Learning Robust Representations for Visual Reinforcement Learning via Task-Relevant Mask Sampling

Anonymous authors
Paper under double-blind review

Abstract

Humans excel at isolating relevant information from noisy data to predict the behavior of dynamic systems, effectively disregarding non-informative, temporally-correlated noise. In contrast, existing visual reinforcement learning algorithms face challenges in generating noise-free predictions within high-dimensional, noise-saturated environments, especially when trained on world models featuring realistic background noise extracted from natural video streams. We propose Task Relevant Masks Sampling (TRMS), a novel approach for identifying task-specific and reward-relevant masks. TRMS utilizes existing segmentation models as a masking prior, which is subsequently followed by a mask selector that dynamically identifies subset of masks at each timestep, selecting those most probable to contribute to task-specific rewards. To mitigate the high computational cost associated with these masking priors, a lightweight student network is trained in parallel. This network learns to perform masking independently and replaces the Segment Anything Model (SAM)-based teacher network after a brief initial phase ($< 10 - 25\%$ of total training). TRMS enhances the generalization capabilities of Soft Actor-Critic agents under distractions, achieves better performance on the RL-Vigen benchmark, which includes challenging variants of the DeepMind Control Suite, Dexterous Manipulation and Quadruped Locomotion tasks.

1 Introduction

Visual Reinforcement Learning (RL) has garnered considerable success in mastering complex behaviors derived directly from high-dimensional, image-based observations (Mnih et al., 2015; Levine et al., 2016; Lee et al., 2020a; Laskin et al., 2020b). Conventionally, it is presumed that environmental observations, often obtained through hand-crafted features, contain only task-relevant information (Ha & Schmidhuber, 2018; Hafner et al., 2020; 2021; Hansen et al., 2024). This assumption enables RL algorithms to function in controlled settings with maximum efficiency, as it eliminates exogenous noise (irrelevant or uncontrollable external factors), such as weather fluctuations or random background movements, that could disrupt or impede the learning process. In the real world, the landscape is vastly different, rich in complex visual information, much of which is irrelevant to a specific task. The true challenge lies in accurately distinguishing task-relevant data while avoiding the unnecessary modeling of exogenous noise. Traditional RL approaches often fails to provide robust representations under noise, consequently failing to generalize. As a result, they inadvertently incorporate irrelevant data into their representations, leading to the modeling of noise dynamics.

Recent approaches have sought to address this by selectively masking irrelevant parts of the input to focus learning on relevant information. Focus then Decide (FTD) by Chen et al. (2024) leverages the Segment Anything Model (SAM) (Kirillov et al., 2023) to select relevant segments via attention scores and attains high rewards. However, using SAM throughout the training process incurs high computational costs, making FTD impractical for complex tasks that require numerous masks. Contrary to this, methods like SGQN (Bertoin et al., 2022) use binarized attribution maps as masks to enforce consistency between the Q-values of masked and original images, highlighting relevant areas, but provides sub-optimal rewards and is extremely sensitivity to hyperparameter choices. InfoGating (Tomar et al., 2023) focuses only on offline RL experiments, using a multi-step inverse dynamics model and U-Nets (Ronneberger et al., 2015) to mask irrelevant features.

Finally, SAM-G (Wang et al., 2023) employs the SAM model but depends on human intervention for mask selection. However, it remains unclear how combining multiple encoders yields keypoints for task-relevant masks.

To achieve robust performance and yield high rewards in noise-prone environmental settings, we propose TRMS, a novel algorithm that leverages existing masking techniques to learn task-relevant masks and to filter out irrelevant segments. This approach improves agent's generalization capabilities with respect to noise in multiple environments. The core idea of TRMS lies in employing pre-existing masking algorithms that extract meaningful substructures within an image by segmentation. Much like how the human brain processes visual information: decomposing scene into objects and selectively attending to task-relevant elements (Kaiser et al., 2016; Seidl et al., 2012a; Peters & Kriegeskorte, 2021), our method focuses on high-level abstractions, bypassing pixel-level relevance assessments. By segmenting the image into semantically meaningful subregions, we substantially reduce the complexity of the selector's task, requiring it to identify relevance from a more refined subset of the scene. This approach sharply contrasts with methods that attempt to infer such abstractions from raw pixel data (Bertoin et al., 2022; Hansen et al., 2021; Grooten et al., 2024), which inevitably suffer from less efficient learning (as shown in the evaluations through empirical results). Instead of evaluating every pixel individually, our method simplifies the process to determining whether a mask (representing a smaller subset of pixels) is correct or not. A selector network, utilizing a Convolutional Neural Network (CNN), provides a binary output for each mask, classifying it as relevant or irrelevant.

To improve the computational efficiency, we include the student network in our training procedure. The pre-trained segmentation model remains frozen throughout training and is used solely to generate output masks during inference. The approach is executed in two key phases: (i) Student Network is used to learn the mask generation over a subset of the batch to mitigate the overhead of processing the entire batch. (ii) After T_{train} steps, the student network, a lightweight CNN architecture, replaces the teacher network, enabling faster computations. An empirical evaluation of wall time is shown in Appendix Section D. Since the student network is co-optimized with the encoder, no auxiliary loss function is required beyond the initial masking loss during the first phase. The encoder's loss alone is sufficient to guide the optimization process.

To evaluate the performance of TRMS, we conducted experiments across modified version of **eleven** environments from three benchmarks in RL-ViGen (Yuan et al., 2023): the Deepmind Control Generalization Benchmark (Hansen et al., 2021), Quadruped Locomotion (Hansen et al., 2021) and Dexterous Manipulation (Rajeswaran et al., 2018). Moreover, TRMS outperforms **eight** well-established methods in various tasks in vision-based reinforcement learning (Hansen et al., 2021; Yuan et al., 2022b; Huang et al., 2022; Bertoin et al., 2022; Yarats et al., 2021; 2022; Laskin et al., 2020a; Wang et al., 2022), achieving better performance across multiple environments.

Our main contributions are:

- We propose TRMS, a novel algorithm with an actor-critic architecture that enhances task-relevant masking by leveraging pre-trained segmentation for providing semantically meaningful subregions, with a selector that identifies task-relevant information from these subregions, improving both generalization and robustness in visually complex environments.
- We further optimize TRMS through a two-phase training process, where a lightweight student network incrementally replaces a teacher network, enabling faster mask learning and improving computational efficiency as compared to solely relying on heavy segmentation model (SAM).
- We validate TRMS through comprehensive experiments across eleven modified environments from the RL-ViGen benchmarks, where it surpasses relevant existing methods in most of the environments in vision-based reinforcement learning tasks.

2 Related Work

Generalization in Visual RL. RL agents struggle with severe generalization limitations, where performance degrades sharply in unfamiliar environments due to overfitting and insufficient adaptability to unseen variations (Kirk et al., 2023; Jiang et al., 2023; Raileanu et al., 2021; Zhang et al., 2018; Cobbe et al., 2019). Numerous methods have been developed to improve generalization in reinforcement learning, including domain adaptation (Xing et al., 2021b; Li et al., 2022; Sun et al., 2022), domain randomization (Mehta et al., 2020; Lee et al., 2020b; Tobin et al., 2017), and curriculum learning (Narvekar et al., 2020; Gupta et al., 2022). Contrastive learning (Laskin et al., 2020a; Agarwal et al., 2021; Liu et al., 2023a; Yang et al., 2022), bisimulation metrics (Ferns et al., 2011; Zhang et al., 2021; Liu et al., 2023b; Sun et al., 2024; Zang et al., 2022), data augmentations (Hansen & Wang, 2021; Laskin et al., 2020b; Raileanu et al., 2021; Yarats et al., 2021; 2022; Mumuni & Mumuni, 2022), keypoints (Wang et al., 2021; 2023), and information-theoretic approaches (Tomar et al., 2023; Fan & Li, 2022; Dave & Rueckert, 2024; You et al., 2022; Wang et al., 2024) improve state representations, whereas imitation learning builds policies robust to perturbations (Fan et al., 2021; Xing et al., 2021a; Wang & Hager, 2024).

Numerous works have proposed different ideas to mitigate the impact of task-irrelevant distractors in reinforcement learning environments (Yarats et al., 2022; Hansen et al., 2021; Huang et al., 2022; Laskin et al., 2020a; Yang et al., 2023; Yuan et al., 2022a; Wang et al., 2024). SODA (Hansen & Wang, 2021) incorporates a BYOL-like (Grill et al., 2020) architecture and augments data by linearly combining supplementary images with observations. TLDA (Yuan et al., 2022a) takes a different approach, recommending the exclusion of task-critical pixels from augmentation, determined through the use of Lipschitz constants. PIEG (Yuan et al., 2022b) employs a pre-trained ResNet (He et al., 2016) as its backbone for generalization under distractors. SRM (Huang et al., 2022) learns representations in frequency-domain and learns to discard certain frequency in the observation to address domain shifts. CG2A (Liu et al., 2023c) identifies potential conflicts among gradients generated by different augmentations and investigates how to better integrate these augmentations.

Masking Distractors in RL. Several approaches have been proposed to enhance the generalization of RL agents by selectively masking parts of the input. SGQN (Bertoin et al., 2022) proposes a saliency-guided method, where binarized attribution maps serve as input masks. It regularizes the value function by enforcing consistency between the Q-values of the masked and original state images, improving learning focus on relevant areas. MLR (Yu et al., 2022) introduces a self-supervised auxiliary objective that performs random masking and reconstructs masked information in the latent space, encouraging dynamic-relevant state representations. InfoGating (Tomar et al., 2023) utilizes a multi-step inverse dynamics model as its primary objective and employs U-Nets to mask irrelevant information, with a focus on offline RL experiments. MaDi (Grooten et al., 2024) closely aligns to our approach in utilising a small CNN for masking (similar to our student network) and using reward-driven supervision to suppress irrelevant pixels, but learns masks end-to-end without leveraging structured segmentation priors. SAM-G (Wang et al., 2023) employs a frozen Segment Anything Model (SAM) (Kirillov et al., 2023) model to generate observation masks. However, it fundamentally differs in its reliance on foundation-model prompts, necessitating human intervention for mask selection. Moreover, the mechanism by which multiple encoders are combined to yield task-relevant masks remains unclear. FTD (Chen et al., 2024) also uses the SAM to select relevant segments via attention scores and regularizes the RL method with inverse dynamics and reward loss. However, the use of SAM throughout the training process results in high computational costs, making it impractical for complex tasks that require numerous masks. These methods rely on unstructured mask learning, costly vision models, or lack spatial selectivity. In contrast to these methods, TRMS combines segmentation priors from SAM with a dynamic selector and lightweight student network.

Relation to Cognitive Perception. Selecting visual information from cluttered real-world scenes requires aligning visual input with the observer's attentional set—an internal representation of objects relevant to current behavioral goals—and as these goals shift, a new attentional set must be instantiated, necessitating the suppression of the previous set to prevent distractions from irrelevant objects (Nasr et al., 2008; Seidl et al., 2012b). Segmentation allows for the tracking and prediction of objects, enabling cognitive functions like memory and action planning independent of sensory input (Scholl, 2001; Brady et al., 2011). Processes such as perceptual grouping, proto-objects, and object files underpin how humans segment and recognize relevant objects in complex scenes (Roelfsema & Ooyen, 2005; Gao et al., 2016). Inspired by human-like segmentation and attention mechanisms, we introduce a segment sampling strategy that leverages masking priors to learn robust and task-relevant representations.

