Normalization Enhances Generalization in Visual Reinforcement Learning

Lu Li*
Tsinghua University
Beijing, China
lilu21@mails.tsinghua.edu.cn

Jiafei Lyu*
Tsinghua University
Beijing, China
lvjf20@mails.tsinghua.edu.cn

Guozheng Ma
Tsinghua University
Beijing, China
mgz21@mails.tsinghua.edu.cn

Zilin Wang
Tsinghua University
Beijing, China
wangzl21@mails.tsinghua.edu.cn

Zhenjie Yang
Shanghai Jiao Tong University
Shanghai, China
yangzhenjie@sjtu.edu.cn

Xiu Li†
Tsinghua University
Beijing, China
li.xiu@sz.tsinghua.edu.cn

Zhiheng Li†
Tsinghua University
Beijing, China
zhhli@tsinghua.edu.cn

ABSTRACT

Recent advances in visual reinforcement learning (RL) have led to impressive success in handling complex tasks. However, these methods have demonstrated limited generalization capability to visual disturbances, which poses a significant challenge to their real-world application and adaptability. Though normalization techniques have demonstrated huge success in supervised and unsupervised learning, their applications in visual RL are still scarce. In this paper, we explore the potential benefits of integrating normalization into visual RL methods with respect to generalization performance. We find that, perhaps surprisingly, incorporating suitable normalization techniques is sufficient to enhance the generalization capabilities, without any additional special design. We utilize the combination of two normalization techniques, CrossNorm and SelfNorm, for generalizable visual RL. Extensive experiments are conducted on DMControl Generalization Benchmark, CARLA, and ProcGen Benchmark to validate the effectiveness of our method. We show that our method significantly improves generalization capability while only marginally affecting sample efficiency. In particular, when integrated with DrQ-v2, our method enhances the test performance of DrQ-v2 on CARLA across various scenarios, from 14% of the training performance to **97%**. Our project page: https://sites.google.com/view/norm-generalization-vrl/home

KEYWORDS

generalization; visual reinforcement learning; normalization

ACM Reference Format:
Lu Li*, Jiafei Lyu*, Guozheng Ma, Zilin Wang, Zhenjie Yang, Xiu Li†, and Zhiheng Li†. 2024. Normalization Enhances Generalization in Visual Reinforcement Learning. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024,* IFAAMAS, 10 pages.

1 INTRODUCTION

Visual reinforcement learning (RL), which leverages high-dimensional visual observations as inputs, has shown potential in a wide range of tasks, such as playing video games [34, 49] and robotic manipulation [27]. However, generalization remains a major challenge for visual RL methods. Even slight alterations, such as color or background changes, can result in considerable performance degradation in the testing environment, which in turn limits the real-world utility of these algorithms. It is vital to develop techniques that can improve the generalization capabilities of visual RL algorithms.

Existing literature mainly enhances the generalization capability of visual RL via data augmentation [13, 17, 50, 56] and domain randomization [36, 37, 47], aiming at learning policies invariant to the changes in the observations. However, recent studies [26, 55] show that certain data augmentation techniques may lead to a decrease in sample efficiency and even cause divergence. Other recent works improve the generalization performance by leveraging pre-trained image encoder [57] or segmenting important pixels from the test environment [2], *etc.* Unfortunately, most of them rely on knowledge or data from outer sources, *e.g.*, ImageNet [10]. We deem that an ideal method for zero-shot generalization should be able to achieve robust performance without relying on any out-of-domain data or prior knowledge of the target domain, and should be able to adapt effectively to a wide variety of environments.

Normalization techniques have achieved huge success in computer vision [45, 48, 52] and natural language processing [1, 51, 53]. Numerous normalization-related methods are proposed to improve the generalization capabilities of deep neural networks [14, 21, 41, 43]. Despite their popularity, normalization techniques have not

*: Equal contribution. †: Corresponding authors.

This work is licensed under a Creative Commons Attribution International 4.0 License.

Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand. © 2024 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

received enough attention in deep RL community. Though previous studies have investigated the effectiveness of normalization methods, *e.g.*, layer normalization [20, 35] and spectral normalization [5, 16, 32, 33], in deep RL algorithms, to the best of our knowledge, it is still unclear whether normalization can aid generalization in visual RL. We hence ask the following question:

Can we develop a visual RL agent that employs normalization techniques and does not rely on prior knowledge and out-of-domain data, enabling it to generalize more effectively to unseen scenarios?

This inquiry drives our exploration of CrossNorm and SelfNorm [45], two normalization methods that have been proven to enhance generalization in computer vision tasks under distribution shifts. Since visual RL algorithms always rely on the encoder to output representations for policy learning, we need to ensure that the learned representation can generalize to unseen scenarios. To fulfill that, we propose to modify the encoder structure of the base visual RL algorithm by incorporating CrossNorm and SelfNorm for the downstream tasks. Our proposed normalization module is plug-and-play, and can be combined with any existing visual RL algorithms.

We assess the effectiveness of our approach using three benchmarks: DeepMind Control Generalization Benchmark [18], a benchmark designed for evaluating generalization capabilities in robotic control tasks; CARLA [11], a realistic simulator for autonomous driving; and ProcGen [8], which features procedurally-generated environments to directly measure an agent's generalization capability. Extensive experimental results demonstrate that when combined with DrQ [55], DrQ-v2 [54], and PPO [42], our proposed normalization module significantly improves their generalization capabilities without requiring any task-specific modifications or prior knowledge. Furthermore, our proposed module demonstrates compatibility and synergy with other generalization algorithms in visual RL (*e.g.*, SVEA [17]), thereby further enhancing their generalization. This indicates the flexibility of our proposed module and its potential to be a valuable addition to the toolset for improving generalization in visual RL tasks. We believe this work offers another chance that allows visual RL algorithms to exhibit greater adaptability and robustness across diverse and dynamic environments. We aspire to propel the field of visual RL forward and broaden the scope of the potential applications of normalization techniques.

