A. The Proof of Theorem 1

In this section, we present a potential method for establishing a connection between Bellman error $\varepsilon(\theta)$ and sub-optimal generalization $d(\pi^*, \pi_\theta)$.

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

$$d(\pi^*, \pi_\theta) \leq 2T\sqrt{C \cdot \varepsilon(\theta)}. \tag{3}$$

Proof. In the following discussions, we introduce the value function $V^\pi(\boldsymbol{s}_0)$ to represent $\mathbb{E}_\pi[Q^\pi(\boldsymbol{s}_0, \pi(\cdot|\boldsymbol{s}_0))]$. For analytical convenience, we assume that policy π is greedy, therefore,

$$\begin{aligned} d(\pi^*, \pi_\theta) &= \mathbb{E}_{\pi_\theta, \pi^*}[Q^{\pi^*}(\boldsymbol{s}_0, \pi^*(\cdot|\boldsymbol{s}_0)) - Q^{\pi_\theta}(\boldsymbol{s}_0, \pi_\theta(\cdot|\boldsymbol{s}_0))] \\ &= V^*(\boldsymbol{s}_0) - V^{\pi_\theta}(\boldsymbol{s}_0) \\ &\leq V^*(\boldsymbol{s}_0) - \mathbb{E}_{\pi^*}[Q^{\pi_\theta}(\boldsymbol{s}_0, \pi^*(\cdot|\boldsymbol{s}_0))] + \mathbb{E}_{\pi_\theta}[Q^{\pi_\theta}(\boldsymbol{s}_0, \pi_\theta(\cdot|\boldsymbol{s}_0))] - V^{\pi_\theta}(\boldsymbol{s}_0). \end{aligned} \tag{11}$$

The non-negative of the two terms added on the right-hand side is due to the greediness of π_θ. Since $Q(\boldsymbol{s}_T, \pi(\cdot|\boldsymbol{s}_T)) = 0$, for any policy π, we have

$$\mathbb{E}_\pi[Q^\pi(\boldsymbol{s}_0, \pi(\cdot|\boldsymbol{s}_0))] - V^\pi(\boldsymbol{s}_0) = \mathbb{E}_\pi[\sum_{t=0}^{T-1}(Q^\pi(\boldsymbol{s}_t, \pi(\cdot|\boldsymbol{s}_t)) - \mathbb{E}_\pi[R_t + Q^\pi(\boldsymbol{s}_{t+1}, \pi(\cdot|\boldsymbol{s}_{t+1}))])]. \tag{12}$$

Therefore, combined with the fact π is the greedy policy, the terms of Equ. (11) can be modified as follows:

$$\begin{aligned} \mathbb{E}_{\pi^*}[Q^{\pi_\theta}(\boldsymbol{s}_0, \pi^*(\cdot|\boldsymbol{s}_0))] - V^*(\boldsymbol{s}_0) &= \mathbb{E}_{\pi^*}[\sum_{t=0}^{T-1}(Q^{\pi_\theta}(\boldsymbol{s}_t, \pi^*(\cdot|\boldsymbol{s}_t)) - \mathbb{E}_{\pi^*}[R_t + Q^{\pi_\theta}(\boldsymbol{s}_{t+1}, \pi^*(\cdot|\boldsymbol{s}_{t+1}))])] \\ &\geq \mathbb{E}_{\pi^*}[\sum_{t=0}^{T-1}(Q^{\pi_\theta}(\boldsymbol{s}_t, \pi^*(\cdot|\boldsymbol{s}_t)) - \mathbb{E}_{\pi^*}[R_t + Q^{\pi^*}(\boldsymbol{s}_{t+1}, \pi^*(\cdot|\boldsymbol{s}_{t+1}))])], \end{aligned} \tag{13}$$

$$\mathbb{E}_{\pi_\theta}[Q^{\pi_\theta}(\boldsymbol{s}_0, \pi_\theta(\cdot|\boldsymbol{s}_0))] - V^{\pi_\theta}(\boldsymbol{s}_0) = \mathbb{E}_{\pi_\theta}[\sum_{t=0}^{T-1}(Q^{\pi_\theta}(\boldsymbol{s}_t, \pi_\theta(\cdot|\boldsymbol{s}_t)) - \mathbb{E}_{\pi_\theta}[R_t + Q^{\pi_\theta}(\boldsymbol{s}_{t+1}, \pi_\theta(\cdot|\boldsymbol{s}_{t+1}))])]. \tag{14}$$