3 Preliminaries

3.1 Visual Reinforcement Learning

Reinforcement Learning (RL) aims to obtain optimal policies for sequential decision-making problems through iterative interactions with the environment (Sutton, 2018). In contexts where agents receive high-dimensional sensory inputs, such as visual observations, these problems are aptly modeled as Partially Observable Markov Decision Processes (POMDPs). A POMDP is formally defined by the tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{O}, P, \Omega, R, \gamma \rangle$, where $\mathcal{S}$ is the set of latent states, $\mathcal{A}$ is the action space, $\mathcal{O}$ is the observation space, $P : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$ is a state transition function, $\Omega : \mathcal{S} \to \mathcal{P}(\mathcal{O})$ is the observation function, $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, and $\gamma \in [0, 1)$ is the discount factor, which attenuates future rewards. To mitigate the challenges associated with partial observability (Kaelbling et al., 1998), we redefine the agent's state s_t as a sequence of k consecutive observations, namely $s_t = (o_t, o_{t-1}, \ldots, o_{t-k+1})$, where each $o_i \in \mathcal{O}$. Although this window does not capture the full action-observation history, it often serves as a practical surrogate in visual domains, enriching the agent's perceptual input and partially recovering temporal dependencies lost in single-frame observations. The agent's objective is to obtain a policy π_{ϕ_a}, parameterized by ϕ_a, that maximizes the expected cumulative discounted return: $\mathbb{E}_{\kappa \sim \pi_{\phi_a}} [\sum_{t=0}^{\infty} \gamma^t R(o_t, a_t)]$, where the trajectory κ is induced by the underlying dynamics of the POMDP.

3.2 Soft Actor-Critic (SAC)

Our methodology builds upon the Soft Actor-Critic (SAC) algorithm (Haarnoja et al., 2018), a model-free, off-policy RL approach that integrates entropy maximization into the policy optimization framework to enhance exploration and stabilize learning. SAC employs a critic network Q_{θ_q} to approximate the soft state-action value function, seeking to estimate the optimal action-value function $Q^*(s, a)$ in the context of stochastic policies. The actor is instantiated as a stochastic policy π_{ϕ_a} that aims to maximize both the expected return and the entropy of the policy, thereby encouraging exploration of the action space. The shared encoder maps the high-dimensional observation space $\mathcal{O}$ into a lower-dimensional representations. To ensure the stability of the learning process, the critic and shared encoder have target networks start with the same parameters $\theta_{tgt} = \theta_q$. These target networks are updated via an exponential moving average (EMA), $\theta_{tgt} \leftarrow (1-\epsilon)\theta_{tgt} + \epsilon\theta_q$, where $\epsilon \in (0, 1)$. The EMA update serves to temper abrupt fluctuations in parameter values, thereby contributing to the stability of the training process.

3.3 Generalization in Visual RL

Our work focuses on the challenge of generalization in visual reinforcement learning, where the agent is trained on the environment without distractors (including augmentations) and then evaluated on previously unseen environment with distractors. The goal is to obtain consistent behavious under domain (environment distribution) shift. Formally, we consider a family of POMDPs, denoted by $\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, ..., \mathcal{M}_k\}$. Each POMDP $\mathcal{M}_i \sim \mathcal{M}$ shares the same underlying dynamics and reward structure but differs in its observation space $\mathcal{O}_i$, typically due to variations in visual appearances. Our objective is to learn a policy π that maximizes the expected cumulative return across POMDPs sampled from $\mathcal{M}$ in a zero-shot manner i.e. without additional training or fine-tuning on the test environments. The goal is to find a policy π that maximizes the expected discounted return: $\eta_{\mathcal{M}}(\pi) = \mathbb{E}_{(o_t, a_t) \sim (\mathcal{M}, \pi)} \left[\sum_{t=0}^{T-1} \gamma^t R(o_t, a_t) \right]$. We denote the training environment as $\mathcal{M}_{\text{train}}$ and the set of test environments as $\mathcal{M}_{\text{test}}$. The generalization performance of the policy can be quantified by the generalization gap (Kirk et al., 2023; Wang et al., 2024), defined as $\mathcal{L}_{\text{gen}} = \eta_{\mathcal{M}_{\text{test}}}(\pi) - \eta_{\mathcal{M}_{\text{train}}}(\pi)$.

4 Method

We propose Task-Relevant Segment Masking (TRMS), an algorithm designed to enhance generalization and robustness in noise-prone environments by masking out irrelevant segments. In this section, we provide a detailed overview of TRMS's architectural design, training procedure and its inclusion within the reinforcement learning framework.

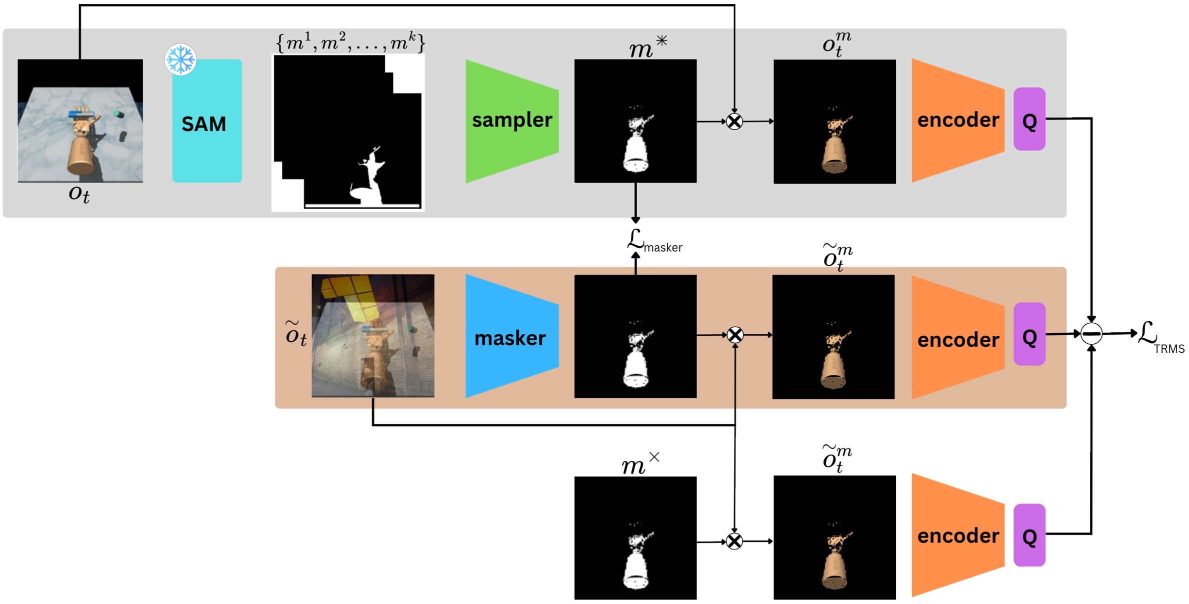


Figure 1: Depiction of the Task-Relevant Mask Sampling (TRMS) architecture, detailing its components and sequential training phases. Initially, k-masks are generated by a frozen segmentation model, followed by sampling and a logical OR operation to produce a single mask m^*. This mask is then applied to the non-augmented image via a Hadamard product $\otimes$, producing the masked image o_t^m, which progresses through the encoder and Q-network to yield the corresponding q-value. Simultaneously, m^* is utilized to mask the augmented image $\tilde{o}_t^m$ through the Hadamard operation. During the initial teaching phase, T_{Teach}, only the row in ⬜ is activated. Beyond this phase, the student masking network replaces the segmentation and sampler, and is trained utilizing the 🟫 row.

4.1 Masking Prior

TRMS explicitly leverages existing pre-trained segmentation models as a backbone to extract masks from non-augmented images. In this case, we employ the Fast Segment Anything Model (FastSAM) (Zhao et al., 2023)[1], known for its compact design that reduces memory usage and provides fast inference. As shown in Figure 1, this network remains frozen throughout the training phase, operating solely as an inference model.

Let o_t and o_{t+1} denote the observation at time step t and $t+1$ respectively. Following the approach in existing methods (Hansen & Wang, 2021), we apply an augmentation ψ by overlaying a random image onto these observations, resulting in augmented pairs $\{\tilde{o}_t, \tilde{o}_{t+1}\} = \{\psi(o_t), \psi(o_{t+1})\}$. Since these augmented images hinder accurate mask extraction from SAM, we rely on non-augmented images for precise and consistent masks (as shown in ⬜ in Fig 1). Provided an non-augmented image o, we utilize a frozen mask extractor (SAM), denoted by M, to obtain $k-$masks, $\{m^1, m^2, ..., m^k\}$.

4.2 Mask Sampling Network

These k masks are fed into a CNN-based mask selection module, denoted as $G_\beta(m)$, which generates the probability of selection for each mask $w(m^i)$. The goal of the mask selector is to identify task-relevant masks, specifically those with selection probabilities exceeding the threshold $1/k$ and to disregard all others. Formally, this selection is represented as follows:

$$p(m^i) = \begin{cases} 1, & \text{if } w(m^i) \geq 1/k \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

[1] As FastSAM is a variant of SAM, here we use FastSAM and SAM interchangably.

The objective of the mask selector is to encourage the probabilities associated with task-relevant masks to exceed $1/k$. Subsequently, we compute the Hadamard product across all selected masks, resulting in a final composite mask. This mask is then applied to both the original and augmented images, isolating the task-relevant regions of the observations,

$$m^* = \bigvee_{i=1}^{k} \left[p(m^i) \cdot m^i \right], \tag{2}$$

where $\bigvee$ denotes the Logical OR operation. Following this, the Hadamard product is calculated with the original and augmented observations, respectively, as follows,

$$o_t^m = m^* \otimes o_t, \tag{3}$$

$$\tilde{o}_t^m = m^* \otimes \tilde{o}_t, \tag{4}$$

where $\otimes$ represents the Hadamard product. o_t^m and $\tilde{o}_t^m$ are the images obtained by applying the same sampled masks on non-augmented and augmented observations. This process is applied to the observations at both time steps, t and $t+1$, ensuring that the task-relevant features are preserved across temporal frames for both the original and augmented observations.

