Appendix

A Visual Reinforcement Learning Baselines

DrQ: This model-free, off-policy reinforcement learning algorithm, is based on Soft Actor-Critic (SAC) [19]. DrQ enhances training stability via applying data augmentation to regularize the Q value of state-action pairs. The key of DrQ is to promote similarity between augmented state-action pairs. The Q-regularization technique is shown in Eq 1, where K is the number of samples, $\mathcal{T}$ is the collection of augmentation.

$$\mathbb{E}_{\substack{s\sim\mu(\cdot)\\ a\sim\pi(\cdot|s)}}[Q(s,a)] \approx \frac{1}{K}\sum_{k=1}^{K} Q\left(f\left(s^*, \nu_k\right), a_k\right) \text{ where } \nu_k \in \mathcal{T} \text{ and } a_k \sim \pi\left(\cdot \mid f\left(s^*, \nu_k\right)\right) \tag{1}$$

DrQ-v2: An improved version of DrQ. DrQ-v2 fuses essential elements from the DDPG algorithm with data augmentation to strengthen visual RL agents' performance. DrQ-v2 also incorporates techniques such as n-step return and target critic, leading to commendable results in most of the medium and hard level DM-Control tasks. The TD-target is defined as follows, where x_{t+n} is the n-step observation, $\boldsymbol{a}_{t+n}$ is the n-step action, and $\bar{\theta}_{1,2}$ is the Q-target networks:

$$y = \sum_{i=0}^{n-1} \gamma^i r_{t+i} + \gamma^n \min_{k=1,2} Q_{\bar{\theta}_k}\left(aug(x_{t+n}), \boldsymbol{a}_{t+n}\right) \tag{2}$$

CURL: CURL integrates contrastive learning methods into the reinforcement learning training process. The auxiliary contrastive loss (Eq 3) allows the agent to obtain better image representation during training, thus mitigating the optimization difficulty under high-dimensional inputs. In our implementation, we only apply a single encoder to produce visual representations instead of two polyak-averaging encoders. This alteration improves the sample efficiency of CURL and put it on a comparable performance with DrQ-v2. More experiments are shown in Appendix F.3.

$$\mathcal{L}_q = \log \frac{\exp\left(q^T W k_+\right)}{\exp\left(q^T W k_+\right) + \sum_{i=0}^{K-1} \exp\left(q^T W k_i\right)} \tag{3}$$

PIE-G: PIE-G proposes a simple yet effective method, combining ImagNet pre-trained visual representations with the early layer and updates of BatchNorm statistical parameters to further enhance the generalization ability of the agent.

SVEA: SVEA finds that heavy data augmentation introduces additional high variance to agent training, which can lead to instability or even divergence. SVEA suggests that using the Q-values of non-augmented images as the target of estimated Q-values for augmented images (Eq 4), thus stabilizing the variance of the value estimation.

$$\left\| Q_\theta\left(aug(x_t), \mathbf{a}_t\right) - q_t^{\text{tgt}} \right\|_2^2 \tag{4}$$

SRM: SRM proposes a novel data augmentation method that operates in the frequency domain. It helps diversify data and alleviate distribution shift issues under various visual scenarios. During the training stage, SRM randomly discards parts of the frequency information from observations, forcing the policy to select suitable actions based on the remaining information. The augmentation method is shown in Eq 5, where $\mathcal{F}$ is the fast Fourier transform, $\mathbf{M}$ is a binary masking matrix, and $\mathbf{Z}$ is a random noise image.

$$\hat{\mathcal{F}}\left(o_i\right) = \mathbf{M} \cdot \mathcal{F}\left(o_i\right) + (\mathbf{1} - \mathbf{M}) \cdot \mathcal{F}(\mathbf{Z}) \tag{5}$$

SGQN: This algorithm introduces the saliency map for the use of augmenting images. Saliency maps, a tool used in computer vision, offers an interpretability analysis of encoders. SGQN retains only agent's focusing areas and removes the visual background by the generated saliency map.

This approach utilizes the augmentation objectives in SVEA [23] to further improve the model's generalization performance. The auxiliary objective is shown in Eq 6, where $M_\rho((o,a),a)$ is the binary masking matrix introduced from the saliency map.

$$L_C(\theta) = \|Q_\theta\left(o,a\right) - Q_\theta\left(o \odot M_\rho(o,a),a\right)\|^2 \tag{6}$$

B Implementation Details

Indoor navigation: Habitat serves as the simulator and extends a variety of indoor navigation tasks. We select *ImageNav* as the test env, whose goal is defined by the image of target location in the chosen map. Due to the complexity in the default training and validation episode settings, which demands extensive training periods to achieve a satisfactory standard, we simplify the setup to 500 initial positions and 1 target position. Meanwhile, we utilize the 3D scenes from the Gibson dataset as our map for all experiments.

Autonomous driving: We choose the stable version of CARLA 0.9.10 for simulation. The reward function is adopted from Zhang et al. [61]. We also implement the wrapping methods from Huang et al. [24] for novel CARLA environments. Moreover, to enhance exploration and ensure stable training, we standardized the *std_schedule* across all algorithms. Each difficulty level contains two weathers, *Easy level:* soft_high_light, soft_noisy_low_light; *Medium level:* HardRainSunset, SoftRainSunset; *Hard level:* hard_low_light, hard_noisy_low_light. The aggregated return is calculated by averaging over the weather at the same level. Further details can be accessed in the documentation provided within our GitHub repository.

Dexterous manipulation: In RL-ViGen, we select three single-view Adroit tasks. Given that tasks in Adroit typically necessitate demonstrations for successful completion, we employ VRL3, the state-of-the-art baseline for these tasks. Since the update process of VRL3 is based on DrQ-v2, it allows a seamless transfer of our algorithms to VRL3's codebase. There are three stages for VRL3 training: stage1 responses to gain a basic perception ability via pretraining on ImageNet; stage2 utilizes offline RL training with expert demonstrations; stage3 executes online training. It is noteworthy to mention that the experiment of VRL3 demonstrates that in single view tasks, only applying stage3 is sufficient to accomplish Adroit tasks with high sample efficiency. Therefore, to compare each algorithm more effectively, we exclude the use of stage1 and stage2. The aggregated success rate is calculated by averaging over all three tasks.

