Don't Touch What Matters: Task-Aware Lipschitz Data Augmentation for Visual Reinforcement Learning

Zhecheng Yuan[1], **Guozheng Ma**[1], **Yao Mu**[2], **Bo Xia**[1], **Bo Yuan**[1], **Xueqian Wang**[1], **Ping Luo**[2], **Huazhe Xu**[3*]
[1]Tsinghua University
[2] Hongkong University
[3]Stanford University
yuanzc20@mails.tsinghua.edu.cn, huazhexu@stanford.edu

Abstract

One of the key challenges in visual Reinforcement Learning (RL) is to learn policies that can generalize to unseen environments. Recently, data augmentation techniques aiming at enhancing data diversity have demonstrated proven performance in improving the generalization ability of learned policies. However, due to the sensitivity of RL training, naively applying data augmentation, which transforms each pixel in a task-agnostic manner, may suffer from instability and damage the sample efficiency, thus further exacerbating the generalization performance. At the heart of this phenomenon is the diverged action distribution and high-variance value estimation in the face of augmented images. To alleviate this issue, we propose **T**ask-aware **L**ipschitz **D**ata **A**ugmentation (TLDA) for visual RL, which explicitly identifies the task-correlated pixels with large Lipschitz constants, and only augments the task-irrelevant pixels for stability. We verify the effectiveness of our approach on DeepMind Control suite, CARLA and DeepMind Manipulation tasks. The extensive empirical results show that TLDA improves both sample efficiency and generalization; it outperforms previous state-of-the-art methods across 3 different visual control benchmarks.

1 Introduction

Deep Reinforcement Learning (DRL) from visual observations has carved out brilliant paths in many domains such as video games [Mnih *et al.*, 2015], robotics manipulation [Kalashnikov *et al.*, 2018], and visual navigation [Zhu *et al.*, 2017]. However, it remains challenging to obtain generalizable policies across different environments with visual variations due to overfitting [Zhang *et al.*, 2018].

Data Augmentation [Shorten and Khoshgoftaar, 2019] and Domain Randomization [Tobin *et al.*, 2017] based approaches are widely used to learn generalizable visual representations. However, recent work [Hansen *et al.*, 2021] find

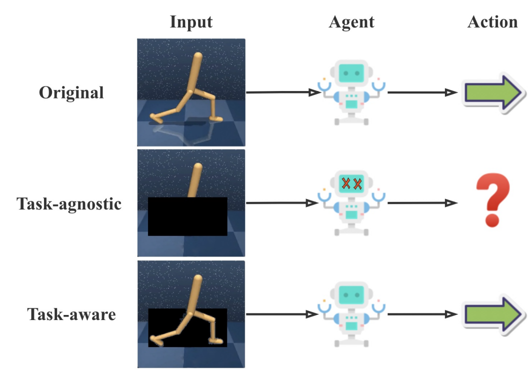


Figure 1: Augmenting observation in a task-agnostic manner (in the *middle*) distracts the agent's decision, hence it will damage agent's asymptotic performance. This problem can be alleviated by task-aware data augmentation (in the *bottom*).

that in visual RL, there is a dilemma: heavy data augmentations are vital for better generalization, but it will cause a significant decrease in both the sample efficiency and the training stability. One of the main reasons is that data augmentation conventionally perform pixel-level transformation, where each pixel is transformed in a task-agnostic manner. However, each pixel in the observation has different relevance to the task and the reward function. Hence, it is worth rethinking data augmentation in the new context of visual RL.

To better understand the effect of data augmentation in visual RL, we visualize the action distribution output from policies trained with various data augmentation choices in Figure 2. We find that the agent's actions vary dramatically when faced with different augmentation methods. Specifically, when weak augmentation such as *shifting* is applied, the action distribution remains closer to the original distribution that has no augmentation (Figure 2(c)); however, when strong augmentation e.g., *random convolution* is applied, the action distribution drastically changes (Figure 2(a)) and the Q-estimation yields the discrepancy with the un-augmented data, as shown in Figure 3. This comparison reveals the severe problem that causes instability when data augmentation is applied blindly without knowing the task information.

In this work, we propose a task-aware data augmentation method in visual RL that learns to augment the pixels less

