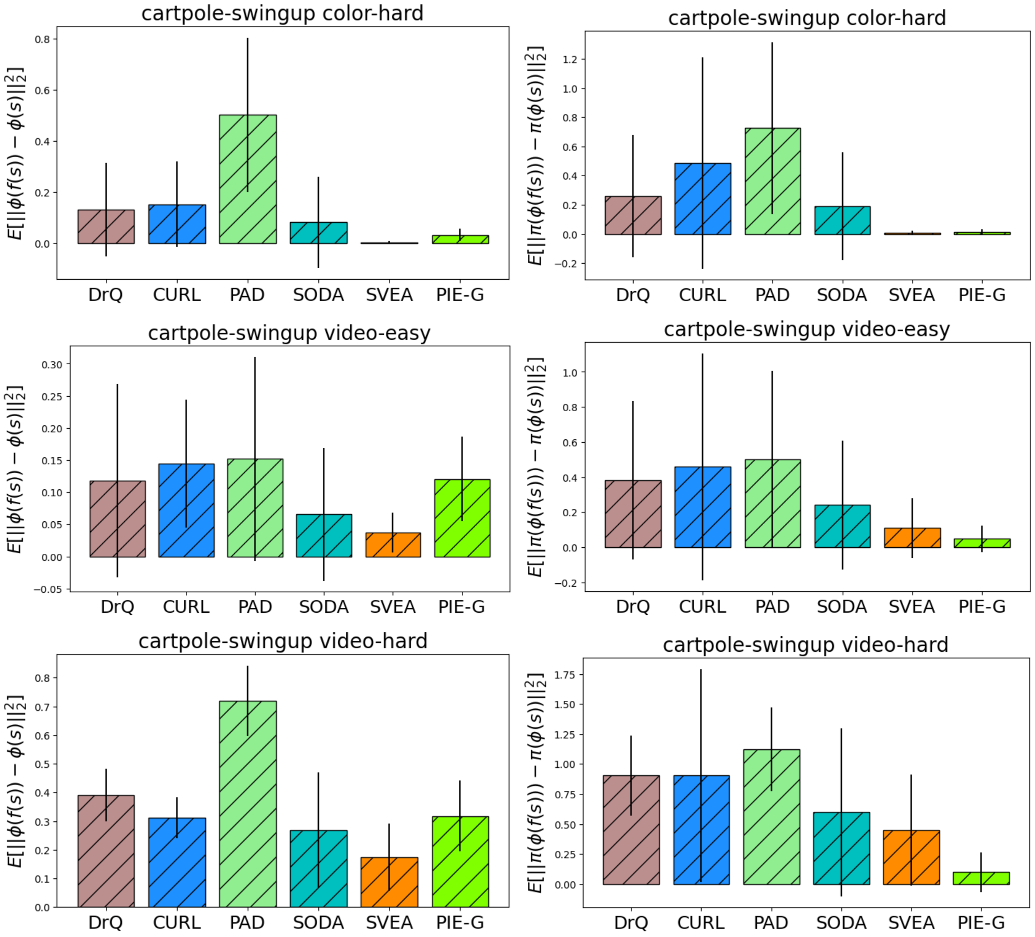


Figure 7: **Evidence that our theoretical results can explain empirical algorithms.** We present comparison of average representation deviation ($\mathbb{E}[\|\phi(s) - \phi(s')\|_2^2]$, left column) and average policy deviation ($\mathbb{E}[\|\pi(\phi(s)) - \pi(\phi(s'))\|_2^2]$, right column) of 6 typical methods on color-hard, video-easy, and video-hard settings of cartpole-swingup task from DMC-GB. The results are averaged over the trajectory and across 5 varied random seeds. The error bar denotes the average standard deviation along the trajectory and 5 seeds.

deviation difference between DrQ and PIE-G is small, and on walker-stand video-hard, PIE-G has a larger representation deviation. This is because the encoders of the two algorithms are trained on *different* domains, leading to different parameters and representation space. It is then understandable that on some tasks, PIE-G has a similar representation deviation as DrQ. Therefore, a better measurement on the generalization gap may be the policy deviation, which is a joint measurement of $L_{\pi_1}\varrho$, and it turns out that PIE-G is better in most of the cases. Moreover, we emphasize that DrQ often has a larger standard deviation on many tasks than SVEA and PIE-G in terms of both policy deviation and representation