2 RELATED WORK

Generalization in Visual RL. Over the past few years, considerable strides have been made towards narrowing the generalization gap in visual RL. An elementary strategy for improving generalization is to employ regularization techniques, initially developed for supervised learning [29]. These techniques include ℓ_2 regularization [15], entropy regularization [60], and dropout [19]. Unfortunately, these conventional regularization techniques exhibit limited effectiveness in improving generalization of visual RL and, in some cases, they may even have a negative impact on sample efficiency [9, 22]. As a result, recent studies have shifted their focus towards learning robust representations by leveraging bisimulation metrics [24, 58], multi-view information bottleneck (MIB) [12], pre-trained image encoder [57], *etc.* From an orthogonal perspective, data augmentation has demonstrated significant efficacy in enhancing generalization by leveraging prior knowledge as an inductive bias for the agent [17, 26, 31, 56]. However, the effectiveness of data augmentation-based techniques is significantly constrained by their highly task-specific nature and the requirement for substantial expert knowledge [25, 38]. On the one hand, applying appropriate data augmentation techniques demands domain-specific knowledge, which limits their applicability to unfamiliar or novel environments. On the other hand, these techniques face challenges in generalizing to new domains due to their reliance on the alignment between augmentations and domain characteristics. In this study, our objective is to explore the utilization of normalization techniques to enhance the generalizability of visual RL, without relying on specific prior knowledge of the shift characteristics between the train and test environments. We note that two recent studies [25, 31] present a comprehensive analysis of the generalization challenges in RL and the application of data augmentation in visual RL, which can be a nice reference.

Normalization. Normalization techniques play a crucial role in training deep neural networks [30, 39, 52]. They notably enhance optimization by normalizing input features, which is particularly advantageous for first-order optimization algorithms such as Stochastic Gradient Descent [6, 62], known to excel in more isotropic landscapes [7]. Batch Normalization [4, 23, 40] (BN) is a method that normalizes intermediate feature maps using statistics computed from mini-batch samples. It has been found to significantly aid in the training of deep networks. Drawing inspiration from the success of BN, a variety of normalization techniques have since been introduced to accommodate different learning scenarios, *e.g.*, layer normalization [1, 44, 59], spectral normalization [28, 33], *etc.*

Despite the huge success and wide applications of normalization techniques, they are not commonly employed in deep RL. This is largely attributed to the online learning nature of RL, which leads to a non-independent and identically distributed (non-i.i.d) input data distribution. Such distribution does not align with the requirements of many normalization techniques. [3] shows that direct application of BN and LN proves to be ineffective for RL. Instead, it introduces cross-normalization, which computes mean feature subtraction using both on-policy and off-policy state-action pairs, leading to better sample efficiency. Moreover, spectral normalization has been found to be effective in stabilizing the training process of RL [5, 16].

It is interesting to ask: since normalization techniques have shown benefits for generalization to new tasks in computer vision, then whether normalization techniques have the potential to enhance the generalization ability of the visual RL algorithms. To the best of our knowledge, none of the prior work explores this issue, and our goal in this work is to answer this question.

3 PRELIMINARY

3.1 Visual Reinforcement Learning

We consider learning in a Partially Observable Markov Decision Processes (POMDPs) specified by the tuple $\mathcal{M} : \langle \mathcal{S}, \mathcal{O}, \mathcal{A}, \mathcal{P}, r, \gamma \rangle$, where $\mathcal{S}$ is the state space, $\mathcal{O}$ is the observation space, $\mathcal{A}$ is the action space, $\mathcal{P}(\cdot|s, a) : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ is the transition probability, $r(s, a) : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ is the scalar reward function, $\gamma \in [0, 1)$ is the discount factor. In the context of the generalization setting, we have a set of such POMDPs $M = \{\mathcal{M}_0, \mathcal{M}_1, \ldots, \mathcal{M}_n\}$ while our agent only has access to one fixed POMDP among them, denoted as $\mathcal{M}_0$.

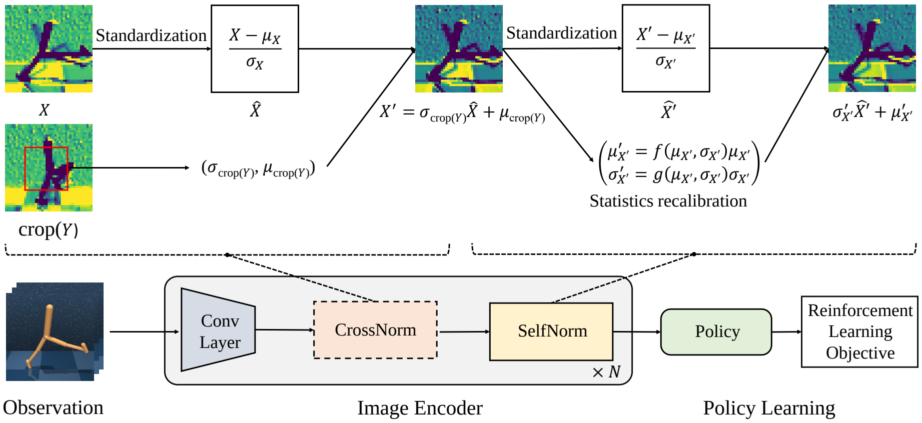
Standardization
$\frac{X-\mu_X}{\sigma_X}$
X
$\hat{X}$
crop(Y)
$(\sigma_{\text{crop}(Y)}, \mu_{\text{crop}(Y)})$
$X' = \sigma_{\text{crop}(Y)}\hat{X} + \mu_{\text{crop}(Y)}$
Standardization
$\frac{X'-\mu_{X'}}{\sigma_{X'}}$
$\widehat{X'}$
$\sigma'_{X'}\widehat{X'} + \mu'_{X'}$
$\begin{pmatrix}\mu'_{X'} = f(\mu_{X'}, \sigma_{X'})\mu_{X'} \\ \sigma'_{X'} = g(\mu_{X'}, \sigma_{X'})\sigma_{X'}\end{pmatrix}$
Statistics recalibration
Conv Layer
CrossNorm
SelfNorm
× N
Policy
Reinforcement Learning Objective
Observation
Image Encoder
Policy Learning

Figure 1: The pipeline of our method. CrossNorm is positioned after the convolutional layer and is followed by SelfNorm. Each CrossNorm layer is randomly activated during training and becomes inactive during testing. Instead, SelfNorm is adopted during training and remains functional during testing. Our method notably does not introduce new learning objectives or utilize out-of-domain (OOD) data. A comprehensive visualization is available in Appendix A for further details.

We aim to train an RL agent to learn a policy $\pi_\theta(\cdot|s)$ parameterized by the parameter θ in $\mathcal{M}_0$, with the objective of maximizing the expected cumulative return $J(\theta) = \mathbb{E}_{a_t\sim\pi_\theta(\cdot|s_t), s_t\sim\mathcal{P}}[\sum_{t=0}^{T} \gamma^t r(s_t, a_t)]$ across the entire set of POMDPs in a zero-shot manner, where T is the horizon of the POMDP.

