Focus On What Matters: Separated Models For Visual-Based RL Generalization

Di Zhang **Bowen Lv** **Hai Zhang** **Feifan Yang** **Junqiao Zhao** *
Hang Yu **Chang Huang** **Hongtu Zhou** **Chen Ye** **Changjun Jiang**
Department of Computer Science, Tongji University, Shanghai, China
MOE Key Lab of Embedded System and Service Computing, Tongji University, Shanghai, China
{2331922, 2151769, zhanghai12138, 2153299, zhaojunqiao}@tongji.edu.cn
{2053881, 2130790, zhouhongtu, yechen, cjjiang}@tongji.edu.cn

Abstract

A primary challenge for visual-based Reinforcement Learning (RL) is to generalize effectively across unseen environments. Although previous studies have explored different auxiliary tasks to enhance generalization, few adopt image reconstruction due to concerns about exacerbating overfitting to task-irrelevant features during training. Perceiving the pre-eminence of image reconstruction in representation learning, we propose SMG (Separated Models for Generalization), a novel approach that exploits image reconstruction for generalization. SMG introduces two model branches to extract task-relevant and task-irrelevant representations separately from visual observations via cooperatively reconstruction. Built upon this architecture, we further emphasize the importance of task-relevant features for generalization. Specifically, SMG incorporates two additional consistency losses to guide the agent's focus toward task-relevant areas across different scenarios, thereby achieving free from overfitting. Extensive experiments in DMC demonstrate the SOTA performance of SMG in generalization, particularly excelling in video-background settings. Evaluations on robotic manipulation tasks further confirm the robustness of SMG in real-world applications. Source code is available at `https://anonymous.4open.science/r/SMG/`.

1 Introduction

Visual-based Reinforcement Learning (RL) has demonstrated remarkable success across various tasks, including Atari games [27, 11, 18], robotic manipulation [23, 9], and autonomous navigation [26, 46]. However, deploying visual-based RL algorithms in real-world applications requires a high generalization ability due to numerous factors that can induce distribution shifts between training and deployment scenarios, such as variations in lighting conditions, camera viewpoints, and backgrounds. Many visual-based RL algorithms are prone to overfitting to the training observations [5, 34, 44], limiting their applicability in scenarios where fine-tuning with deployment observations is not allowed.

To address the generalization gap in visual-based RL, current studies primarily focus on utilizing data augmentation techniques [19, 20, 33] and exploring various auxiliary tasks [12, 3, 13]. However, few of the previous works successfully incorporate reconstruction loss to this field, which is commonly adopted in standard visual-based RL settings and has been demonstrated to improve the sample efficiency of RL agents [41, 10, 7]. This is because reconstructing the entire input observation can exacerbate the overfitting problem to task-irrelevant features and thus weaken the generalization ability. Although several works also explored extracting task-relevant features from visual observations

*Corresponding author

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

 https://doi.org/10.52202/079017-3713

[6, 39, 45], little attention has been paid to the potential of leveraging these features in improving generalization.

In this paper, we propose SMG (Separated Models for Generalization), a method that utilizes a reconstruction-based auxiliary task to extract task-relevant representations from visual observations and further strengthens the generalization ability of RL agents with the help of two consistency losses. The core mechanisms behind SMG can be summarized in two parts: First, we introduce two model branches to disentangle foreground and background representations underlying in the visual observations. This separated model framework circumvents the risk of overfitting task-irrelevant features inherent in a single model structure by prudently designing the reconstruction paths, allowing our model to benefit from reconstruction loss without sacrificing generalization ability. Second, we introduce two consistency losses to align the agent's focus on the task-relevant features between raw and augmented observations. This approach enables the foreground model to extract more robust task-relevant representations, which substantially boost the generalization capability of RL agents across diverse deployment scenarios.

We evaluate SMG's effectiveness across a range of challenging visual-based RL tasks, including five tasks from DMControl [36] and two more realistic robotic manipulation tasks [17]. We also adapt different evaluation settings with random-color and video-background modifications. Through comparisons with strong baseline methods, SMG demonstrates state-of-the-art performance in terms of generalization, particularly showcasing superiority in video-background settings and robotic manipulation tasks.

In summary, the main contributions of this paper are as follows:

- We present SMG, a novel approach that aims to enhance the zero-shot generalization ability of RL agents. SMG is designed as a plug-and-play method that seamlessly integrates with existing standard off-policy RL algorithms.
- SMG emphasizes the significance of task-relevant features in visual-based RL generalization and successfully incorporates a reconstruction loss into this setting.
- Extensive experimental results demonstrate that SMG achieves state-of-the-art performance across various visual-based RL tasks, particularly excelling in video-background settings and robotic manipulation tasks.

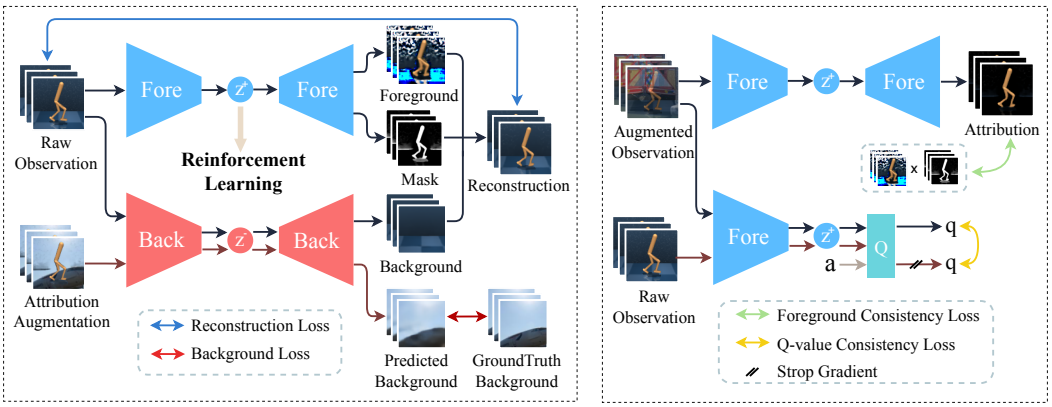
2 Background

A Markov Decision Process (MDP) can be defined as a tuple $(\mathcal{S}, \mathcal{A}, p, r, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $p : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the state transition probability function, $r : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the reward function, and $\gamma \in [0, 1]$ is the discount factor. At each time step t, the agent receives a state $s_t \in \mathcal{S}$, selects an action $a_t \in \mathcal{A}$, and then receives a reward $r_t \in \mathbb{R}$. The agent's goal is to learn a optimal policy $\pi(a_t|s_t)$ that maximizes the expected return $\mathbb{E}_{(s_t,a_t)\sim\rho_\pi}[\sum_{t=0}^{\infty} \gamma^t r_t]$, where ρ_π defines the discounted state-action visitation of π.

Learning an optimal policy from visual observations poses a substantial challenge for RL agents due to the inherent partial observability of the environment, a characteristic of POMDPs (Partially Observed MDP). For one thing, at each timestep t, the visual observation o_t can only capture partial information about the true state s_t, as certain elements may be obscured in the image. For another, the dimension of o_t is much higher than that of s_t, which makes it difficult to utilize o_t directly for policy learning.

To infer the true underlying state from visual observations, existing methods usually employ a parameterized encoder f to map a stacked frame sequence $x_t = (o_{t'}, o_{t'+1}, ..., o_t)$ to a compact low-dimensional latent vector z_t, which is then used as input by policy and value function. However, training the encoder solely to rely on the reward signal is demonstrated to sample inefficiency and may lead to suboptimal performance [41]. To tackle this issue, various auxiliary tasks have been proposed to enhance encoder training, with one common choice being to extract features from pixels via image reconstruction loss [7, 21, 2]. By adding another parameterized image decoder g, the reconstruction loss is defined by maximizing the likelihood function:

$$L_{\text{recon}} = -\mathbb{E}_{o_t\sim\mathcal{D}}[\mathbb{E}_{z_t\sim f(o_t)}[\log g(o_t|z_t)]] \tag{1}$$



(a) Learning Task-Relevant Representations With SMG (b) Improving GeneralizationWith SMG

Figure 1: Architecture of SMG. One-way arrows represent different types of data flows with the same input. Two-way arrows represent different types of loss.

3 Approach

3.1 What Matters in a Reinforcement Learning Task?

Learning to generalize is hard for RL agents, particularly when utilizing an image reconstruction loss. While images are rich in information, requiring the agent to reconstruct the entire input observation can lead the autoencoder network to overfit to features that are unrelated to the task (e.g. colors, textures, and backgrounds). In contrast, humans can accurately figure out what matters visually when learning a new task. Even when colors or backgrounds are changed, humans can still leverage the prior knowledge to complete the task by focusing on task-relevant features. Considering a robotic manipulation task where the agent must move the arm to the red target (Figure 2), despite variations in background colors and textures across four test scenarios on the left, only the arm's orientation and the target position should be focused on this task. We aim for our RL agent to learn an optimal policy that solely relies on these task-relevant features while disregarding irrelevant regions.