*Corresponding Author

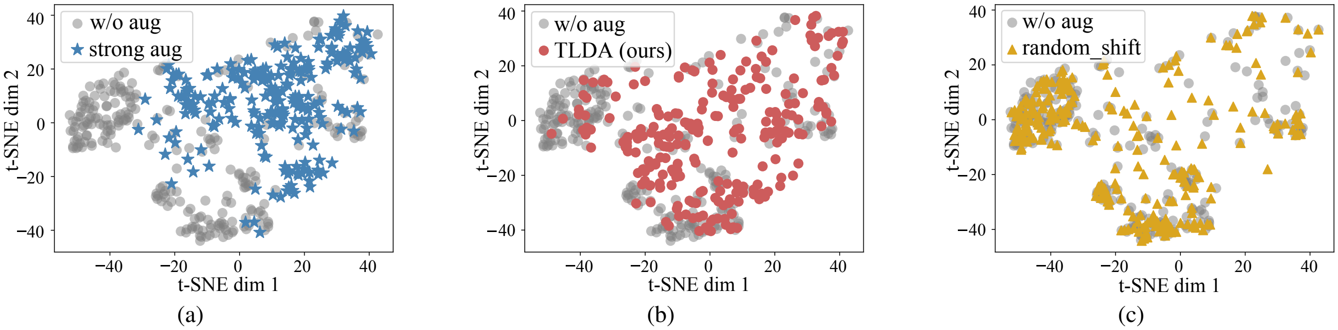


Figure 2: **Action Distribution.** We use t-SNE to show high-dimensional actions employed by the same agent. The **grey dots** are the actions given the observations without augmentation (*w/o aug*); **the blue dots** (a) and **orange dots** (c) are the actions given the same observations under strong (*random_conv*) and weak augmentation (*random_shift*), respectively. The visualized results demonstrate that there is a significant action distribution shift under strong augmentation, while under weak augmentation, the policy is closer to the original distribution. The **red dots** (b) are TLDA under strong augmentation, which comes up with an action distribution that remains similar to **the grey dots**.

correlated to the task, namely **T**ask-aware **L**ipschitz **D**ata **A**ugmentation (TLDA), as shown in Figure 1. A desirable quality for such a method is that it maintains a stable policy output even on augmented observations. Following this insight, we introduce the *Lipschitz constant* that measures the relevance between the pixel and the task, then guides the augmentation strategy. Specifically, we first impose a perturbation on a certain pixel, and calculate the corresponding Lipschitz constant for the pixel via the policy change before and after the perturbation. Then, to avoid the occurrence of drastic policy changes, we treat the pixels with larger Lipschitz constant as the task-relevant ones and avoid augmenting them. Therefore, the output could be more stable while keeping the diversity of augmented data.

We conduct experiments on 3 benchmarks: DMControl Generalization Benchmark (DMC-GB) [Hansen and Wang, 2021], CARLA [Dosovitskiy *et al.*, 2017], and DMControl manipulation tasks [Tunyasuvunakool *et al.*, 2020]. We train agents in a fixed environment and evaluate on the unseen environments. Extensive experiments show that TLDA outperforms the prior state-of-the-art methods due to more stable and efficient training and robust generalization performance.

Our main contributions are summarized as follows: (i) We propose **T**ask-aware **L**ipschitz **D**ata **A**ugmentation (TLDA), which can be implemented on downstream visual RL algorithm easily without adding auxiliary objectives or additional learnable parameters; (ii) We provide theoretical understanding and experiments to show TLDA can alleviate the action distribution shift and high variance Q-estimation problems effectively; (iii) TLDA achieves competitive or better sample efficiency and generalization ability than previous state-of-the-art methods in 3 different kinds of benchmarks.

2 Related Work

Generalization in RL. Researchers have been investigated RL generalization from various perspectives, such as visual appearances [Cobbe *et al.*, 2019], dynamics [Packer *et al.*, 2018] and environment structures [Cobbe *et al.*, 2020]. In this paper, we focus on generalization over different visual appearances. Two popular paradigms are proposed to address the overfitting issue in current visual RL research. The first is to regard generalization as a representation learning problem. Bi-simulation metric [Ferns *et al.*, 2011] is implemented to learn robust representation features [Zhang *et al.*, 2020]. The other paradigm is to design auxiliary tasks. SODA [Hansen and Wang, 2021] adds a BYOL-like [Grill *et al.*, 2020] architecture and introduces an auxiliary loss which encourages the representation to be invariant to task-irrelevant properties of the environment. In contrast to the previous efforts, our method does not require to employ a specific metric to learn representation, nor to introduce additional modules.

Data Augmentation for RL. Data Augmentation is an efficient method to improve the generalization of visual RL. RAD [Laskin *et al.*, 2020] compares different data augmentation methods and reveals that the benefits of different augmentation methods to RL tasks are not the same. SECANT [Fan *et al.*, 2021] mentions that weak augmentation can improve sample efficiency but not generalization ability. It also shows that the simple use of strong augmentation is likely to cause training divergence, though generalization ability is improved. Automatic data augmentation is proposed in [Raileanu *et al.*, 2021] to make better use of data augmentation. We advocate this paradigm and believe that one crucial factor for improving sample efficiency and generalization lies in the design of data augmentation, namely, how we can diversify the input as much as possible while maintaining the invariance of output. We show that how strong augmentation affects action distribution shifts and causes high variance of Q estimation, and illustrate that our approach is effective in alleviating these two problems.