3.2 CrossNorm and SelfNorm

CrossNorm and SelfNorm [45] were initially introduced to improve generalization capabilities in the face of distribution shifts within computer vision tasks. To expand the training distribution, CrossNorm interchanges the mean and standard deviation between channels. Specifically, given a batch of feature maps, for each feature map x, randomly select another feature map y. For each channel i within both x and y, we define 'channel $\mathcal{X}$' from x and 'channel $\mathcal{Y}$' from y to be the corresponding channels. The mean and standard deviation of channel $\mathcal{X}$ are represented as $\mu_{\mathcal{X}}$ and $\sigma_{\mathcal{X}}$, respectively. Similarly, for channel $\mathcal{Y}$, they are denoted as $\mu_{\mathcal{Y}}$ and $\sigma_{\mathcal{Y}}$. Essentially, CrossNorm replaces the μ and σ values of channels $\mathcal{X}$ with those of $\mathcal{Y}$, as delineated in the subsequent equation 1:

$$\mathcal{X} = \sigma_{\mathcal{Y}} \frac{\mathcal{X} - \mu_{\mathcal{X}}}{\sigma_{\mathcal{X}}} + \mu_{\mathcal{Y}} \tag{1}$$

While CrossNorm enlarges the training distribution, the motivation of SelfNorm is to bridge the train-test distribution gap. To that end, SelfNorm replaces $\mathcal{X}$ with recalibrated mean $\mu'_{\mathcal{X}} = f(\mu_{\mathcal{X}}, \sigma_{\mathcal{X}})\mu_{\mathcal{X}}$ and standard deviation $\sigma'_{\mathcal{X}} = g(\mu_{\mathcal{X}}, \sigma_{\mathcal{X}})\sigma_{\mathcal{X}}$, where f and g are the attention functions. The adjusted feature becomes as Equation 2:

$$\hat{\mathcal{X}} = \sigma'_{\mathcal{X}} \frac{\mathcal{X} - \mu_{\mathcal{X}}}{\sigma_{\mathcal{X}}} + \mu'_{\mathcal{X}} \tag{2}$$

As f and g learn to scale $\mu_{\mathcal{X}}$ and $\sigma_{\mathcal{X}}$ based on their values, the method adapts to the specific characteristics of the data. While CrossNorm expands the data distribution, SelfNorm aims to emphasize the discriminative styles shared by both training and test distributions while de-emphasizing the insignificant styles.

4 METHOD

4.1 Enhancing Generalization in Visual RL via Normalization

The primary challenge in visual RL generalization stems from distribution shifts in observations. This issue is particularly prominent due to the diverse and dynamic nature of environments in RL tasks. Recognizing the proven effectiveness of CrossNorm and SelfNorm in bolstering generalization under distribution shift in computer vision tasks, we explore the possibilities of these normalization techniques in visual RL. By integrating CrossNorm and SelfNorm, we aim to enhance the generalization capability of visual RL, fostering the learning of more robust and generalizable representations.

Although computer vision tasks and visual RL tasks both involve the representation learning of visual input, their respective data distributions can be quite different. While CrossNorm is inspired by the observation that computer vision datasets are typically rich and diverse, stemming from a variety of sources, visual RL generally involves training the agent within a single task and environment. This situation results in a notably limited data distribution. In other words, the difference between the mean and standard deviation of channel $\mathcal{X}$ and channel $\mathcal{Y}$ tends to be small, thus diminishing the effect of the CrossNorm. Hence, it becomes crucial to further diversify and expand the data distribution. To achieve this, we utilize random cropping during the computation of the channel's mean μ and standard deviation σ, as illustrated in Equation 3. This technique can result in a wider distribution of the mean and the standard deviation values, further contributing to its ability to adapt to various data distributions.

$$\mathcal{X}' = \sigma_{\text{crop}(\mathcal{Y})} \frac{\mathcal{X} - \mu_{\mathcal{X}}}{\sigma_{\mathcal{X}}} + \mu_{\text{crop}(\mathcal{Y})} \tag{3}$$

We present the pipeline of our proposed method in Figure 1, where our core contribution is the proposal of a plug-and-play module that is equipped with cropped CrossNorm and SelfNorm. Notably, we arrange CrossNorm immediately after the convolution layer, followed by SelfNorm. This sequence is designed to optimally leverage the effects of these two operations, with CrossNorm augmenting the feature diversity before SelfNorm performs intra-instance normalization. Considering their characteristics, CrossNorm is activated only during the training phase, whereas SelfNorm is used during the training phase and remains functional during the testing phase.

During each forward pass in the training process, a predetermined number of CrossNorm layers are randomly activated. For these activated layers, each instance in the mini-batch has its μ and σ values for every channel swapped with those of the same channels of another randomly chosen instance. The remaining CrossNorm layers stay inactive during this process. Generally, how many CrossNorm layers can be activated strongly depends on how many hidden layers the encoder of the base algorithm has. We allow a dynamic utilization of the CrossNorm layers because unlike supervised learning, where the model usually has a strong supervised signal and various methods can be applied to learn task-relevant representations, visual RL lacks sufficient supervised signals. It is thus difficult for it to effectively capture important knowledge from the pixels. As a result, the training process in visual RL is often more fragile and susceptible to disruptions. By selecting an appropriate number of active CrossNorm layers during the training process, we can effectively manage the learning difficulty, ensuring more stable training dynamics in the learning process.

The role of CrossNorm can be seen as a form of data augmentation. However, unlike traditional data augmentation methods that have been used in visual RL, CrossNorm operates directly on the feature maps rather than the raw observations. This distinction allows CrossNorm to facilitate more diverse alterations. On the other hand, similar to traditional data augmentation methods, CrossNorm improves generalization at the cost of sample efficiency, while SelfNorm aims to offset this trade-off, thereby ensuring a more stable learning process. Our method does not introduce new learning objectives or require any out-of-domain data or prior knowledge. This makes it a self-contained and flexible approach to generalization. Moreover, our method is not only compatible with standard RL algorithms but can also be seamlessly integrated with other techniques aimed at enhancing the generalization of visual RL, and can further improve the robustness of these methods. This versatility further underscores the generality of our approach.

5 EXPERIMENTS

Our experiments are aimed to investigate the following questions: (a) Does our method enhance the generalization capabilities of vanilla visual RL methods and to what extent does it impact the training performance? (b) Is our proposed method general enough to be integrated with existing generalization methods in visual RL to further enhance their capability?

5.1 Generalization on CARLA Autonomous Driving Tasks

5.1.1 Experimental setup. To assess our method in realistic scenarios and better gauge its effectiveness and generalization capabilities, we evaluate the performance of our method in the CARLA autonomous driving simulator, which offers realistic observations and complex driving scenarios.

We build our method upon DrQ-v2 [54] and compare the generalization ability of DrQ-v2+CNSN with state-of-the-art methods and strong baselines: **DrQ-v2** [54]: our base visual RL algorithm, which is the prior state-of-the-art model-free visual RL algorithm in terms of sample efficiency. It demonstrates superior performance on a variety of tasks while maintaining high sample efficiency, making it a suitable foundation for our research in developing more generalizable visual RL methods. **SVEA** [17]: the previous state-of-the-art data augmentation based method for generalization, which achieves improved performance by reducing Q-variance through the use of an auxiliary loss.