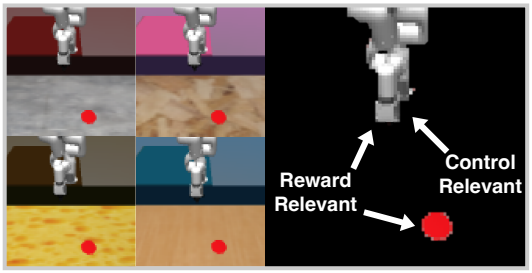


Figure 2: A robotic manipulation task explanation for task-relevant parts in the environment.

Formally, we decompose the latent representation z_t into task-relevant part z_t^+ and task-irrelevant part z_t^-. These two representations are independent, as $p(z_t|o_t) = p(z_t^+|o_t)p(z_t^-|o_t)$. The task-relevant representation can be further subdivided into the "control-relevant" part, which is directly affected by the agent's actions (the arm); and the "reward-relevant" part, which is associated with the reward signal (the arm and the target), both are crucial for policy learning.

3.2 Learning Task-Relevant Representations with Separated Models

3.2.1 Separated Models and Reconstruction

The representation learning objective of SMG is to maximize the mutual information $I(o_t; z_t)$ between the observation o_t and the latent representation z_t, and we further derive an image reconstruction objective incorporating the combination of task-relevant representation z_t^+ and task-irrelevant representation z_t^- as follows:

$$L_{\text{recon}} = -I(o_t; z_t) \leq -\mathbb{E}_{o_t \sim \mathcal{D}}[\mathbb{E}_{z_t^+ \sim f^+(o_t), z_t^- \sim f^-(o_t)}[\log q(o_t|z_t^+, z_t^-)]] \tag{2}$$

Inspired by previous works [6, 30] that explore how to mitigate background distractions, we implement the reconstruction process by introducing the foreground encoder f^+ and the background

https://doi.org/10.52202/079017-3713

encoder f^- to extract different types of representations simultaneously, which forms a separated models architecture. We also incorporate two decoders. The foreground decoder g^+ is employed to reconstruct the foreground image o_t^+ and predict a mask M_t with values between $(0, 1)$. The background decoder g^- is employed to reconstruct the background image o_t^-. The full image o_t is then reconstructed by o_t^+, o_t^- and the mask M_t via $o_t' = o_t^+ \odot M_t + o_t^- \odot (1 - M_t)$ ($\odot$ denotes the Hadamard product), the reconstruction process is illustrated by the black arrows in Figure 1a. Notably, the area where the agent is focusing can be visualized as $o_t^+ \odot M_t$, which we term the "attribution" of the agent, formally defined as $Attrib(o_t)$.

3.2.2 Additional Loss Terms

Based on the separated models architecture, we define four additional loss terms to enhance the model's ability to distinguish between two types of representations. These include the mask ratio loss and background reconstruction loss, which supervise the model's pixel outputs; along with the Q-value loss and empowerment loss, designed to consider the two properties of task-relevant representation.

Mask ratio loss. To further refine the accuracy of mask prediction, we introduce a hyperparameter ρ, termed the mask ratio, to constrain the proportion of the foreground part in the mask. As shown in Equation 3, we regard L_{mask} as an explicit form of an information bottleneck, as the percentage ρ determines the number of pixels of o_t^+ retained in the final reconstruction. This constraint forces f^+ to prioritize the task-relevant parts of the observation during encoding. Empirical results in Section 4.4 demonstrate that L_{mask} facilitates learning a more precise mask.

$$L_{mask} = (\frac{\sum_{i,j} M_t(i,j)}{\text{image\_size}^2} - \rho)^2 \tag{3}$$

Background reconstruction loss. Improving the precision of background prediction can consequently enhance the foreground as well. Since the foreground and background are complementary, providing supervision for the background prevents the foreground from learning all parts of the observation. Therefore, we add additional supervision to the task-irrelevant representation z_t^-. To achieve this, we propose a new type of data augmentation called attribution augmentation tailored for SMG, as illustrated in Figure 3b. This augmentation involves augmenting the raw observation o_t with its corresponding predicted mask M_t via $\tau_{\text{attrib}}(o_t) = o_t \odot M_t + \epsilon \odot (1 - M_t)$, where ϵ represents a randomly sampled image. This simulates the video-background setting in deployment scenarios. We define the background reconstruction loss L_{back} as follows:

$$L_{\text{back}} = -\mathbb{E}_{o_t \sim \mathcal{D}}[\mathbb{E}_{z_t^- \sim f^-(\tau_{\text{attrib}}(o_t))}[\log g^-(\epsilon | z_t^-)]] \tag{4}$$

Q-value loss. Recall that the task-relevant representation z_t^+ has two key properties: reward-relevant and control-relevant. Satisfying the former is relatively straightforward, as the representation z_t^+ is used for policy learning. Through the Bellman residual update objective [35] outlined in Equation 5, z_t^+ will progressively enhance its correlation with the reward signal.

$$L_{\text{q}} = \mathbb{E}_{\tau \sim \mathcal{D}}[(Q(z_t^+, a_t) - (r_t + \gamma V(z_{t+1}^+)))^2] \tag{5}$$

Empowerment loss. For the control-relevant property, we integrate an empowerment term $I(a_t, z_{t+1}^+ | z_t^+)$ [28] based on conditional mutual information, which quantifies the relevance between the action and latent representation. Maximizing the empowerment term further leads to maximizing a variational lower bound $q(a_t | z_{t+1}^+, z_t^+)$ as shown in Equation 6. This objective necessitates that a_t is predictable when two neighboring representations are known. We implement this objective by incorporating an inverse dynamic model.

$$L_{\text{action}} = -I(a_t, z_{t+1}^+ | z_t^+) \leq -\mathbb{E}_{p(a_t, z_{t+1}^+, z_t^+)}[\log q(a_t | z_{t+1}^+, z_t^+)] \tag{6}$$

The whole separated models architecture is shown in figure 1a.

3.3 Generalize Task-Relevant Representations with Separated Models

Utilizing the separated models architecture, SMG can successfully extract task-relevant representations from raw observations. Nevertheless, the agent still lacks the ability to generalize effectively

and may struggle to extract meaningful features from scenarios with transformed styles. To address this issue, we treat the task-relevant representation under raw observations as the ground truth and train SMG on more diversely augmented samples. Instead of directly optimizing the distance between the representations under raw and augmented observations, we introduce two types of consistency losses, considering both attribution and Q-values for more explainable supervision. By doing so, the foreground model can learn to extract task-relevant representations across different deployment scenarios.

Foreground consistency loss. To force the agent to focus on the same task-relevant area in transformed scenarios, we train the foreground models to predict the attribution under augmented observation $Attrib(\tau(o_t))$ with the supervision of the ground truth attribution $Attrib(o_t)$ (as $Attrib(o_t)$ is relatively easier to converge to an accurate value, and we discuss it in detail in Appendix F). The foreground consistency loss $L_{\text{fore\_consist}}$ is defined as Equation 7 (where **sg** means the stop-gradient operation).

$$L_{\text{fore\_consist}} = \mathbb{E}_{o_t \sim \mathcal{D}}[|Attrib(\tau(o_t)) - \mathbf{sg}(Attrib(o_t))|] \tag{7}$$

Q-value consistency loss. In addition to the attributions, the Q-values obtained from transformed observations also exhibit high variance [14], indicating instability in both the extracted representations and the Q function. To address this, we regularize the Q-values under augmented observations to be consistent with those under raw observations, as shown in Equation 8. This approach also regularizes the agent to learn an accurate task-relevant representation, as the gradient of $L_{q\_consist}$ is back-propagated to the latent space.

$$L_{\text{q\_consist}} = \mathbb{E}_{o_t, a_t \sim \mathcal{D}}[[Q(f^+(\tau(o_t)), a_t) - \mathbf{sg}(Q(f^+(o_t), a_t))]^2] \tag{8}$$

The above two consistency losses are illustrated in Figure 1b.

3.4 Overall Objective

Our proposed separated models architecture can seamlessly integrate as a plug-and-play module into any existing off-policy RL algorithms. In this work, we leverage SAC [8] as the base algorithm. Throughout the training phase, SMG iteratively performs exploration, critic update, policy update, and auxiliary task update. We define the critic loss L_{critic} as the sum of the Q-value loss L_{q} and the Q-value consistency loss $L_{\text{q\_consist}}$:

$$L_{\text{critic}} = L_{\text{q}} + \lambda_{\text{q\_consist}} L_{\text{q\_consist}} \tag{9}$$

Additionally, the auxiliary loss L_{aux} comprises five previously mentioned loss terms:

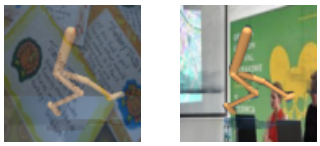
$$L_{\text{aux}} = \lambda_{\text{recon}} L_{\text{recon}} + \lambda_{\text{mask}} L_{\text{mask}} + \lambda_{\text{back}} L_{\text{back}} + \lambda_{\text{action}} L_{\text{action}} + \lambda_{\text{fore\_consist}} L_{\text{fore\_consist}} \tag{10}$$

Although L_{aux} contains five loss terms, experimental results show that using average weights for the first four terms and a smaller weight for the last term can achieve satisfactory performance. Detailed information about hyperparameters tuning is provided in Appendix C.3. The detailed derivation of Equation 2 and Equation 6 are provided in Appendix A.