Table-top manipulation: SECANT [12] previously employed Robosuite for generalization testing. Building on its codebase, we adopt one of the latest versions - Robosuite 1.4.0 and mujoco 2.3.0 as well as simplified the installation process. Meanwhile, we also introduce a range of new classes of visual generalization. For each difficulty level, we deploy a variety of scenarios, and each trained agent is evaluated within each environment 10 times (in a total of 100 evaluations). The aggregated return is calculated by averaging over all three tasks.

Locomotion: In addition to the locomotion tasks from DM-Control (1.0.8 version), we also incorporate models from Mujocoreie [7], and carefully designed corresponding rewards, enabling them to accomplish *walk* or *stand* tasks. Furthermore, building on DMC-GB, we have added additional generalization categories for further enriching RL-ViGen. The aggregated return is calculated by averaging over two tasks.

Our experiments are all conducted with TeslaA40 or TeslaA100 GPU and AMD EPYC 7542 32-Core Processor CPU. More details can be founded in `https://github.com/gemcollector/RL-ViGen`.

C Hyper-parameters

We use the same hyper-parameters as the original papers and perform a small-scale grid search to achieve better performance of certain algorithms. The common hyper-parameters are listed in Table 2.

The individual hyper-parameters are listed in the following Tables. The additional hyper-parameters introduced by SGQN are listed as well.

Table 2: Common hyper-parameters in RL-ViGen.

Hyper-parameters	Value
Input size	84×84
Discount factor γ	0.99
Replay Buffer size	int(1e7)
Feature dim	DrQ(v2), CURL: 50, otherwise: 256
Action repeat	Robosuite: 1, otherwise: 2
N-step return	DrQ: 1, otherwise: 3
Optimizer	Adam
Hidden dim	1024
Frame stack	3

Table 3: CARLA hyper-parameters in RL-ViGen.

Hyper-parameters	Value
Training Frames	int(1e6)
Learning Rate	PIE-G: 5e-5, DrQ: 5e-4, otherwise: 1e-4
N-step return	1
SGQN quantile	0.9
SGQN critic weight	0.5
SGQN aux lr	8e-5

D Visualization of each difficulty level

To gain a better understanding of our setting and RL-ViGen, we visualize the images under various generalization settings and difficulty levels as mentioned in the experiment section.

D.1 Visual Appearances and Lighting Changes

D.1.1 Robosuite

In the context of Robosuite, each difficulty level comprises 10 distinct scenes. We perform 10 trials for each of these scenes (100 trials in total).

The *Easy* level includes changes of the background appearance, while the *Hard* level contains additional complexities of moving light and alterations to the robotic arm's color. The *Extreme* level further employs a dynamic video background to evaluate the trained agents' generalization abilities. The visualized figures are shown in Figure 10.

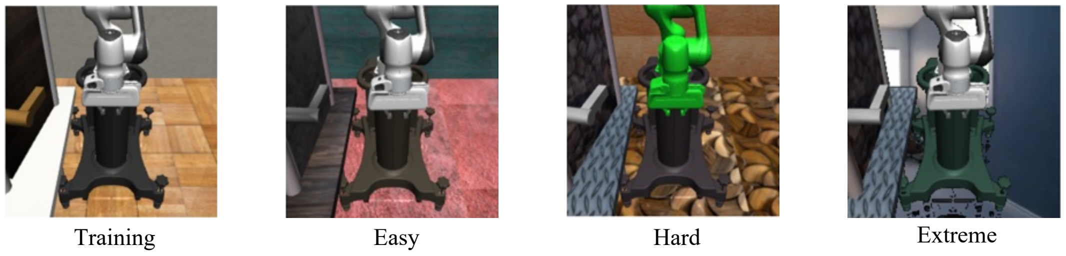


Figure 10: **The visualization of various difficulty levels of Robosuite.** This figure shows examples from the *Door* task. As the difficulty level increases, more types of distracting factors are introduced.

D.1.2 CARLA

Apart from the default weather settings in CARLA, we implement a series of challenging new scenarios. In our CARLA setup, each level of difficulty is characterized by two specific weather conditions. The *Easy* level includes *soft_noisy_low_light* and *soft_high_light*, while the *Medium*

Table 4: Habitat hyper-parameters in RL-ViGen.

Hyper-parameters	Value
Training Frames	int(7e5)
Learning Rate	1e-4
N-step return	1
SGQN quantile	0.93
SGQN critic weight	0.9
SGQN aux lr	8e-5

Table 5: Adroit hyper-parameters in RL-ViGen.

Hyper-parameters	Task	Value
Training Frames	Hammer	int(1e6)
	Door	int(1e6)
	Pen	int(2e6)
Learning Rate	Hammer	1e-4
	Door	1e-4
	Pen	1e-4
SGQN quantile	Hammer	0.9
	Door	0.9
	Pen	0.9
SGQN critic weight	Hammer	0.9
	Door	0.5
	Pen	0.9
SGQN aux lr	Hammer	8e-5
	Door	8e-5
	Pen	8e-5

level is defined by the *HardRainSunset* and *SoftRainSunset* conditions. The *Hard* level contains *hard_low_light* and *hard_noisy_low_light*. As the disparity between the novel scenarios and the training images increases, the difficulty level of generalization grows. RL-ViGen also encompasses challenging conditions such as rainy, overcast, and slippery road surfaces. The visualized figures are shown in Figure 11.