3 Preliminaries

We consider learning in a Markov Decision Process (MDP) formulated by the tuple $\langle \mathcal{S}, \mathcal{A}, r, \mathcal{P}, \gamma \rangle$ where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $r : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ is a reward function, $\mathcal{P}(s_{t+1} \mid s_t, a_t)$ is the state transition function, $\gamma \in [0, 1)$ is the discount factor. The goal is to learn a policy π^* to maximize the expected cumulative return $\pi^* = \operatorname{argmax}_\pi \mathbb{E}_{a_t \sim \pi(\cdot|s_t), s_t \sim \mathcal{P}} \left[\sum_{t=1}^{T} \gamma^t r(s_t, a_t) \right]$, starting from

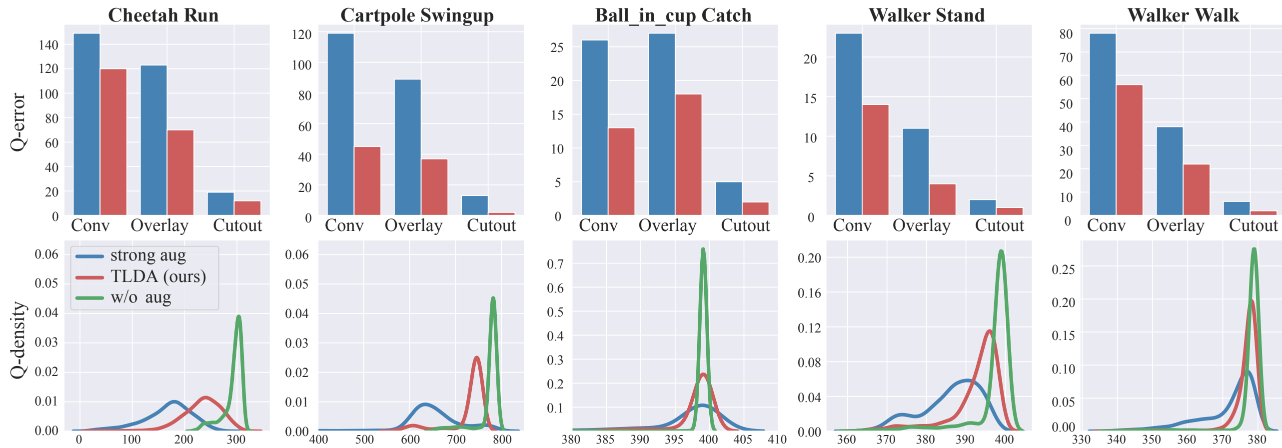


Figure 3: **Q-estimation error**. *Top:* We measure the Q-estimation mean square error of the different augmented observation vs.the non-augmented observation. **The blue bar** and **the red bar** are the error between strong augmented data and TLDA-augmented data vs non-augmented data, separately. It shows that TLDA can significantly reduce the Q-estimation error to alleviate the high-variance estimation problems. *Bottom:* The distribution of Q-estimation. TLDA comes up with a closer Q-estimation distribution with the original one.

an initial state $s_0 \in \mathcal{S}$ and following the policy $\pi_\theta(\cdot \mid s_t)$ which is parameterized by a set of learnable parameters θ. Meanwhile, we expect the learned policy π_θ^* can be well generalized to new environments, which have the same structure and definition of the original MDP, but with different observation space $\mathcal{O}$ constructed from the same state space $\mathcal{S}$.

3.1 Data Augmentation

Definition 1 (Optimality-Invariant State Transformation) *Given an MDP $\mathcal{M}$, we define an augmentation method $\phi : \mathcal{O} \to \mathcal{O}'$ as an optimality-invariant transformation if $\forall o \in \mathcal{O}, a \in \mathcal{A}, \phi(o) \in \mathcal{O}'$, where $\mathcal{O}'$ is a new set of observation satisfies:*

$$Q(o,a) = Q(\phi(o),a) \quad \pi(\cdot \mid o) = \pi(\cdot \mid \phi(o)) \tag{1}$$

A desirable quality for data augmentation is to satisfy the form of Optimality-Invariant State Transformation while distortion or distracting noise is added to the observation.

3.2 Lipschitz Constant

The Lipschitz constant is frequently utilized to measure the robustness of a model, we introduce Lipschitz continuity of the policy here. A function f: $\mathbb{R}^n \to \mathbb{R}^m$ is Lipschitz continuous on $\mathcal{X} \subseteq \mathbb{R}^n$ if there exists a non-negative constant $K \geq 0$ such that

$$\|f(x) - f(y)\| \leq K\|x - y\| \text{ for all } x, y \in \mathcal{X} \tag{2}$$

The smallest such K is called the Lipschitz constant of f [Pauli *et al.*, 2021].