Multiple masks may be relevant to the task; thus, the selector must be capable of selecting several masks simultaneously. Empirical observations suggest that using only a Softmax activation function at the output layer often results in higher selection probabilities for only a few masks (typically one or two). However, complex scenes frequently require the use of multiple masks. To address this limitation, we employ the Gumbel-Softmax distribution (Gumbel, 1954; Jang et al., 2017) at the output layer, which helps mitigate this selection bias. By adjusting the temperature parameter τ, we can encourage a more uniform probability distribution, making it easier for the selector by requiring only a slight increase in probability for the relevant masks to be chosen. The Gumbel-Softmax is defined as

$$y_i = \frac{\exp\left((\log(x_i) + g_i)/\tau\right)}{\sum_{j=1}^{k} \exp\left((\log(x_j) + g_j)/\tau\right)} \tag{5}$$

where g_i are i.i.d. samples from a Gumbel distribution, typically computed as $g_i = -\ln(-\ln(u_i))$ with $u_i \sim \text{Uniform}(0,1)$. This formulation enables more robust selection in complex scenes with multiple relevant areas by reducing the chances of a single mask's probability dominating the distribution. More details are provided in the Supplementary Material. Since we perform a thresholding operation in Eq. equation 1, we apply Straight-Through Estimators (STE) to address the backpropagation challenges associated with discrete operations (Bengio et al., 2013).

4.3 Q-relevant Mask Sampling

To effectively select task-relevant masks, we assume that the task is already well-performed in the original, non-augmented environment. If the algorithm cannot solve the task without augmentation, then solving it would be infeasible. Therefore, we consider the Q-values obtained from the non-augmented environment as expert Q-values. For each pair of masked augmented and non-augmented observations, denoted as o^m and $\tilde{o}^m$, we obtain their corresponding state representations s and $\tilde{s}$. The target q function can be defined as $q_{\text{tgt}} = r(s_t, a_t) + \gamma \max_{a_t'} Q_{\text{tgt}}(s_{t+1}, a_t')$. The Q-loss function for the non-augmented images is then given by

$$\mathcal{L}_Q(\theta_q, \beta) = \mathbb{E}_{(s_t, a_t, r_t, s_{t+1}) \sim \mathcal{B}} \left[\frac{1}{2} \left(q_{\text{tgt}} - Q_{\theta_q}(s_t, a_t) \right)^2 \right], \tag{6}$$

where $\mathcal{B}$ denotes the replay buffer. In a similar way, the q-target for the augmented images can be written as $q_{\text{tgt}} = r(\tilde{s}_t, a_t) + \gamma \max_{a_t'} Q_{\text{tgt}}(\tilde{s}_{t+1}, a_t')$ and the corresponding Q-loss is

$$\mathcal{L}_{\tilde{Q}}(\theta_q, \beta) = \mathbb{E}_{(s_t, a_t, r_t, s_{t+1}) \sim \mathcal{B}} \left[\frac{1}{2} \left(\tilde{q}_{\text{tgt}} - Q_{\theta_q}(\tilde{s}_t, a_t) \right)^2 \right]. \tag{7}$$

The Task-Relevant Mask Selection (TRMS) loss is defined as the average of these two losses:

$$\mathcal{L}_{\text{TRMS}}(\theta_q, \beta) = \frac{1}{2}\left(\mathcal{L}_Q(\theta_q, \beta) + \mathcal{L}_{\tilde{Q}}(\theta_q, \beta)\right). \tag{8}$$

This TRMS loss simultaneously optimizes the critic's accuracy over both augmented and non-augmented observations, thereby promoting robustness to input augmentations when identical masks are applied. If task-irrelevant segments are selected, the loss will increase. By discarding visually irrelevant segments, the mask selector reduces representational variance from distractor pixels and encourages the policy to focus on task-relevant features. Consequently, this allows networks to consistently observe the same pixel value across different augmentations, thus maintaining a coherent state representation regardless of the augmentation applied. The Total Loss of the overall architecture can be defined as

$$\mathcal{L}_{\text{Total}} = \mathcal{L}_{\text{TRMS}} + \mathcal{L}_{\text{masker}} \tag{9}$$

4.4 Student Masking Networks

Although it is technically feasible to run the algorithm for 500K steps using the FastSAM models and a mask sampler, this approach is highly computationally intensive due to the considerable time required by the masking models. These models, while accurate, are not optimized for speed and may impose significant delays. Our empirical results shows that the model utilizing only SAM can take upto 3 days on the *Walker Walk* task (Appendix D). Moreover, FTD (Chen et al., 2024), which also relies on SAM, can require up to 3-6 days to process a single seed, depending on both the task complexity and the number of masks to be generated. To mitigate this computational burden, we introduce a Student Masking Network, a CNN-based network that effectively mimics the behavior of the Prior Masking model but only for a defined initial period, denoted as T_{teach}. During this period, the student network learns directly from the teacher model, replicating its outputs. Details regarding this initial phase are elaborated upon in the Supplementary Material.

The training objective for this student network capitalizes on the binary nature of the mask outputs. We employ a Binary Cross-Entropy (BCE) loss to measure the discrepancy between the teacher network's output mask, m^* (as defined in Eq. equation 2), and the mask generated by the student network, $M^*_\alpha(o_t)$, which is parameterized by α. Formally, this is expressed as

$$\mathcal{L}_{\text{masker}} = \text{BCE}(m^*, M^*_\alpha(\tilde{o}_t)). \tag{10}$$

This BCE loss is then exclusively backpropagated through the student network, enabling it to gradually learn the teacher's masking strategy during the initial training period T_{teach}. After completing these T_{teach} steps, the teacher network is omitted, and only the student network is utilised to generate task-relevant masks, maintaining operational efficiency while significantly reducing computation time (Appendix Section D). An additional strategy to further optimize training time is implemented after T_{teach}, we increase the batch size. This adjustment expedites the learning process by enabling the student network to process more data per training iteration. Detailed training description is provided in the Supplementary Material. The details about the training of the entire architecture in Fig. 1 is provided in Algorithm 1[2]. We augmented the SAC algorithm with TRMS components, shown in blue.

5 Experiments

In this section, we present our experimental evaluations conducted on generalization benchmarks from RL-ViGen (Yuan et al., 2023). These benchmarks were selected as they encompass a wide range of environments: **(1)** DeepMind Control Generalization Benchmark (Hansen et al., 2021) for evaluating continuous control agents across tasks with complex dynamics and diverse rewards, essential for testing generalization capabilities; **(2)** Quadruped Locomotion (Hansen et al., 2021), which includes a Unitree quadruped robot, challenging agents with intricate balance and control requirements in dynamic environments; **(3)** Dexterous Manipulation (Rajeswaran et al., 2018), featuring multi-object interactions and sparse rewards that require

[2] Temperature update in SAC, double critics and target network updates are omitted for clarity.

Algorithm 1 Training Algorithm for SAC with TRMS

```
Require: $\mathrm{E}_{\text{step}}$, $\psi, k, \lambda_q, \lambda_\pi, \lambda_\alpha$          ▷ Variables Initialization
Require: $\phi_a, \theta_q$, M, $\mathrm{M}^*_\alpha$, $\mathrm{G}_\beta$          ▷ Networks Initialization
$D \leftarrow \emptyset$          ▷ Initialize replay buffer
for each initial collection step do
    $a_t \sim \pi_{\text{random}}(\cdot|o_t)$          ▷ Sample random action
    $o_{t+1}, r_{t+1} \sim \mathrm{E}_{\text{step}}(a_t)$          ▷ Apply action
    $D \leftarrow D \cup (o_{t+1}, a_t, r_{t+1})$          ▷ Append to buffer
end for
for every training step do
    $\{(o_t, a_t, r_t, o_{t+1})\}_{t=k}^{L+k} \sim D$          ▷ Sample minibatch
    $a_t \sim \pi_{\phi_a}(a_t|o_t)$          ▷ Sample action
    $o_{t+1}, r_t \sim \mathrm{E}_{\text{step}}(a_t)$
    $D \leftarrow D \cup (o_t, a_t, r_t, o_{t+1})$
    $\tilde{o}_t, \tilde{o}_{t+1} \leftarrow \psi(o_t), \psi(o_{t+1})$          ▷ Augmentation
    for each gradient step do
        if step $\leq T_{\text{teach}}$ then
            $\tilde{o}^m_t, \tilde{o}^m_{t+1} \leftarrow \tilde{o}_t \otimes G_\beta(M(o_t)), \tilde{o}_{t+1} \otimes G_\beta(M(o_{t+1}))$          ▷ SAM
        else
            $\tilde{o}^m_t, \tilde{o}^m_{t+1} \leftarrow \tilde{o}_t \otimes M^*_\alpha(\tilde{o}_t), \tilde{o}_{t+1} \otimes M^*_\alpha(\tilde{o}_{t+1})$          ▷ CNN Masking
        end if
        Update Encoder and Mask Sampler (Eq. equation 8)
        $\theta_q \leftarrow \theta_q - \lambda_q \nabla \mathcal{L}_{\text{TRMS}}(o_t, o_{t+1}, \tilde{o}^m_t, \tilde{o}^m_{t+1}; \theta_q, \beta)$
        $\phi_a \leftarrow \phi_a - \lambda_\pi \nabla \mathcal{L}_\pi(\phi_a)$          ▷ Update policy
        $\alpha \leftarrow \alpha - \lambda_\alpha \nabla \mathcal{L}_{\text{masker}}(o_t, \tilde{o}_t; \alpha)$          ▷ Update Masker
    end for
end for
```

precise control strategies to handle sophisticated manipulation tasks. We provide a comprehensive description of our experimental configurations and compare the performance of TRMS against relevant existing approaches. This analysis demonstrates the effectiveness of TRMS in enhancing generalization across diverse tasks in vision-based reinforcement learning. An evaluation of wall time comparing TRMS with the Only-SAM method is presented in Appendix Section D. The analysis demonstrates that incorporating the student network substantially enhances computational efficiency compared to relying solely on a heavily parameterised segmentation model.

Baselines. Our method is compared against several prominent Visual RL algorithms that are specifically designed for generalization. DrQ (Yarats et al., 2021) improves SAC (Haarnoja et al., 2018) by augmenting visual inputs while updating the TD loss. DrQ-v2 (Yarats et al., 2022), a DDPG (Lillicrap et al., 2016) and DrQ-based model-free algorithm. CURL (Laskin et al., 2020a) enhances visual representations by using a contrastive learning approach similar to SimCLR (Chen et al., 2020) i.e. aligning augmented views of the same observation. SVEA (Hansen et al., 2021) stabilizes learning by using un-augmented images for the target Q-value, while applying augmentation to reduce Q-value variance. SRM (Huang et al., 2022) learns representations in frequency-domain and learns to discard certain frequency in the observation to address domain shifts. PIE-G (Yuan et al., 2022b) incorporates ResNet (He et al., 2016) pre-trained models to enhance generalization, while SGQN (Bertoin et al., 2022) uses saliency maps to focus on key pixels crucial for decision-making. VRL3 (Wang et al., 2022) is SOTA algorithm for Adroit tasks, utilizing human demonstrations.

Augmentation. All of our baselines, except for SAC, leverage specific data augmentation techniques during training. TRMS uses an augmentation strategy inspired by SVEA (Hansen et al., 2021), which has proven effective for handling distracting video backgrounds through overlay augmentation. In this approach, we overlay a randomly selected image n from the Places dataset (Zhou et al., 2018) onto our observation frame as $\tilde{o}_t = \delta \cdot o_t + (1 - \delta) \cdot n$, where δ is a weighting factor that controls the degree of image overlay. In all the experiments, we set $\delta = 0.5$.