Our experimental setup in CARLA is adapted from [58]. Specifically, we employ three cameras mounted on the vehicle's roof, each offering a 60-degree field of view. To train the RL agent and evaluate the final performance across various methods, we define the reward function as given in Equation 4:

$$r_t = \mathbf{v}_{\text{ego}}^{\top}\hat{\mathbf{u}}_{\text{highway}} \cdot \Delta t - \lambda_i \cdot \text{impulse} - \lambda_s \cdot |\text{steer}| \qquad (4)$$

where $\mathbf{v}_{\text{ego}}$ is the velocity vector of the ego vehicle, projected onto the highway's unit vector $\hat{\mathbf{u}}_{\text{highway}}$, and multiplied by the time discretization $\Delta t = 0.05$ to measure highway progression in meters. Collisions result in impulses, measured in Newton-seconds. A steering penalty is also applied, with steer $\in [-1, 1]$. The weights used in the reward function are $\lambda_i = 10^{-4}$ and $\lambda_s = 1$.

Due to the fact that the encoder in DrQ-v2 has four hidden layers, the maximum number of activated CrossNorm modules in DrQ-v2+CNSN is four. In CARLA experiments, all four CrossNorm layers are activated during the training phase. All agents are trained under one fixed weather condition for 200,000 environment steps. Their performance is then assessed across various other weather conditions within the same map and task, as shown in Figure 2. Moreover, it's worth noting that not only the visual observations change with different weather conditions, but also the dynamics of POMDP might vary due to factors like rain.

Since we employ DrQ-v2 as our base visual RL algorithm and baseline method, we also adapt and reimplement SVEA using the DrQ-v2 structure to ensure a fair comparison. We then train the two variations of SVEA on CARLA, one applying random convolution as data augmentation and the other employing random overlay with images from Places365 dataset [61], respectively.

5.1.2 Generalization performance. The generalization performance results are shown in Table 1. The results indicate that DrQ-v2 cannot adapt to new weather with different lighting, humidity, *etc.* However, by combing it with CNSN, DrQ-v2+CNSN is enough to generalize well on most of the unseen complicated scenes without a performance drop. Notably, DrQ-v2+CNSN significantly improves the test average performance from DrQ-v2's 14% of the training performance to **97%** of the training performance.

Moreover, it can be seen that both variants of SVEA, using random convolution and random overlay respectively, exhibit a significant performance drop in unseen weather conditions. For example, SVEA(conv) trained under HardRainNoon achieves an average return of 53 when tested under WetNoon, while DrQ-v2+CNSN attains an average performance of **173**, despite the fact that DrQ-v2+CNSN has lower training performance than SVEA(conv). The primary reason for the significant performance drop of SVEA is that

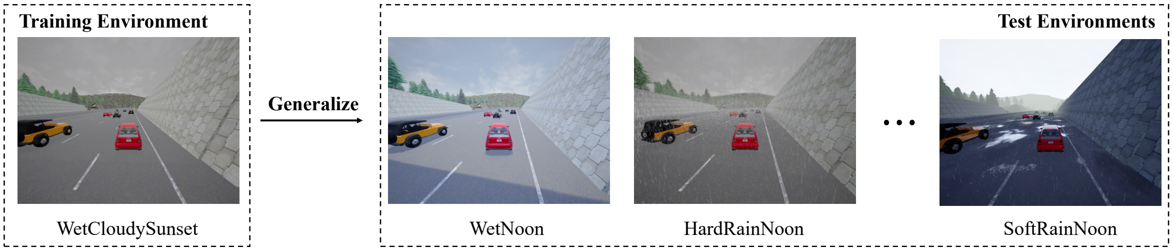

Figure 2: In CARLA autonomous driving simulator, agents are trained under one fixed weather condition. Then they are evaluated on unseen weather conditions in a zero-shot manner. These weather conditions vary in aspects such as lighting, humidity, and other factors, leading to differences not only in visual observation but also in the dynamics of the environment.

Table 1: CARLA generalization results. Training and testing performance (episode return) of methods trained in one fixed weather and evaluated on other 6 unseen weather conditions. We separately conduct training under two distinct weather conditions: WetCloudySunset (WCS) and HardRainNoon (HRN). SVEA(conv) refers to the variant of SVEA that utilizes random convolution for data augmentation, while SVEA(overlay) denotes the variant that employs random overlay for data augmentation. For a fair comparison, we have reimplemented these two versions of SVEA using DrQ-v2. The results presented are performance averaged over 5 random seeds, with each seed corresponding to 50 evaluation episodes for each weather condition.

Method	DrQ-v2		DrQ-v2+CNSN		SVEA(conv)		SVEA(overlay)	
Training Weather	WCS	HRN	WCS	HRN	WCS	HRN	WCS	HRN
Training	249±23	249±34	225±11	225±14	221±25	243±28	173±87	204±11
WetCloudySunset	249±23	118±43	225±11	**211±9**	221±25	184±18	173±87	30±21
MidRainSunset	184±18	−2±11	**233±32**	**208±11**	184±44	59±91	160±24	68±22
HardRainSunset	36±26	−3±10	**230±21**	**221±16**	169±41	79±93	148±31	87±18
WetNoon	2±6	5±4	**210±9**	**173±43**	82±85	51±53	1±6	−1±2
SoftRainNoon	−2±7	−6±8	**232±40**	**205±19**	101±90	59±69	57±50	14±26
MidRainyNoon	89±38	−3±8	**237±27**	**215±17**	190±38	69±95	143±29	166±36
HardRainNoon	145±20	249±34	**237±25**	225±14	190±36	243±28	146±25	204±11
Average test return	76±74	18±49	**230±29**	**206±27**	153±75	81±88	109±67	61±61

the two data augmentation techniques it employs do not align well with the test environments. Consequently, these augmentations do not provide sufficient generalization capability for unseen weather conditions, which ultimately limits SVEA's robustness in these scenarios. This finding underscores the necessity for more adaptable and versatile visual RL techniques that can effectively cope with the dynamic and intricate nature of real-world environments. Instead, our method does not rely on any task-specific data augmentation or prior knowledge, and can lead to more robust performance in a wide range of real-world scenarios.

5.2 Generalization on DMControl Generalization Benchmark

5.2.1 Experimental setup. We also assess our method on the DeepMind Control Generalization Benchmark (DMC-GB) [18], a well-established benchmark for evaluating the generalization capabilities of visual RL algorithms, based on DeepMind Control Suite [46]. In DMC-GB, agents are trained in standard DeepMind Control environments and subsequently evaluated in visually disturbed environments. These disturbances include changes in color (*color hard*) and the replacement of backgrounds with moving videos (*video easy*, *video hard*), as shown in Figure 3.