Furthermore, by applying the Cauchy-Schwartz inequality and considering the condition of Assumption 1 for any policy π, we obtain the following result:

$$\begin{aligned} &\left|\mathbb{E}_\pi[\sum_{t=0}^{T-1}(Q^\pi(\boldsymbol{s}_t, \pi(\cdot|\boldsymbol{s}_t)) - \mathbb{E}_\pi[R_t + Q^\pi(\boldsymbol{s}_{t+1}, \pi(\cdot|\boldsymbol{s}_{t+1}))])]\right| \\ &\leq \sqrt{T\sum_{t=0}^{T-1}\mathbb{E}_\pi[\|Q^\pi(\boldsymbol{s}_t, \pi(\cdot|\boldsymbol{s}_t)) - \mathbb{E}_\pi[R_t + Q^\pi(\boldsymbol{s}_{t+1}, \pi(\cdot|\boldsymbol{s}_{t+1}))]\|_2^2]} \\ &\leq \sqrt{C}T\sqrt{\frac{1}{T}\sum_{t=0}^{T-1}\mathrm{d}\mu_t \cdot \|Q^\pi(\boldsymbol{s}_t, \pi(\cdot|\boldsymbol{s}_t)) - y_t^*\|_2^2} \\ &= \sqrt{C}T\sqrt{\varepsilon(\theta)}, \end{aligned} \tag{15}$$

where $y_t^* = \mathbb{E}_{\boldsymbol{s}_{t+1}\sim\mathcal{P}}[R_t + \gamma Q^{\pi^*}(\boldsymbol{s}_{t+1}, \pi^*(\cdot|\boldsymbol{s}_{t+1}))]$. Incorporating Equ. (13), Equ. (14), and Equ. (15) into Equ. (11) yields the expression:

$$\begin{aligned} d(\pi^*, \pi_\theta) &= V^*(\boldsymbol{s}_0) - V^{\pi_\theta}(\boldsymbol{s}_0) \\ &\leq -\mathbb{E}_{\pi^*}[\sum_{t=0}^{T-1}(Q^{\pi_\theta}(\boldsymbol{s}_t, \pi^*(\cdot|\boldsymbol{s}_t)) - \mathbb{E}_{\pi^*}[R_t + Q^{\pi^*}(\boldsymbol{s}_{t+1}, \pi^*(\cdot|\boldsymbol{s}_{t+1}))])] + \\ &\quad \mathbb{E}_{\pi_\theta}[\sum_{t=0}^{T-1}(Q^{\pi_\theta}(\boldsymbol{s}_t, \pi_\theta(\cdot|\boldsymbol{s}_t)) - \mathbb{E}_{\pi_\theta}[R_t + Q^{\pi_\theta}(\boldsymbol{s}_{t+1}, \pi_\theta(\cdot|\boldsymbol{s}_{t+1}))])] \\ &\leq 2T\sqrt{C \cdot \varepsilon(\theta)}, \end{aligned} \tag{16}$$

which finishes the proof. This inequality above shows that the generalization error is bounded by policy divergence and Bellman error in test environments.