Definition 2 (Lipschitz constant of the policy) *Assume the state space is equipped with a distance metric $d(\cdot,\cdot)$. Under a certain augmentation method ϕ, the Lipschitz constant of a policy π is defined as follows:*

$$K_\pi = \sup_{s \in \mathcal{S}} \frac{D_{TV}\left(\pi\left(\cdot \mid \phi(s)\right) \| \pi\left(\cdot \mid s\right)\right)}{d(\phi(s), s)} \tag{3}$$

where $D_{TV}(P\|Q) = \frac{1}{2}\sum_{a \in \mathcal{A}} |P(a) - Q(a)|$ is the total variation distance between distributions. If K_π is finite, the policy π is Lipschitz continuous.

For a certain model, a smaller Lipschitz constant represents higher stability against the variance of input [Finlay *et al.*, 2018]. The following proposition illustrates that the estimation error of Q-value can be bounded by Lipschitz constant:

Proposition 1 *We consider an MDP $\mathcal{M}$, a policy π and an augmentation method ϕ. Suppose the rewards are bounded by r_{max} and state space is equipped with a distance metric $d(\cdot,\cdot)$, such that $\forall a \in \mathcal{A}, \forall s \in \mathcal{S}, |r(s,a)| \leq r_{max}$, the following inequality holds, where $\|d(\phi)\|_\infty = \sup_{s \in \mathcal{S}} d(\phi(s), s)$:*

$$|Q^\pi(s,a) - Q^\pi(\phi(s),a)| \leq 2r_{max}\frac{(K_\pi \|d(\phi)\|_\infty + 1)}{1-\gamma} \tag{4}$$

This proposition indicates that if a smaller Lipschitz constant under one specific augmentation is acquired, we will have a tighter bound of the Q-value estimation with a lower variance while implementing data augmentation.

4 Method

To maintain the training stability and improve the generalization ability, we propose: **T**ask-aware **L**ipschitz **D**ata **A**ugmentation (TLDA), an efficient and general task-aware data augmentation method for visual RL.

4.1 Construct the *K-matrix*

We first calculate the Lipschitz constant from perturbed input images. By using a kernel to perturb the original image $o \in \mathbb{R}^{H \times W}$, we obtain the perturbed image denoted as $A(o)$. Next, we choose the pixels centered with the location (i,j) of $A(o)$ as in the Eq (5), denoted as $\Phi(o,i,j)$. Specifically, we use the Hadamard product $\odot$ to choose the perturbed pixels around location(i,j) by an image mask $M(i,j) \in (0,1)^{H \times W}$:

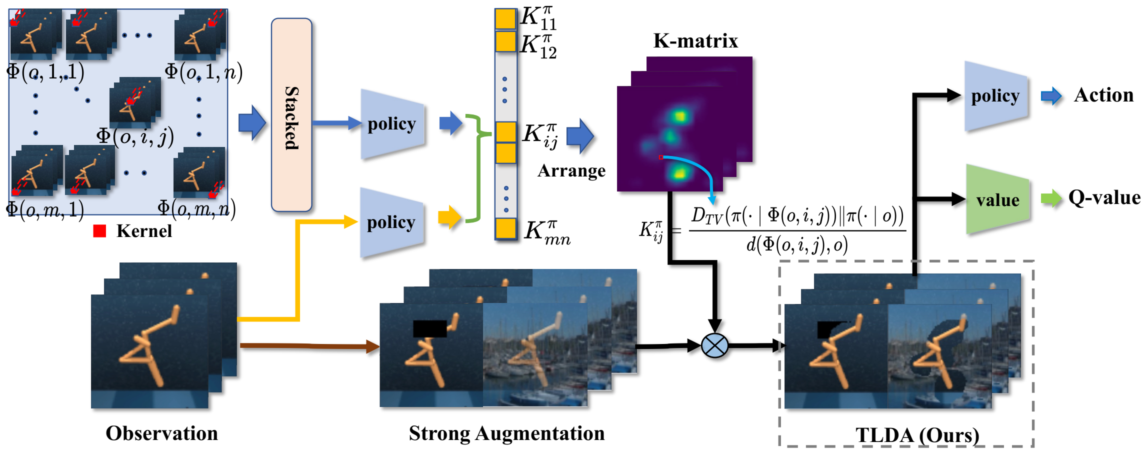


Figure 4: **Overview of TLDA**. This figure shows two examples and the pipeline of TLDA. The agent generates the *K-matrix* in a frame, and then preserves the larger Lipschitz constant areas under strong augmentation. The preserved areas are highlighted in the *K-matrix*.

