Improving Visual Generalization in Model-Based Reinforcement Learning

Anonymous Authors[1]

Abstract

Learning a generalizable reinforcement learning (RL) agent to the unseen visual image in a zero-shot manner enables further deployments of deep RL into the real world. The field has witnessed significant progress in the prior literature by leveraging data augmentation and auxiliary representation learning techniques. However, simultaneously achieving superior sample efficiency and generalization ability still remains challenging for visual RL agents. In this work, we devise **Vi**sual **G**eneralization in **MO**del-Based RL (**ViGMO**), a novel model-based RL method to encourage visual generalization with superior sample efficiency by blending a popular model-based RL architecture with groundbreaking recipes from the prior literature in model-free RL. Our key idea is to constrain the model to exhibit a consistent prediction ability regardless of visual perturbations during training. We provide extensive empirical results on the sample efficiency and generalization ability of visual RL agents in diverse environments and tasks.

1. Introduction

Deep reinforcement learning (DRL), an interconnected field of deep neural network and reinforcement learning, has pioneered diverse sequential decision-making problems, including games (Silver et al., 2016; Van Hasselt et al., 2016; Wang et al., 2016; Hessel et al., 2018) and robotic locomotion (Lillicrap, 2015; Schulman, 2015; Schulman et al., 2017; Haarnoja et al., 2018; Fujimoto et al., 2018). Visual reinforcement learning has achieved impressive successes by expanding the low-dimensional state space of a DRL agent to a high-dimensional pixel image space across complex continuous control problems (Laskin et al., 2020a; Yarats et al., 2021b).

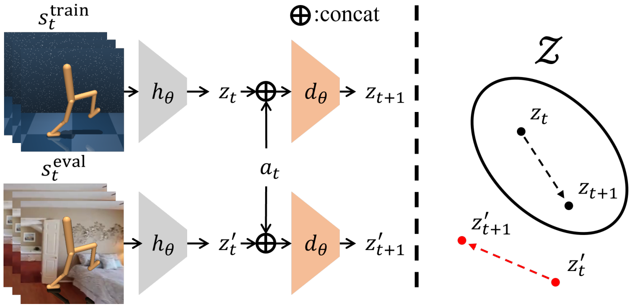


Figure 1. **Out-of-distributional representation.** Distribution shift occurs when sampled states between training and evaluation distribution differ. s_t^{train} and s_t^{eval} are example states. h_θ and d_θ are the encoder and transition dynamics, z and z' are extracted representations from in-distributional and out-of-distributional states, respectively. a is an action and $\mathcal{Z}$ is the distribution of z where representations are projected from only the training distribution. Subscript t represents a time step of the environment transition.

Learning a visuomotor policy that derives an optimal action from pixel image input presents unique challenges; for instance, the generalization of the learned model. While the inductive bias of the visual feature extractor (i.e., CNN) enables efficient behavior learning of a DRL agent, the inherent nature of CNN also hampers the broader generalization capability of the learned policy. Alleviating this generalization issue has induced numerous challenges since deep RL often couples policy learning and representation learning. Previous approaches have fabricated novel solutions by learning robust representation to visual disturbances (Laskin et al., 2020b; Nair et al., 2022; Yuan et al., 2022; Wang et al., 2023; Yang et al., 2024), applying stronger data augmentations (Laskin et al., 2020a; Hansen & Wang, 2021; Hansen et al., 2021; Huang et al., 2022), and stabilizing value function learning (Hansen et al., 2021; Liu et al., 2023; Huang et al., 2024).

While visual RL agents have demonstrated impressive visual generalization performance across diverse continuous control tasks, those agents still suffer from poor sample efficiency. Interestingly, a common ground shared across these approaches is that they fall into the model-free RL category where the agent mainly relies on the Q value function for policy learning. However, the nature of model-free RL that updates the policy incrementally and contains weak

[1] Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

inductive bias (Botvinick et al., 2019) essentially decreases the sample efficiency of the RL agent. Furthermore, the broken randomness (Xu et al., 2023) or high-dimensional state and action spaces (Yarats et al., 2021c) worsen this problem. Alongside model-free RL, groundbreaking ideas in model-based RL have proven their superior performance and sample efficiency in diverse and challenging continuous control suites in recent years (Hafner et al., 2019b;a; Hansen et al., 2022; 2023; Hafner et al., 2023). By learning a latent transition dynamics model with additional components regarding the model, current model-based RL has validated scalability to higher dimensions and brilliant performance on more complex domains. However, naively adopting model-based RL into visual generalization can become problematic; the learned latent transition dynamics model would be conditioned on out-of-distributional representations, leading to a collapse of model-based RL in Figure 1.