For the easy tasks in DeepMind Control Suites, we utilize DrQ as our base visual reinforcement learning algorithm. For medium tasks that DrQ struggles to solve, we employ DrQ-v2 due to its capability to address complex locomotion tasks using pixel observations, providing a more effective solution for these more challenging tasks. To ensure a fair comparison, we have re-implemented SVEA using DrQ-v2 as its base algorithm for medium tasks, considering that the original SVEA was implemented based on DrQ. Our experimental setting mainly follows that of SVEA [17]. For the easy tasks, all agents were trained for 500,000 steps in the vanilla training environments without visual alteration. Meanwhile, for the medium tasks, the training process is extended to 1,500,000 steps for all methods. Note that DrQ contains 11 hidden layers in its encoder while DrQ-v2 only has 4. Across our experiments, we randomly activate 5 out of 11 CrossNorm layers in DrQ-CNSN during the training phase and activate all 4 CrossNorm layers for DrQ-v2. Furthermore, we recognize PIE-G [57] as a state-of-the-art baseline, particularly effective in addressing the challenging *video*

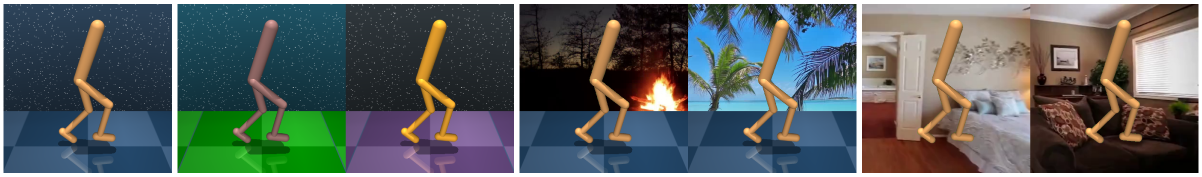

Figure 3: Examples of training and testing environments in DMC-GB. From left to right: training environment, *color hard* test environment, *video easy* test environment, and *video hard* test environment.

Table 2: DMC-GB generalization results. Performance on *video easy* and *video hard* testing environments. SVEA refers to the implementation of SVEA that utilizes random overlay as data augmentation method. All the results are averaged over 5 random seeds. *color hard* results can be found in Appendix A.

Easy tasks-*video easy*	DrQ	+CNSN	SVEA	+CNSN	PIE-G	+CNSN	RAD	SODA
Walker Walk	682±89	**792**±67	819±71	**842**±58	**917**±15	**923**±8	606±63	635±48
Walker Stand	873±83	**957**±12	961±8	**967**±6	**961**±7	**956**±9	745±146	903±56
Cartpole Swingup	485±105	**498**±26	**782**±27	752±26	**421**±76	353±40	373±72	474±143
Ball in cup Catch	318±157	**584**±83	871±106	**913**±45	854±54	**892**±43	481±26	539±111
Medium tasks-*video easy*	DrQ-v2	+CNSN	SVEA	+CNSN	PIE-G	+CNSN	RAD	SODA
Cheetah Run	42±19	**274**±35	**408**±78	**404**±29	327±54	**347**±34	203±1	317±37
Walker Run	124±31	**452**±22	**611**±20	**609**±18	520±16	**541**±17	178±11	505±38
Easy tasks-*video hard*	DrQ	+CNSN	SVEA	+CNSN	PIE-G	+CNSN	RAD	SODA
Walker Walk	104±22	**166**±28	377±93	**480**±46	633±59	**669**±42	80±10	312±32
Walker Stand	289±49	**492**±62	834±46	**871**±23	**902**±38	856±38	229±45	736±132
Cartpole Swingup	138±9	**171**±13	393±45	**417**±31	285±45	**309**±19	152±29	403±17
Ball in cup Catch	92±23	**199**±138	403±174	**691**±72	**741**±108	721±7	98±40	381±163
Medium tasks-*video hard*	DrQ-v2	+CNSN	SVEA	+CNSN	PIE-G	+CNSN	RAD	SODA
Cheetah Run	21±5	**49**±4	68±9	**88**±9	153±40	**162**±23	23±10	66±13
Walker Run	24±2	**43**±2	120±8	**148**±8	252±7	**281**±5	40±3	111±24

hard scenarios. We utilize the ResNet+CNSN pre-trained model, deactivating all CrossNorm and SelfNorm during the RL agent's training. For PIE-G+CNSN, a ResNet50+CNSN pre-trained model from [45] is employed. Meanwhile, for PIE-G, we use a ResNet50 pre-trained model from the *torchvision* package.

5.2.2 Generalization performance. To further assess the effectiveness and flexibility of the CrossNorm and SelfNorm in aiding the generalization ability of the visual RL policies, we build CrossNorm and SelfNorm on top of four visual RL algorithms, DrQ, DrQ-v2, SVEA, and PIE-G. We activate 5 out of 11 CrossNorm layers for SVEA on easy tasks and all 4 CrossNorm layers for SVEA (like DrQ-v2) on medium tasks. We assess the testing performance of DrQ+CNSN, DrQ-v2+CNSN, SVEA+CNSN, and PIE-G+CNSN across the following settings: *color hard*, *video easy*, and *video hard*, where *color hard* tasks have randomly jittered color, *video easy* and *video hard* tasks replace the background with the unseen moving videos. Notably, the most challenging one is *video hard*, where the reference plane of the ground is also removed. We adopt SVEA with random overlay for all these settings and baselines, since it performs better than SVEA(conv) on *video easy* and *video hard* environments. This enables us to investigate whether our module (CrossNorm and SelfNorm) can further enhance generalization when integrated with strong data augmentation-based approaches. Additionally, for comparison purposes, we present the results of two other generalization methods in visual RL, namely RAD [26] and SODA [18]. As illustrated in Table 2, incorporating CrossNorm and SelfNorm significantly improves test performance in most of the testing environments compared to the original methods, while maintaining comparable performance in the remaining situations. In particular, when applied to DrQ and DrQ-v2, our method achieves substantial improvements in *video easy* and *video hard* environments, with average performance improvement of **155%** and **80%**, respectively. Additionally, when combined with SVEA, our method yields notable improvements across most environments. Similarly, in combination with PIE-G, our approach registers significant advancements in *video hard* scenarios. These results further substantiate the efficacy and adaptability of our proposed method.