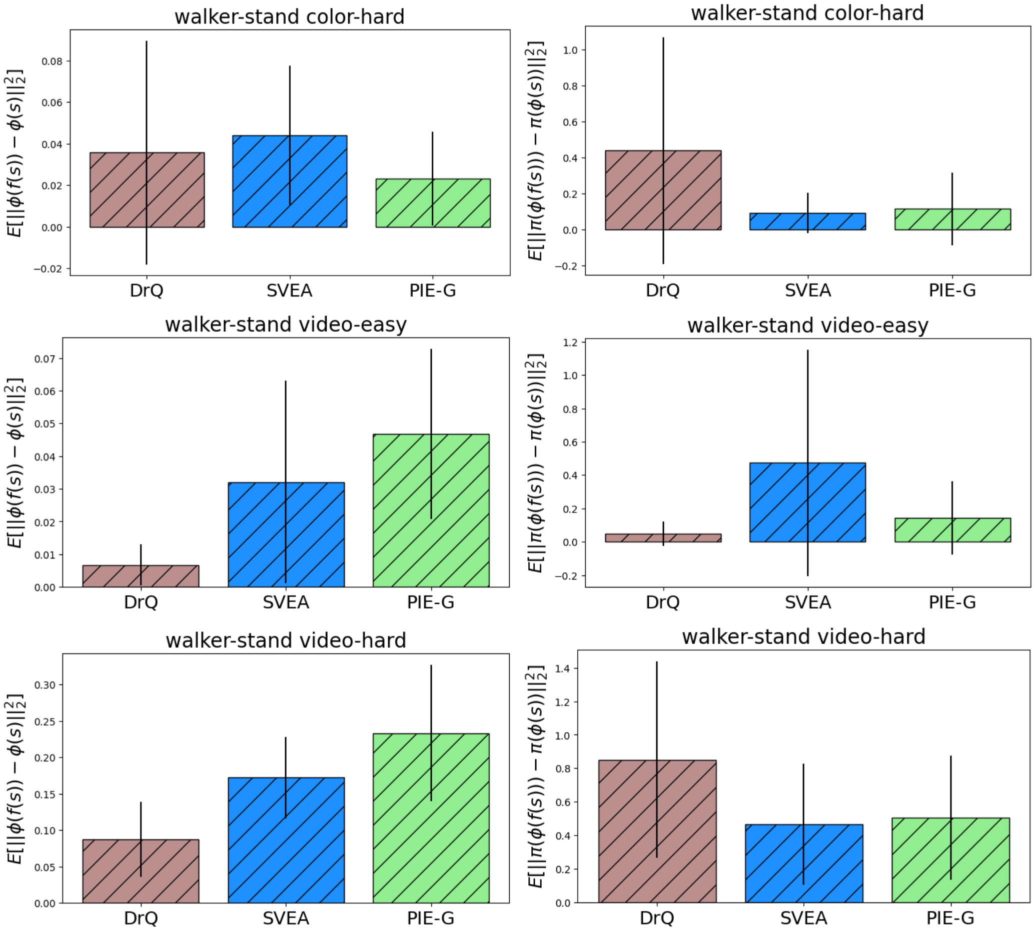


Figure 8: **Evidence that our theoretical results can explain empirical algorithms.** We present comparison of average representation deviation ($\mathbb{E}[\|\phi(s)-\phi(s')\|_2^2]$, left column) and average policy deviation ($\mathbb{E}[\|\pi(\phi(s))-\pi(\phi(s'))\|_2^2]$, right column) of 6 typical methods on color-hard, video-easy, and video-hard settings of walker-stand task from DMC-GB. The results are averaged over the trajectory and across 5 varied random seeds. The error bar denotes the average standard deviation along the trajectory and 5 seeds.

deviation. Such instability and error can be accumulated during executing actions in the testing environments, and finally result in a bad performance. PIE-G and SVEA, instead, can learn more robust policies and encoders.

To summarize, some special cases (e.g., walker-stand color-hard, walker-stand video-hard) showing that the representation deviation of DrQ is smaller while its generalization gap is larger do not make our theory a failure. As when comparing the generalization gap of different algorithms, *it is more meaningful to compare policy deviation because different algorithms have different encoders and update rules for policy*, and the generalization gap is jointly determined by the smoothness of the policy and the representation deviation.

DrQ has poor test performance as it learns an unstable policy (with a large Lipschitz constant). However, by adopting a better data augmentation method and modifying the way of updating Q function, SVEA (which is built upon DrQ) learns a smooth policy and incurs a better generalization ability, which aligns with our theory. Furthermore, the success of PIE-G also strongly supports our theory, since it only replaces the encoder in the vanilla DrQ-v2 with the pre-trained image encoder. We also note that enabling generalizable representation (and smaller representation deviation) across different evaluation scenarios and achieving a smoother policy seems to be strongly connected in practice. Therefore, we believe our claims and insights hold.

Appendix C. Missing Results on Deviations along the Trajectory

We summarize the comparison results of representation deviation and policy deviation along the evaluation trajectories between DrQ, SVEA and PIE-G in Figure 9, and the representation deviation comparison between DrQ, CURL, SODA and SVEA in Figure 10 and policy deviation comparison in Figure 11. The comparison is conducted on two DMC-GB environments, walker-walk and finger-spin.