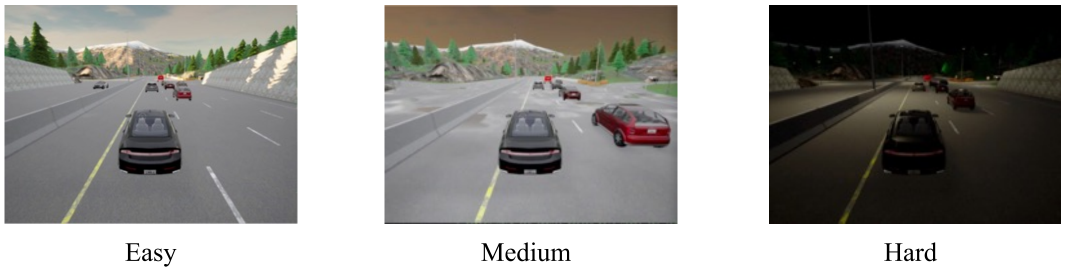


Figure 11: **The visualization of various difficult level of CARLA.** The higher the disparity from the training observations, the more challenging the new scenario.

D.1.3 Habitat

For Habitat, the Gestaltor 3D model editor is applied to modify the appearance of the scene's 3D models. A total of 10 distinct scenarios are created. The visualized figures are shown in Figure 12.

Table 6: Robosuite hyper-parameters in RL-ViGen.

Hyper-parameters	Task	Value
Training Frames	Door	int(6e5)
	Lift	int(8e5)
	TwoArmPegInhole	int(8e5)
Learning Rate	Door	1e-4
	Lift	DrQ(v2), CURL: 1e-4, otherwise: 8e-5
	TwoArmPegInhole	SGQN: 1e-4, otherise: 8e-5
Level	Door	Easy
	Lift	Medium
	TwoArmPegInhole	Medium
SGQN quantile	Door	0.9
	Lift	0.9
	TwoArmPegInhole	0.87
SGQN critic weight	Door	0.7
	Lift	0.7
	TwoArmPegInhole	0.7
SGQN aux lr	Door	8e-5
	Lift	8e-5
	TwoArmPegInhole	8e-5

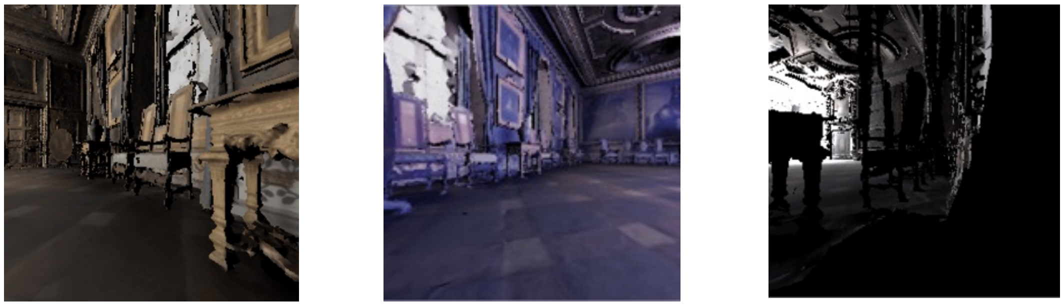


Figure 12: **The visualization of Habitat.** We create 10 distinct scenarios for the generalization of visual appearances.

D.1.4 Locomotion

For DM-Control, we further augment numerous new types of generalizations on the basis of DMC-GB. For the unitree series tasks, *Easy* and *Hard* denote two levels of difficulty regarding light color, light position, changes of light's movement and objects' color. The visualized figures are shown in Figure 13.

D.1.5 Adroit

In the Adroit environment, we provide four generalization scenarios. The *Color* setting changes the background, object color, and table texture, while the *Video* setting utilizes a dynamic background and introduces moving light. As illustrated in Figure 14, each scenario is configured with two levels of difficulty.

D.2 Camera Views

For camera view generalization, we implement alternations to the camera view through the modification of camera's orientation, position, and FOV. The visualized figure is shown in Figure 15.

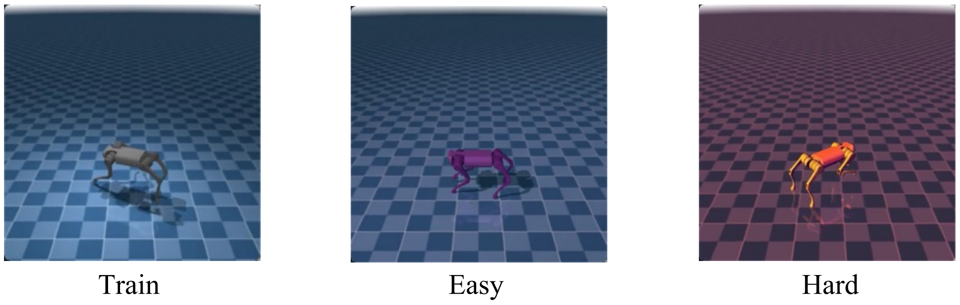


Figure 13: **The visualization of various difficulty level of DM-Control.** The figure above show examples from unitree tasks. Factors such as light color, light position, movement of light, and object color are varied.

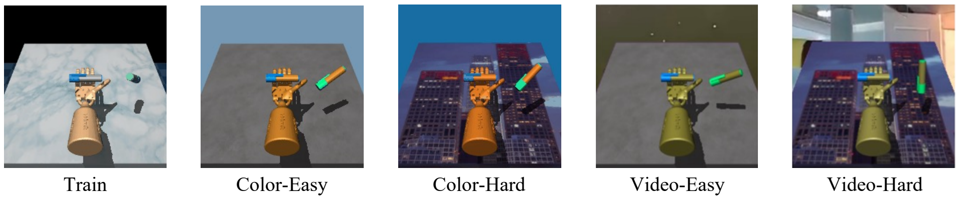


Figure 14: **The visualization of various difficulty levels of Adroit.** This figure demonstrates examples from the *Pen* task. We show four generalization scenarios provided in RL-ViGen.