Zero-shot Evaluation. To assess generalization, we perform zero-shot evaluations of the trained agents on a range of unseen environments with different distraction intensities. Specifically, we evaluate performance on the video-easy and video-hard configurations across all environments. Each seed undergoes evaluation over 100 episodes, corresponding to the designated noise levels.

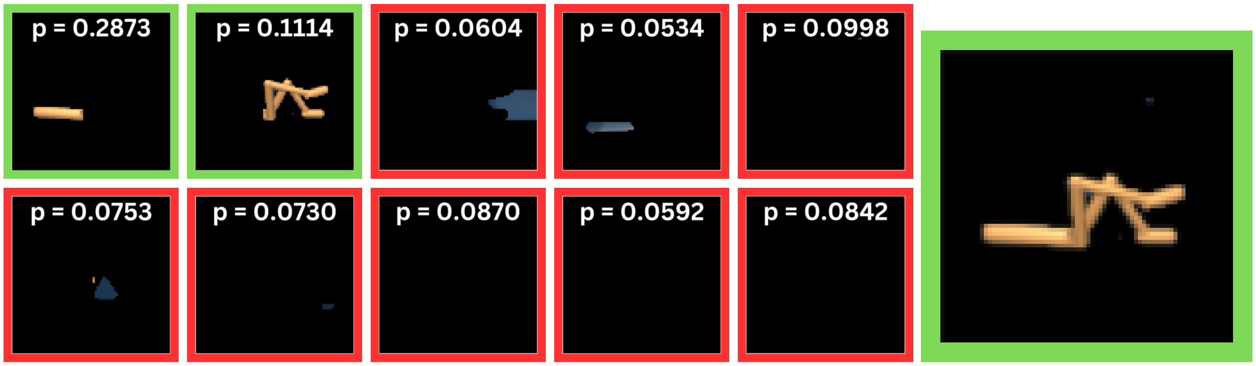
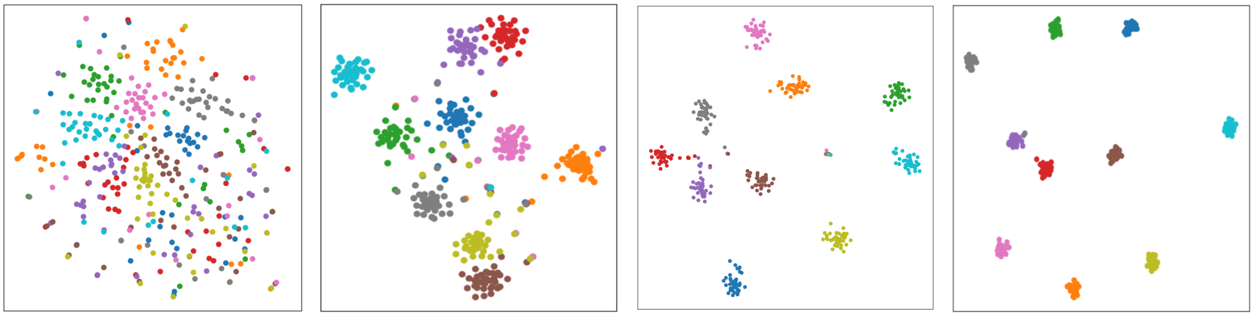


Figure 2: The figure on the left shows the probability of each mask out of the 10 masks provided. As there are 10 masks, only the masks above the probability of 0.1 (1/10) are selected, resulting into the image on the right.

5.1 Deepmind Control Suite

We evaluate our algorithm on the DMC-GB (Hansen & Wang, 2021) benchmark, spanning six tasks: Walker Walk, Walker Stand, Ball in Cup Catch, Finger Spin, Cartpole Swingup, and Cheetah Run. RL-ViGen integrates DMC-GB with two difficulty levels: video-easy (10 background videos) and video-hard (100 background videos without surface), where natural video backgrounds are used to rigorously test generalization under varying visual distractions.

Generalization Performance. We evaluate generalization by running five seeds per task and calculating the mean and standard deviation of returns. As shown in Table 1, TRMS surpasses all the baselines in



SRM: 535 PIEG: 641 SGQN: 655 TRMS (ours): 719

Figure 3: t-SNE visualization of clustering results for TRMS and three baselines. TRMS demonstrates a more distinct and well-separated clustering pattern, with each cluster representing identical agent poses with distinct backgrounds.

Table 1: Performance on DMC Benchmark Environment in Video-Hard (VH) and Video-Easy (VE) settings. S-up: Swingup.

Task (VE)	**SAC**	**DrQ**	**DrQ-v2**	**CURL**	**SVEA**	**SRM**	**PIEG**	**SGQN**	**TRMS**	**Δ**
Cartpole S-up	398±60	485±105	267±41	404±67	**782±27**	724±75	482±51	717±35	**787±35**	+5 (0.63%)
Walker Walk	245±165	682±89	175±117	556±133	819±81	854±42	**871±22**	860±53	863±74	-8 (0.91%)
Walker Stand	389±131	873±83	560±48	852±75	961±8	**966±42**	957±12	955±9	**967±5**	+1 (0.10%)
Ball in Cup	192±157	318±157	871±106	316±119	871±106	924±35	910±37	761±171	**938±10**	+14 (1.15%)
Finger Spin	206±169	533±119	456±15	502±19	808±33	853±76	837±107	609±61	**868±24**	+31 (3.70%)
Cheetah Run	87±21	102±30	64±22	104±24	249±20	257±21	**287±20**	269±33	207±83	-80 (27.87%)
Average	253	499	457	456	757	763	724	697	**772**	+9 (1.18%)

Task (VH)	**SAC**	**DrQ**	**DrQ-v2**	**CURL**	**SVEA**	**SRM**	**PIEG**	**SGQN**	**TRMS**	**Δ**
Cartpole S-up	158±17	138±9	130±3	114±15	393±45	475±75	323±24	488±18	**514±102**	+26 (5.33%)
Walker Walk	122±47	104±22	34±11	58±18	377±93	535±35	641±63	655±45	**747±63**	+92 (14.05%)
Walker Stand	231±57	289±49	151±13	45±5	834±46	863±57	852±56	851±24	**906±20**	+43 (4.98%)
Ball in Cup	101±37	100±40	97±27	115±33	403±174	566±135	773±74	782±57	**837±20**	+55 (7.05%)
Finger Spin	13±10	91±13	21±4	27±21	335±58	419±32	762±59	554±8	**791±54**	+29 (5.19%)
Cheetah Run	10±5	32±13	23±5	21±7	105±37	115±24	154±17	144±34	**189±86**	+35 (22.72%)
Average	106	126	76	63	408	496	584	579	**664**	+80 (13.70%)

10 out of 12 environments. Notably, TRMS demonstrates an advantage in the video-hard setting, the most challenging environment due to its complex video perturbations, where it outperforms all the selected relevant baselines. In the video-easy setting, TRMS exhibits an improvement of 1.18% over the second-best performing method, SRM, and 5% over the average of the next four best methods (SVEA,SRM,PIEG and SGQN). Interestingly, in the video-hard settings, TRMS not only outperforms all baselines across every environment but also achieves an impressive average improvement of 13.70% over the second-best method, SGQN, as shown by the Δ in Table 1. Collectively, these results underscore TRMS's robust generalization capabilities, particularly under high-noise and complex video conditions, establishing it as a reliable solution across diverse test environments.

Mask Sampling Probabilities Visualization. The mask selection probability from the masker is illustrated in Figure 2. Given that there are 10 masks, the selector's objective is to increase the probability of task-relevant segments above the threshold of 1/10 (i.e., 0.1) while reducing the probability of irrelevant segments below this threshold. this case, only the first two segments are selected (i.e., only their masks, without the overlayed image). These selected segments are then subjected to the Logical OR operation and the Hadamard product, as depicted in Eq. equation 2 and Eq. equation 3 respectively. The image on the right of the figure represents the resulting output, clearly demonstrating the selective focus on relevant regions.

Representations under Distractors. To demonstrate that TRMS' capability of learning domain-invariant representations, we employ t-SNE (van der Maaten & Hinton, 2008) to visualize the features extracted by the encoder. We select 10 distinct observations from different states and replace their backgrounds with 40 unseen images. These images are then encoded using the learned representations from each algorithm. Observations corresponding to the same state are marked with identical colors. As illustrated in Figure 3, representations of images with varying backgrounds (distractors) but identical agent poses are embedded into tight clusters, indicating that the learned embeddings capture pose-invariant features. Notably, TRMS exhibits the most compact clustering and achieves the highest rewards under distracting backgrounds, suggesting a stronger ability to learn domain-invariant representations.

Robustness to Distraction Levels. Figure 4 illustrates the absolute difference in average rewards as the distraction level increases from video-easy to video-hard. The figure reveals that all methods experience considerable performance drops under higher distraction levels. Notably, TRMS exhibits the smallest fluctuation, closest to SGQN, while achieving considerable higher rewards than it, underscoring its resilience to elevated noise levels.

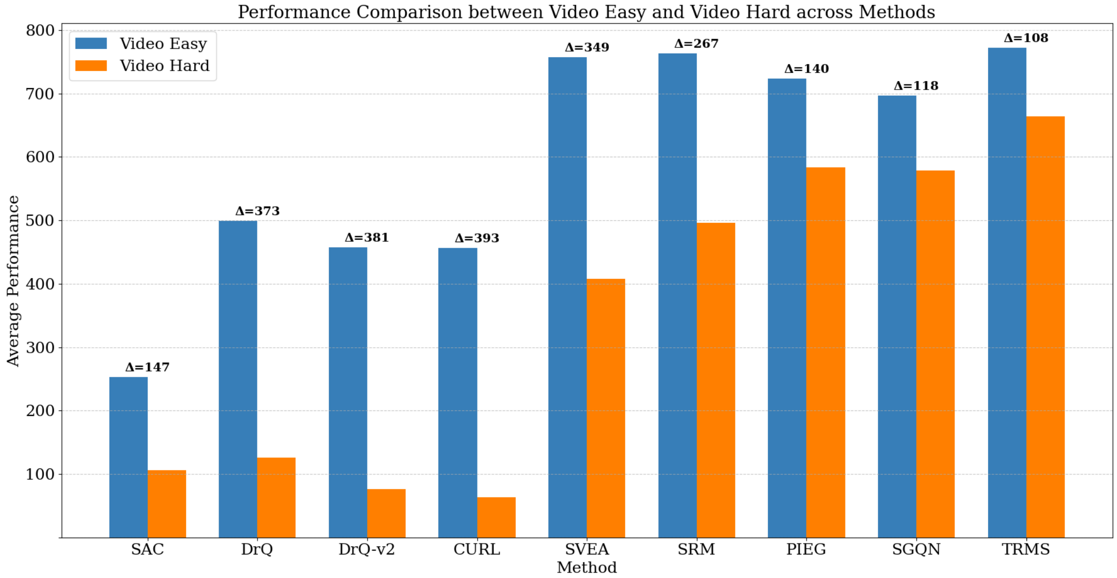


Figure 4: For each method, paired bars display the average performance under Video Easy and Video Hard conditions, and the annotated Δ indicates the absolute difference between the two settings. Standard deviations are omitted, as the underlying tasks differ significantly, making such variance comparisons uninformative.