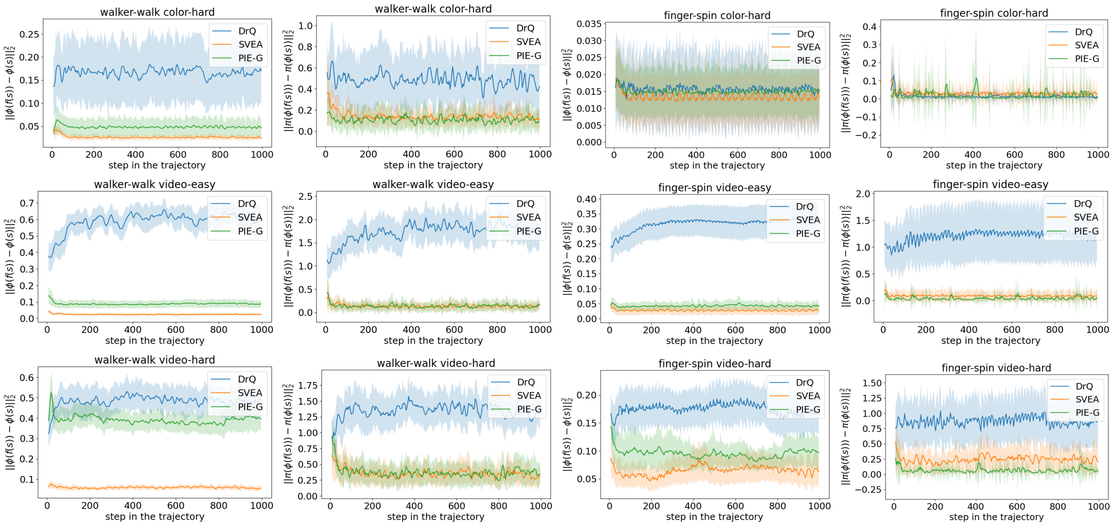


Figure 9: **Detailed plots of representation deviation and policy deviation comparison.** We show the comparison between SVEA, PIE-G, and DrQ on color-hard, video-easy, and video-hard settings of walker-walk and finger-spin tasks from DMC-GB. The results are averaged over 5 varied random seeds.

Appendix D. Compute Infrastructure

In Table 1, we list the compute infrastructure that we use to run all of the algorithms.

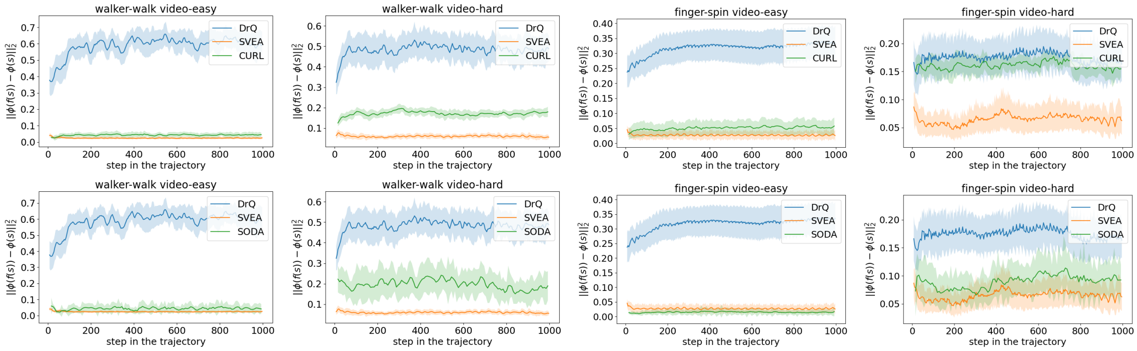


Figure 10: **Detailed plots of representation deviation comparison.** We show the comparison between DrQ, CURL, SODA, and SVEA on video-easy and video-hard settings of walker-walk and finger-spin tasks from DMC-GB. The results are averaged over 5 varied random seeds.

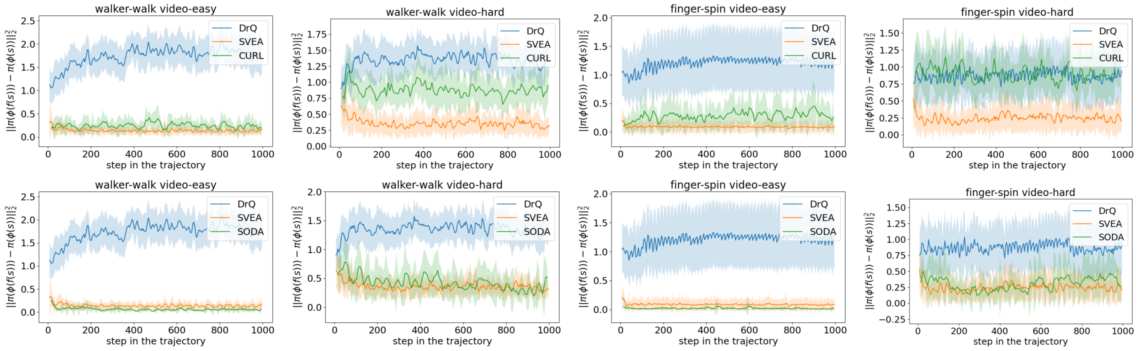


Figure 11: **Detailed plots of policy deviation comparison.** We show the comparison between DrQ, CURL, SODA, and SVEA on video-easy and video-hard settings of walker-walk and finger-spin tasks from DMC-GB. The results are averaged over 5 varied random seeds.