D.3 Scene Structures

As shown in Figure 16, we established a variety of road scenarios in CARLA, including roundabouts, narrow paths, tunnels, etc., which can be also utilized in conjunction with other adjustable parameters. As the experiment illustrated in Figure 7, we employ the same weather conditions as those during training.

D.4 Cross Embodiments

In terms of cross-embodiment generalization, we modify the type of the robotic arm in Robosuite. In addtion, by leveraging the OSC_POSE control method, the input actions are interpreted as delta values from the current state, thus facilitating to maintain the action space dimensions and corresponding meanings.

E Visualization of each difficulty level

To gain a better understanding of our setting and RL-ViGen, we visualize the images under various generalization settings and difficulty levels as mentioned in the experiment section.

At first, we will structure the general modifications made to each generalization type. A more detailed configuration of each environment will be specified subsequently.

- Visual appearances: The generalization type of visual appearances are divided into two variations: static and dynamic changes.
  - Color: Drawing on DMC-GB [22], we categorize color variations into two levels (*easy* and *hard*) of difficulty, each containing 100 color combinations. In both levels, alterations are made to three environmental attributes: *body*, *grid*, and *skybox*. In the *easy* level, the contrast between the color combinations and the original training scene's color combinations is 0.07, while in the *hard* level, the contrast increases to 0.14.
  - Video: We divide video variations into two levels of difficulty by replacing the environment's *skybox* with videos. The *easy* level contains 10 videos, while the *hard* level

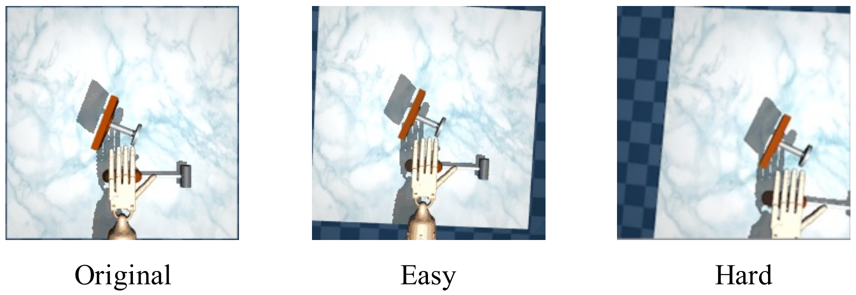


Figure 15: **The visualization of camera views of Adroit.** The larger the deviation angle of the camera, the higher the difficulty of generalization.

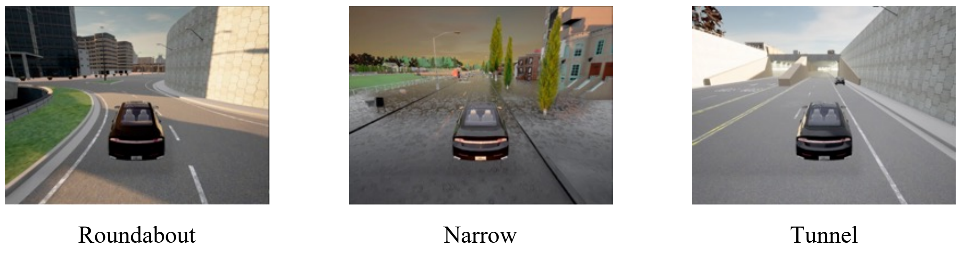


Figure 16: **The visualization of scene structures of CARLA.** We selected certain locations within different maps to serve as scenarios for scene structure generalization.

consists of 100 videos with increased contrast. Additionally, the hard level also remove the reference plane of the ground.

- Lighting conditions: The generalization type of lighting conditions are divided into two variations: static and dynamic changes. Each variations are categorized into two difficulty levels.
  - Static: We adjust the static lighting conditions by altering the position, intensity, and color.
    - Position: *Easy*: the coordinates x,y will be determined by adding values randomly sampled from the interval [-0.5, 0.5] via uniform sampling to the original light source coordinates, and the z coordinate will be selected from the range [-0.2, 0.2]; *Hard*: the coordinates x,y will be determined by adding values randomly sampled from the interval [-1, 1] via uniform sampling to the original light source coordinates, and the z coordinate will be selected from the range [-0.5, 0.5].
    - Intensity: *Easy:* the value of intensity will be uniformly sampled from the interval [0.7, 1.4]; *Easy:* the value of intensity will be uniformly sampled from the interval [1.4, 2.5].
    - Color: *Easy:* the RGB values will be determined by adding values randomly sampled from the interval [-0.1, 0.1] via uniform sampling to each channel; *Hard:* the RGB values will be determined by adding values randomly sampled from the interval [-0.2, 0.2] via uniform sampling to each channel. It should be noted that the RGB values are normalized to [0, 1].
  - Dynamic: The alteration of y coordinate is defined by adding a value within the interval [-1.2, 1.2] to the original coordinate value, with an incremental change of 0.04 at each step.
- Camera views: The generalization type of camera views are divided into two variations: static and dynamic changes. Each variations are categorized into two difficulty levels.
  - Position: *Easy*: the coordinates x,y,z will be determined by adding values randomly sampled from the interval [-0.03, 0.03] via uniform sampling to the original camera positions; *Hard*: the coordinates x,y,z will be determined by adding values randomly

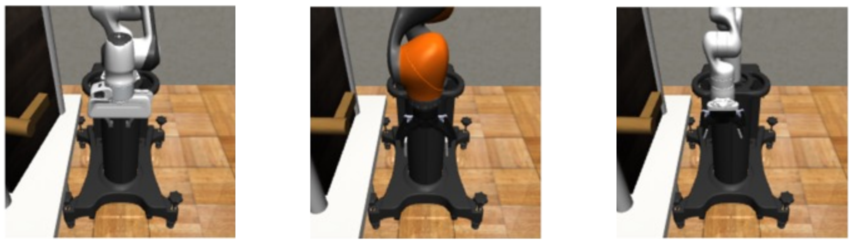


Figure 17: **The visualization of cross embodiment of Robosuite.** This figure shows examples from the *Door* task. Here, we demonstrate our modification of the style of the robotic arm for cross-embodiment generalization.

sampled from the interval [-0.07, 0.07] via uniform sampling to the original camera positions.