In this paper, we propose ViGMO; improving **Vi**sual **G**eneralization in **MO**del-based reinforcement learning, a model-based RL that empirically demonstrates strong generalization ability over unseen image input without sacrificing sample efficiency by employing recipes from model-free RL. ViGMO consists of three key factors for improving performance: (1) applying weak and strong data augmentations to given image input for sample efficiency and generalization, (2) predicting a consistent latent representation simulated by the latent transition dynamics, and (3) regularizing the encoder to extract consistent representations over differently augmented input. We perform extensive experiments to verify our design choice contributes to superior performance on the generalization benchmark (Yuan et al., 2024) across DM-Control (Tassa et al., 2018) and Robosuite (Zhu et al., 2020) benchmarks. Through a comprehensive ablation study, we prove that the proposed design becomes the best fit for solving visual generalization with model-based RL.

2. Related Works

2.1. Visual Generalization in Deep RL

Learning a policy that outputs an action maximizing the expected cumulative return under different observation spaces between training and evaluation produces a unique challenge. Visual generalization refers to how an agent trained with visual input maximizes the cumulative return during evaluation where the input images for evaluation are visually augmented with perturbations (e.g., background color) and unseen during training. Previous approaches often incorporate model-free value-based algorithms with representation learning (Laskin et al., 2020b; Nair et al., 2022; Bertoin et al., 2022; Yuan et al., 2022; Wang et al., 2023; Yang et al., 2024), data augmentation (Lee et al., 2019; Laskin et al., 2020a; Hansen & Wang, 2021; Hansen et al., 2021; Huang et al., 2022), and stabilization of value learning (Hansen et al., 2021; Liu et al., 2023; Huang et al., 2024). Regarding data augmentation, enlarging the limited dataset with *weakly* augmented, i.e. random shift augmentation (Yarats et al., 2021a), visual data contributes to the significant sample-efficient RL with visual input (Laskin et al., 2020a; Yarats et al., 2021b;a), whereas employing a relatively *strong* augmentation, e.g. random convolution or overlay, improves generalization capability of the agent over unseen image inputs during training (Hansen & Wang, 2021; Hansen et al., 2021). Since jointly learning low-dimensional compact representation from a high-dimensional raw image while capturing optimal behavior from reward signal in an end-to-end manner usually necessitates a large quantity of dataset (Pari et al., 2021; Stooke et al., 2021), learning an encoder that can extract helpful information for downstream RL training from data plays a critical role in visual generalization. In this work, we address the visual generalization problem in RL similarly to previous works. However, we jointly focus on the sample efficiency problem during RL training in addition to generalization performance, where most prior works have been overlooked. We contend that considering the sample efficiency problem is as significant as the generalization performance since we are given only a limited set of training images according to problem formulation, which exacerbates when a pool of evaluation images increases.

2.2. Model-Based Reinforcement Learning

Expanding previous value-based RL methods (Sutton, 2018) with the deep neural network has enabled successful adoptions of conventional RL to challenging domains, including a high-dimensional state or continuous action space. However, a prerequisite of a huge bucket of experience replay to learn a well-performing policy becomes a primary bottleneck for RL practitioners (Yarats et al., 2021c). Model-based RL has been introduced as an alternative approach that trains a proxy of the environment transition model and exploits the learned model to generate synthetic data for further policy learning (Sutton, 1991; Deisenroth & Rasmussen, 2011), allowing the agent to simulate future states and plan the best action to maximize the expected return. Since the proxy model is trained via a limited pool of transitions, using the ensembles of the trained model (Buckman et al., 2018; Kurutach et al., 2018; Janner et al., 2019) alleviates the uncertainty arising from the imperfect model. Learning a world model that simulates future states usually from high-dimensional observations with a latent sequential transition model (Ha & Schmidhuber, 2018) demonstrates superior sample efficiency and downstream RL performance. Formally, learning a recurrent transition model while reconstructing future images with encoder-decoder structure (Hafner et al., 2019b;a; 2023) or combining the planning with model predictive controller without reconstructions (Hansen et al., 2022; 2023; Zhao et al., 2023) proves successful adoption to continuous control of more complicated domains. In this work, we choose TD-MPC2 (Hansen et al., 2023) as a backbone model-based RL architecture for visual generalization, which has demonstrated superior sample efficiency over another state-of-the-art architecture, DreamerV3 (Hafner et al., 2023). We provide further discussions concerning model-based RL in Appendix C.