References

Agarwal, A., Hsu, D. J., Kale, S., Langford, J., Li, L., & Schapire, R. E. (2014). Taming the Monster: A Fast and Simple Algorithm for Contextual Bandits. In *International Conference on Machine Learning*.

Agarwal, R., Machado, M. C., Castro, P. S., & Bellemare, M. G. (2021). Contrastive

Table 1: Compute infrastructure.

CPU	**GPU**	**Memory**
AMD EPYC 7452	RTX3090×8	288GB

Behavioral Similarity Embeddings for Generalization in Reinforcement Learning. In *International Conference on Learning Representations*.

Agrawal, S., & Goyal, N. (2012). Thompson Sampling for Contextual Bandits with Linear Payoffs. In *International Conference on Machine Learning*.

Bahl, S., Mukadam, M., Gupta, A. K., & Pathak, D. (2020). Neural Dynamic Policies for End-to-End Sensorimotor Learning. In *Neural Information Processing Systems*.

Bauer, M., & Mnih, A. (2021). Generalized Doubly Reparameterized Gradient Estimators. In *International Conference on Machine Learning*.

Bertoin, D., Zouitine, A., Zouitine, M., & Rachelson, E. (2022). Look where you look! Saliency-guided Q-networks for generalization in visual Reinforcement Learning. In *Neural Information Processing Systems*.

Bertrán, M., Martínez, N., Phielipp, M., & Sapiro, G. (2020). Instance based Generalization in Reinforcement Learning. In *Neural Information Processing Systems*.

Bradtke, S. J. (1992). Reinforcement Learning Applied to Linear Quadratic Regulation. In *Neural Information Processing Systems*.

Cao, Y., & Ren, W. (2010). Optimal Linear-Consensus Algorithms: An LQR Perspective. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *40*, 819–830.

Cao, Y., Li, Z., Yang, T., Zhang, H., Zheng, Y., Li, Y., Hao, J., & Liu, Y. (2022). Galois: Boosting deep reinforcement learning via generalizable logic synthesis. *ArXiv*, *abs/2205.13728*.

Chebotar, Y., Handa, A., Makoviychuk, V., Macklin, M., Issac, J., Ratliff, N. D., & Fox, D. (2018). Closing the Sim-to-Real Loop: Adapting Simulation Randomization with Real World Experience. In *2019 International Conference on Robotics and Automation (ICRA)*.

Ciosek, K., & Whiteson, S. (2020). Expected Policy Gradients for Reinforcement Learning. *Journal of Machine Learning Research*, *21*(52), 1–51.

Cobbe, K., Klimov, O., Hesse, C., Kim, T., & Schulman, J. (2018). Quantifying Generalization in Reinforcement Learning. In *International Conference on Machine Learning*.

Cole, E., Yang, X. S., Wilber, K., Aodha, O. M., & Belongie, S. J. (2021). When Does Contrastive Visual Representation Learning Work?. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Dabney, W., Ostrovski, G., Silver, D., & Munos, R. (2018). Implicit Quantile Networks for Distributional Reinforcement Learning. In *International Conference on Machine Learning*.

Dai, T., Arulkumaran, K., Tukra, S., Behbahani, F. M. P., & Bharath, A. A. (2019). Analysing Deep Reinforcement Learning Agents Trained with Domain Randomisation. *Neurocomputing*, *493*, 143–165.

Delfosse, Q., Shindo, H., Dhami, D. S., & Kersting, K. (2023). Interpretable and explainable logical policies via neurally guided symbolic abstraction. In *Neural Information Processing Systems*.

Delfosse, Q., Sztwiertnia, S., Stammer, W., Rothermel, M., & Kersting, K. (2024). Interpretable concept bottlenecks to align reinforcement learning agents. *ArXiv*, *abs/2401.05821*.

Derman, E., Mankowitz, D., Mann, T., & Mannor, S. (2020). A bayesian approach to robust reinforcement learning. In *Uncertainty in Artificial Intelligence*.

Dittadi, A., Träuble, F., Wüthrich, M., Widmaier, F., Gehler, P., Winther, O., Locatello, F., Bachem, O., Schölkopf, B., & Bauer, S. (2021). The Role of Pretrained Representations for the OOD Generalization of Reinforcement Learning Agents. *arXiv*, *arXiv/2107.05686*.