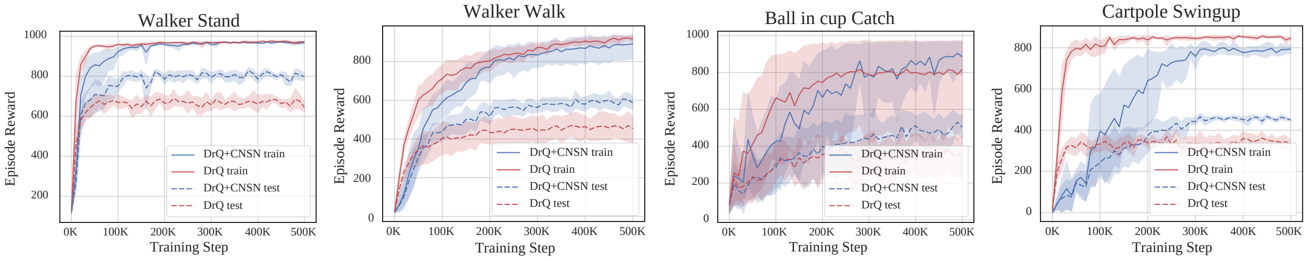

Figure 4: Training and testing performance of DrQ+CNSN against DrQ. The red line is DrQ and blue one is DrQ+CNSN. The test performance is calculated as the average across the three test settings of DMC-GB, *i.e., color hard, video easy, video hard.*

Table 3: ProcGen generalization results. All the results are averaged over 5 random seeds.

	PPO train	+CNSN train	PPO test	+CNSN test
Jumper	8.5±0.2	8.6±0.2	5.2±0.3	**6.7±0.2**
Starpilot	31.1±2.5	31.0±3.3	27.8±3.2	**28.5±3.4**
Caveflyer	6.8±0.4	7.3±0.5	5.4±0.3	**5.8±0.5**

5.2.3 Sample efficiency and generalization gap. We present the learning curves of DrQ and DrQ+CNSN on four tasks in Figure 4. One can find that the generalization gap is significantly reduced by incorporating CrossNorm and SelfNorm. Despite that adopting normalization techniques harms the sample efficiency in the training environments, such sacrifice is tolerable since the difference in the training curves on most of the tasks are marginal, while the generalization capability of the agent is largely boosted.

5.3 Generalization on ProcGen Benchmark

To further validate our method's efficacy, we experimented on the ProcGen benchmark [8]. Unlike prior experiments, ProcGen environments utilize discrete action spaces. We adopted Proximal Policy Optimization (PPO) [42] as our baseline and evaluated on three ProcGen environments: Jumper, Starpilot, and Caveflyer. We placed three CNSN modules after the convolutional layers, each activated with a probability $p = 0.5$ during training. The results, averaged across five runs, are presented in Table 3. Evidently, the integration of our CNSN technique with PPO resulted in a 13% performance enhancement on average across the three environments, without compromising the performance in training environments. These findings underscore the adaptability of our method in environments characterized by discrete action spaces.

5.4 Ablation Study

To validate the essentiality of the design choices incorporated into our method, we perform a series of ablation studies to delve deeper into the understanding of our proposed approach.

Ablation of CrossNorm and SelfNorm. Our proposed module is a combination of (cropped) CrossNorm (CN) and SelfNorm (SN). To investigate the individual contributions of CN and SN to generalization capability, we evaluate DrQ+CN and DrQ+SN on several tasks from the DMC-GB and CARLA. This analysis will help us understand the impact of each component on the overall performance of our method. The results are shown in Table 4.

Unlike computer vision datasets that originate from diverse sources, visual RL agents typically train in a single environment, leading to a relatively narrow data distribution. Therefore, it's understandable that using SelfNorm alone aids computer vision tasks but could reduce the robustness of visual RL. The μ and σ of the feature maps tend to be relatively stable, causing SelfNorm to overfit, which ultimately leads to a decrease in generalization performance.

It seems that using CrossNorm alone upon DrQ sometimes results in comparable test performance against DrQ+CNSN. However, in more complex autonomous driving scenarios, we observe that relying solely on CrossNorm does not yield performance as good as using both CrossNorm and SelfNorm. The results suggest that SelfNorm may only be effective in visual RL tasks with the existence of CrossNorm. Furthermore, the empirical results in CARLA scenarios also validate that. It is interesting to note here that it seems that for complex real-world applications, it is beneficial to combine the above two normalization techniques.

Ablation on the random cropping of CrossNorm. We also investigate how random cropping of CrossNorm (Equation 3) helps the generalization in DMC-GB tasks, as shown in Table 4. The results show that the inclusion of random cropping when calculating μ and σ in CrossNorm significantly improves generalization performance compared to cases without cropping.

Ablation on the Placement of CNSN. We further investigate the effect of CNSN placement within the architecture.While our approach places the CNSN after the convolutional layer, we have also tested its performance when placed before the convolutional layer, on two DMCGB tasks. The results are illustrated in Table 5. The results reveal that positioning the CNSN module post the convolutional layer results in superior performance. Furthermore, during the Cartpole Swingup task, introducing the CNSN module prior to the convolutional layer disrupted the training process.

Generalization Performance of Other Normalization Techniques. In addition to CrossNorm and SelfNorm, we investigate two other normalization techniques prevalent in deep learning: batch normalization (BN) and spectral normalization (SpecN). We integrate them into the image encoder of DrQ separately to assess their potential to enhance the generalization performance. BN layers are positioned after every convolution layer in the image encoder. When utilizing SpecN, we follow the conclusion from [16]

Table 4: Ablation study results. This table presents the impact of various components on the performance of our method. w/o Crop refers to DrQ+CNSN without using random cropping in CrossNorm. The results of the CARLA benchmark were obtained by training in the WetCloudySunset weather condition and testing in 6 other unseen weather conditions.

Tasks	Setting	Method				
		DrQ	+CN	+SN	+CNSN	w/o Crop
Walker Walk	*color hard*	520±91	**823±21**	188±34	815±65	634±124
	video easy	682±89	829±60	207±39	**842±58**	664±121
	video hard	104±22	**196±41**	89±24	166±28	130±35
Walker Stand	*color hard*	770±71	**951±27**	525±66	942±19	841±50
	video easy	873±83	945±33	445±113	**957±12**	857±129
	video hard	289±49	461±81	223±22	**492±62**	322±46
Cartpole Swingup	*color hard*	586±52	**695±38**	187±34	679±35	560±134
	video easy	485±105	**515±29**	135±9	498±26	410±89
	video hard	138±9	**183±4**	111±22	171±13	155±20
Ball in cup Catch	*color hard*	365±210	885±73	174±6	**894±78**	463±89
	video easy	318±157	**599±29**	161±33	584±83	391±116
	video hard	92±23	146±54	75±44	**199±138**	104±35
CARLA	unseen weather	76±74	183±91	71±70	**230±29**	185±94

Table 5: Generalization performance comparison across various normalization techniques and CNSN placements.