- Orientation: *Easy*: each component of quaternion will be determined by adding values randomly sampled from the interval [-0.03, 0.03] via uniform sampling to the original camera quaternion; *Hard*: each component of quaternion will be determined by adding values randomly sampled from the interval [-0.08, 0.08] via uniform sampling to the original camera quaternion.
- FOV: *Easy*: the coordinates x,y,z will be determined by adding values randomly sampled from the interval [-5, 5] via uniform sampling to the original FOV; *Hard*: the coordinates x,y,z will be determined by adding values randomly sampled from the interval [-10, 10] via uniform sampling to the original FOV.

Due to the challenges in defining the difficulties associated with the generalization types of the scene structures and cross embodiments, and the complexity in numerically describing the differences between scenes, we don't provide additional quantification to these two types.

F Additional Results

F.1 Generalization Evaluation

F.1.1 Locomotion

Built upon DM-Control, which has included numerous locomotion tasks, we extend this benchmark by integrating real-world robot models from Mujocoreie [7] with corresponding tasks. Moreover, RL-ViGen also augments DMC-GB with more tasks and generalization types. Here we evaluate the performance of each algorithm on the Unitree series tasks. Figure 18 demonstrates that all generalization algorithms exhibit comparable performance. More specifically, SVEA outperforms other techniques in the *Easy* setting, where the other generalization techniques do not show any advantages. In the *Hard* setting, where the agent's color closely resembles that of the surrounding environment, SGQN may not effectively capture the agent's outline, leading to a performance decline.

F.1.2 Table-top Manipulation

In Robosuite, three tasks, including single-arm and dual-arm settings, are selected in RL-ViGen: *Door*, *Lift*, and *TwoArmPegInhole*. Additionally, we create multiple difficulty levels, incorporating various visual scenarios, and dynamic backgrounds. In the *Easy* and *Medium* test environments, where considerable variations in visual colors and lighting changes are introduced, the results in Figure 19 show that PIE-G demonstrates slightly better performance than that of SGQN and SRM in *Easy* and *Medium* settings. However, when faced with the *Hard* setting that integrates dynamic video backgrounds, SRM, which mainly resorts to static frequency-based augmentation, is unable to adapt effectively to such scenarios for completing the manipulation tasks. Figure 19 further indicates that the remaining algorithms struggle to demonstrate generalization abilities in this environment.

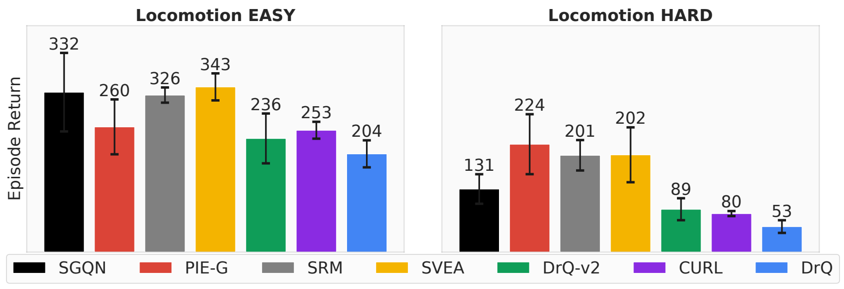


Figure 18: **Generalization score of Locomotion.** The generalization algorithms show comparable performance at two difficulty levels.

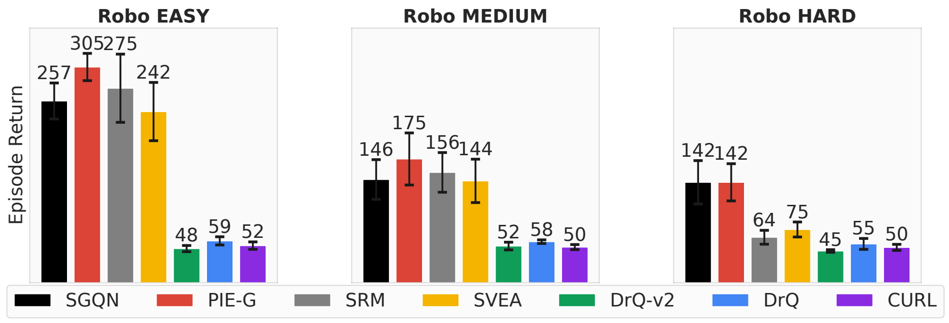


Figure 19: **The aggregated generalization score of table-top manipulation.** We present the aggregated return of three tasks for each method. PIE-G shows better generalization performance of table-top manipulation tasks when facing unseen visual scenarios.

F.2 Wall Time

So far, our main focus has been the comparison of generalization performance of each method across various tasks. In this section, we turn our attention to the comparison of each algorithm's wall-clock training time. We choose *Walker walk* task from DMControl for evaluation. This task requires a large batch size for training, thus is suitable for better demonstrating the wall-time efficiency of each approach. Frames-per-second (FPS) is selected to be the evaluation metric. Figure 20 illustrates that DrQ-v2 owns the least computational cost. Conversely, for the algorithms that utilize additional data for augmentation purposes, they tend to exhibit lower frames-per-second (FPS) rates. SGQN builds the saliency maps during every training step, which takes extra costs. Meanwhile, PIE-G utilizes the ImageNet pre-trained ResNet model to convert high-dimensional images into representations, thus adding more burden on the model's inference compared to other algorithms.