Doersch, C., Gupta, A. K., & Efros, A. A. (2015). Unsupervised Visual Representation Learning by Context Prediction. In *2015 IEEE International Conference on Computer Vision (ICCV)*.

D'Oro, P., & Jaśkowski, W. (2020). How to Learn a Useful Critic? Model-based Action-Gradient-Estimator Policy Optimization. In *Neural Information Processing Systems*.

Fan, L. J., Wang, G., Huang, D.-A., Yu, Z., Fei-Fei, L., Zhu, Y., & Anandkumar, A. (2021). SECANT: Self-Expert Cloning for Zero-Shot Generalization of Visual Policies. In *International Conference on Machine Learning*.

Fazlyab, M., Robey, A., Hassani, H., Morari, M., & Pappas, G. J. (2019). Efficient and Accurate Estimation of Lipschitz Constants for Deep Neural Networks. In *Neural Information Processing Systems*.

Figurnov, M., Mohamed, S., & Mnih, A. (2018). Implicit Reparameterization Gradients. In *Neural Information Processing Systems*.

Finn, C., Tan, X. Y., Duan, Y., Darrell, T., Levine, S., & Abbeel, P. (2015). Learning Visual Feature Spaces for Robotic Manipulation with Deep Spatial Autoencoders. *ArXiv*, *abs/1509.06113*.

Grooten, B., Sokar, G., Dohare, S., Mocanu, E., Taylor, M. E., Pechenizkiy, M., & Mocanu, D. C. (2023). Automatic noise filtering with dynamic sparse training in deep reinforcement learning. In *Adaptive Agents and Multi-Agent Systems*.

Gumbel, E. J. (1954). Statistical theory of extreme values and some practical applications : A series of lectures. Technical report.

Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018a). Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In *International Conference on Machine Learning*.

Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., Abbeel, P., & Levine, S. (2018b). Soft Actor-Critic Algorithms and Applications. *ArXiv*, *abs/1812.05905*.

Hafner, D., Lillicrap, T. P., Fischer, I. S., Villegas, R., Ha, D. R., Lee, H., & Davidson, J. (2018). Learning Latent Dynamics for Planning from Pixels. In *International Conference on Machine Learning*.

Hansen, N., Jangir, R., Sun, Y., Alenyà, G., Abbeel, P., Efros, A. A., Pinto, L., & Wang, X. (2021a). Self-Supervised Policy Adaptation during Deployment. In *International Conference on Learning Representations*.

Hansen, N., Su, H., & Wang, X. (2021b). Stabilizing Deep Q-Learning with ConvNets and Vision Transformers under Data Augmentation. In *Neural Information Processing Systems*.

Hansen, N., & Wang, X. (2020). Generalization in Reinforcement Learning by Soft Data Augmentation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*.

Hansen, N., Wang, X., & Su, H. (2022). Temporal Difference Learning for Model Predictive Control. In *International Conference on Machine Learning*.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Heess, N. M. O., Wayne, G., Silver, D., Lillicrap, T. P., Erez, T., & Tassa, Y. (2015). Learning Continuous Control Policies by Stochastic Value Gradients. In *Neural Information Processing Systems*.

Huang, Y., Peng, P., Zhao, Y., Chen, G., & Tian, Y. (2022). Spectrum Random Masking for Generalization in Image-based Reinforcement Learning. In *Neural Information Processing Systems*.

Huijben, I. A. M., Kool, W., Paulus, M. B., & van Sloun, R. J. G. (2021). A Review of the Gumbel-max Trick and its Extensions for Discrete Stochasticity in Machine Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *45*, 1353–1371.

Jaderberg, M., Mnih, V., Czarnecki, W. M., Schaul, T., Leibo, J. Z., Silver, D., & Kavukcuoglu, K. (2017). Reinforcement Learning with Unsupervised Auxiliary Tasks. In *International Conference on Learning Representations*.

Jaksch, T., Ortner, R., & Auer, P. (2008). Near-optimal Regret Bounds for Reinforcement Learning. *Journal of Machine Learning Research*, *11*, 1563–1600.

Jang, E., Gu, S., & Poole, B. (2017). Categorical Reparameterization with Gumbel-Softmax. In *International Conference on Learning Representations*.

Jiang, Z., & Luo, S. (2019). Neural logic reinforcement learning. In *International Conference on Machine Learning*.

Joo, W., Kim, D., Shin, S.-J., & Moon, I.-C. (2020). Generalized Gumbel-Softmax Gradient Estimator for Various Discrete Random Variables. *ArXiv*, *abs/2003.01847*.

Jost, M., Pannocchia, G., & Mönnigmann, M. (2017). Accelerating Linear Model Predictive Control by Constraint Removal. *European Journal of Control*, *35*, 42–49.

Kingma, D. P., Salimans, T., & Welling, M. (2015). Variational Dropout and the Local Reparameterization Trick. In *Neural Information Processing Systems*.