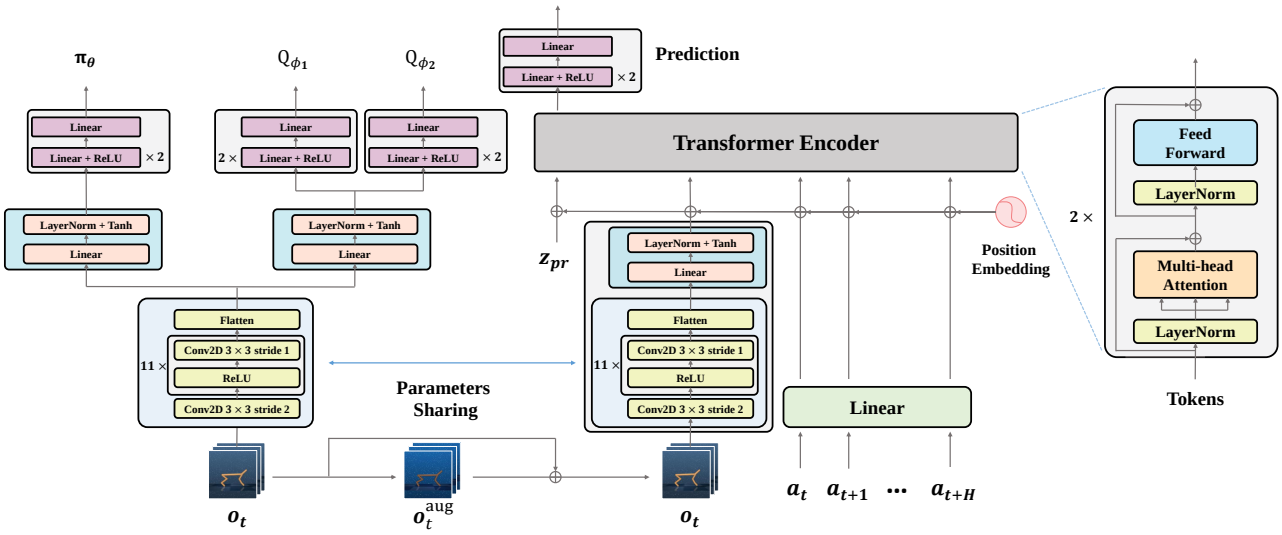


Figure 6: **TRP neural network architecture.**

B. Implementation Details

In this section, we describe the implementation details of TRP, including network architecture and important hyperparameters. Since the setup of all experiments is similar, we illustrate the specifics of DMC as a representative example. Additionally, we present details of our architecture in Figure 6.

Network architectures

Shared Observation Encoder. We adopt the network architecture introduced by SVEA without modifying the architecture or hyperparameters for a fair comparison. For shared observation encoder f_ψ, we employ an 11-layer convolutional neural network. This network processes stacked RGB images of size $3 \times 84 \times 84$ into a $32 \times 21 \times 21$ feature map without padding, where 32 represents channels. Specifically, the first convolutional layer uses 3×3 kernels, 32 filters, and 2 strides. Subsequent convolutional layers employ 3×3 kernels, 32 filters, and 1 stride. Note that independent policy, value, and prediction networks follow observation encoder f_ψ.

Policy and Value Networks. For policy network π_θ, following SAC, we adopt a fully connected network with three layers, each with a hidden size of 1024. Specifically, we first project the $32 \times 21 \times 21$ feature map into a 100-dimensional embedding with Layer Normalization. Then, we feed it into a three-layer fully connected network to obtain the mean and variance of the action distribution. The reparameterization trick is then employed to formulate the policy as a Gaussian distribution. For value network Q_ϕ, we adopt a double Q network to alleviate the overestimation of Q values. Each Q network employs a 3-layer fully connected network. Analogous to the policy network, the $32 \times 21 \times 21$ feature map is projected to a 100-dimensional embedding with Layer Normalization at the beginning.