$$\Phi(o,i,j) = o \odot (1 - M(i,j)) + A(o) \odot M(i,j) \tag{5}$$

To derive the Lipschitz constant, we use the notation $d(\Phi(o,i,j),o)$ to represent the distance between input o and $\Phi(o,i,j)$ under metric $d(\cdot,\cdot)$. As in Definition 2, for a given observation o, the Lipschitz constant of the pixel (i,j) can be computed as follows:

$$K_{ij}^{\pi} = \frac{D_{TV}\left(\pi\left(\cdot \mid \Phi(o,i,j)\right) \| \pi(\cdot \mid o)\right)}{d(\Phi(o,i,j),o)} \tag{6}$$

where the numerator can be interpreted as distance between two action distributions: $\pi(\cdot \mid \Phi(o,i,j))$, $\pi(\cdot \mid o)$, and the denominator is the distance between the original observation and the perturbed one. With the per-pixel Lipschitz constant in hand, we then construct the matrix that can reflect the task-relevance information and be applied on the whole observation. By arranging K_{ij}^{π} into a matrix which have the same size as o following Eq (7), we denote this matrix as the *K-matrix*:

$$\textit{K-matrix} \triangleq \begin{bmatrix} K_{11}^{\pi} & K_{12}^{\pi} & \cdots & K_{1n}^{\pi} \\ K_{21}^{\pi} & K_{22}^{\pi} & \cdots & K_{2n}^{\pi} \\ \vdots & \vdots & \ddots & \vdots \\ K_{m1}^{\pi} & K_{m2}^{\pi} & \cdots & K_{mn}^{\pi} \end{bmatrix} \tag{7}$$

We aim to capture the task-related locations with large Lipschitz constants which tend to cause high variance in the policy/value output during the same level of perturbation.

4.2 Task-Aware Lipschitz Augmentation (TLDA) with the *K-matrix*

Intuitively, data augmentation operations should not modify the task-related pixels indicated by large Lipschitz constants. We follow this intuition and propose a simple yet effective way to decide which areas can be modified. We use the mean value of the *K-matrix* as a threshold, and binarize the *K-matrix* by the following way, where N is the number of pixels $(H \times W)$, $K^{mean} = \frac{1}{N} \times \sum_{ij} K_{ij}^{\pi}$:

$$M_{ij}^{K} = \begin{cases} 1, & if \quad K_{ij}^{\pi} \geq K^{mean} \\ 0, & else \end{cases} \tag{8}$$

The obtained mask M^K is used to decide which pixels can be augmented. For any data augmentation method $o' = \mathrm{Aug}(o)$, we apply the following operation:

$$\tilde{o} = M^K \odot o + (1 - M^K) \odot o' \tag{9}$$

We note that the output $\tilde{o}$ is only modified in the areas that have low relevance to the task.

As mentioned above, TLDA tends to preserve the pixels with large K_{ij}^{π} and augment only the pixels associated with the small ones, which adds an implicit constraint to maintain the stable output of the policy and value network. Hence, it echoes with the Optimality-Invariant State Transformation as in Definition 1. Figure 4 demonstrates the overall framework of TLDA. During the training process, the *K-matrix* is calculated on the fly against every training step on augmented observations. Take *cutout* (adding a black patch to the image) in Figure 4 as an example, since the corresponding *K-matrix* shows that the upper part of the robot's body features large Lipschitz constants, therefore, blindly augmenting the image might touch the pixels in this area and cause catastrophic action/value changes. In contrast, TLDA preserves the critical parts of the original observations indicated by *K-matrix*.

4.3 Reinforcement Learning with TLDA

We use soft-actor-critic (SAC) as the base reinforcement learning algorithm for TLDA. Similar to previous work, we also include a regularization term $\mathcal{R}_Q(\theta)$ to the SAC critic loss $\mathcal{J}_Q(\theta)$ to handle augmented data. Our critic loss $\mathcal{L}_Q(\theta)$ is as follows, where s_t^{aug} is calculated by Eq (9), and $\hat{Q}(s_t,a_t) = r(s_t,a_t) + \gamma \mathbb{E}_{s_{t+1}\sim\mathcal{P}}[V(s_{t+1})]$:

$$\mathcal{L}_Q(\theta) = \mathcal{J}_Q(\theta) + \lambda \mathcal{R}_Q(\theta) \tag{10}$$

with

$$\mathcal{J}_Q(\theta) = \mathbb{E}_{(s_t,a_t)\sim\mathcal{D}}\left[\frac{1}{2}\left(Q_\theta(s_t,a_t) - \hat{Q}(s_t,a_t)\right)^2\right]$$

$$\mathcal{R}_Q(\theta) = \mathbb{E}_{(s_t,a_t)\sim\mathcal{D}}\left[\frac{1}{2}\left(Q_\theta(s_t^{\text{aug}},a_t) - \hat{Q}(s_t,a_t)\right)^2\right]$$