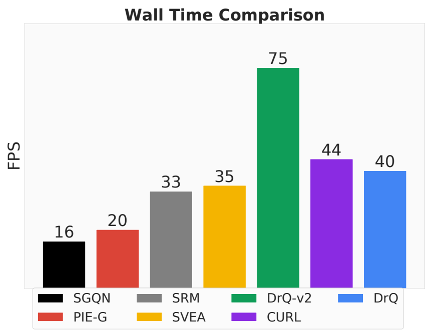


Figure 20: **Wall Time Comparison.** DrQ-v2 enjoys the lowest computational cost.

F.3 The re-implementation of CURL

CURL [33], which adopts contrastive loss as an auxiliary objective, is frequently mentioned in numerous works [57, 32, 35], yet the effectiveness of contrastive loss appears to be less pronounced [35, 36]. Distinct from prior studies, we do not utilize a target encoder and remove the update of momentum parameters related to the encoder. As shown in Figure 21, comparing to the state-of-the-art approach DrQ-v2 and the results reported in previous work [57], the use of a single shared encoder for achieving representations seems to yield more favorable results when leveraging contrastive loss.

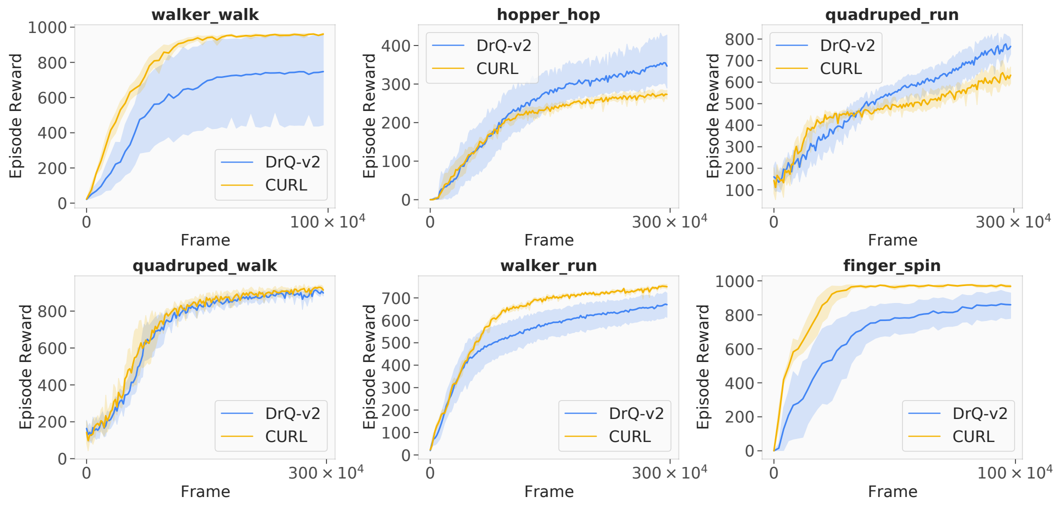


Figure 21: **The sample efficiency comparison between CURL and DrQ-v2.** Our re-implementation of CURL can achieve comparable sample efficiency with DrQ-v2.

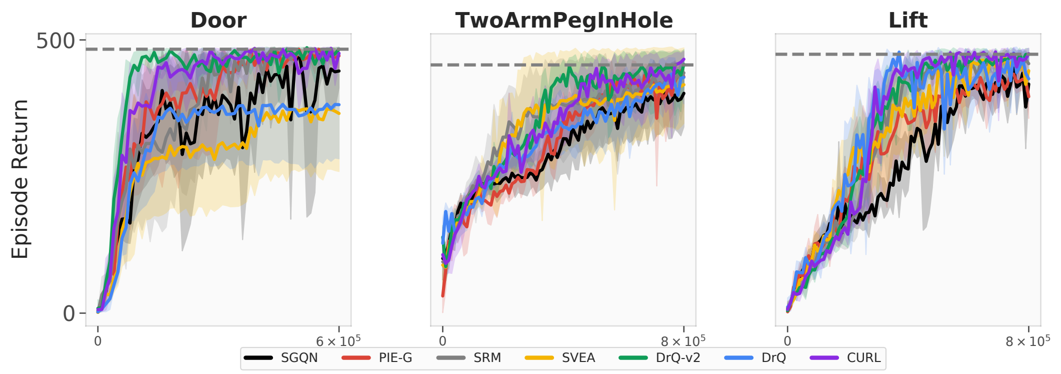


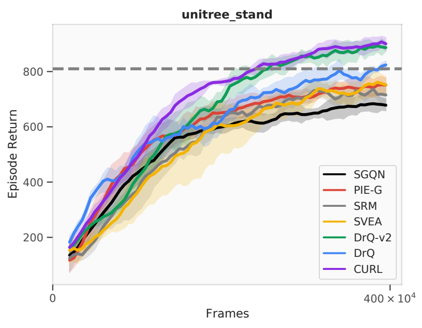
Figure 22: **Sample efficiency of Robosuite.** The episode return of each algorithm. We normalize the training steps into (0, 1). DrQ-v2 and CURL show better sample efficiency.

F.4 Sample efficiency

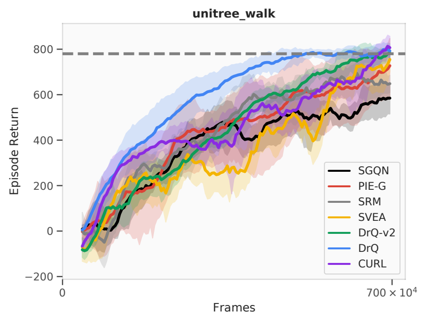
In this section, we compare the sample efficiency of various visual RL algorithms. As one of the state-of-the-art visual RL algorithms, DrQ-v2 serves as a baseline for evaluating the training performance of various algorithms across different tasks. In each figure, the convergence performance of DrQ-v2 is marked with a gray dashed line.As shown in Figure 22 and Figure 23, DrQ-v2 and CURL obtain advantageous sample efficiency in locomotion and table-top manipulation tasks. A shared attribute between these two types of tasks is that the agent is positioned at the center of observation. Hence, the additional noise introduced by data augmentation tends to exacerbate training instability.