Transformer based Predictor. The standard Transformer processes a 1D sequence of tokens as input. VIT reshapes image $\boldsymbol{o} \in \mathbb{R}^{H \times W \times C}$ into a sequence of flattened 2D patches with dimensions $N \times P^2 \times C$, where P is the patch size. Different from both standard Transformer and VIT, we combine the prediction token, observation token, and action tokens into a sequence $(\boldsymbol{z}_{pr}, \boldsymbol{o}_t, \boldsymbol{a}_t, \boldsymbol{a}_{t+1}, \cdots, \boldsymbol{a}_{t+H})$ that serves as the input. Notably, each token is represented by a 100-dimensional embedding. After that, we feed the input into the Transformer Encoder and map the initial 100 dimensions of the output to a truncated return with the prediction head. Specifically, we utilize the observation encoder, as described earlier, to get the observation token. Action tokens are obtained from the action encoder (a single linear layer). The prediction token is a learnable embedding. Furthermore, position embeddings are added to the tokens to retain positional information. The Transformer Encoder includes two stack units. Each unit contains two sub-layers (fully connected layers or Multi-head Attentions), followed by Layer Normalization. A residual connection is applied around each of these sub-layers. The prediction head comprises a three-layer fully connected network.

Hyperparameters

Most hyperparameters are inherited from SVEA. TRP introduces an important hyperparameter, *i.e.*, the length H of the sampled truncated trajectories. We maintain most hyperparameters consistent across different environments, adjusting only a few

Table 4: **Hyperparameters** adopted in DMControl-GB.

Hyperparameter	Value
Frame rendering	$84 \times 84 \times 3$
Stacked frames	3
Action repeat	4 (Walker run, Cheetah run, Walker walk), 8 (Cartpole swingup), 2 (Finger spin)
Random shift	Up to ± 4 pixels
Discount factor γ	0.99
Episode length	1000
Number of frames	$0.5M$
Replay buffer size	$0.5M$
Optimizer (actor θ, critic ϕ)	Adam ($lr = 1e-3, \beta_1 = 0.9, \beta_2 = 0.999$)
Optimizer (α_{entropy} of SAC)	Adam ($lr = 1e-4, \beta_1 = 0.5, \beta_2 = 0.999$)
Optimizer (σ of auxiliary learning updates)	Adam ($lr = 3e-4, \beta_1 = 0.9, \beta_2 = 0.999$)
Batch size	256
Length of truncated trajectory	$200/$action repeat
Target networks update frequency	2
Target networks momentum coefficient	0.05 (encoder), 0.01 (critic)
Auxiliary updates frequency	2
TRP coefficients	$\alpha = 0.5, \beta = 0.5$

parameters related to the image size based on the task requirements. As an illustrative example, we adopt the hyperparameter configuration used in DMControl-GB. Table 4 summarizes the crucial hyperparameters used in all experiments.

C. Environment Details

DeepMind Control Suite

As shown in Figure 7, DMControl-GB is a benchmark modified from DMC, specifically designed for evaluating visual RL generalization. The environments support random color variations and the replacement of backgrounds with dynamic videos.

Robotic Manipulation Task

As shown in Figure 8, Robotic Manipulation tasks are introduced in SQGN. We consider two tasks, *Reach* and *PegBox*. We train each method for 250000 steps and test over 30 episodes to calculate the average return. Notably, the observations have dimensions of $3 \times 84 \times 84$, while the action repeat is set to 4. Importantly, each episode consists of 1000 steps.

CARLA

Carla is a popular autonomous driving simulator. In our experiment, we adopt a stable version of CALAR0.9.6 and choose Town04 as the driving environment. We train our method under a fixed weather condition and evaluate the performance under different weather scenarios. This task aims to drive as far as possible within 1000 steps. Meanwhile, the agent must avoid colliding with 20 other vehicles or barriers. The observations are depicted in Figure 9. Similar to DBC, we utilize five cameras with a 300-degree view as our observation sensors. Each camera captures images of dimensions $3 \times 84 \times 84$. Thus, the observations consist of images with dimensions $3 \times 84 \times 420$. The action involves two continuous controls, thrusting and steering. We train all methods for 100000 steps and test over 20 episodes to calculate the average return. Following the setup of DBC, the reward function encourages the vehicle to move forward while penalizing collisions. This formulation is as follows:

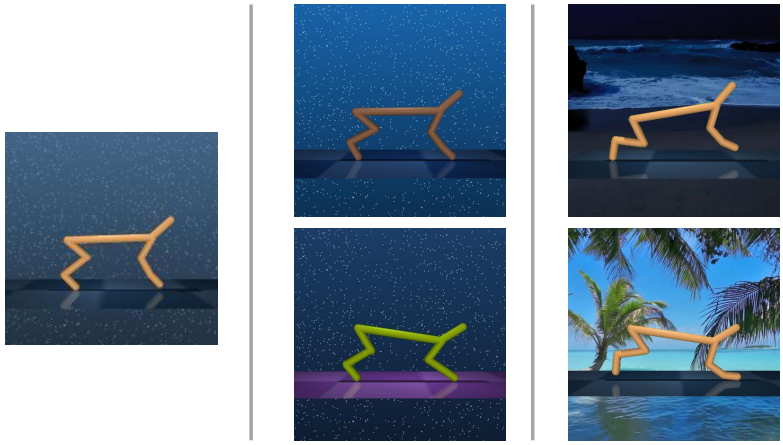
$$R_t = \mathbf{v}_{\text{ego}}^{\top} \hat{\mathbf{u}}_{\text{highway}} \cdot \Delta t - \lambda_i \cdot \text{impulse} - \lambda_s \cdot |\text{steer}|\,, \tag{17}$$

where $\mathbf{v}_{\text{ego}}$ indicates the velocity vector of the ego vehicle projected to the highway, $\hat{\mathbf{u}}_{\text{highway}}$ is the unit vector of the highway, $\Delta t = 0.05$ is the unit time, impulses measure the collisions in Newton-seconds, and $|\text{steer}|$ denotes the penalty of steering angle. We adopt $\lambda_i = 10 - 4$ and $\lambda_s = 1$ in our experiment.

D. Additional Results

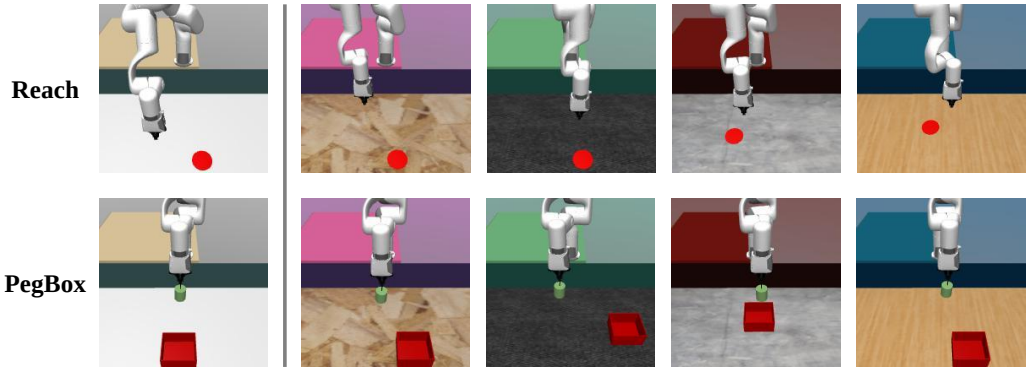
Evaluation on CARLA

To demonstrate the generality of TRP, we conduct a series of experiments on the Carla simulator. The experimental results are presented in Table 5. Specifically, we utilize the reward function and driving distance as our key evaluation metrics. We run three times for each algorithm and calculate the average value. The experimental results indicate that our method travels the longest distance and achieves the highest return. SGQN fails to achieve competitive generalization performance, possibly due to the need to address more complex information within Carla compared to DMControl-GB or Robotic Manipulation. Therefore, the acquisition of a pixel-level mask by SGQN becomes challenging.



(a) Train (b) Color Hard (c) Video Easy

Figure 7: **Examples of DMControl-GB.**



(a) Train (b) Test1 (c) Test2 (d) Test3 (e) Test4

Figure 8: **Examples of Robotic Manipulation.**

Hyperparameter Sensitivity

We conduct experiments on DMControl-GB to investigate the hyperparameters sensitivity of TRP, *i.e.*, trajectory length H. As shown in Table 6, the results demonstrate that different trajectory lengths have little impact on the overall generalization performance. However, the longer the trajectory length, the better the experimental results exhibit. We believe the phenomenon is due to the truncated return being more inaccurate for a shorter trajectory.