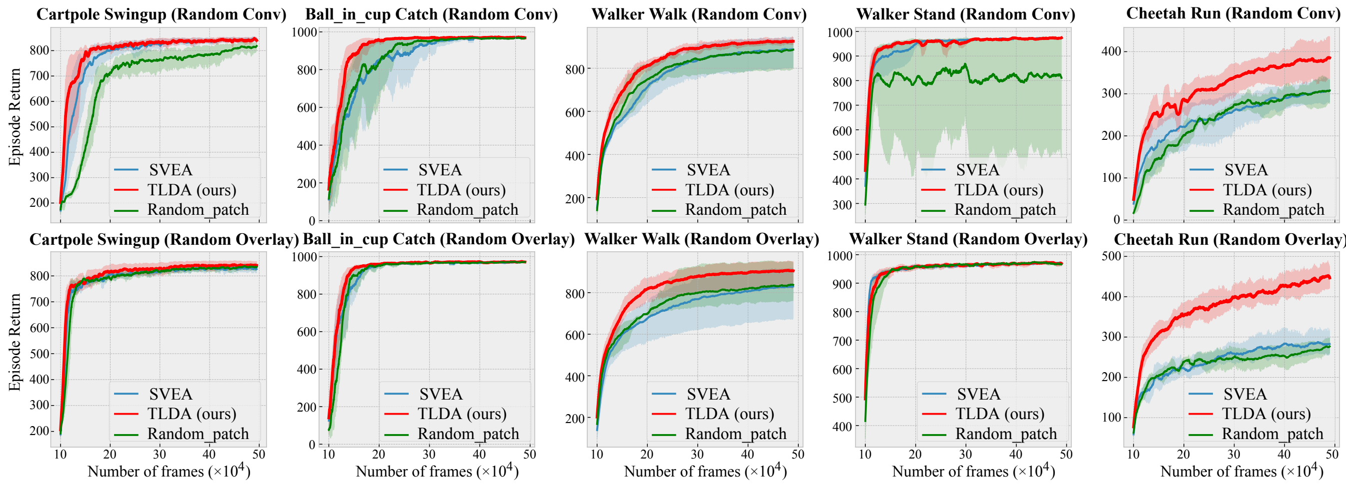


Figure 5: **Sample efficiency in training environment**. We compare TLDA, SVEA, and *random patch* under two kinds of augmentations. *Top row* and *Bottom Row* are corresponding to *Random Conv* and *Random Overlay* training curves of the episode return respectively. TLDA (*red line*) shows better sample efficiency on the training period. Mean and standard deviation of 5 runs.

5 Experiment

In this section, we explore how TLDA can affect the agent's sample efficiency and generalization performance. We compare our method with other baselines on a wide spectrum of tasks including DeepMind control suite, CARLA simulator, as well as DeepMind Manipulation tasks.

5.1 Evaluation on DeepMind Control Suite

Setup. For comparison, we mainly consider two augmentation ways applied in the prior state-of-the-art methods: *random convolution* (passing input through a random convolutional layer) and *random overlay* (linearly combining the observation o with the extra image $\mathcal{I}$, $\phi(o) = \alpha o + (1-\alpha)\mathcal{I}$).

Baselines. We benchmark TLDA against the following state-of-the-art methods: (1) **DrQ** [Kostrikov *et al.*, 2020]: SAC with weak augmentation (*random shift*); (2) **PAD** [Hansen *et al.*, 2020]: adding an auxiliary task for adapting to the unseen environment; (3) **SODA** [Hansen and Wang, 2021]: maximizing the mutual information between latent representation by employing a BYOL-like [Grill *et al.*, 2020] architecture; (4) **SVEA** [Hansen *et al.*, 2021]: modifying the form of Q-target.

Sample efficiency under strong augmentations. We compare the sample efficiency with SVEA to exhibit the effectiveness of TLDA. We also include another baseline that preserves random patches from the un-augmented observation as opposed to TLDA that preserves task-related parts. We call this baseline *random patch*. By contrast, SVEA only uses the strong augmentation method but retains no raw pixel. Figure 5 demonstrates that TLDA achieves better or comparable asymptotic performance in the training environment than baselines on DM-control suite while having better sample efficiency. The results also indicate that *random patch* will hinder the performance in some tasks. We reckon that since *random patch* does not have any pixel-to-task relevance knowledge, it inevitably destroys the image's integrity and even leads to further distortion to the observations after data augmentation. Therefore, blindly keeping the original observation's information cannot improve the agent's training performance. It is the retention of areas with larger Lipschitz constants, instead of random original areas, that boosts the sample efficiency of training agents.

Generalization Performance. We evaluate the agent's generalization ability on two settings from DMControl-GB [Hansen and Wang, 2021]. Results are shown in Table 1. TLDA outperforms prior state-of-the-art methods in **7** out of **10** instances. The agent trained with TLDA is able to acquire a good robust policy in different unseen environments. Meanwhile, we notice that prior methods are sensitive to augmentation methods, which makes their testing performance varies dramatically. On the contrary, our method with task-aware observations is more stable and not susceptible to this issue.