Tasks	Setting	Method				
		DrQ	+BN	+SpecN	+CNSN(after Conv)	+CNSN(before Conv)
Walker Walk	*color hard*	520±91	257±89	525±64	**815±65**	697±211
	video easy	682±89	479±109	739±19	**792±67**	619±239
	video hard	104±22	57±12	145±20	**166±28**	126±35
Cartpole Swingup	*color hard*	586±52	164±48	512±107	**679±35**	114±52
	video easy	485±105	182±66	375±14	**498±26**	117±30
	video hard	138±9	113±13	130±2	**171±13**	98±18

that using too many SpecN layers can decrease the capacity of networks and be detrimental to learning. Hence, we only place SpecN layers after the second, third, and fourth convolution layers of the image encoder. We train these agents on two DMC-GB tasks and evaluate their generalization performance in three settings.

As shown in Table 5, the results show that both BN and SpecN do not improve the generalization performance. Furthermore, BN leads to a significant decrease in generalization capabilities. This can be attributed to the fact that BN assumes the test data distribution is the same as the training data distribution, which can result in performance degradation when facing distribution shift. Previous literature suggests that SpecN is effective in maintaining a stable learning process for RL, particularly for very deep neural networks. Based on our results, It appears that SpecN does not significantly affect the generalization performance when faced with visual disturbances.

6 CONCLUSION

In this paper, we explore the potential benefits of normalization techniques on the generalization capabilities of visual RL and propose a novel normalization module containing CrossNorm and SelfNorm for generalizable RL. By conducting extensive experiments upon different base algorithms across diverse tasks in three generalization benchmarks, DMC-GB, CARLA autonomous driving simulator, and ProcGen, we demonstrate that our method is able to enhance generalization capability without the help of out-of-domain data and prior knowledge. These characteristics establish our approach as a self-contained method for achieving generalizable visual RL. Our method can be integrated with any visual RL algorithm, making it a valuable approach for tackling unpredictable environments.

ACKNOWLEDGMENTS

This work was supported by the National Key R&D Program of China under Grant 2022YFB2502905 and the STI 2030-Major Projects under Grant 2021ZD0201404.

REFERENCES

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450* (2016).

[2] David Bertoin, Adil Zouitine, Mehdi Zouitine, and Emmanuel Rachelson. 2022. Look where you look! Saliency-guided Q-networks for generalization in visual Reinforcement Learning. In *Neural Information Processing Systems*.

[3] Aditya Bhatt, Max Argus, Artemij Amiranashvili, and Thomas Brox. 2019. CrossNorm: Normalization for Off-Policy TD Reinforcement Learning. arXiv:1902.05605 [cs.LG]

[4] Johan Bjorck, Carla P. Gomes, and Bart Selman. 2018. Understanding Batch Normalization. In *Neural Information Processing Systems*.

[5] Nils Bjorck, Carla P Gomes, and Kilian Q Weinberger. 2021. Towards deeper deep reinforcement learning with spectral normalization. *Advances in Neural Information Processing Systems* 34 (2021), 8242–8255.

[6] Léon Bottou. 2010. Large-Scale Machine Learning with Stochastic Gradient Descent. In *International Conference on Computational Statistics*.

[7] Stephen P Boyd and Lieven Vandenberghe. 2004. *Convex optimization*. Cambridge university press.

[8] Karl Cobbe, Chris Hesse, Jacob Hilton, and John Schulman. 2020. Leveraging procedural generation to benchmark reinforcement learning. In *International conference on machine learning*. PMLR, 2048–2056.

[9] Karl Cobbe, Oleg Klimov, Chris Hesse, Taehoon Kim, and John Schulman. 2019. Quantifying generalization in reinforcement learning. In *International Conference on Machine Learning*. PMLR.

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255.

[11] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. 2017. CARLA: An Open Urban Driving Simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*. 1–16.

[12] Jiameng Fan and Wenchao Li. 2022. Dribo: Robust deep reinforcement learning via multi-view information bottleneck. In *International Conference on Machine Learning*. PMLR, 6074–6102.

[13] Linxi Fan, Guanzhi Wang, De-An Huang, Zhiding Yu, Li Fei-Fei, Yuke Zhu, and Animashree Anandkumar. 2021. SECANT: Self-Expert Cloning for Zero-Shot Generalization of Visual Policies. In *Proceedings of the 38th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, 3088–3099. https://proceedings.mlr.press/v139/fan21c.html

[14] Xinjie Fan, Qifei Wang, Junjie Ke, Feng Yang, Boqing Gong, and Mingyuan Zhou. 2021. Adversarially adaptive normalization for single domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8208–8217.

[15] Jesse Farebrother, Marlos C Machado, and Michael Bowling. 2018. Generalization and regularization in DQN. *arXiv preprint arXiv:1810.00123* (2018).

[16] Florin Gogianu, Tudor Berariu, Mihaela C Rosca, Claudia Clopath, Lucian Busoniu, and Razvan Pascanu. 2021. Spectral normalisation for deep reinforcement learning: an optimisation perspective. In *International Conference on Machine Learning*. PMLR, 3734–3744.

[17] Nicklas Hansen, Hao Su, and Xiaolong Wang. 2021. Stabilizing deep Q-learning with ConvNets and vision transformers under data augmentation. *Advances in Neural Information Processing Systems* 34 (2021).

[18] Nicklas Hansen and Xiaolong Wang. 2021. Generalization in Reinforcement Learning by Soft Data Augmentation. In *International Conference on Robotics and Automation*.

[19] Matthew Hausknecht and Nolan Wagener. 2022. Consistent Dropout for Policy Gradient Reinforcement Learning. *arXiv preprint arXiv:2202.11818* (2022).

[20] Takuya Hiraoka, Takahisa Imagawa, Taisei Hashimoto, Takashi Onishi, and Yoshimasa Tsuruoka. 2021. Dropout Q-Functions for Doubly Efficient Reinforcement Learning. *ArXiv* abs/2110.02034 (2021).

[21] Lei Huang, Jie Qin, Yi Zhou, Fan Zhu, Li Liu, and Ling Shao. 2023. Normalization techniques in training dnns: Methodology, analysis and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).

[22] Maximilian Igl, Kamil Ciosek, Yingzhen Li, Sebastian Tschiatschek, Cheng Zhang, Sam Devlin, and Katja Hofmann. 2019. Generalization in reinforcement learning with selective noise injection and information bottleneck. *Advances in neural information processing systems* 32 (2019).

[23] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*. pmlr, 448–456.

[24] Mete Kemertas and Tristan Aumentado-Armstrong. 2021. Towards robust bisimulation metric learning. *Advances in Neural Information Processing Systems* 34 (2021), 4764–4777.