4 Experimental Results

4.1 Setup

We benchmark SMG against the following baselines: (1) SAC [8], serving as the foundational algorithm for all other baselines; (2) DrQ [19], utilizing random shift augmentation; (3) SODA [12], incorporating a consistency loss on latent representations; (4) SVEA [14], focusing on stabilizing Q-values; (5) SRM [15], proposing a novel data augmentation technique; (6) SGQN [3], the previous SOTA method integrating saliency maps into RL tasks. We reproduce the results using the same settings reported in the original papers, with the exception of setting the batch size to 64 for all methods. Additionally, all results are calculated by four random seeds.



(a) Overlay (b) Attribution

Figure 3: Two types of data augmentations using in SMG.

https://doi.org/10.52202/079017-3713

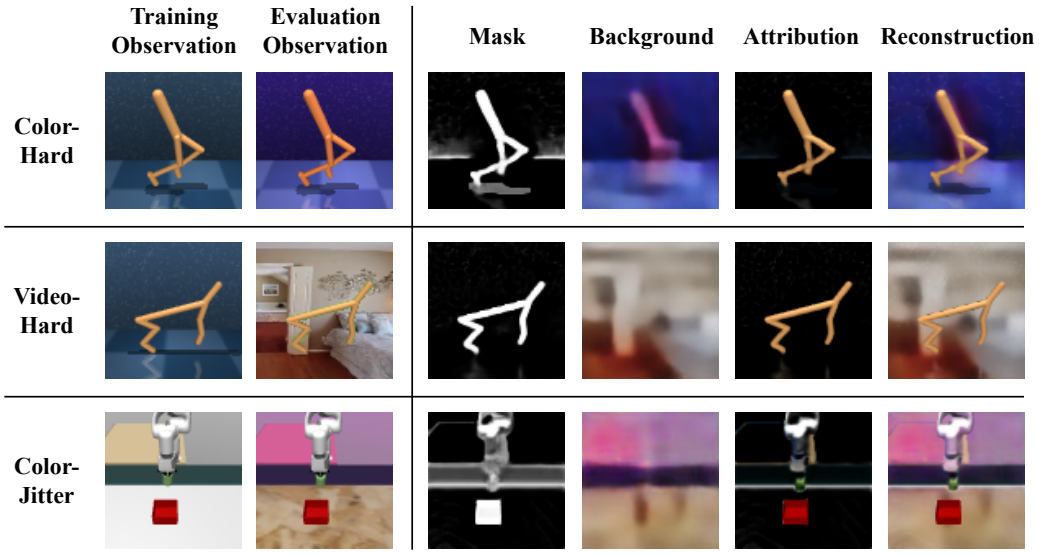
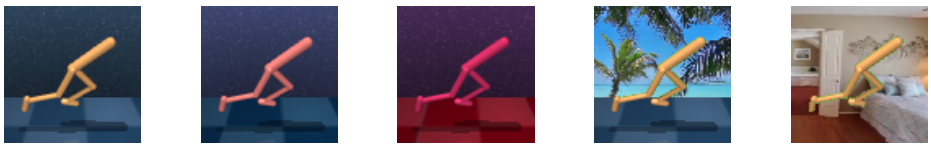


Figure 4: Visualizing the reconstruction process of SMG in different tasks (from top to bottom: *walker-walk*, *cheetah-run*, *peg in box*).

To achieve stable performance across various evaluation settings, we train SMG using a hybrid data augmentation approach for $\tau(o_t)$, involving random overlay [14] and attribution augmentation for all tasks (each time we randomly select a type of data augmentation, as shown in Figure 3). The network design for SMG and more detailed experiment settings are reported in Appendix C.

4.2 DMControl Results

We first conduct experiments on five selected tasks from DMControl [36] and adopt the same evaluation setting as DMControl Generalization Benchmark [12] (DMC-GB) used, which contains random-colors and video-background modifications across four different levels: *color-easy*, *color-hard*, *video-easy* and *video-hard*. Figure 5 shows an example in *walker-walk* task. We train all methods for 500k steps (except *walker-stand* for 250k, as it converges faster) on the training setting and evaluate the zero-shot generalization performance on the four evaluation settings.



(a) *Training* (b) *Color-easy* (c) *Color-hard* (d) *Video-easy* (e) *Video-hard*

Figure 5: Example of training and testing observation for DMC-GB (*walker-walk*). (a) is the training observation. (b-c) indicates different degrees of color change; (d-e) replaces the background with random videos, with (e) additionally removing the floor and the walker's shadow.

To provide a clear explanation of how SMG reconstructs images, we present the image outputs of *walker-walk* and *cheetah-run* after 500k training steps of training in the first two rows of Figure 4. The last four columns illustrate the model outputs necessary for reconstructing the evaluation observations. The predicted attribution (the fifth column) highlights the extracted task-relevant area, which shows SMG accurately depicts the attribution of the input observation while omitting the task-irrelevant elements such as the skybox, the floor, and even the random color variation. This indicates that the task-relevant representation z_t^+ contains only the information required to accomplish the task, which is crucial for generalization. Note that we aim to maintain the similarity between $Attrib(\tau(o_t))$ and $Attrib(o_t)$, even in random-color settings. As shown by the first row of *color-hard* setting, SMG predicts a yellow attribution despite the input evaluation observation being orange.

Table 1: DMControl results in video-background settings. We evaluate each seed five times and calculate the mean value. Then, we calculate the mean and standard deviation with four random seeds. Red indicates the best and blue indicates the second-best. Δ = improvement of SMG over the second best.

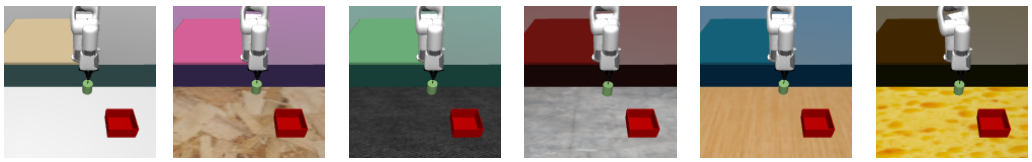
DMControl (*video-easy*)	SAC	DrQ	SODA	SVEA (overlay)	SRM	SGQN	**SMG** (ours)	Δ
`cartpole, swingup`	175 ±23	606 ±31	617 ±76	**718** ±101	645 ±108	717 ±77	**839** ±16	+121 17%
`finger, spin`	171 ±37	511 ±192	615 ±56	817 ±94	642 ±101	**860** ±82	**952** ±48	+92 11%
`walker, stand`	484 ±185	908 ±38	924 ±28	928 ±50	947 ±14	**949** ±10	**961** ±19	+12 1%
`walker, walk`	325 ±26	720 ±69	518 ±92	691 ±120	662 ±75	**830** ±58	**904** ±34	+74 9%
`cheetah, run`	179 ±65	241 ±25	215 ±15	278 ±51	253 ±27	**308** ±34	**348** ±28	+40 13%

DMControl (*video-hard*)	SAC	DrQ	SODA	SVEA (overlay)	SRM	SGQN	**SMG** (ours)	Δ
`cartpole, swingup`	156 ±16	168 ±35	346 ±59	510 ±177	254 ±69	**599** ±112	**764** ±32	+165 28%
`finger, spin`	22 ±10	54 ±44	310 ±72	353 ±71	131 ±89	**710** ±159	**910** ±61	+200 28%
`walker, stand`	212 ±41	278 ±79	406 ±68	814 ±57	558 ±139	**870** ±78	**955** ±9	+85 10%
`walker, walk`	132 ±26	110 ±33	175 ±31	348 ±80	165 ±99	**634** ±136	**814** ±51	+180 28%
`cheetah, run`	56 ±30	38 ±26	118 ±40	105 ±13	87 ±24	**135** ±44	**303** ±46	+168 124%

Table 1 reports the generalization performance of SMG and all baseline methods with the video-background modification, which is the most challenging evaluation setting. The table shows that SMG outperforms all baselines in all ten tasks. Particularly impressive is SMG's superiority in *video-hard*; when removing the floor and the walker's shadow, the performance of all baseline methods drops significantly. However, SMG is less affected by this substantial distribution shift and maintains a stable performance across all tasks, with episode returns boosted more than 160 over the second-best in four out of five tasks (as *walker-stand* is a much easier task to train), showcasing its exceptional generalization capability.

4.3 Robotic Manipulation Results

To further validate SMG's applicability to more realistic tasks, we conduct experiments on two goal-reaching robotic manipulation tasks [17], including *peg-in-box* and *reach*, and following similar generalization settings used in [3]. As illustrated in Figure 6, there are five different testing settings with different colors and textures for the background and the table. We train all methods for 250k steps and use random convolutions [22] as the data augmentation for baseline methods, as it aligns better with the testing scenarios. SMG continued to use hybrid augmentation as previously mentioned.