Effect on Action Distribution and Q-estimation. In this section, we analyze how TLDA influences the output of the policy and value networks. Given a DrQ agent trained in the original environment, we assess the Q-value estimation and the action distribution under different augmentation. To get a better understanding of this issue, we visualize the action distribution of the agent under different augmentation methods, as shown in Figure 2. For weak augmentation, although its action distribution is closest to the un-augmented one (Figure 2(c)), it cannot improve generalization, as shown in Table 1(DrQ). Strong augmentation, on the other hand, will cause an obvious distribution shift (Figure 2(a)), thus significantly hindering the training process. TLDA has a closer action distribution than simply applying strong augmentation (Figure 2 (b)) by using the Lipschitz constant to identify and preserve the task-aware areas. Furthermore, as shown in Figure 3, we find that the Q-estimation of TLDA has a lower variance than that of naively applying strong augmentation. These two results illustrate that TLDA has the potential to

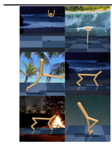
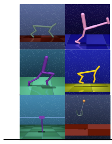
Setting	DMControl	DrQ	PAD	SVEA (conv)	SVEA (overlay)	SODA (conv)	SODA (overlay)	TLDA (conv)	TLDA (overlay)
	Cartpole-Swingup	485±105	521±76	606±85	**782±27**	474±143	758±62	607±74	671±57
	Walker-Stand	873±83	935±20	795±70	961±8	903±56	955±13	962±15	**973±6**
	Walker-Walk	682±89	717±79	612±144	819±71	635±48	768±38	**873±34**	868±63
	Ball_in_cup-Catch	318±157	436±55	659±110	871±106	539±111	875±56	**892±68**	863±56
	Cheetah-Run	102±30	206±34	292±32	249±20	229±29	223±32	356±52	**366±57**
	Cartpole-Swingup	586±52	630±63	**837±23**	832±23	831±21	805±28	748±40	760±60
	Walker-Stand	770±71	797±46	**942±26**	933±24	930±12	893±12	919±24	**947±26**
	Walker-Walk	520±91	468±47	760±145	749±61	697±66	692±68	743±83	**823±58**
	Ball_in_cup-Catch	365±210	563±50	**961±7**	959±5	892±37	949±19	932±32	930±40
	Cheetah-Run	100±27	159±28	264±51	273±23	294±34	238±28	**371±51**	358±25

Table 1: **DMC-GB Generalization Performance.** Episode return in test environments. The agents are trained on a fixed environment and evaluated on *random colors* (*Bottom*) and *video backgrounds* (*Top*) these two unseen test environments. Mean and std.deviation of 5 runs.

achieve higher sample efficiency in training and learn a more robust policy to perform well in unseen environments.

5.2 Evaluation on Autonomous Driving in CARLA

To further evaluate the TLDA's performance, we apply this method in the tasks with more realistic observations: autonomous driving in the CARLA simulator. In our experiment, we use one camera as our input observation for driving tasks, where the goal of the agent is to drive along a curvy road as far as possible in 1000 time-steps without colliding with the moving vehicles, pedestrians and barriers. We adapt the reward function and train an agent under the weather with the same setting from previous work [Zhang *et al.*, 2020]. We find that our method achieves the best training sample efficiency. For generalization, CARLA provides different weather conditions with built-in parameters. We evaluate our method in 4 kinds of weather with different lighting conditions, realistic raining and slipperiness. Results are in Table 2, where we choose the success rate for reach 100m distance as the driving evaluation metric. TLDA outperforms all base algorithms in both sample efficiency and generalization ability with a more stable driving policy.

Setting	DrQ	SVEA	Ours
Training	24%	49%	**52%**
Wet Noon	0.8%	8.8%	**18%**
SoftRain noon	0.4%	1.2%	**7.6%**
Wet Sunset	0.8%	1.6%	**9.2%**
MidRain Sunset	0.0%	5.2%	**12%**

Table 2: **CARLA Driving**. We report the success rate for reaching 100m distance under the unseen weather during 250 episodes across 5 seeds for each weather. (50 episodes for each seed)

5.3 Evaluation on DMC Manipulation Tasks

Robot manipulation is another set of challenging and meaningful tasks for visual RL. DM control [Tunyasuvunakool *et al.*, 2020] provides a set of configurable manipulation tasks with a robotic Jaco arm and snap-together bricks. We consider two tasks for experiments: *reach* and *push*. All the agents are trained on the default background and evaluated on different colors of arms and platforms. The generalization performance are shown in Table 3. The results show that our method can be adapted to the unseen environments more appropriately. The *Modified Platform* and *Modified Both* setting are challenging for agents to discern the target objects from the noisy backgrounds. SVEA under strong data augmentation suffers from instability and divergence for training, while TLDA can augment the pixel in a task-aware manner, thus further maintaining the training stability. Despite that DrQ shows better training performance, it barely generalizes to the environments with different visual layouts. In summary, sample efficiency and generalization performance contribute to exhibiting the superiority of the proposed algorithm.