Visualization of HeatMaps

We utilize Gradient Class Activation Map (Grad-CAM) to visualize heatmaps of observations from DMControl-GB tasks. Figure 10 illustrates the visualizations of tasks in the *color-hard* setting. In the *walker run* task, the agent assigns heightened attention to the specific points of its feet, knees, and calves. Similarly, in the *cheetah run* task, greater attention is directed towards the trunk and feet. In the *walker walk* task, attention is distributed across both legs, knees, and the torso. During the *cartpole swingup* task, the key focus areas are the apex of the swing bar and the cart. Lastly, in the *finger spin* task, the agent emphasizes the fingertip and the unconstrained part of the body. In addition, we present the visualization results for the *video-easy* setting, as shown in Figure 11. Note that the attention patterns for various tasks remain similar to those observed in the *color-hard* setting. This consistency suggests that our method demonstrates effective generalization across different visual distractions. However, we observe that the agent in the *video-easy* setting also exhibits some interest in the background, which may explain the performance degradation.

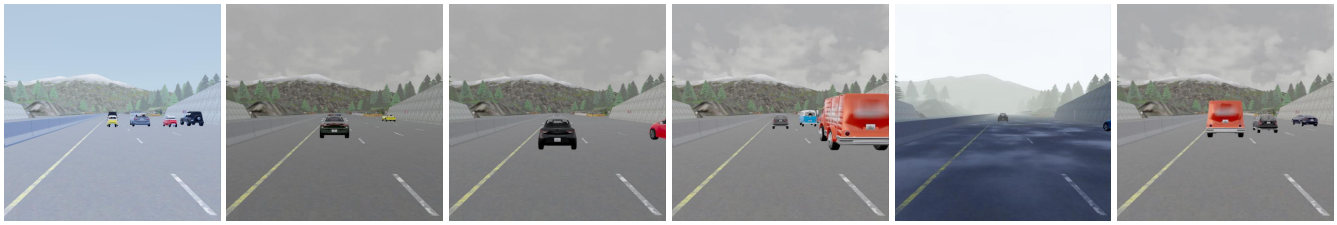


Figure 9: **Examples of Carla simulator.**

Table 5: **Results in Carla.**

Weather	metrics	SVEA	SGQN	TRP
HardRainNoon	Reward	193	62	**210**
	Distance	213	73	**236**
WetCloudySunset	Reward	187	78	**206**
	Distance	207	91	**226**
MidRainSunset	Reward	202	81	**203**
	Distance	**224**	89	**224**
SoftRainNoon	Reward	188	111	**208**
	Distance	209	118	**229**
HardRainSunset	Reward	191	79	**204**
	Distance	212	87	**226**

Table 6: **Results of parameter sensitivity.**

DMControl-GB (color-hard)	$H = 1$	$H = 10$	$H = 200$/action repeat
walker, run	360 ± 30	337 ± 36	$\mathbf{372 \pm 25}$
cheetah, run	457 ± 25	463 ± 28	$\mathbf{476 \pm 21}$
walker, walk	801 ± 53	806 ± 51	$\mathbf{823 \pm 28}$
cartpole, swingup	747 ± 55	768 ± 31	$\mathbf{799 \pm 40}$
finger, spin	891 ± 12	870 ± 50	$\mathbf{918 \pm 34}$
DMControl-GB (video-easy)	$H = 1$	$H = 10$	$H = 200$/action repeat
walker, run	403 ± 29	373 ± 37	$\mathbf{426 \pm 44}$
cheetah, run	424 ± 41	437 ± 20	$\mathbf{443 \pm 25}$
walker, walk	$\mathbf{889 \pm 30}$	871 ± 27	871 ± 18
cartpole, swingup	643 ± 109	706 ± 38	$\mathbf{733 \pm 62}$
finger, spin	813 ± 51	814 ± 43	$\mathbf{816 \pm 26}$