Kingma, D. P., & Welling, M. (2013). Auto-encoding Variational Bayes. *arXiv*, *arXiv/1312.6114*.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollár, P., & Girshick, R. B. (2023). Segment Anything. *ArXiv*, *abs/2304.02643*.

Kolesnikov, A., Beyer, L., Zhai, X., Puigcerver, J., Yung, J., Gelly, S., & Houlsby, N. (2019a). Big Transfer (BiT): General Visual Representation Learning. In *European Conference on Computer Vision*.

Kolesnikov, A., Zhai, X., & Beyer, L. (2019b). Revisiting Self-Supervised Visual Representation Learning. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kulhánek, J., Derner, E., de Bruin, T., & Babuka, R. (2019). Vision-based Navigation Using Deep Reinforcement Learning. In *2019 European Conference on Mobile Robots (ECMR)*.

Laskin, M., Lee, K., Stooke, A., Pinto, L., Abbeel, P., & Srinivas, A. (2020). Reinforcement Learning with Augmented Data. In *Neural Information Processing Systems*.

Le Lan, C., Tu, S., Oberman, A., Agarwal, R., & Bellemare, M. G. (2022). On the Generalization of Representations in Reinforcement Learning. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*.

Lecarpentier, E., Abel, D., Asadi, K., Jinnai, Y., Rachelson, E., & Littman, M. L. (2020). Lipschitz Lifelong Reinforcement Learning. In *AAAI Conference on Artificial Intelligence*.

Lee, A. X., Nagabandi, A., Abbeel, P., & Levine, S. (2019a). Stochastic Latent Actor-Critic: Deep Reinforcement Learning with a Latent Variable Model. In *Neural Information Processing Systems*.

Lee, K., Lee, K., Shin, J., & Lee, H. (2019b). Network Randomization: A Simple Technique for Generalization in Deep Reinforcement Learning. In *International Conference on Learning Representations*.

Li, H., Pan, S. J., Wang, S., & Kot, A. C. (2018). Domain Generalization with Adversarial Feature Learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5400–5409.

Li, L., Lyu, J., Ma, G., Wang, Z., Yang, Z., Li, X., & Li, Z. (2023). Normalization Enhances Generalization in Visual Reinforcement Learning. *ArXiv*, *abs/2306.00656*.

Li, Z., Ratliff, L. J., Nassif, H., Jamieson, K. G., & Jain, L. P. (2022). Instance-optimal PAC Algorithms for Contextual Bandits. In *Neural Information Processing Systems*.

Lin, X., Baweja, H. S., Kantor, G. A., & Held, D. (2019). Adaptive Auxiliary Task Weighting for Reinforcement Learning. In *Neural Information Processing Systems*.

Liu, H.-T. D., Williams, F., Jacobson, A., Fidler, S., & Litany, O. (2022). Learning Smooth Neural Functions via Lipschitz Regularization. In *ACM SIGGRAPH 2022 Conference Proceedings*.

Lorberbom, G., Johnson, D. D., Maddison, C. J., Tarlow, D., & Hazan, T. (2021). Learning Generalized Gumbel-max Causal Mechanisms. In *Neural Information Processing Systems*.

Luo, L., Zhang, G., Xu, H., Yang, Y., Fang, C., & Li, Q. (2024). Insight: End-to-end neuro-symbolic visual reinforcement learning with language explanations. *ArXiv*, *abs/2403.12451*.

Lyle, C., Rowland, M., Ostrovski, G., & Dabney, W. (2021). On The Effect of Auxiliary Tasks on Representation Dynamics. In *International Conference on Artificial Intelligence and Statistics*.

Lyu, J., Wan, L., Li, X., & Lu, Z. (2024). Off-policy rl algorithms can be sample-efficient for continuous control via sample multiple reuse. *Information Sciences*, *666*, 120371.

Maddison, C. J., Mnih, A., & Teh, Y. W. (2016). The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. *ArXiv*, *abs/1611.00712*.

Majumdar, A., Yadav, K., Arnaud, S., Ma, Y. J., Chen, C., Silwal, S., Jain, A., Berges, V.-P., Abbeel, P., Malik, J., Batra, D., Lin, Y., Maksymets, O., Rajeswaran, A., & Meier, F. (2023). Where are we in the search for an Artificial Visual Cortex for Embodied Intelligence?. *ArXiv*, *abs/2303.18240*.

Maksymets, O., Cartillier, V., Gokaslan, A., Wijmans, E., Galuba, W., Lee, S., & Batra, D. (2021). THDA: Treasure Hunt Data Augmentation for Semantic Navigation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*.

Mitrano, P., & Berenson, D. (2022). Data Augmentation for Manipulation. *ArXiv*, *abs/2205.02886*.

Miyato, T., Kataoka, T., Koyama, M., & Yoshida, Y. (2018). Spectral Normalization for Generative Adversarial Networks. In *International Conference on Learning Representations*.