Task	Setting	DrQ	SVEA	Ours
Reach	Training	**136** ±20	49 ±48	124 ±32
	M Arm	**68** ±20	21 ±25	55 ±21
	M Platform	0.8 ±1.3	24 ±25	**89** ±40
	M Both	1 ±2	13 ±14	**36** ±25
Push	Training	**141** ±47	42 ±40	109 ±27
	M Arm	**88** ±52	21 ±16	60 ±43
	M Platform	4 ±1	34 ±28	**95** ±33
	M Both	5 ±1	32 ±20	**56** ±42

Table 3: **DMC Manipulation Tasks**. M in the *Setting* column means: ***Modified***. TLDA can better focus on the aim objects in the noisy and colorful visual backgrounds.

6 Conclusion

In this paper, we propose **T**ask-aware **L**ipschitz **D**ata **A**ugmentation (TLDA) for visual RL, which can reliably identify and augment pixels that are not strongly correlated with the learning task while keeping task-related pixels untouched. This technique aims to provide a principled mechanism for boosting the generalization ability of RL agents and can be seamlessly incorporated into various existing visual RL frameworks. Experimental results on 3 challenging benchmarks confirm that, compared with the baselines, TLDA not only features higher sample efficiency but also helps the agents generalize well to the unseen environments.

References

[Cobbe *et al.*, 2019] Karl Cobbe, Oleg Klimov, Chris Hesse, Taehoon Kim, and John Schulman. Quantifying generalization in reinforcement learning. In *International Conference on Machine Learning*, pages 1282–1289. PMLR, 2019.

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

[Dosovitskiy *et al.*, 2017] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017.

[Fan *et al.*, 2021] Linxi Fan, Guanzhi Wang, De-An Huang, Zhiding Yu, Li Fei-Fei, Yuke Zhu, and Anima Anandkumar. Secant: Self-expert cloning for zero-shot generalization of visual policies. *arXiv preprint arXiv:2106.09678*, 2021.

[Ferns *et al.*, 2011] Norm Ferns, Prakash Panangaden, and Doina Precup. Bisimulation metrics for continuous markov decision processes. *SIAM Journal on Computing*, 40(6):1662–1714, 2011.

[Finlay *et al.*, 2018] Chris Finlay, Jeff Calder, Bilal Abbasi, and Adam Oberman. Lipschitz regularized deep neural networks generalize and are adversarially robust. *arXiv preprint arXiv:1808.09540*, 2018.

[Grill *et al.*, 2020] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

[Hansen and Wang, 2021] Nicklas Hansen and Xiaolong Wang. Generalization in reinforcement learning by soft data augmentation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13611–13617. IEEE, 2021.

[Hansen *et al.*, 2020] Nicklas Hansen, Rishabh Jangir, Yu Sun, Guillem Alenyà, Pieter Abbeel, Alexei A Efros, Lerrel Pinto, and Xiaolong Wang. Self-supervised policy adaptation during deployment. *arXiv preprint arXiv:2007.04309*, 2020.

[Hansen *et al.*, 2021] Nicklas Hansen, Hao Su, and Xiaolong Wang. Stabilizing deep q-learning with convnets and vision transformers under data augmentation. *arXiv e-prints*, pages arXiv–2107, 2021.

[Kalashnikov *et al.*, 2018] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Scalable deep reinforcement learning for vision-based robotic manipulation. In *Conference on Robot Learning*, pages 651–673. PMLR, 2018.

[Kostrikov *et al.*, 2020] Ilya Kostrikov, Denis Yarats, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *arXiv preprint arXiv:2004.13649*, 2020.

[Laskin *et al.*, 2020] Misha Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. Reinforcement learning with augmented data. *Advances in Neural Information Processing Systems*, 33, 2020.

[Mnih *et al.*, 2015] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[Packer *et al.*, 2018] Charles Packer, Katelyn Gao, Jernej Kos, Philipp Krähenbühl, Vladlen Koltun, and Dawn Song. Assessing generalization in deep reinforcement learning. *arXiv preprint arXiv:1810.12282*, 2018.

[Pauli *et al.*, 2021] Patricia Pauli, Anne Koch, Julian Berberich, Paul Kohler, and Frank Allgower. Training robust neural networks using lipschitz bounds. *IEEE Control Systems Letters*, 2021.

[Raileanu *et al.*, 2021] Roberta Raileanu, Maxwell Goldstein, Denis Yarats, Ilya Kostrikov, and Rob Fergus. Automatic data augmentation for generalization in reinforcement learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[Shorten and Khoshgoftaar, 2019] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):1–48, 2019.

[Tobin *et al.*, 2017] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *IROS*, pages 23–30. IEEE, 2017.

[Tunyasuvunakool *et al.*, 2020] Saran Tunyasuvunakool, Alistair Muldal, Yotam Doron, Siqi Liu, Steven Bohez, Josh Merel, Tom Erez, Timothy Lillicrap, Nicolas Heess, and Yuval Tassa. dm_control: Software and tasks for continuous control. *Software Impacts*, 6:100022, 2020.

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

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

[Zhu *et al.*, 2017] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *ICRA*, pages 3357–3364. IEEE, 2017.