(a) Training (b) Test 1 (c) Test 2 (d) Test 3 (e) Test 4 (f) Test 5

Figure 6: Examples of training and testing observation for the robotic environment (*Peg-in-box*). (b-f) indicates five different evaluation settings varying in background colors and table textures.

https://doi.org/10.52202/079017-3713

Table 2 presents the evaluation results for *peg-in-box*, a task where a robot must insert a peg tied to its arm into a box. SMG achieves dominant performance across all evaluation settings, boosting an average improvement of 102% over the second-best method. Impressively, SMG exhibits remarkable stability across the six evaluation settings, with a standard deviation of only 7, while baseline methods all fail in some evaluation settings. This underscores SMG's generalization capability. These results also highlight SMG's superiority in realistic tasks, as its reconstruction-based auxiliary loss can capture more detailed features in the image, which is hard for methods that mainly rely on data augmentation techniques.

Table 2: Robotic manipulation results in *peg-in-box*. Red indicates the best and blue indicates the second-best. Δ = improvement of SMG over the second best. The last row reports the average performance over all six evaluation settings.

Robtic-Manipulation (*peg-in-box*)	SAC	DrQ	SODA	SVEA (overlay)	SRM	SGQN	**SMG** (ours)	Δ
`train`	31 ±73	**233** ±14	232 ±20	212 ±39	227 ±15	232 ±19	**237** ±16	+4 2%
`test1`	−33 ±25	**63** ±99	34 ±143	−18 ±59	55 ±98	−67 ±28	**237** ±18	+174 276%
`test2`	−42 ±31	−40 ±77	76 ±119	85 ±68	11 ±54	**194** ±51	**219** ±37	+25 13%
`test3`	−8 ±46	15 ±107	66 ±147	67 ±73	147 ±114	**198** ±34	**237** ±15	+39 20%
`test4`	−42 ±51	72 ±28	80 ±122	109 ±98	**112** ±123	−51 ±46	**237** ±17	+125 112%
`test5`	−52 ±31	−54 ±30	−104 ±51	−26 ±102	**143** ±122	−108 ±24	**237** ±15	+94 66%
Average	−24 ±28	48 ±95	64 ±98	72 ±80	**116** ±69	66 ±143	**234** ±7	+118 102%

4.4 Ablation Study

In order to explore the role played by different loss terms in SMG, we conduct an ablation study in DMControl tasks. Table 3 presents the performance drop without each loss term compared to the full model in the *video-hard* setting. The results indicate that every loss term contributes significantly to the final performance. Notably, $L_{q\_consist}$ exhibits the most substantial impact on performance, highlighting the importance of maintaining stable Q-value estimation in generalization tasks. Moreover, the performance drop without L_{back} or L_{mask} is around 20% to 30%, underlining the importance of attribution augmentation in enhancing SMG's generalization in video-background settings, as the two loss terms directly affect the quality of the attribution augmentation. Additionally, L_{action} aids in learning a better task-relevant representation. As for $L_{fore\_consist}$, it also contributes to improving generalization ability, particularly in relatively challenging tasks where the performance improvement ranges from 15% to 25%.

Table 3: Ablation study in DMControl (*video-hard*). Red indicates the performance drop of the ablated model compared to the full model.

DMControl (video hard)	SMG (full)	w/o $L_{\text{fore\_consis}}$	w/o L_{action}	w/o L_{back}	w/o L_{mask}	w/o $L_{\text{q\_consis}}$
`cartpole, swingup`	764 ± 32	720 ± 100 −44 (6%)	631 ± 92 −133 (17%)	763 ± 44 −1 (0%)	590 ± 84 −174 (23%)	302 ± 30 −462 (60%)
`finger, spin`	910 ± 61	695 ± 103 −215 (24%)	609 ± 352 −301 (33%)	412 ± 170 −498 (55%)	731 ± 130 −179 (20%)	509 ± 83 −401 (44%)
`walker, stand`	955 ± 9	885 ± 45 −70 (7%)	855 ± 96 −100 (10%)	775 ± 144 −180 (19%)	836 ± 127 −119 (12%)	432 ± 210 −523 (55%)
`walker, walk`	814 ± 51	642 ± 63 −172 (21%)	670 ± 22 −144 (18%)	657 ± 103 −157 (19%)	416 ± 98 −398 (49%)	282 ± 34 −532 (65%)
`cheetah, run`	303 ± 46	247 ± 40 −56 (18%)	212 ± 52 −91 (30%)	233 ± 110 −70 (23%)	162 ± 100 −141 (47%)	130 ± 37 −173 (57%)
Average		−15%	−22%	−23%	−30%	−56%

To better grasp the significance of L_{mask} and L_{back} in SMG, we showcase the predicted masks and their corresponding attribution augmentations in Figure 7. When L_{mask} is removed, the model generates an almost white mask, indicating that the foreground model overly captures irrelevant features without the constraint of mask ratio loss. Consequently, only a few parts are replaced by a random image in the attribution augmentation. In contrast, removing L_{back} causes the background model to learn all features excessively, resulting in attribution augmentation images devoid of task-relevant information. The ablation results underscore that both L_{mask} and L_{back} are vital in crafting meaningful attribution augmentations, which in turn are utilized by the two consistency losses and impact the representation learning process. We conduct more experiments in Appendix E to reveal that L_{mask} serves as a guiding factor in mask learning and SMG is not significantly influenced by variations in the hyperparameter mask ratio ρ.

(a) SMG (b) without L_{mask} (c) without L_{back}

Figure 7: Predicted masks and corresponding attribution augmentations. (a) is the full model, (b) and (c) are the models without L_{mask} and L_{mask} respectively.

5 Related Work

Improving generalization ability of RL agents has drawn increasing attention in recent years. Researchers primarily explore two aspects: using data augmentation techniques to inject useful priors when training [20, 15, 16, 22, 14, 32, 38] and employing various auxiliary tasks to guide the learning process [13, 3, 1, 42, 40, 12]. For example, Hansen and Wang [12] regularize the representations between observations with its augmented view through an auxiliary prediction task; Hansen et al. [14] stabilize Q-values via delicately design the data augmentation process; Bertoin et al. [3] introduce saliency maps to visualize the focus of Q-functions; Wang et al. [40] extract the foreground objects by employing a segment anything model. Orthogonal to existing works, we argue that focusing the RL agent on task-relevant features across diverse deployment scenarios can substantially boost the generalization capability. We propose a novel reconstruction-based auxiliary task to achieve this goal.

Decision-making based on task-relevant features can substantially enhance the performance and robustness of RL agents [4, 45, 43, 29]. Bharadhwaj et al. [4] use an empowerment term to distill control-relevant features from the task; Zhu et al. [45] bolster the resilience of RL agents by regularizing the posterior predictability; Zhang et al. [43] learns compact representations by bisimulation metrics. Additionally, methods utilizing separated model architectures to extract different types of features simultaneously have been proposed [6, 39, 30, 25, 37]. For instance, Wang et al. [39] decompose the latent state into four parts based on their interaction with actions and rewards; Pan et al. [30] leverage both controllable and non-controllable states in policy learning; Wan et al. [37] apply task-relevant features to imitation learning. Our work also employs separated models. However, we prudently design this architecture in a model-free setting and propose novel loss terms to enhance the accuracy of image predictions.

A detailed comparison between SMG and other methods is provided in Appendix F.2.

6 Conclusion and Future Work

In this paper, we propose SMG for visual-based RL generalization and show its superiority in sample efficiency, stability, and generalization through extensive experiments. The success of SMG can be attributed to two key factors: (i) a delicately designed reconstruction-based auxiliary task with separated models architecture, which enables the RL agent to extract task-relevant and task-irrelevant representations from visual observations simultaneously; (ii) two consistency losses to further guide the RL agent's focus under deployment scenarios. We believe that the proposed method can be applied to a wide range of tasks.

https://doi.org/10.52202/079017-3713

SMG is particularly well-suited for robotic manipulation tasks in realistic scenarios. However, when the observation contains too many task-relevant objects, the complexity of accurately learning a mask increases. This can lead to a decline in SMG's performance. For instance, in an autonomous navigation task, the presence of numerous pedestrians in the view makes it challenging to accurately mask all of them.

The future work includes exploring more advanced backbones for task-relevant feature extraction, taking into account the generalization on non-static camera viewpoints and the test of SMG on realistic tasks to verify its generalization ability in real applications.

Acknowledgments and Disclosure of Funding

This work is supported by the National Key Research and Development Program of China (No. 2020YFA0711402).

References

[1] Rishabh Agarwal, Marlos C Machado, Pablo Samuel Castro, and Marc G Bellemare. Contrastive behavioral similarity embeddings for generalization in reinforcement learning. *arXiv preprint arXiv:2101.05265*, 2021.

[2] Brandon Amos, Samuel Stanton, Denis Yarats, and Andrew Gordon Wilson. On the model-based stochastic value gradient for continuous reinforcement learning. In *Learning for Dynamics and Control*, pages 6–20. PMLR, 2021.