[25] Robert Kirk, Amy Zhang, Edward Grefenstette, and Tim Rocktäschel. 2021. A Survey of Generalisation in Deep Reinforcement Learning. *arXiv preprint arXiv:2111.09794* (2021).

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

[27] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. 2016. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research* 17, 1 (2016), 1334–1373.

[28] Zinan Lin, Vyas Sekar, and Giulia C. Fanti. 2020. Why Spectral Normalization Stabilizes GANs: Analysis and Improvements. In *Neural Information Processing Systems*.

[29] Zhuang Liu, Xuanlin Li, Bingyi Kang, and Trevor Darrell. 2020. Regularization Matters in Policy Optimization-An Empirical Study on Continuous Control. In *International Conference on Learning Representations*.

[30] Ekdeep Singh Lubana, Robert P. Dick, and Hidenori Tanaka. 2021. Beyond BatchNorm: Towards a Unified Understanding of Normalization in Deep Learning. In *Neural Information Processing Systems*.

[31] Guozheng Ma, Zhen Wang, Zhecheng Yuan, Xueqian Wang, Bo Yuan, and Dacheng Tao. 2022. A Comprehensive Survey of Data Augmentation in Visual Reinforcement Learning. *arXiv preprint arXiv:2210.04561* (2022).

[32] K. Mehta, Anuj Mahajan, and Priyesh Kumar. 2022. Effects of Spectral Normalization in Multi-agent Reinforcement Learning. *ArXiv* abs/2212.05331 (2022).

[33] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. 2018. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957* (2018).

[34] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602* (2013).

[35] Emilio Parisotto, H. Francis Song, Jack W. Rae, Razvan Pascanu, Çaglar Gülçehre, Siddhant M. Jayakumar, Max Jaderberg, Raphael Lopez Kaufman, Aidan Clark, Seb Noury, Matthew M. Botvinick, Nicolas Manfred Otto Heess, and Raia Hadsell. 2019. Stabilizing Transformers for Reinforcement Learning. In *International Conference on Machine Learning*.

[36] Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. 2018. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 3803–3810.

[37] Lerrel Pinto, Marcin Andrychowicz, Peter Welinder, Wojciech Zaremba, and Pieter Abbeel. 2017. Asymmetric Actor Critic for Image-Based Robot Learning. arXiv:1710.06542 [cs.RO]

[38] Roberta Raileanu, Maxwell Goldstein, Denis Yarats, Ilya Kostrikov, and Rob Fergus. 2021. Automatic data augmentation for generalization in reinforcement learning. *Advances in Neural Information Processing Systems* 34 (2021), 5402–5415.

[39] Tim Salimans and Diederik P. Kingma. 2016. Weight Normalization: A Simple Reparameterization to Accelerate Training of Deep Neural Networks. In *Neural Information Processing Systems*.

[40] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. 2018. How Does Batch Normalization Help Optimization?. In *Neural Information Processing Systems*.

[41] Amartya Sanyal, Philip H. S. Torr, and Puneet Kumar Dokania. 2019. Stable Rank Normalization for Improved Generalization in Neural Networks and GANs. *ArXiv* abs/1906.04659 (2019).

[42] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[43] Seonguk Seo, Yumin Suh, Dongwan Kim, Jongwoo Han, and Bohyung Han. 2019. Learning to Optimize Domain Specific Normalization for Domain Generalization. In *European Conference on Computer Vision*.

[44] Jiacheng Sun, Xiangyong Cao, Hanwen Liang, Weiran Huang, Zewei Chen, and Zhenguo Li. 2020. New interpretations of normalization methods in deep learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[45] Zhiqiang Tang, Yunhe Gao, Yi Zhu, Zhi Zhang, Mu Li, and Dimitris Metaxas. 2021. CrossNorm and SelfNorm for generalization under distribution shifts. In *ICCV 2021*. https://www.amazon.science/publications/crossnorm-and-selfnorm-for-generalization-under-distribution-shifts

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

[47] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. 2017. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 23–30.

[48] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. 2017. Instance Normalization: The Missing Ingredient for Fast Stylization. arXiv:1607.08022 [cs.CV]

[49] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* 575, 7782 (2019), 350–354.

[50] Kaixin Wang, Bingyi Kang, Jie Shao, and Jiashi Feng. 2020. Improving generalization in reinforcement learning with mixture regularization. *Advances in Neural Information Processing Systems* 33 (2020), 7968–7978.

[51] Zhiguo Wang, Patrick Ng, Xiaofei Ma, Ramesh Nallapati, and Bing Xiang. 2019. Multi-passage BERT: A Globally Normalized BERT Model for Open-domain Question Answering. In *Conference on Empirical Methods in Natural Language Processing*.

[52] Yuxin Wu and Kaiming He. 2018. Group Normalization. *International Journal of Computer Vision* 128 (2018), 742–755.

[53] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu. 2020. On Layer Normalization in the Transformer Architecture. In *International Conference on Machine Learning*.

[54] Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. 2021. Mastering Visual Continuous Control: Improved Data-Augmented Reinforcement Learning. *arXiv preprint arXiv:2107.09645* (2021).

[55] Denis Yarats, Ilya Kostrikov, and Rob Fergus. 2021. Image Augmentation Is All You Need: Regularizing Deep Reinforcement Learning from Pixels. In *International Conference on Learning Representations*. https://openreview.net/forum?id=GY6-6sTvGaf

[56] Zhecheng Yuan, Guozheng Ma, Yao Mu, Bo Xia, Bo Yuan, Xueqian Wang, Ping Luo, and Huazhe Xu. 2022. Don't Touch What Matters: Task-Aware Lipschitz Data Augmentationfor Visual Reinforcement Learning. *arXiv preprint arXiv:2202.09982* (2022).

[57] Zhecheng Yuan, Zhengrong Xue, Bo Yuan, Xueqian Wang, Yi Wu, Yang Gao, and Huazhe Xu. 2022. Pre-Trained Image Encoder for Generalizable Visual Reinforcement Learning. In *Neural Information Processing Systems*.

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

[59] Biao Zhang and Rico Sennrich. 2019. Root Mean Square Layer Normalization. *ArXiv* abs/1910.07467 (2019).

[60] Chiyuan Zhang, Oriol Vinyals, Remi Munos, and Samy Bengio. 2018. A study on overfitting in deep reinforcement learning. *arXiv preprint arXiv:1804.06893* (2018).

[61] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. 2017. Places: A 10 million Image Database for Scene Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2017).

[62] Martin A. Zinkevich, Markus Weimer, Alex Smola, and Lihong Li. 2010. Parallelized Stochastic Gradient Descent. In *Neural Information Processing Systems*.