Regarding Habitat and CARLA, as shown in Figure 24, the difference of sample efficiency across diverse algorithms is minimal. This may be attributed to the fact that both two environments employ first-person view rendered images, which makes them more robust to the extra noise. Besides, it should be noted that in CARLA, agents are required to execute fast action changes on the roads to avert collisions with surrounding vehicles. Therefore, Figure 24b demonstrates that DrQ is prone to entropy collapse, while SGQN struggles to extract salient information with many distracted factors.

In terms of Adroit, as mentioned in Section 4.1.3, the safe Q mechanism is able to endow the trained agent with robustness against noise. The sample efficiency of each algorithm is shown in Figure 5.

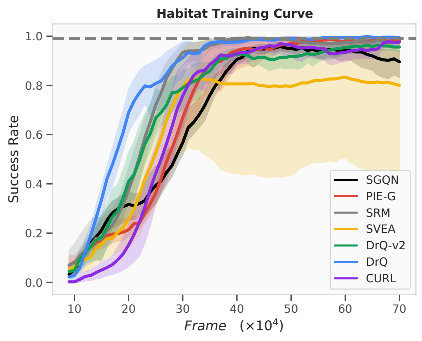


(a)

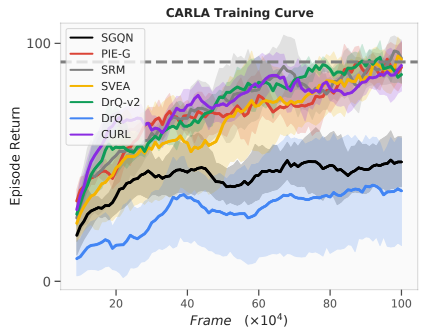


(b)

Figure 23: **Sample efficiency of Unitree tasks.** The episode return of each method. The agent, positioned at the center of observation in these tasks, is subjected to additional noise due to data augmentation.



(a)



(b)

Figure 24: **Sample efficiency of Habitat and CARLA.** We show the success rate of Habitat and the episode return of CARLA accordingly. The first-person view observations are more robust to the augmentation of adding additional noise.

F.5 Training on the full-distribution scenarios

To investigate the performance of agents under test and full-distribution scenarios, we select three locomotion tasks: *walker walk*, *finger spin*, and *walker stand*. In terms of algorithmic choices, we employ three algorithms, DrQ-v2, SRM, and SVEA, to validate the training and generalization efficacy.

We have devised three distinct visual scenarios, denoted as Scenario 1, Scenario 2, and Scenario 3 for training agents and evaluating their generalization performance. Scenario 1, which serves as the training environment within the Section 4, is constructed in a static and uncluttered visual setting. Scenario 2, employed as the testing scenario for visual appearances in our work, introduces dynamic complexity by integrating video backgrounds. Scenario 3, characterized as a full distribution scenario, further amplifies this complexity by incorporating additional visual generalization types, such as changes in camera view and lighting conditions. The visualized figure are shown in Figure 25.

First, We explore the training performance of different algorithms across various visual scenarios. As shown in Figure 27, it demonstrates that as the distribution expands and the incorporated variations increase, a noticeable decline is observed in both the sample efficiency and asymptotic performance across all algorithms.

Subsequently, we investigate the generalization performance of each trained agent across three visual environments. During the generalization testing, we conduct evaluations in a zero-shot manner. The complete generalization performance are shown in Figure 28. With the increasing complexity of the testing scenarios, the generalization scores tend to decline. Moreover, the generalization scores are directly correlated with the training performance; despite the fact that Scenario 3 incorporates the

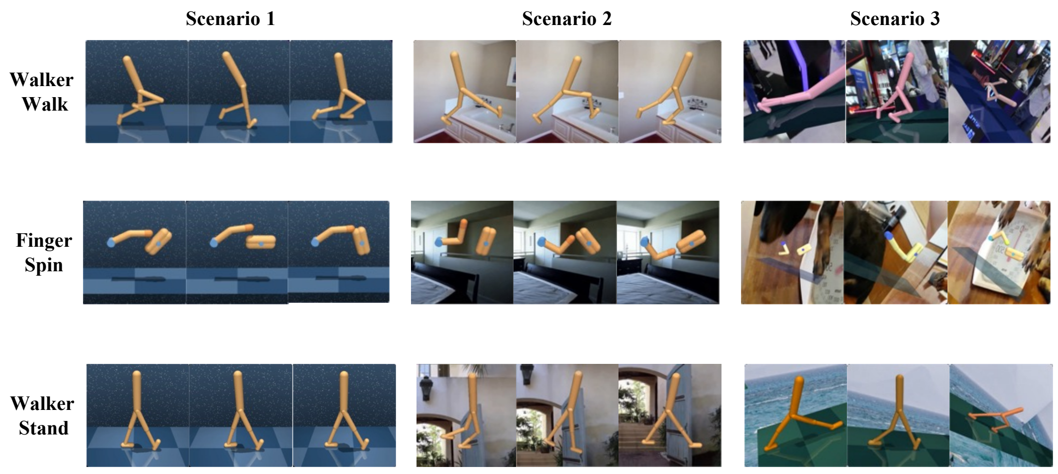


Figure 25: **Visualization of three trained tasks in different visual scenarios.** The scenario 1 is constructed in a static and clean visual scenario. Conversely, the scenario 2 introduces dynamic variations by employing a video background that alters with each episode, encompassing a total of 110 distinct videos. The scenario 3 further extends this complexity by incorporating additional visual generalization types, such as changes in camera view and lighting conditions.