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018). *Foundations of Machine Learning*. MIT press.

Morimoto, J., & Doya, K. (2001). Robust Reinforcement Learning. In *Neural Information Processing Systems*.

Mostafa, H., & Wang, X. (2019). Parameter Efficient Training of Deep Convolutional Neural Networks by Dynamic Sparse Reparameterization. In *International Conference on Machine Learning*.

Motiian, S., Piccirilli, M., Adjeroh, D. A., & Doretto, G. (2017). Unified Deep Supervised Domain Adaptation and Generalization. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 5716–5726.

Nair, A., Pong, V. H., Dalal, M., Bahl, S., Lin, S., & Levine, S. (2018). Visual reinforcement learning with imagined goals. In *Neural Information Processing Systems*.

Oberst, M., & Sontag, D. A. (2019). Counterfactual Off-Policy Evaluation with Gumbel-Max Structural Causal Models. In *International Conference on Machine Learning*.

Packer, C., Gao, K., Kos, J., Krähenbühl, P., Koltun, V., & Song, D. X. (2018). Assessing Generalization in Deep Reinforcement Learning. *ArXiv*, *abs/1810.12282*.

Peng, X. B., Andrychowicz, M., Zaremba, W., & Abbeel, P. (2017). Sim-to-Real Transfer of Robotic Control with Dynamics Randomization. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*.

Perkins, T. J., & Barto, A. G. (2001). Lyapunov-Constrained Action Sets for Reinforcement Learning. In *International Conference on Machine Learning*.

Pinto, L., Andrychowicz, M., Welinder, P., Zaremba, W., & Abbeel, P. (2017). Asymmetric Actor Critic for Image-Based Robot Learning. *ArXiv*, *abs/1710.06542*.

Polvara, R., Patacchiola, M., Hanheide, M., & Neumann, G. (2020). Sim-to-Real Quadrotor Landing via Sequential Deep Q-Networks and Domain Randomization. *Robotics*, *9*, 8.

Potapczynski, A., Loaiza-Ganem, G., & Cunningham, J. P. (2019). Invertible Gaussian Reparameterization: Revisiting the Gumbel-Softmax. *ArXiv*, *abs/1912.09588*.

Raileanu, R., Goldstein, M., Yarats, D., Kostrikov, I., & Fergus, R. (2021). Automatic Data Augmentation for Generalization in Reinforcement Learning. In *Neural Information Processing Systems*.

Recht, B. (2019). A Tour of Reinforcement Learning: The View from Continuous Control. *Annual Review of Control, Robotics, and Autonomous Systems*, *2*, 253–279.

Ruiz, F. J. R., Titsias, M. K., & Blei, D. M. (2016). The Generalized Reparameterization Gradient. In *Neural Information Processing Systems*.

Salimans, T., & Kingma, D. P. (2016). Weight Normalization: A Simple Reparameterization to Accelerate Training of Deep Neural Networks. In *Neural Information Processing Systems*.

Scaman, K., & Virmaux, A. (2018). Lipschitz Regularity of Deep Neural Networks: Analysis and Efficient Estimation. In *Neural Information Processing Systems*.

Schwarzer, M., Anand, A., Goel, R., Hjelm, R. D., Courville, A. C., & Bachman, P. (2020). Data-Efficient Reinforcement Learning with Self-Predictive Representations. In *International Conference on Learning Representations*.

Sehnke, F., Osendorfer, C., Rückstiess, T., Graves, A., Peters, J., & Schmidhuber, J. (2008). Policy Gradients with Parameter-Based Exploration for Control. In *International Conference on Artificial Neural Networks*.

Shalev-Shwartz, S., & Ben-David, S. (2014). *Understanding machine learning: From theory to algorithms*. Cambridge university press.

Simchowitz, M., & Foster, D. J. (2020). Naive Exploration is Optimal for Online LQR. In *International Conference on Machine Learning*.

Slaoui, R. B., Clements, W. R., Foerster, J. N., & Toth, S. (2019). Robust Domain Randomization for Reinforcement Learning. *ArXiv*, *abs/1910.10537*.

Slivkins, A. (2011). Contextual bandits with similarity information. In *Proceedings of the 24th annual Conference On Learning Theory*.

Song, X., Jiang, Y., Tu, S., Du, Y., & Neyshabur, B. (2020). Observational Overfitting in Reinforcement Learning. In *International Conference on Learning Representations*.

Srinivas, A., Laskin, M., & Abbeel, P. (2020). CURL: Contrastive Unsupervised Representations for Reinforcement Learning. In *International Conference on Machine Learning*.

Stone, A., Ramirez, O., Konolige, K., & Jonschkowski, R. (2021). The Distracting Control Suite - A Challenging Benchmark for Reinforcement Learning from Pixels. *ArXiv*, *abs/2101.02722*.