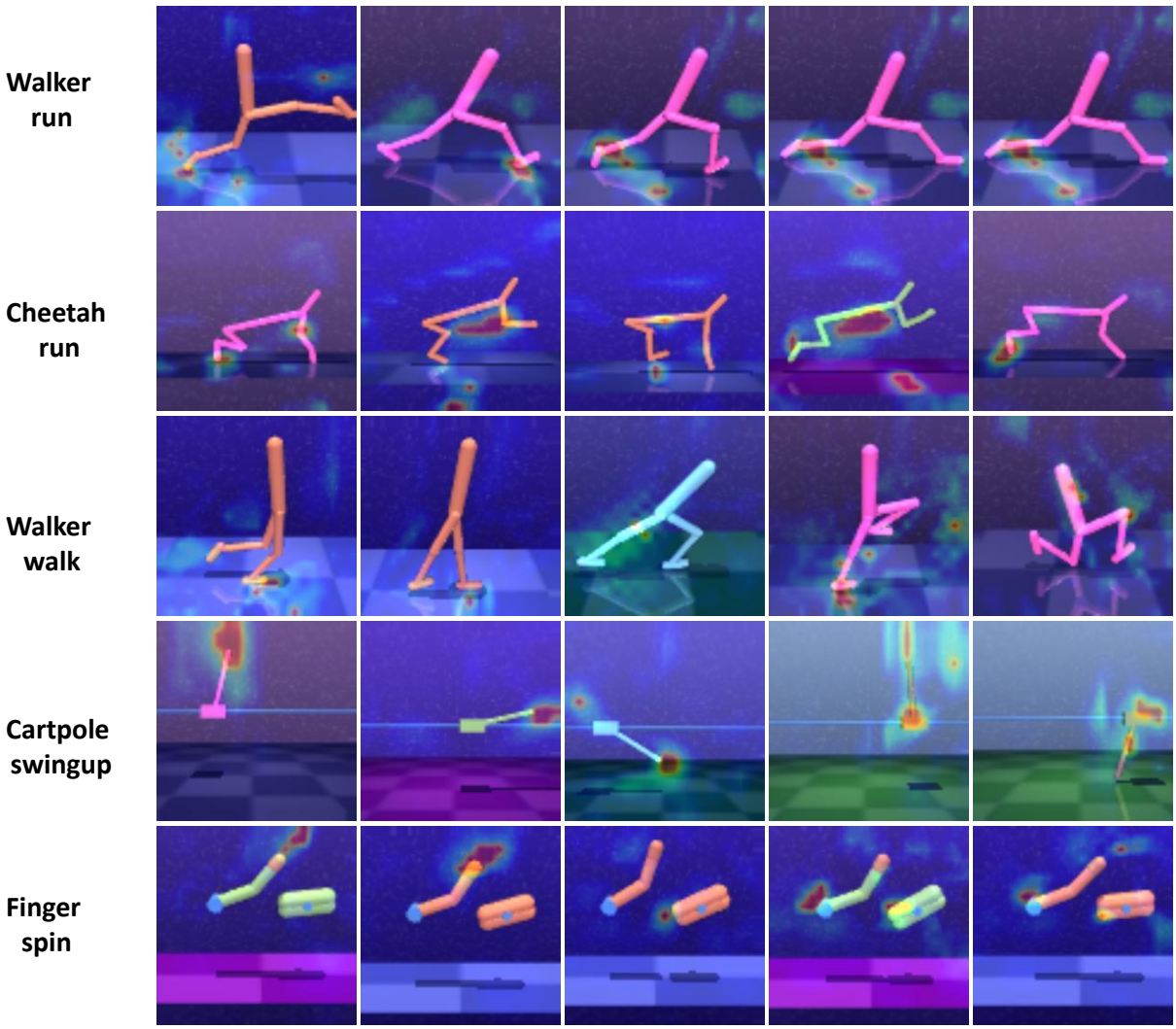


Figure 10: **Visualizations of heatmaps over *color-hard*.**