5.2 Locomotion

For Locomotion tasks, we utilise Unitree Series tasks (Hansen et al., 2021): Unitree Stand and Unitree Walk. The training and evaluation are performed in a similar way as described in Section 5.1 for DMC settings.

Generalization Performance. We evaluate generalization by running three seeds per task and computing the mean and standard deviation of returns. As shown in Table 2, TRMS outperforms the baselines across 3 out of 4 environments. In the video-easy setting, TRMS demonstrates a substantial 33.5% improvement in the Unitree Walk task, though it lags in Unitree Stand, resulting into lower average performance as compared to the baselines. However, as distractions intensify in the video-hard setting, TRMS consistently outperforms all baselines, showcasing remarkable resilience to distractor noise. Overall, TRMS achieves a 77.46% increase in performance in video-hard setting relative to the second-best method.

Table 2: Performance of various methods on Unitree Walk and Unitree Stand tasks for Video-Easy (VE) and Video-Hard (VH)

Task (VE)	**DrQ**	**DrQ-v2**	**CURL**	**SVEA**	**SRM**	**PIEG**	**SGQN**	**TRMS (ours)**
Walk	67.4±9.2	97.8±15.7	74.8±14.2	98.4±28.3	98.0±9.4	140.2±63.9	151.7±87.1	**202.6±47.2** (+33.55%)
Stand	341.4±19.8	374.8±64.7	431.4±38.3	**587.0±39.6**	553.2±27.9	379.6±65.8	447.0±50.3	315.4±105.8 (-46.26%)
Average	204.4	236.3	253.1	342.7	325.6	259.9	332.0	**259.0** (-24.42%)
Task (VH)	**DrQ**	**DrQ-v2**	**CURL**	**SVEA**	**SRM**	**PIEG**	**SGQN**	**TRMS (ours)**
Walk	39.6±22.3	83.0±24.2	61.2±25.9	73.8±52.2	72.4±29.0	203.7±75.6	122.8±68.2	**214.4±31.5** (+5.25%)
Stand	65.6±25.7	95.8±37.4	99.4±25.3	279.3±10.7	**300.0±34.5**	202.0±43.1	139.8±47.0	**305.4±101.7** (+1.80%)
Average	52.6	89.4	80.3	176.6	186.2	202.8	131.3	**259.9** (+77.46%)

5.3 Dexterous Manipulation

Adroit (Rajeswaran et al., 2018) is an environment specifically designed for complex dexterous hand manipulation tasks, requiring substantial exploration and detailed feature extraction due to its sparse reward structure and the intricacy of its high-dimensional action space. In our experiments, we consider three of its tasks from a single view in RL-ViGen: Door, Hammer and Pen. This environment involves a magnitude of objects that needs to be masked, which makes the task extremely difficult.

TRMS achieved the highest average performance across tasks with scores of 59.5 in the video-easy setting and 54.9 in the video-hard setting. GQN matched TRMS at 59.5 in video-easy but scored lower in video-hard (30.3) due to heavy distractions. Notably, SGQN excelled on specific tasks like Door and Hammer, where TRMS's scores were comparatively lower, suggesting that while TRMS provides robust overall performance. However, there is a room for improvement in these environments. See the future directions below.

Table 3: Performance comparison of various methods on Adroit tasks with Video Easy (VE) and Video Hard (VH) background.

Task (VE)	**VRL3**	**SVEA**	**SGQN**	**PIE-G**	**TRMS**
Pen	1.7±0.6	46.7±3.8	64.0±9.0	53.6±4.7	**72.4±9.1** (+13.10%)
Door	0.0±0.0	44.8±8.5	**58.2±12.3**	56.6±11.1	50.7±12.3 (-12.90%)
Hammer	0.0±0.0	8.4±8.6	**56.3±6.3**	44.3±13.0	55.3±5.3 (-1.80%)
Average	0.6	33.3	**59.5**	51.5	**59.5**

Task (VH)	**VRL3**	**SVEA**	**SGQN**	**PIE-G**	**TRMS**
Pen	2.7±1.5	41.7±6.1	56.0±2.4	54.0±9.4	**67.3±5.2** (+20.20%)
Door	0.0±0.0	7.6±1.8	20.3±6.1	**52.6±3.3**	43.7±5.4 (-17.0%)
Hammer	0.0±0.0	4.2±3.7	14.6±4.7	46.0±4.6	**53.7±7.6** (+16.70%)
Average	0.9	17.8	30.3	50.8	**54.9** (+8.10%)

6 Conclusion

To address the challenge of task-irrelevant distracting visual features in Visual Reinforcement Learning, we introduce TRMS, a method that utilizes existing masking strategies to extract masks from the visual scene. It segments and samples only the task-relevant masks. This approach eliminates the need for additional segmentation labels for individual tasks. We bypass the heavy computation time and resources by employing student network that learns these masks in few training steps. We evaluate TRMS on the RL-ViGen (Yuan et al., 2023) benchmark, covering tasks from the DeepMind Control Suite, Unitree locomotion, and Dexterous manipulation under varied distractions. TRMS achieves a 13.70% higher average reward in video-hard DeepMind tasks and surpasses baselines in 10 out of 12 tasks. It also yields a 77.46% improvement in locomotion tasks and comparable performance in dexterous manipulation, all while demonstrating robust resilience to increasing noise levels compared to baselines.

Limitations and Future Directions. Currently, the prior masking approach, FastSAM, generates masks independently for each state. Incorporating recent advancements like SAM2 (Ravi et al., 2024), which leverages temporal dependencies to refine mask extraction, could greatly enhance sampling efficiency. We need to select $k-$masks and initial teaching steps T_{Teach} for distinct environments, depending on the complexity of the environment. An extension for automated selection of this hyperparameter would be extremely useful. Utilizing segmentation models with human-in-the-loop guidance via natural language descriptions (Zhang et al., 2024) can significantly enhance masking performance, especially in complex environments (Rajeswaran et al., 2018). Extending TRMS to non-agent-centric environments, such as CARLA, presents unique challenges, as masking-based methods often face limitations in these domains, warranting further investigation. Additionally, future research will explore the applicability of TRMS in more complex, real-world robotic applications, an area that remains largely underexplored in this domain.

References

Rishabh Agarwal, Marlos C. Machado, Pablo Samuel Castro, and Marc G Bellemare. Contrastive behavioral similarity embeddings for generalization in reinforcement learning. In *International Conference on Learning*

Representations, Virtual, 2021. PMLR. URL https://openreview.net/forum?id=qda7-sVg84.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation, 2013.

David Bertoin, Adil Zouitine, Mehdi Zouitine, and Emmanuel Rachelson. Look where you look! saliency-guided q-networks for generalization in visual reinforcement learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 30693–30706, New Orleans, Louisiana, 2022. Curran Associates, Inc. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/c5ee2a08fbe743b171b0b4b2bdfd6f86-Paper-Conference.pdf.

Timothy F Brady, Talia Konkle, and George A Alvarez. A review of visual memory capacity: Beyond individual items and toward structured representations. *Journal of vision*, 11(5):4–4, 2011.

Chao Chen, Jiacheng Xu, Weijian Liao, Hao Ding, Zongzhang Zhang, Yang Yu, and Rui Zhao. Focus-then-decide: Segmentation-assisted reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(10):11240–11248, Mar. 2024. doi: 10.1609/aaai.v38i10.29002. URL https://ojs.aaai.org/index.php/AAAI/article/view/29002.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1597–1607, Vienna, Austria, 13–18 Jul 2020. PMLR. URL https://proceedings.mlr.press/v119/chen20j.html.

Karl Cobbe, Oleg Klimov, Chris Hesse, Taehoon Kim, and John Schulman. Quantifying generalization in reinforcement learning. In *International conference on machine learning*, pp. 1282–1289, California, 2019. PMLR.

Vedant Dave and Elmar Rueckert. Information-theoretic world model learning for denoised predictions, 2024. URL https://openreview.net/forum?id=MdHDUsP2lt.

Jiameng Fan and Wenchao Li. DRIBO: Robust deep reinforcement learning via multi-view information bottleneck. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 6074–6102, Baltimore, USA, 17–23 Jul 2022. PMLR. URL https://proceedings.mlr.press/v162/fan22b.html.

Linxi Fan, Guanzhi Wang, De-An Huang, Zhiding Yu, Li Fei-Fei, Yuke Zhu, and Animashree Anandkumar. Secant: Self-expert cloning for zero-shot generalization of visual policies. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 3088–3099, Virtual, 18–24 Jul 2021. PMLR. URL https://proceedings.mlr.press/v139/fan21c.html.

Norm Ferns, Prakash Panangaden, and Doina Precup. Bisimulation metrics for continuous markov decision processes. *SIAM Journal on Computing*, 40(6):1662–1714, 2011. doi: 10.1137/10080484X. URL https://doi.org/10.1137/10080484X.

Zaifeng Gao, Shixian Yu, Chengfeng Zhu, Rende Shui, Xuchu Weng, Peng Li, and Mowei Shen. Object-based encoding in visual working memory: Evidence from memory-driven attentional capture. *Scientific Reports*, 6(1):22822, 2016.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent - a new approach to self-supervised learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 21271–21284, Virtual,

2020. Curran Associates, Inc. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/f3ada80d5c4ee70142b17b8192b2958e-Paper.pdf.

Bram Grooten, Tristan Tomilin, Gautham Vasan, Matthew E. Taylor, A. Rupam Mahmood, Meng Fang, Mykola Pechenizkiy, and Decebal Constantin Mocanu. Madi: Learning to mask distractions for generalization in visual deep reinforcement learning. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '24, pp. 733–742, Richland, SC, 2024. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9798400704864.

Emil Julius Gumbel. *Statistical theory of extreme values and some practical applications: a series of lectures*, volume 33. US Government Printing Office, USA, 1954.

Kashish Gupta, Debasmita Mukherjee, and Homayoun Najjaran. Extending the capabilities of reinforcement learning through curriculum: A review of methods and applications. *SN Computer Science*, 3:1–18, 2022.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, Montréal, Canada, 2018. Curran Associates, Inc. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/2de5d16682c3c35007e4e92982f1a2ba-Paper.pdf.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870, Stockholm, 2018. PMLR.

Danijar Hafner, Timothy P. Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, Addis Ababa, Ethiopia, 2020. OpenReview.net. URL https://openreview.net/forum?id=S1lOTC4tDS.

Danijar Hafner, Timothy P Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *International Conference on Learning Representations*, Virtual, 2021. JMLR. URL https://openreview.net/forum?id=0oabwyZbOu.

Nicklas Hansen and Xiaolong Wang. Generalization in reinforcement learning by soft data augmentation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 13611–13617, Xi'an, China, 2021. IEEE Press. doi: 10.1109/ICRA48506.2021.9561103. URL https://doi.org/10.1109/ICRA48506.2021.9561103.