[3] David Bertoin, Adil Zouitine, Mehdi Zouitine, and Emmanuel Rachelson. Look where you look! saliency-guided q-networks for generalization in visual reinforcement learning. *Advances in Neural Information Processing Systems*, 35:30693–30706, 2022.

[4] Homanga Bharadhwaj, Mohammad Babaeizadeh, Dumitru Erhan, and Sergey Levine. Information prioritization through empowerment in visual model-based rl. *arXiv preprint arXiv:2204.08585*, 2022.

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

[6] Xiang Fu, Ge Yang, Pulkit Agrawal, and Tommi Jaakkola. Learning task informed abstractions. In *International Conference on Machine Learning*, pages 3480–3491. PMLR, 2021.

[7] David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

[8] 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*, pages 1861–1870. PMLR, 2018.

[9] Tuomas Haarnoja, Ben Moran, Guy Lever, Sandy H Huang, Dhruva Tirumala, Markus Wulfmeier, Jan Humplik, Saran Tunyasuvunakool, Noah Y Siegel, Roland Hafner, et al. Learning agile soccer skills for a bipedal robot with deep reinforcement learning. *arXiv preprint arXiv:2304.13653*, 2023.

[10] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019.

[11] Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193*, 2020.

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

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

[14] Nicklas Hansen, Hao Su, and Xiaolong Wang. Stabilizing deep q-learning with convnets and vision transformers under data augmentation. *Advances in neural information processing systems*, 34:3680–3693, 2021.

[15] Yangru Huang, Peixi Peng, Yifan Zhao, Guangyao Chen, and Yonghong Tian. Spectrum random masking for generalization in image-based reinforcement learning. *Advances in Neural Information Processing Systems*, 35:20393–20406, 2022.

[16] Stephen James, Paul Wohlhart, Mrinal Kalakrishnan, Dmitry Kalashnikov, Alex Irpan, Julian Ibarz, Sergey Levine, Raia Hadsell, and Konstantinos Bousmalis. Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12627–12637, 2019.

[17] Rishabh Jangir, Nicklas Hansen, Sambaran Ghosal, Mohit Jain, and Xiaolong Wang. Look closer: Bridging egocentric and third-person views with transformers for robotic manipulation. *IEEE Robotics and Automation Letters*, 7(2):3046–3053, 2022.

[18] Lukasz Kaiser, Mohammad Babaeizadeh, Piotr Milos, Blazej Osinski, Roy H Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, et al. Model-based reinforcement learning for atari. *arXiv preprint arXiv:1903.00374*, 2019.

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

[20] 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, 2020.

[21] Alex X Lee, Anusha Nagabandi, Pieter Abbeel, and Sergey Levine. Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. *Advances in Neural Information Processing Systems*, 33:741–752, 2020.

[22] Kimin Lee, Kibok Lee, Jinwoo Shin, and Honglak Lee. Network randomization: A simple technique for generalization in deep reinforcement learning. *arXiv preprint arXiv:1910.05396*, 2019.

[23] 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.

[24] Lanqing Li, Hai Zhang, Xinyu Zhang, Shatong Zhu, Junqiao Zhao, and Pheng-Ann Heng. Towards an information theoretic framework of context-based offline meta-reinforcement learning. *arXiv preprint arXiv:2402.02429*, 2024.

[25] Yuren Liu, Biwei Huang, Zhengmao Zhu, Honglong Tian, Mingming Gong, Yang Yu, and Kun Zhang. Learning world models with identifiable factorization. *Advances in Neural Information Processing Systems*, 36, 2024.

[26] Piotr Mirowski, Razvan Pascanu, Fabio Viola, Hubert Soyer, Andrew J Ballard, Andrea Banino, Misha Denil, Ross Goroshin, Laurent Sifre, Koray Kavukcuoglu, et al. Learning to navigate in complex environments. *arXiv preprint arXiv:1611.03673*, 2016.

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

[28] Shakir Mohamed and Danilo Jimenez Rezende. Variational information maximisation for intrinsically motivated reinforcement learning. *Advances in neural information processing systems*, 28, 2015.

 https://doi.org/10.52202/079017-3713

[29] Tung D Nguyen, Rui Shu, Tuan Pham, Hung Bui, and Stefano Ermon. Temporal predictive coding for model-based planning in latent space. In *International Conference on Machine Learning*, pages 8130–8139. PMLR, 2021.

[30] Minting Pan, Xiangming Zhu, Yunbo Wang, and Xiaokang Yang. Iso-dream: Isolating and leveraging noncontrollable visual dynamics in world models. *Advances in neural information processing systems*, 35:23178–23191, 2022.

[31] Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *International Conference on Machine Learning*, pages 5171–5180. PMLR, 2019.

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

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

[34] Xingyou Song, Yiding Jiang, Stephen Tu, Yilun Du, and Behnam Neyshabur. Observational overfitting in reinforcement learning. *arXiv preprint arXiv:1912.02975*, 2019.

[35] Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3:9–44, 1988.

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

[37] Shenghua Wan, Yucen Wang, Minghao Shao, Ruying Chen, and De-Chuan Zhan. Semail: eliminating distractors in visual imitation via separated models. In *International Conference on Machine Learning*, pages 35426–35443. PMLR, 2023.

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

[39] Tongzhou Wang, Simon S Du, Antonio Torralba, Phillip Isola, Amy Zhang, and Yuandong Tian. Denoised mdps: Learning world models better than the world itself. *arXiv preprint arXiv:2206.15477*, 2022.

[40] Ziyu Wang, Yanjie Ze, Yifei Sun, Zhecheng Yuan, and Huazhe Xu. Generalizable visual reinforcement learning with segment anything model. *arXiv preprint arXiv:2312.17116*, 2023.

[41] Denis Yarats, Amy Zhang, Ilya Kostrikov, Brandon Amos, Joelle Pineau, and Rob Fergus. Improving sample efficiency in model-free reinforcement learning from images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10674–10681, 2021.

[42] 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. *arXiv preprint arXiv:2202.09982*, 2022.

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

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

[45] Chuning Zhu, Max Simchowitz, Siri Gadipudi, and Abhishek Gupta. Repo: Resilient model-based reinforcement learning by regularizing posterior predictability. *Advances in Neural Information Processing Systems*, 36, 2024.

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

A Derivations

We formulate the representation learning objective as a variational lower bound of the mutual information [31, 24] between the observation o_t and the representation z_t. By considering the independence between the task-relevant and task-irrelevant representations, we can decompose the mutual information as:

$$\begin{aligned} I(o_t; z_t) &= \mathbb{E}_{p(o_t, z_t)}[\log p(o_t|z_t) - \log p(o_t)] \\ &\geq \mathbb{E}_{p(o_t, z_t)}[\log p(o_t|z_t)] \\ &\geq \mathbb{E}_{p(o_t, z_t)}[\log p(o_t|z_t)] - \mathbb{E}_{p(z_t)}[\mathbb{D}_{KL}(p(o_t|z_t)||q(o_t|z_t))] \\ &= \mathbb{E}_{p(z_t, o_t)}[\log q(o_t|z_t)] \\ &= \mathbb{E}_{q(z_t|o_t)p(o_t)}[\log q(o_t|z_t)] \\ &= \mathbb{E}_{q(z_t^+|o_t)q(z_t^-|o_t)p(o_t)}[\log q(o_t|z_t^+, z_t^-)] \\ &= \mathbb{E}_{o_t \sim \mathcal{D}}[\mathbb{E}_{z_t^+ \sim f^+(o_t), z_t^- \sim f^-(o_t)}[\log q(o_t|z_t^+, z_t^-)]] \end{aligned} \tag{11}$$

We use the empowerment term $I(a_t, z_{t+1}^+|z_t^+)$ introduced in [28] to quantify the information contained in the representation z_{t+1}^+ about the selected action a_t, in goal of enhance the control-relevant property of the task-relevant representation z_t^+. We derive the variational lower bound of the empowerment term as:

$$\begin{aligned} I(a_t, z_{t+1}^+|z_t^+) &= \mathbb{E}_{p(a_t, z_{t+1}^+, z_t^+)}[\log \frac{p(a_t|z_{t+1}^+, z_t^+)}{p(a_t|z_t^+)}] \\ &= \mathbb{E}_{p(a_t, z_{t+1}^+, z_t^+)}[\log \frac{q(a_t|z_{t+1}^+, z_t^+)}{p(a_t|z_t^+)} + \log \frac{p(a_t|z_{t+1}^+, z_t^+)}{q(a_t|z_{t+1}^+, z_t^+)}] \\ &\geq \mathbb{E}_{p(a_t, z_{t+1}^+, z_t^+)}[\log \frac{q(a_t|z_{t+1}^+, z_t^+)}{p(a_t|z_t^+)}] \\ &= \mathbb{E}_{p(a_t, z_{t+1}^+, z_t^+)}[\log q(a_t|z_{t+1}^+, z_t^+)] - \int p(z_t^+)p(a_t|z_t^+)p(z_{t+1}^+|z_t^+, a_t) \log p(a_t|z_t^+) \\ &= \mathbb{E}_{p(a_t, z_{t+1}^+, z_t^+)}[\log q(a_t|z_{t+1}^+, z_t^+)] + \mathbb{E}_{p(z_t^+)p(z_{t+1}^+|z_t^+, a_t)}[H(p(a_t|z_t^+))] \\ &\geq \mathbb{E}_{p(a_t, z_{t+1}^+, z_t^+)}[\log q(a_t|z_{t+1}^+, z_t^+)] \end{aligned} \tag{12}$$