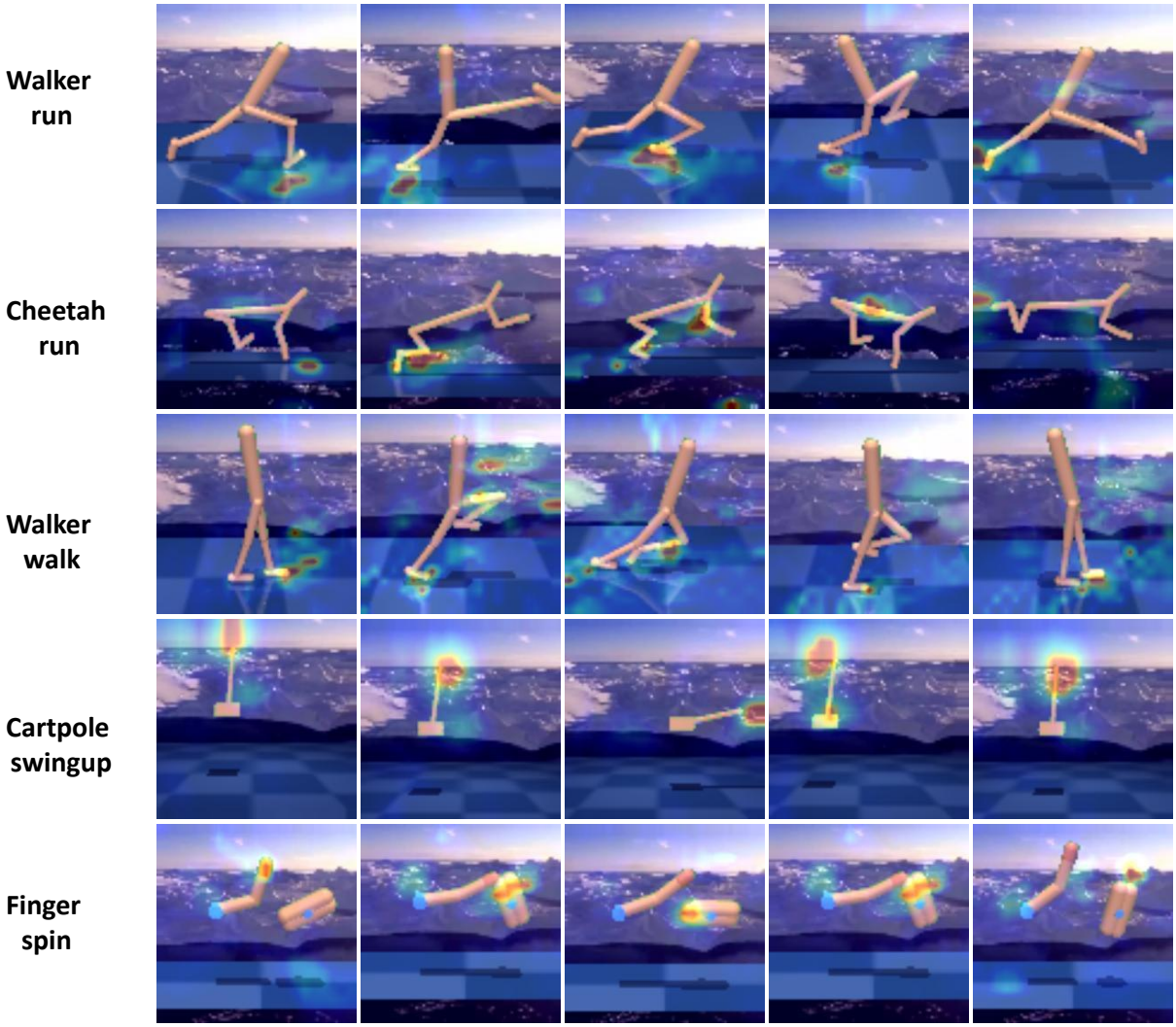


Figure 11: **Visualizations of heatmaps over *video-easy*.**