Nicklas Hansen, Hao Su, and Xiaolong Wang. Stabilizing deep q-learning with convnets and vision transformers under data augmentation. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 3680–3693, virtual, 2021. Curran Associates, Inc. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/1e0f65eb20acbfb27ee05ddc000b50ec-Paper.pdf.

Nicklas Hansen, Hao Su, and Xiaolong Wang. Td-mpc2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations (ICLR)*, Vienna,Austria, 2024. PMLR.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Nevada, United States, June 2016. IEEE.

Yangru Huang, Peixi Peng, Yifan Zhao, Guangyao Chen, and Yonghong Tian. Spectrum random masking for generalization in image-based reinforcement learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 20393–20406, New Orleans, USA, 2022. Curran Associates, Inc. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/802a4350ca4fced76b13b8b320af1543-Paper-Conference.pdf.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, Toulon, France, 2017. JMLR. URL https://openreview.net/forum?id=rkE3y85ee.

Yiding Jiang, J. Zico Kolter, and Roberta Raileanu. On the importance of exploration for generalization in reinforcement learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 12951–12986, New Orleans, USA, 2023. Curran Associates, Inc. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/2a4310c4fd24bd336aa2f64f93cb5d39-Paper-Conference.pdf.

Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1):99–134, 1998. ISSN 0004-3702. doi: https://doi.org/10.1016/S0004-3702(98)00023-X. URL https://www.sciencedirect.com/science/article/pii/S000437029800023X.

Daniel Kaiser, Nikolaas N. Oosterhof, and Marius V. Peelen. The neural dynamics of attentional selection in natural scenes. *Journal of Neuroscience*, 36(41):10522–10528, 2016. ISSN 0270-6474. doi: 10.1523/JNEUROSCI.1385-16.2016. URL https://www.jneurosci.org/content/36/41/10522.

Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4015–4026, Paris, France, October 2023. IEEE.

Robert Kirk, Amy Zhang, Edward Grefenstette, and Tim Rocktäschel. A survey of zero-shot generalisation in deep reinforcement learning. *Journal of Artificial Intelligence Research*, 76:201–264, 2023.

Michael Laskin, Aravind Srinivas, and Pieter Abbeel. CURL: Contrastive unsupervised representations for reinforcement learning. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5639–5650, virtual, 13–18 Jul 2020a. PMLR. URL https://proceedings.mlr.press/v119/laskin20a.html.

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

Alex X. Lee, Anusha Nagabandi, Pieter Abbeel, and Sergey Levine. Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. In *Advances in Neural Information Processing Systems (NeurIPS)*, virtual, 2020a. PMLR.

Kimin Lee, Kibok Lee, Jinwoo Shin, and Honglak Lee. Network randomization: A simple technique for generalization in deep reinforcement learning. In *International Conference on Learning Representations*, virtual, 2020b. PMLR. URL https://openreview.net/forum?id=HJgcvJBFvB.

Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(39):1–40, 2016.

Dongfen Li, Lichao Meng, Jingjing Li, Ke Lu, and Yang Yang. Domain adaptive state representation alignment for reinforcement learning. *Information Sciences*, 609:1353–1368, 2022.

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In Yoshua Bengio and Yann LeCun (eds.), *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, pp. 14, San Juan, Puerto Rico, 2016. PMLR. URL http://arxiv.org/abs/1509.02971.

Minsong Liu, Luntong Li, Shuai Hao, Yuanheng Zhu, and Dongbin Zhao. Soft contrastive learning with q-irrelevance abstraction for reinforcement learning. *IEEE Transactions on Cognitive and Developmental Systems*, 15(3):1463–1473, 2023a. doi: 10.1109/TCDS.2022.3218940.

Qiyuan Liu, Qi Zhou, Rui Yang, and Jie Wang. Robust representation learning by clustering with bisimulation metrics for visual reinforcement learning with distractions. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23, Washington, USA, 2023b. AAAI Press. ISBN 978-1-57735-880-0. doi: 10.1609/aaai.v37i7.26063. URL https://doi.org/10.1609/aaai.v37i7.26063.

Siao Liu, Zhaoyu Chen, Yang Liu, Yuzheng Wang, Dingkang Yang, Zhile Zhao, Ziqing Zhou, Xie Yi, Wei Li, Wenqiang Zhang, and Zhongxue Gan. Improving generalization in visual reinforcement learning via conflict-aware gradient agreement augmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 23436–23446, Vancouver, Canada, October 2023c. IEEE.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Bhairav Mehta, Manfred Diaz, Florian Golemo, Christopher J. Pal, and Liam Paull. Active domain randomization. In Leslie Pack Kaelbling, Danica Kragic, and Komei Sugiura (eds.), *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pp. 1162–1176, virtual, 30 Oct–01 Nov 2020. PMLR. URL https://proceedings.mlr.press/v100/mehta20a.html.

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.

Alhassan Mumuni and Fuseini Mumuni. Data augmentation: A comprehensive survey of modern approaches. *Array*, 16:100258, 2022. ISSN 2590-0056. doi: https://doi.org/10.1016/j.array.2022.100258. URL https://www.sciencedirect.com/science/article/pii/S2590005622000911.

Sanmit Narvekar, Bei Peng, Matteo Leonetti, Jivko Sinapov, Matthew E Taylor, and Peter Stone. Curriculum learning for reinforcement learning domains: A framework and survey. *Journal of Machine Learning Research*, 21(181):1–50, 2020.

Shahin Nasr, Ali Moeeny, and Hossein Esteky. Neural correlate of filtering of irrelevant information from visual working memory. *PLoS One*, 3(9):e3282, 2008.

Benjamin Peters and Nikolaus Kriegeskorte. Capturing the objects of vision with neural networks. *Nature human behaviour*, 5(9):1127–1144, 2021.

Roberta Raileanu, Maxwell Goldstein, Denis Yarats, Ilya Kostrikov, and Rob Fergus. Automatic data augmentation for generalization in reinforcement learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 5402–5415, virtual, 2021. Curran Associates, Inc. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/2b38c2df6a49b97f706ec9148ce48d86-Paper.pdf.

Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. In *Proceedings of Robotics: Science and Systems*, Pittsburgh, Pennsylvania, June 2018. Robotics: Science and Systems. doi: 10.15607/RSS.2018.XIV.049.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos, 2024. URL https://arxiv.org/abs/2408.00714.

Pieter R Roelfsema and Arjen van Ooyen. Attention-gated reinforcement learning of internal representations for classification. *Neural computation*, 17(10):2176–2214, 2005.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi (eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pp. 234–241, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24574-4.

Brian J Scholl. Objects and attention: The state of the art. *Cognition*, 80(1-2):1–46, 2001.

Katharina N. Seidl, Marius V. Peelen, and Sabine Kastner. Neural evidence for distracter suppression during visual search in real-world scenes. *Journal of Neuroscience*, 32(34):11812–11819, 2012a. doi: 10.1523/JNEUROSCI.1693-12.2012. URL `https://www.jneurosci.org/content/32/34/11812`.

Katharina N Seidl, Marius V Peelen, and Sabine Kastner. Neural evidence for distracter suppression during visual search in real-world scenes. *Journal of Neuroscience*, 32(34):11812–11819, 2012b.

Ruixiang Sun, Hongyu Zang, Xin Li, and Riashat Islam. Learning latent dynamic robust representations for world models. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 47234–47260, Vienna, 21–27 Jul 2024. PMLR. URL `https://proceedings.mlr.press/v235/sun24n.html`.

Yanchao Sun, Ruijie Zheng, Xiyao Wang, Andrew E Cohen, and Furong Huang. Transfer RL across observation feature spaces via model-based regularization. In *International Conference on Learning Representations*, pp. 23, virtual, 2022. PMLR. URL `https://openreview.net/forum?id=7KdAoOsI81C`.

Richard S Sutton. Reinforcement learning: An introduction. *A Bradford Book*, 2(2):548, 2018.

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 *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, number 1 in IEEE, pp. 23–30, Vancouver, British Columbia, Canada, 2017. IEEE. doi: 10.1109/IROS.2017.8202133.

Manan Tomar, Riashat Islam, Matthew Taylor , Sergey Levine, and Philip Bachman. Ignorance is bliss: Robust control via information gating. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 38624–38641, New Orleans , USA, 2023. Curran Associates, Inc. URL `https://proceedings.neurips.cc/paper_files/paper/2023/file/797be96e4481c3fe5d675c1ba5352969-Paper-Conference.pdf`.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL `http://jmlr.org/papers/v9/vandermaaten08a.html`.

Che Wang, Xufang Luo, Keith Ross, and Dongsheng Li. Vrl3: A data-driven framework for visual deep reinforcement learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 32974–32988, New Orleans, USA, 2022. Curran Associates, Inc. URL `https://proceedings.neurips.cc/paper_files/paper/2022/file/d4cc7a2d0d70736e29a3b48c3729bc06-Paper-Conference.pdf`.

Shuo Wang, Zhihao Wu, Jinwen Wang, Xiaobo Hu, Youfang Lin, and Kai Lv. How to learn domain-invariant representations for visual reinforcement learning: An information-theoretical perspective. In Kate Larson (ed.), *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pp. 1389–1397, Jeju, Korea, 8 2024. International Joint Conferences on Artificial Intelligence Organization. doi: 10.24963/ijcai.2024/154. URL `https://doi.org/10.24963/ijcai.2024/154`. Main Track.

Weiyao Wang and Gregory D. Hager. Domain adaptation of visual policies with a single demonstration. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 17208–17215, Yokohama, Tokyo, 2024. IEEE. doi: 10.1109/ICRA57147.2024.10610569.

Xudong Wang, Long Lian, and Stella X. Yu. Unsupervised visual attention and invariance for reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6677–6687, Virtual, June 2021. IEEE.

Ziyu Wang, Yanjie Ze, Yifei Sun, Zhecheng Yuan, and Huazhe Xu. Generalizable visual reinforcement learning with segment anything model, 2023.

Eliot Xing, Abhinav Gupta, Sam Powers, and Victoria Dean. Kitchenshift: Evaluating zero-shot generalization of imitation-based policy learning under domain shifts. In *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*, Virtual, 2021a. NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications. URL `https://openreview.net/forum?id=DdglKo8hBq0`.

Jinwei Xing, Takashi Nagata, Kexin Chen, Xinyun Zou, Emre Neftci, and Jeffrey L Krichmar. Domain adaptation in reinforcement learning via latent unified state representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 10452–10459, virtual, 2021b. AAAI Press.

Rui Yang, Jie Wang, Zijie Geng, Mingxuan Ye, Shuiwang Ji, Bin Li, and Feng Wu. Learning task-relevant representations for generalization via characteristic functions of reward sequence distributions. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, pp. 2242–2252, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393850. doi: 10.1145/3534678.3539391. URL `https://doi.org/10.1145/3534678.3539391`.