In practice, we integrate a parameterized inverse dynamic model to predict the action a_t based on the two continuous representations z_t^+ and z_{t+1}^+. We employ the Mean Squared Error (MSE) loss to guide the training of the inverse dynamic model.

https://doi.org/10.52202/079017-3713

B Pseudocode

Algorithm 1 SAC with Separated Models

Denote network parameters θ, mask ratio ρ, batch size N, replay buffer $\mathcal{B}$
Denote policy network π_θ, foreground encoder f_θ^+, background encoder f_θ^-
foreach *iteration time step* **do**
  $a, o', r \sim \pi_\theta(f_\theta^+(o)), \mathcal{P}(o,a), \mathcal{R}(o,a)$
  $\mathcal{B} \leftarrow \mathcal{B} \cup (o, a, r, o')$
  foreach *update time step* **do**
    $\{o_i, a_i, r_i, o_i'\}_{i\in[1,N]} \sim \mathcal{B}$
    $o_i^+, mask_i \sim f_\theta^+(o_i)$
    $o_i^- \sim f_\theta^-(o_i)$
    $o_i^{aug} \leftarrow o_i^+ * mask_i + \epsilon * (1 - mask_i)$ // ϵ is sampled from image dataset
    $L_{recon} \leftarrow L(o_i, o_i^+ * mask_i + o_i^- * (1 - mask_i))$ // Equation 2
    $L_{fore\_consist} \leftarrow L(o_i^+, f_\theta^+(o_i^{aug}))$ // Equation 7
    $L_{back} \leftarrow L(\epsilon, f_\theta^-(o_i^{aug}))$ // Equation 4
    $L_{action} \leftarrow L(o_i, o_i', a)$ // Equation 6
    $L_{mask} \leftarrow L(mask_i, \rho)$ // Equation 3
    $L_{q\_consist} \leftarrow L(Q_\theta(f_\theta^+(o_i), a), Q_\theta(f_\theta^+(o_i^{aug}), a))$ // Equation 8
    $L_{aux} \leftarrow L_{recon} + L_{fore\_consist} + L_{back} + L_{action} + L_{mask}$ // auxiliary loss
    $L_{critic} \leftarrow L_q + L_{q\_consist}$ // critic loss
    update θ with $L_{actor}, L_{critic}, L_{aux}$
  end for
end for
$L_{\text{q}}, L_{\text{actor}}$ are defined by SAC

C More Experiment Details

C.1 Computing Hardware

We conduct all experiments on a single machine equipped with an AMD EPYC 7B12 CPU (64 cores), 512GB RAM, and eight NVIDIA GeForce RTX 3090 GPUs (24 GB memory). We report the training wall time of different methods on DMControl tasks in Table 4.

Table 4: Wall time comparison of different methods on DMControl tasks.

Algorithm	Wall Time (DMControl, 500k)
SAC	$\sim$ 10 hours
DrQ	$\sim$ 13 hours
SODA	$\sim$ 12 hours
SVEA	$\sim$ 12 hours
SRM	$\sim$ 8 hours
SGQN	$\sim$ 12 hours
SMG (ours)	$\sim$ 22 hours

C.2 Network Architecture

We reproduce all baseline methods with the official code of DMC-GB (`https://github.com/nicklashansen/dmcontrol-generalization-benchmark`) published by Nicklas Hansen, and we build our model on top of the SAC implementation. We use the same encoder and decoder architecture as the baseline methods to ensure a fair comparison.

Figure 8 provides a detailed view of the encoder and decoder architecture. The input observation shape is $9 \times 84 \times 84$, achieved by stacking three continuous frames. The encoder network contains 12 stacked convolutional layers, each with 32 filters of size 3×3. The stride is set to 1 for the first layer and 2 for the subsequent ones, facilitating down-sampling of the visual input. Then, after a flatten operation and a fully connected layer, an embedding of size *embedding_size* $\times$ 1 is obtained. Before decoding, SMG first expands the embedding into triples of the same size, aiming to decode three stacked input images separately. These three embeddings are then individually fed into the same decoder network, which consists of two groups of convolutional and upsampling layers to reconstruct the observation. The foreground decoder outputs the reconstructed foreground and a mask, while the background decoder outputs only the reconstructed background. For the inverse dynamic model, we adopt the architecture from [13], which utilizes multi-layer perceptions to project the concatenation of two embeddings into the action space.

The number of parameters in SMG is approximately double that of the baseline methods due to the use of two model branches. However, the performance improvement is primarily due to the novel model architecture rather than the increase in the number of parameters, as we use encoder and decoder networks similar to those in the baseline methods.

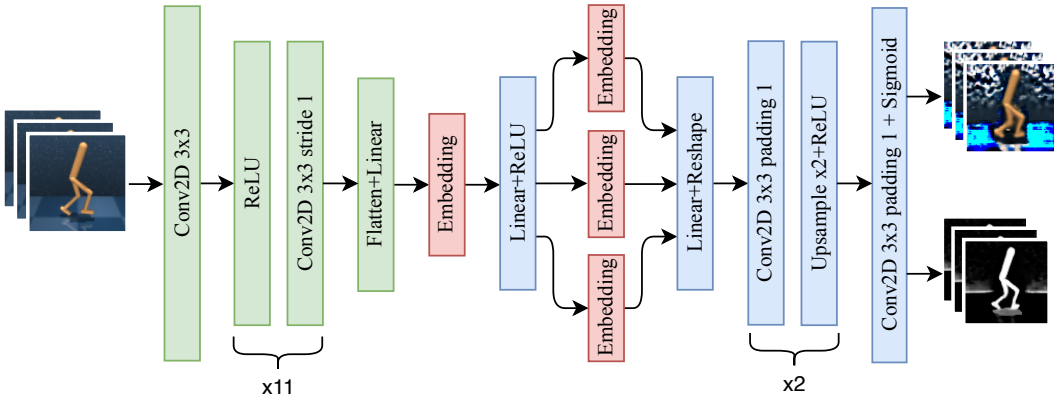


Figure 8: SMG network architecture (foreground encoder + foreground decoder).

C.3 Hyperparameters

We report the hyperparameters used in our experiments in Table 5. We use the same hyperparameters for all seven tasks, except the action repeat and the mask ratio ρ. The L_{aux} in SMG comprises five loss terms, which seems challenging to balance the weights. However, through experiments, we found that setting average weights for $L_{recon}, L_{mask}, L_{action}, L_{back}$ is sufficient to achieve good performance (except the λ_{back} is set to 2 since the background model should train to fit more complex images). Regarding the L_{fore}, a too-large weight would lead to the model overfitting the inaccurate attribution predictions in the early stage (as we use the model output under raw observation as ground truth), so we set it to 0.1.

D More Experiment Results

D.1 Training Curves

We present the training curves for all seven tasks in Figure 11, including four evaluation settings of DMControl and Robotic Manipulation tasks. As depicted in the figure, SMG demonstrates notably faster convergence and higher asymptotic performance across nearly all training and evaluation settings, showcasing the effectiveness of the reconstruction-based auxiliary task in enhancing sample efficiency. SMG exhibits superiority, particularly in the *video-hard* setting of DMControl tasks, where the performance of other methods drops evidently when random videos replace the background. Additionally, the figure underscores the considerable challenge posed by Robotic Manipulation tasks, with only SMG and SGQN successfully achieving zero-shot generalization in evaluation settings.

https://doi.org/10.52202/079017-3713

Table 5: Hyperparameters.

Hyperparameter	**Value**
Observation size	84×84
Frame stack	3
Discount factor γ	0.99
Batch size	64
Embedding size	256
Action repeat	8 (*cartpole-swingup*), 4 (*walker-walk*, *walker-stand*, *cheetah-run*) 2 (*finger-spin*), 1 (*reach*, *peg-in-box*)
Train steps	250k (*walker-stand*, *reach*, *peg-in-box*), 500k (others)
Replay buffer size	500k
Actor optimizer	Adam ($lr = 1\text{e}-3, \beta_1 = 0.9, \beta_2 = 0.999$)
Critic optimizer	Adam ($lr = 1\text{e}-3, \beta_1 = 0.9, \beta_2 = 0.999$)
Auxiliary task optimizer	Adam ($lr = 1\text{e}-3, \beta_1 = 0.9, \beta_2 = 0.999$)
Auxiliary task update frequency	2
Reconstruction loss weight λ_{recon}	1
Background reconstruction loss weight λ_{back}	2
Mask ratio loss weight λ_{mask}	1
Empowerment loss weight λ_{action}	1
Q-value consistency loss weight $\lambda_{q\_consist}$	0.5
Foreground consistency loss weight $\lambda_{fore\_consist}$	0.1
Mask ratio ρ	0.12 (*reach*, *peg-in-box*), 0.06 (*walker-walk*, *walker-stand*, *cheetah-run*) 0.04 (*cartpole-swingup*, *finger-spin*)

Moreover, SMG shows more stable performance across different evaluation settings, which is crucial for real-world applications.