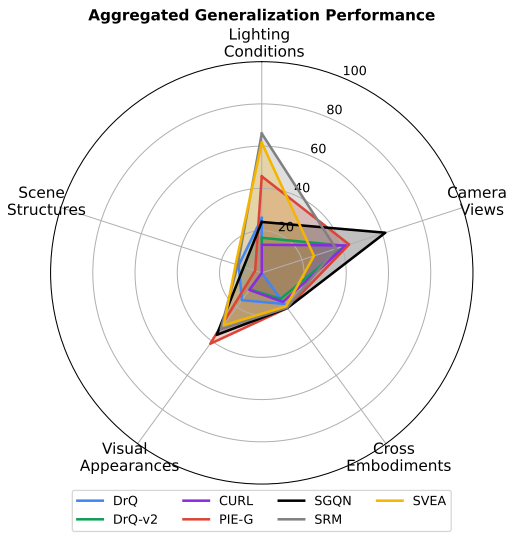


Figure 26: **Aggregated score of each algorithm.** We show the aggregated combine metric for each agent. There is no single algorithm can lead all generalization types.

most extensive array of generalization types, the inferior training performance of the agent leads to suboptimal generalization performance across various scenarios.

F.6 The aggregated score of each algorithm

Additionally, we present an aggregated metric for each algorithm across different types of generalization. We normalize the score of each environment into 0-100, and then take the average score across all environments for each generalization type. As shown in Figure 26, no single visual RL algorithm has proven capable of adeptly handling all types of generalization, especially in the types of scene structures and cross embodiments settings. In the Discussion section, we have analyse the underlying causes of these challenges. It provides a direction for guiding future efforts towards seeking approaches that can achieve further generalization advancements in these two aspects.

G Additional Related Work

Honor of Kings Arena [52] serves as another benchmark for RL generalization. Distinct from RL-ViGen, it functions as a multi-agent platform, focusing mainly on the generalization of targets and opponents rather than visual aspects. While platforms like MineRL [18] and Malmo [26], built on Minecraft [42], are capable of handling a variety of tasks, the construction of these tasks tends to be relatively simplistic without fine-grained modeling of agents and objects. Crafter [20] and the Obstacle Tower [27], on the other hand, still utilize discrete actions, and the task types they offer are limited and lack diversity. The benchmarks such as BEHAVIOR [37] and ThreeDWorld [16] present photo-realistic environments, but their task visual scenarios are also relatively narrow and are not applicable for visual generalization evaluation.

H Additional Discussion

The augmentations during training. Both DrQ and DrQ-v2 employ data augmentation techniques like random shift or random crop. Such augmentations are referred to as weak augmentations, which only introduce minor changes to the image such as slight cropping and shifting. Numerous studies [35, 57, 32] have shown that weak augmentations are indispensable for image-based RL to achieve high sample efficiency. Absence of such weak augmentations could easily fail on most tasks [35]. However, when it comes to generalization, weak augmentations cannot help agents to obtain generalization abilities [35]. On the other hand, the augmentation method utilizing extra datasets falls into the category of strong augmentations [23, 21], which substantially distorts the image. Contrary to weak augmentations, this approach is imperative to foster superior generalization capabilities, but it will hinder agent's training performance.Therefore, most generalization algorithms utilize both types of augmentations [21, 23] to achieve generalization ability while maintaining high sample efficiency.

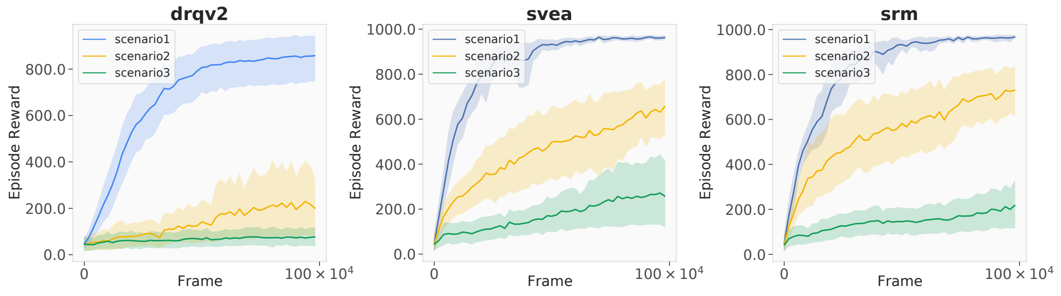
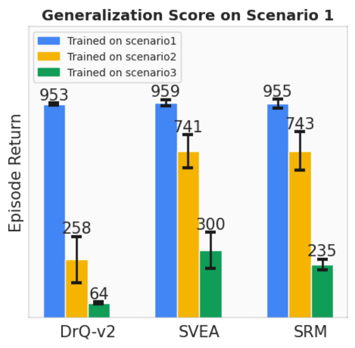
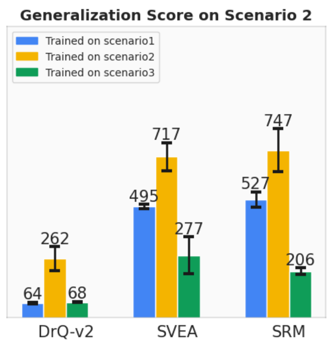


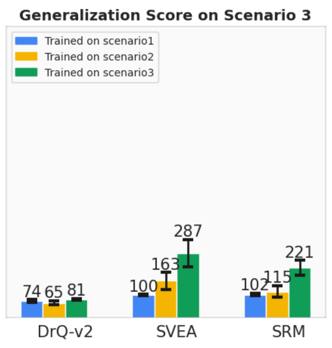
Figure 27: **Training Curves of three algorithms in different visual scenarios.** The legends are defined to represent various training curves under different scenarios. As the distribution expands and the incorporated variations increase, a noticeable decline is observed in both the sample efficiency and asymptotic performance across all algorithms.



(a)



(b)



(c)

Figure 28: **Generalization performance on different visual scenarios.** In this figure, the generalization performance of each agent is evaluated across three types of visual scenarios. Specifically, the blue, yellow, and green bars represent the generalization scores evaluated in a certain scene for the agents trained under Scenario1, Scenario2 and Scenario3. Each bar represents the average performance across the three tasks.