Sizhe Yang, Yanjie Ze, and Huazhe Xu. Movie: Visual model-based policy adaptation for view generalization. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 21507–21523, New Orleans, USA, 2023. Curran Associates, Inc. URL `https://proceedings.neurips.cc/paper_files/paper/2023/file/43b77cef2a83a25aa27d3271d209e4fd-Paper-Conference.pdf`.

Denis Yarats, Ilya Kostrikov, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. In *International Conference on Learning Representations*, virtual, 2021. PMLR. URL `https://openreview.net/forum?id=GY6-6sTvGaf`.

Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning. In *International Conference on Learning Representations*, virtual, 2022. pmlr. URL `https://openreview.net/forum?id=_SJ-_yyes8`.

Bang You, Jingming Xie, Youping Chen, Jan Peters, and Oleg Arenz. Self-supervised sequential information bottleneck for robust exploration in deep reinforcement learning. *arXiv preprint arXiv:2209.05333*, 1(1): 1, 2022.

Tao Yu, Zhizheng Zhang, Cuiling Lan, Yan Lu, and Zhibo Chen. Mask-based latent reconstruction for reinforcement learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 25117–25131, New Orleans, USA, 2022. Curran Associates, Inc. URL `https://proceedings.neurips.cc/paper_files/paper/2022/file/a0709efe5139939ab69902884ecad9c1-Paper-Conference.pdf`.

Zhecheng Yuan, Guozheng Ma, Yao Mu, Bo Xia, Bo Yuan, Xueqian Wang, Ping Luo, and Huazhe Xu. Don't touch what matters: Task-aware lipschitz data augmentation for visual reinforcement learning. In Lud De Raedt (ed.), *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pp. 3702–3708, Vienna, 7 2022a. International Joint Conferences on Artificial Intelligence Organization. doi: 10.24963/ijcai.2022/514. URL `https://doi.org/10.24963/ijcai.2022/514`. Main Track.

Zhecheng Yuan, Zhengrong Xue, Bo Yuan, Xueqian Wang, YI WU, Yang Gao, and Huazhe Xu. Pre-trained image encoder for generalizable visual reinforcement learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 13022–13037, New Orleans, Louisiana, 2022b. Curran Associates, Inc. URL `https://proceedings.neurips.cc/paper_files/paper/2022/file/548a482d4496ce109cddfbeae5defa7d-Paper-Conference.pdf`.

Zhecheng Yuan, Sizhe Yang, Pu Hua, Can Chang, Kaizhe Hu, and Huazhe Xu. Rl-vigen: A reinforcement learning benchmark for visual generalization. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 6720–6747, New Orleans, USA, 2023. Curran Associates, Inc. URL `https://proceedings.neurips.cc/paper_files/paper/2023/file/15c9f64ec172b046470d2a4d2b7669fc-Paper-Datasets_and_Benchmarks.pdf`.

Hongyu Zang, Xin Li, and Mingzhong Wang. Simsr: Simple distance-based state representations for deep reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(8):8997–9005, Jun. 2022. doi: 10.1609/aaai.v36i8.20883. URL `https://ojs.aaai.org/index.php/AAAI/article/view/20883`.

Amy Zhang, Nicolas Ballas, and Joelle Pineau. A dissection of overfitting and generalization in continuous reinforcement learning. *arXiv preprint arXiv:1806.07937*, 1(1), 2018.

Amy Zhang, Rowan Thomas McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. In *International Conference on Learning Representations*, pp. 17, Virtual, 2021. PMLR. URL `https://openreview.net/forum?id=-2FCwDKRREu`.

Yuxuan Zhang, Tianheng Cheng, Lianghui Zhu, Rui Hu, Lei Liu, Heng Liu, Longjin Ran, Xiaoxin Chen, Wenyu Liu, and Xinggang Wang. Evf-sam: Early vision-language fusion for text-prompted segment anything model. *arXiv preprint arXiv:2406.20076*, 2024.

Xu Zhao, Wenchao Ding, Yongqi An, Yinglong Du, Tao Yu, Min Li, Ming Tang, and Jinqiao Wang. Fast segment anything, 2023.

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6): 1452–1464, 2018. doi: 10.1109/TPAMI.2017.2723009.

A Implementation Details

We provide additional details on our implementation of TRMS. Table 4 summarizes the hyperparameters utilized in our study. For the DMC Generalization Benchmark (Hansen & Wang, 2021) and Locomotion Task (Hansen et al., 2021), we used the default hyperparameters specified in RL-ViGen (Yuan et al., 2023) for all the respective baselines.

Table 4: Hyperparameters for TRSM in DMC-GB.

Hyperparameter	**Values**
Input size	84×84
Optimizer	Adam (Kingma, 2014)
Learning Rate (Actor,Critic, Masker)	10^{-4}
Adam β_1, β_2 (All Networks)	0.9, 0.999
Discount (γ)	0.99
Frame Stack	3
Action Repeat	2
Initial Batch Size	8
Batch Size	256
Feature Dimension	256
Initial Sampling Steps	2000
Replay Buffer Size	150K
Environment Steps	500K
Gradient Steps Per Training Step	1
Target Update Interval	2
Target Smoothing Coefficient Critic	0.01
Target Smoothing Coefficient Encoder	0.05
Initial Temperature (α in SAC)	0.1
Temperature Learning Rate	10^{-4}
Mask Sampler	AdamW (Loshchilov & Hutter, 2019)
Learning Rate	10^{-5}
Number of Masks	4
Gumbel Softmax temperature (τ)	5.0
Segmentation Model	FastSAM (Zhao et al., 2023) (default)
Image Size	640
IoU Threshold	0.75
Confidence Threshold	0.40
Overlap Mask	False
Initial Teaching Steps (T_{teach})	25K (DMC-GB) and 50K (Otherwise)

The common Reinforcement Learning Hyperparameters were kept consistent with those used for TRMS, while method-specific parameters followed the configurations provided in their respective papers. Details on the specific hyperparameters are listed in Table 5.

Table 5: Hyperparameters for baselines in DMC-GB

Hyper-parameters	**Value**
Feature dim	DrQ-v2, CURL: 50; otherwise: 256
N-step return	DrQ: 1; otherwise: 3
Optimizer	Adam
Hidden dim	1024
Frame stack	3
SGQN Quantile Threshold	0.95 or 0.98
Critic Weight Decay	10^{-5}
SGQN Auxiliary Learning Rate	8e-5

For the Dexterous Manipulation tasks, we use the same hyperparameters utilised for the respective baselines as mentioned in their papers and for environments as mentioned in RL-ViGen (Yuan et al., 2023). They are described in the Table 6.

Table 6: Adroit Hyperparameters.

Hyper-parameter	**Task**	**Value**
Training Frames	Hammer	10^6
	Door	10^6
	Pen	2×10^6
Learning Rate	Hammer	10^{-4}
	Door	10^{-4}
	Pen	10^{-4}
$k-$Masks	Hammer	15
	Door	15
	Pen	10
SGQN Quantile	Hammer	0.9
	Door	0.9
	Pen	0.9
SGQN Critic Weight	Hammer	0.9
	Door	0.5
	Pen	0.9
SGQN Auxiliary Learning Rate	Hammer	8×10^{-5}
	Door	8×10^{-5}
	Pen	8×10^{-5}

B Soft Actor-Critic Algorithm

The Soft Actor-Critic (SAC) (Haarnoja et al., 2018) algorithm is an off-policy reinforcement learning method that maximizes cumulative rewards while promoting entropy to encourage exploration. The objective for SAC is given by

$$\pi^* = \arg\max_{\pi_{\phi_a}} \mathbb{E}_{(s_t,a_t)\sim\pi_{\phi_a}} \left[\sum_{t=0}^{T} r(s_t, a_t) + \alpha\mathcal{H}(\pi_{\phi_a}(\cdot|s_t)) \right] \tag{11}$$

where α is a temperature parameter that balances reward and entropy terms. SAC employs two critic networks, $Q_{\theta_{q_1}}$ and $Q_{\theta_{q_2}}$, trained to minimize the soft Bellman residual

$$L(\theta_{q_i}) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\left[\left(Q_{\theta_{q_i}}(s,a) - \left(r + \gamma\,\mathbb{E}_{a'\sim\pi_{\phi_a}}\left[\min_{j=1,2} Q_{\theta_{q_j}}(s',a') - \alpha\log\pi(a'|s')\right]\right)\right)^2\right] \tag{12}$$

The actor network π_{ϕ_a} is updated to maximize the expected Q-value regularized by an entropy term, defined as

$$\mathcal{L}_\pi(\phi_a) = \mathbb{E}_{s\sim D, a\sim\pi_{\phi_a}}\left[\alpha\log\pi_{\phi_a}(a|s) - Q_{\theta_q}(s,a)\right], \tag{13}$$

where α is the entropy temperature coefficient. The corresponding policy gradient is given by

$$\nabla\mathcal{L}_\pi(\phi_a) = \mathbb{E}_{s,a}\left[\nabla_{\phi_a}\log\pi_{\phi_a}(a|s)\left(\alpha - Q_{\theta_q}(s,a)\right)\right]. \tag{14}$$

To balance exploration and exploitation, SAC adapts α by minimizing the temperature objective:

$$\mathcal{L}(\alpha) = \mathbb{E}_{a\sim\pi_{\phi_a}}\left[-\alpha\log\pi_{\phi_a}(a|s) - \alpha\mathcal{H}\right], \tag{15}$$

where $\mathcal{H}$ denotes the target entropy, encouraging diverse action sampling in high-dimensional action spaces.

C Architecture Details

Encoder. The encoder network consists of two main components: a shared convolutional module and a subsequent linear projection. The shared convolutional module is an 11-layer network designed to process input observations composed of 3 stacked RGB frames, with dimensions $[9, 84, 84]$, ultimately generating spatial feature maps. The first layer employs a 3×3 convolutional kernel with a stride of 2 and 32 output channels, allowing for an early reduction in spatial resolution. The remaining layers are structured as sequential ReLU-convolution blocks, each composed of a ReLU activation followed by a 3×3 convolution with a stride of 1 and maintaining 32 channels across all layers. This uniform channel depth preserves consistency in feature representation throughout the network.

The final convolutional output is then flattened into a feature vector of size $32 \times 21 \times 21$. This vector is subsequently passed through a linear projection layer, which reduces the dimensionality to 512, thus producing a condensed latent representation suitable for further processing. This final representation serves as the input to the policy and value networks, allowing for efficient and effective state encoding. Furthermore, input frames are normalized by scaling to the range $[-0.5, 0.5]$

Actor. The actor network comprises a feature extractor and a policy head. The feature extractor maps the 512-dimensional input representation to a 256-dimensional latent space via a fully connected layer, followed by layer normalization and Tanh activations for normalized, non-linear transformations. The resulting features are then passed through two hidden layers with 1024 units and ReLU activations, capturing complex action-value mappings.

The final layer outputs action means, μ, which are scaled by a Tanh activation to enforce bounded action outputs. The standard deviation, σ, is constant and scaled by an input parameter, std. Together, μ and σ parameterize a Truncated Normal distribution for continuous action sampling.