Stooke, A., Lee, K., Abbeel, P., & Laskin, M. (2020). Decoupling Representation Learning from Reinforcement Learning. In *International Conference on Machine Learning*.

Sun, Y., Wang, X., Liu, Z., Miller, J., Efros, A. A., & Hardt, M. (2019). Test-Time Training with Self-Supervision for Generalization under Distribution Shifts. In *International Conference on Machine Learning*.

Tessler, C., Efroni, Y., & Mannor, S. (2019). Action Robust Reinforcement Learning and Applications in Continuous Control. In *International Conference on Machine Learning*.

Tschannen, M., Bachem, O., & Lucic, M. (2018). Recent Advances in Autoencoder-Based Representation Learning. *ArXiv*, *abs/1812.05069*.

Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P.-A. (2008). Extracting and Composing Robust Features with Denoising Autoencoders. In *International Conference on Machine Learning*.

Wang, H., Zheng, S., Xiong, C., & Socher, R. (2019). On the Generalization Gap in Reparameterizable Reinforcement Learning. In *International Conference on Machine Learning*.

Wang, X., Lian, L., & Yu, S. X. (2021). Unsupervised Visual Attention and Invariance for Reinforcement Learning. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wei, E., Wicke, D., & Luke, S. (2018). Hierarchical Approaches for Reinforcement Learning in Parameterized Action Space. In *AAAI*.

Wu, Y., Fan, Y., Liang, P. P., Azaria, A., Li, Y.-F., & Mitchell, T. M. (2023). Read and reap the rewards: Learning to play atari with the help of instruction manuals. In *Neural Information Processing Systems*.

Xu, M., Quiroz, M., Kohn, R., & Sisson, S. A. (2018). Variance Reduction Properties of the Reparameterization Trick. In *International Conference on Artificial Intelligence and Statistics*.

Yang, M., Levine, S., & Nachum, O. (2022). TRAIL: Near-Optimal Imitation Learning with Suboptimal Data. In *International Conference on Learning Representations*.

Yarats, D., Fergus, R., Lazaric, A., & Pinto, L. (2021). Reinforcement Learning with Prototypical Representations. In *International Conference on Machine Learning*.

Yarats, D., Fergus, R., Lazaric, A., & Pinto, L. (2022). Mastering visual continuous control: Improved data-augmented reinforcement learning. In *International Conference on Learning Representations*.

Ye, W., Liu, S.-W., Kurutach, T., Abbeel, P., & Gao, Y. (2021). Mastering Atari Games with Limited Data. In *Neural Information Processing Systems*.

Yuan, Z., Xue, Z., Yuan, B., Wang, X., Wu, Y., Gao, Y., & Xu, H. (2022). Pre-Trained Image Encoder for Generalizable Visual Reinforcement Learning. In *Neural Information Processing Systems*.

Yue, X., Zhang, Y., Zhao, S., Sangiovanni-Vincentelli, A. L., Keutzer, K., & Gong, B. (2019). Domain Randomization and Pyramid Consistency: Simulation-to-Real Generalization Without Accessing Target Domain Data. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*.

Zambaldi, V. F., Raposo, D., Santoro, A., Bapst, V., Li, Y., Babuschkin, I., Tuyls, K., Reichert, D. P., Lillicrap, T. P., Lockhart, E., Shanahan, M., Langston, V., Pascanu, R., Botvinick, M. M., Vinyals, O., & Battaglia, P. W. (2018). Relational deep reinforcement learning. *ArXiv*, *abs/1806.01830*.

Ze, Y., Hansen, N., Chen, Y., Jain, M., & Wang, X. (2022). Visual Reinforcement Learning With Self-Supervised 3D Representations. *IEEE Robotics and Automation Letters*, *8*, 2890–2897.

Zhang, A., Ballas, N., & Pineau, J. (2018). A Dissection of Overfitting and Generalization in Continuous Reinforcement Learning. *ArXiv*, *abs/1806.07937*.

Zhang, A., McAllister, R. T., Calandra, R., Gal, Y., & Levine, S. (2021). Learning Invariant Representations for Reinforcement Learning without Reconstruction. In *International Conference on Learning Representations*.

Zhang, C., Vinyals, O., Munos, R., & Bengio, S. (2018). A Study on Overfitting in Deep Reinforcement Learning. *ArXiv*, *abs/1804.06893*.

Zhang, H., Chen, H., Boning, D. S., & Hsieh, C.-J. (2021). Robust Reinforcement Learning on State Observations with Learned Optimal Adversary. In *International Conference on Learning Representations*.

Zheng, W., Sharan, S. P., Fan, Z., Wang, K., Xi, Y., & Wang, Z. (2022). Symbolic visual reinforcement learning: A scalable framework with object-level abstraction and differentiable expression search. *ArXiv*, *abs/2212.14849*.

Zhou, K., Liu, Z., Qiao, Y., Xiang, T., & Loy, C. C. (2021). Domain Generalization: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *45*, 4396–4415.