D.2 More Table Results

Table 6: DMControl results in random-color settings.

DMControl-GB (*color-easy*)	SAC	DrQ	SODA	SVEA (overlay)	SRM (SAC)	SGQN	**SMG** (ours)	Δ
`cartpole, swingup`	178 ±24	845 ±29	720 ±109	809 ±40	**856** ±14	764 ±84	**854** ±13	−2 0%
`finger, spin`	296 ±22	827 ±174	761 ±87	**919** ±43	916 ±34	852 ±126	**957** ±52	+38 4%
`walker, stand`	592 ±274	827 ±97	929 ±23	**957** ±4	953 ±5	906 ±50	**965** ±13	+8 1%
`walker, walk`	430 ±33	669 ±68	539 ±51	705 ±124	632 ±93	**805** ±47	**915** ±36	+110 14%
`cheetah, run`	253 ±27	237 ±74	219 ±46	289 ±43	272 ±24	**312** ±34	**346** ±27	+34 11%

DMControl-GB (*color-hard*)	SAC	DrQ	SODA	SVEA (overlay)	SRM (SAC)	SGQN	**SMG** (ours)	Δ
`cartpole, swingup`	184 ±26	717 ±133	585 ±66	**752** ±86	**752** ±103	636 ±110	726 ±62	−26 3%
`finger, spin`	271 ±23	655 ±214	663 ±106	**868** ±74	834 ±90	700 ±219	**841** ±113	−27 3%
`walker, stand`	526 ±259	769 ±182	719 ±138	799 ±118	**807** ±128	788 ±114	**878** ±70	+71 9%
`walker, walk`	379 ±37	456 ±192	396 ±78	571 ±134	483 ±123	**632** ±176	**739** ±31	+107 17%
`cheetah, run`	208 ±54	147 ±80	199 ±38	**238** ±69	203 ±30	210 ±18	**299** ±22	+61 26%

Table 6 shows the generalization performance of SMG and all baseline methods with the random-color modification in DMControl tasks. SMG outperforms all baselines in 7 out of 10 tasks, with the performance gap within 5% in the other three tasks. The results indicate that SMG not only performs well in video-background settings but also exhibits superior generalization capability in random-color settings. This is achieved because overlaying the observation with random images can also introduce color shift.

Table 7: Training and average performance in DMControl.

DMControl-GB (training)	SAC	DrQ	SODA	SVEA (overlay)	SRM (SAC)	SGQN	**SMG** (ours)	Δ
cartpole, swingup	186 ±6	**872** ±10	687 ±175	809 ±42	**871** ±10	805 ±58	858 ±9	−14 2%
finger, spin	306 ±12	884 ±115	801 ±65	923 ±36	**925** ±35	922 ±61	**961** ±44	+36 4%
walker, stand	630 ±224	955 ±18	881 ±51	**959** ±5	959 ±6	952 ±17	**964** ±18	+5 1%
walker, walk	422 ±42	827 ±61	581 ±129	753 ±143	715 ±74	**876** ±45	**924** ±31	+48 5%
cheetah, run	311 ±36	333 ±43	225 ±39	300 ±37	298 ±30	**343** ±37	**357** ±25	+14 4%

DMControl-GB (average)	SAC	DrQ	SODA	SVEA (overlay)	SRM (SAC)	SGQN	**SMG** (ours)	Δ
cartpole, swingup	176 ±11	642 ±255	591 ±132	**720** ±110	676 ±226	704 ±77	**808** ±53	+88 12%
finger, spin	213 ±107	586 ±297	630 ±173	776 ±215	690 ±297	**809** ±88	**924** ±45	+115 14%
walker, stand	489 ±147	747 ±243	772 ±198	891 ±70	845 ±154	**893** ±61	**945** ±33	+52 6%
walker, walk	338 ±109	556 ±254	442 ±147	614 ±146	531 ±199	**755** ±103	**859** ±72	+104 14%
cheetah, run	201 ±85	199 ±100	195 ±40	242 ±72	223 ±75	**262** ±77	**331** ±24	+69 26%

For a more direct measurement of the generalization ability in DMControl, we further calculate the average performance across five evaluation settings (including performance under training observation) and report the results in Table 7. As shown in the table, SMG achieves state-of-the-art zero-shot generalization capability in all five DMControl tasks, surpassing all baseline methods by a margin of up to 26%. The results also demonstrate SMG's stability across different evaluation settings, with standard deviations less than 80 in all tasks. In contrast, the standard deviations of other methods range from 100 to 250.

Table 8: Robotic manipulation results in *reach*.

Robtic-Manipulation (*Reach*)	SAC	DrQ	SODA	SVEA (overlay)	SRM (SAC)	SGQN	**SMG** (ours)	Δ
train	4 ±18	32 ±3	11 ±14	**33** ±2	30 ±2	**33** ±2	30 ±2	−3 9%
test1	−16 ±33	−1 ±23	−26 ±9	−22 ±16	−3 ±25	**19** ±13	**30** ±1	+11 58%
test2	−10 ±22	−9 ±11	−17 ±16	−21 ±22	−8 ±22	**33** ±2	**24** ±6	−9 27%
test3	−32 ±14	−38 ±29	−20 ±34	−13 ±10	24 ±9	**33** ±2	**30** ±2	−3 9%
test4	−19 ±50	10 ±26	−21 ±16	0 ±21	−1 ±30	**24** ±6	**29** ±1	+5 21%
test5	−54 ±11	−33 ±19	−50 ±7	−37 ±27	**-8** ±29	−16 ±22	**29** ±2	+37 462%
Average	−21 ±18	−6 ±24	−20 ±18	−10 ±22	6 ±15	**21** ±17	**29** ±2	+8 38%

The experiment results of robotic manipulation *reach* are reported in Table 8. SMG also shows a stable and superior performance in this task, with an average improvement of 38% over the second-best method.

https://doi.org/10.52202/079017-3713

E More Ablation Study

We report the effect of removing each loss term to the average performance across five evaluation settings in DMControl tasks in Table 9. Compared with Table 3, $L_{q\_consist}$ still exhibits the most substantial impact on performance, though the performance drop is slightly smaller. This may be because the random-color settings do not shift the observations heavily compared to the video-background settings, so the Q-value estimation is less affected. A similar phenomenon is observed in L_{back} and L_{mask}, indicating that attribution augmentation is more crucial in video-background settings.

The mask ratio ρ is a hyperparameter that controls the expected proportion of the foreground area. However, this parameter is an empirical choice and may not precisely match the actual proportion of a given task. To investigate the sensitivity of SMG to the mask ratio, we conduct experiments with different ρ values in the *walker-walk* task of the *video-hard* setting. We select ρ values ranging from 0.02 to 0.1 with an interval of 0.02 and report the average performance across five evaluation settings in Figure 9. The results indicate that variations do not significantly influence SMG in the mask ratio, as ρ values between 0.04 and 0.08 achieve similar performance. Moreover, when ρ is too small (0.02) or too large (0.1), the performance drops around 6% compared to the optimal ρ value (0.06). We also report the predicted masks of different ρ values in the figure. As ρ increases, the predicted masks start to include background areas, so a too high value leads to decreased performance. Conversely, when ρ is too small, the mask depicts an inaccurate foreground area (e.g. the legs of the walker with $\rho = 0.02$), resulting in a performance drop as well.

Table 9: Ablation study in DMControl (average performance).

DMControl (average)	SMG (full)	w/o $L_{\text{fore\_consis}}$	w/o L_{action}	w/o L_{back}	w/o L_{mask}	w/o $L_{\text{q\_consis}}$
`cartpole, swingup`	808 ± 53	763 ± 28 -45 (6%)	762 ± 71 -46 (6%)	795 ± 32 -13 (2%)	758 ± 97 -50 (6%)	646 ± 191 -162 (20%)
`finger, spin`	924 ± 45	815 ± 66 -109 (12%)	791 ± 112 -133 (14%)	640 ± 115 -284 (31%)	866 ± 73 -58 (6%)	773 ± 151 -151 (16%)
`walker, stand`	945 ± 33	918 ± 26 -27 (3%)	874 ± 17 -71 (8%)	915 ± 70 -30 (3%)	930 ± 47 -15 (2%)	598 ± 114 -347 (37%)
`walker, walk`	859 ± 72	727 ± 54 -132 (15%)	757 ± 61 -102 (12%)	756 ± 103 -103 (12%)	693 ± 145 -166 (19%)	613 ± 227 -246 (29%)
`cheetah, run`	331 ± 24	304 ± 29 -27 (8%)	325 ± 58 -6 (2%)	270 ± 25 -61 (18%)	319 ± 82 -12 (4%)	269 ± 110 -62 (19%)
Average		-9%	-8%	-13%	-7%	-24%

(a) $\rho = 0.02$ (800) (b) $\rho = 0.04$ (857) (c) $\rho = 0.06$ (859) (d) $\rho = 0.08$ (823) (e) $\rho = 0.1$ (793)

Figure 9: Ablation study of mask ratio ρ in *walker-walk* of average performance across five evaluation settings. The images and numbers in parentheses indicate the predicted masks and the corresponding performance, respectively.