Critic. The critic network comprises two parallel Q-networks, Q_1 and Q_2, which are employed to estimate state-action values and mitigate overestimation bias. Initially, observations are passed through a shared feature extraction module that projects the input representation of dimension 512 into a 256-dimensional feature vector. This transformation is accomplished through a fully connected layer, followed by layer normalization and Tanh activation, which ensures normalized outputs and reduces the likelihood of activation saturation.

The resulting 256-dimensional feature vector is concatenated with the action input, forming a joint representation that is processed by each Q-network independently. Both Q_1 and Q_2 are structured with two hidden layers, each containing 1024 units, and use ReLU activations to introduce non-linearity, enabling the networks to model complex value functions effectively. The final layer of each Q-network outputs a single scalar, representing the Q-value for the given state-action pair. By using two independent Q-networks, the critic can take the minimum of both Q-value estimates, which reduces overestimation—an issue commonly encountered in value-based reinforcement learning. This design contributes to the stability and robustness of the learned policy.

Mask Sampler. The Mask Sampler network is designed to process multiple input masks and output selection probabilities for each. A single input has the shape of $(k, 84, 84)$, where k is the number of masks. The network comprises three convolutional layers, each with 3×3 kernels and a padding of 1 to maintain spatial dimensions. The first layer maps the input channels, corresponding to the number of masks, to 32 feature maps, followed by batch normalization and ReLU activation. The feature depth is then sequentially increased to 64 and 128 channels by the subsequent convolutional layers, each followed by batch normalization and ReLU to enhance spatial feature extraction.

The output of the final convolutional layer undergoes global average pooling to reduce the spatial dimensions to 1×1, yielding a 128-dimensional vector. This vector is then passed through a fully connected layer with 64 units and a ReLU activation, followed by a final linear layer that outputs logits corresponding to the number of masks. A Gumbel-Softmax activation with subsequent temperature is applied to these logits.

Masker. The Masker network is a convolutional architecture designed to produce a single-channel mask from RGB inputs. The network consists of five convolutional layers, each with a 3×3 kernel and a padding of 1 to maintain spatial dimensions. The first two layers map the input image, with three color channels, to 64 feature channels through successive applications of convolution, ReLU activation, and Batch Normalization, promoting feature extraction while stabilizing training.

The third and fourth layers reduce the feature depth to 32 channels, employing similar ReLU and batch normalization operations to preserve spatial information while refining feature representations. The final convolutional layer outputs a single-channel feature map, which represents the generated mask.

The output feature map is then passed through a sigmoid activation function, designed to constrain output values between 0 and 1. All convolutional layers are initialized with Xavier uniform initialization, ensuring balanced weight distributions. This architecture allows Masker to effectively learn spatial patterns for producing accurate binary masks from RGB input data.

D Wall Time

As TRMS incorporates both SAM and a student network to enhance computational efficiency, here we compare the wall time required for convergence between TRMS and the SAM-only variant. Evaluation is performed across three seeds of the *Walker Walk* task, where the teacher network is employed for the first 25k steps.

As shown, TRMS achieves an average wall time of approximately **15 hours**, whereas the SAM-only variant reaches up to **3 days and 1 hour** (∼73 hours). This substantial reduction demonstrates that TRMS not only accelerates learning early on but also sustains faster convergence through its student network (due to its small architecture), which absorbs knowledge from the heavy parameterised teacher network.

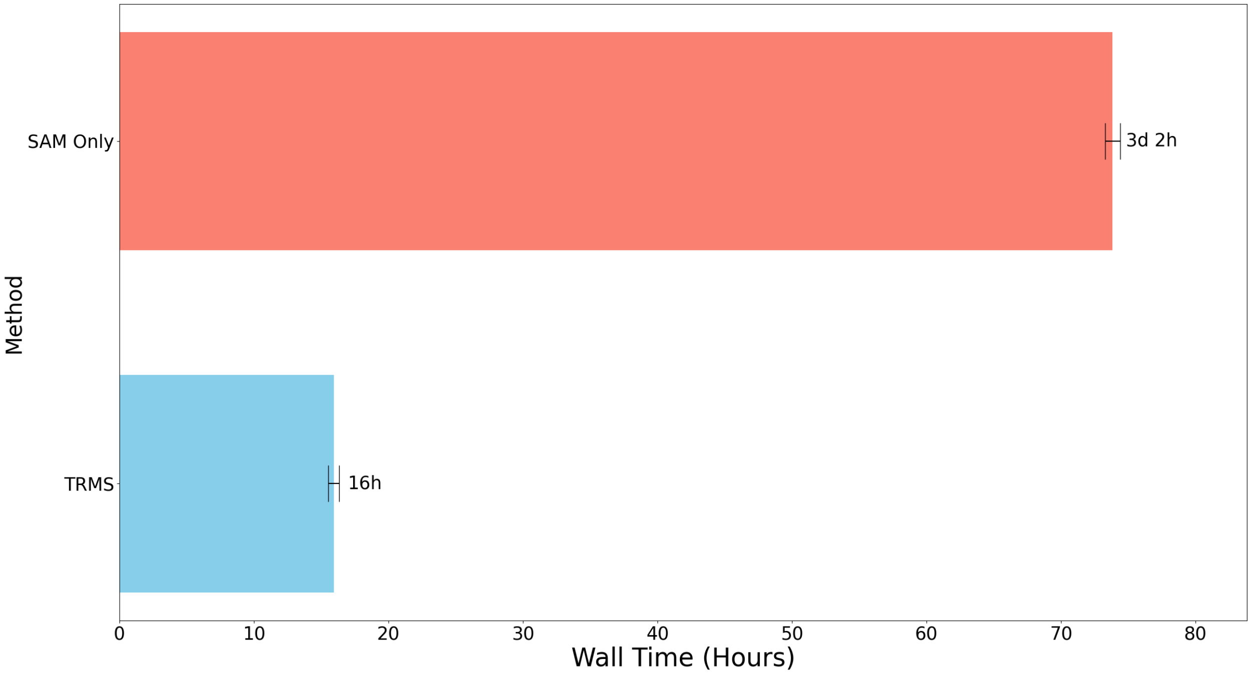
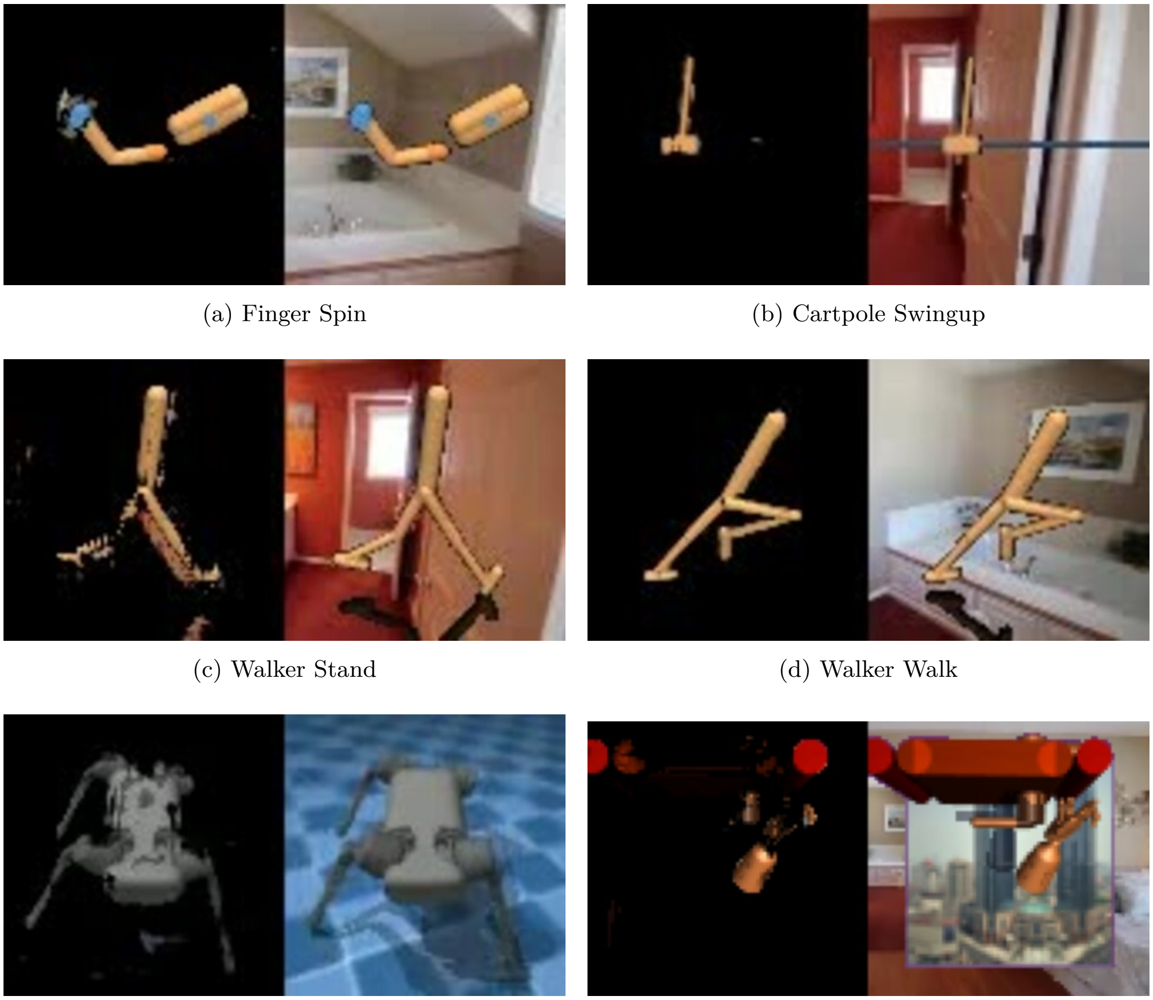


Figure 5: Comparison of TRMS and SAM-Only Wall time for Walker Walk on three seeds.

E Additional Visual Results

The following visual demonstrations showcase selected frames from the video-hard environment. On the left, each image displays the actual observation, while the corresponding masked observation is presented on the right. These comparisons highlight the complexity of the environment, where the masking process isolates relevant features, facilitating the agent's focus on relevant task elements amidst challenging distractors.



(a) Finger Spin

(b) Cartpole Swingup

(c) Walker Stand

(d) Walker Walk

(e) Walker Walk

(f) Door Open

Figure 6: Visual examples from the video-hard environment, showing actual observations (left) and corresponding masked observations (right), highlighting relevant feature isolation.

F Baseline Results:

Baseline results presented in this paper were obtained as follows. For the DMC-GB environments, we re-implemented most baselines ourselves, with the exceptions of DrQ, DrQ-v2, and SVEA, whose performance numbers were directly cited from the PIE-G paper (Yuan et al., 2022b). For the PIE-G baseline, we conducted experiments using the authors' original implementation. SGQN (Bertoin et al., 2022) results were reproduced using the official implementation provided with the RL-ViGen (Yuan et al., 2023) benchmark. For Locomotion and Manipulation environments, all baseline performances are directly cited from the RL-ViGen benchmark [3].

[3] https://github.com/gemcollector/RL-ViGen/tree/master/results