F More Discussion

F.1 Bootstrapping Process in SMG

The attribution augmentation utilized in SMG requires the model to predict an accurate mask, and the foreground consistency loss also requires a precise attribution prediction of the model. This might seem contradictory, as the model struggles to make meaningful predictions in the early stages, which means it cannot satisfy the two requirements immediately. We dig into the training process of SMG

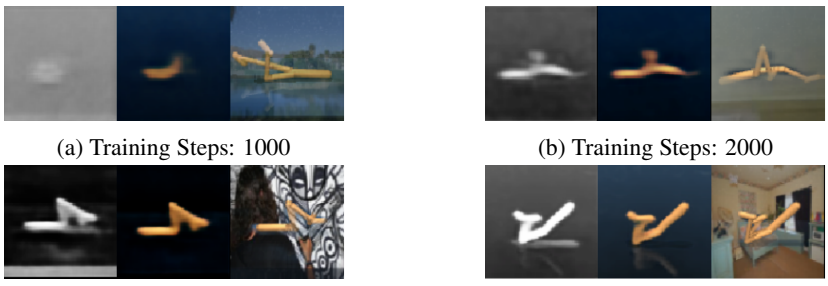
by experiments and provide the model outputs in different training stages in Figure 10. In the very early stage (≤ 1000 steps), the model has difficulty predicting accurate masks, leading the attribution augmentation more likes an overlay augmentation. However, the model rapidly learns to predict relatively accurate masks and generate meaningful attribution augmentation images that can help optimize L_{back} and $L_{\text{fore\_consist}}$ (after 2000 steps), aided by the constraint of L_{q}. Subsequently, with the inclusion of L_{back} and $L_{\text{fore\_consist}}$, the network begins to focus more on task-relevant areas in the observation, thereby in turn comes back to enhance the accuracy of Q-values and foreground predictions. Consequently, we view the training of SMG as a bootstrapping process.

F.2 Comparison with Related Work

TIA [6] also designs two model branches to capture task and distractor features, similar to our separated models architecture. However, SMG differs from TIA in several essential aspects: (i) TIA is a model-based method focusing on eliminating task-irrelevant distractors in training observations, while SMG aims to utilize task-relevant features across diverse deployment scenarios to enhance the generalization capability of RL agents; (ii) SMG operates in a model-free setting, which can be more efficient to train and more flexible for applying data augmentation techniques; (iii) TIA uses a background-only reconstruction loss and requires the background model to reconstruct the full observation, which may cause the background branch to overly fit task-relevant features. In contrast, SMG addresses this issue by introducing attribution augmentation images to supervise the background model; (iv) SMG utilizes mask ratio loss to learn a more precise mask, while the masks in TIA are prone to containing distractors, as reported in its original paper.

SODA [12] also improves the generalization ability of RL agents by regularizing the representations between observations and their augmented views, similar to the consistency losses in SMG. However, SODA implements this by simply minimizing the L2 distance between the two representations, which imposes a too rigid constraint and lacks interpretability. We achieve this by introducing Q-value consistency loss and foreground consistency loss, which provide more explainable supervision and additionally improve the stability of Q-values and predicted attributions.

Note that the core idea underlying the Q-value loss in Equation 8 differs significantly from the consistency regulation objective proposed by SGQN [3]. SGQN focuses on prioritizing pixels that belong to the saliency map during encoding, primarily to enhance the accuracy of Q-value estimation under raw observations. In contrast, SMG treats the Q-values under raw observations as the ground truth and aims to achieve consistency between these Q-values and those obtained under augmented observations. Thus, we additionally use a stop-gradient operation.



(a) Training Steps: 1000 (b) Training Steps: 2000

(c) Training Steps: 6000 (d) Training Steps: 30000

Figure 10: Masks, attributions, and corresponding attribution augmentation images in different training stages.

https://doi.org/10.52202/079017-3713

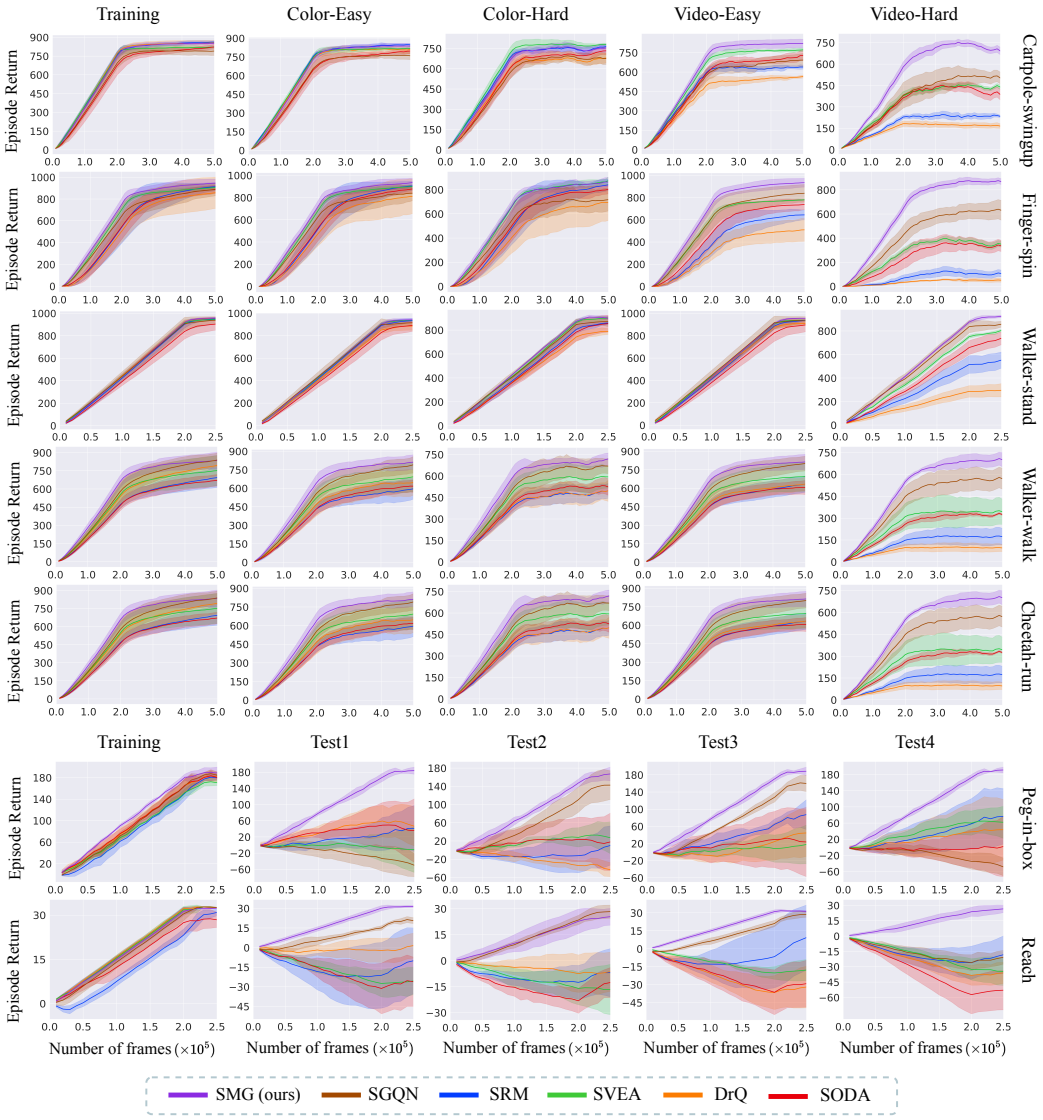


Figure 11: Training curves in all seven tasks. We evaluate each seed three times and then calculate the mean episode return for every 10k training steps, and the variance is shown as the shaded area by calculating four random seeds.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the contributions of the paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations and future work are discussed in the Section 6.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

https://doi.org/10.52202/079017-3713

Justification: The derivations of the representation learning objective and the empowerment term are provided in the Appendix A.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Source code is available at `https://anonymous.4open.science/r/SMG/`, and the experimental setting and details are described in the Appendix C.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open Access to Data and Code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Source code is available at `https://anonymous.4open.science/r/SMG/`.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The experimental setting and details are described in the Appendix C.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The paper reports the mean and standard deviation of the results in the tables and figures.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

https://doi.org/10.52202/079017-3713

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper provides the details of the compute resources in the Appendix C.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and ensured that our research conforms to it.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: There is no societal impact of the work performed.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper does not release data or models that have a high risk for misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for Existing Assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper properly credits the original owners of the assets and mentions the license and terms of use. The official code of DMC-GB (`https://github.com/nicklashansen/dmcontrol-generalization-benchmark`) uses the MIT license.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

https://doi.org/10.52202/079017-3713

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We publish our source code, and the new assets are well documented in this paper.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

https://doi.org/10.52202/079017-3713