3. Preliminaries

3.1. Problem Formulation

We design the visual generalization problem as the Partially Observable Markov Decision Problem (POMDP) (Bellman, 1957). POMDP is defined as a tuple $\langle \mathcal{S}, \mathcal{O}, \mathcal{A}, \mathcal{T}, r, \gamma \rangle$, where $\mathcal{S}$ is the state space, $\mathcal{O}$ is the observational space, $\mathcal{A}$ is the action space, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \mapsto \mathcal{S}$ is the transition dynamics probability, and $r : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ is the reward function. The agent receives not the state directly but the high-dimensional image from the observation space $\mathcal{O}$. Likewise in (Yarats et al., 2021b; Hansen et al., 2021), we define the state s_t as a stack of consequent images for simplicity, i.e. $s_t = \{o_t, o_{t-1}, o_{t_2}, \dots, o_{t-k+1}\}$ where $s_t \in \mathcal{S}$, $o_t \in \mathcal{O}$; t and k is the environment time-step and the number of image stacks, respectively. The goal of the agent is to find an optimal policy π^* that maximizes the cumulative expected return $\mathbb{E}_{a_t \sim \pi(\cdot|s_t)} \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)$ during evaluation where the agent receives perturbed images (e.g., background appearance).

3.2. Temporal Difference learning for MPC

Our method is built upon TD-MPC2 (Hansen et al., 2023), a scalable and robust model-based RL architecture that combines temporal difference learning (Sutton, 2018) for terminal Q value function with the model predictive control (MPC) for planning. TD-MPC2 is a latent space decoder-free world model that jointly learns parameters of the model: (i) a representation $z = h_\theta(s, e)$ by encoding a stack of high-dimensional inputs s into a low-dimensional representation z with an encoder h_θ, (ii) a latent dynamics model $z' = d_\theta(z, a, e)$ that predicts the next latent state z' given current latent state z and action a, (iii) a reward function $\hat{r} = R_\theta(z, a, e)$ that predicts the one-step reward, (iv) a Q value function $\hat{q} = Q_\theta(z, a, e)$ that predicts the state-action value function, and (v) a prior policy $\hat{a} \sim \pi_\theta(z, e)$ that is trained to maximize the Q value function Q_θ and used as a guiding policy for planning. z' and s' are the successor (latent) state while z and s are predecessor (latent) state, respectively. While the original TD-MPC2 can be used for both single-task and multi-task control-centric world models with the task embedding e, we employ TD-MPC2 as a *single-task visual* world model in this work due to prohibitively large budget for of computing cost. Hence, we will omit the notion of e for clarity.

During online training, the world model is trained via minimizing a weighted loss over the prediction horizon given the experience replay $\mathcal{B}$:

$$\begin{aligned}
&\mathcal{L}_{\text{TD-MPC2}}(\theta; \mathcal{L}_{rew}, \mathcal{L}_Q, \mathcal{L}_{dyn}, \mathcal{B}) \\
&= \mathbb{E}_{\Gamma \sim \mathcal{B}} \left[\sum_{i=t}^{t+H} \lambda^{i-t} \mathcal{L}_{\text{TD-MPC2}}(\theta; \mathcal{L}_{rew}, \mathcal{L}_Q, \mathcal{L}_{dyn}, \Gamma) \right] \\
&= \mathbb{E}_{\Gamma \sim \mathcal{B}} \left[\sum_{i=t}^{t+H} \lambda^{i-t} \Big(c_1 \mathcal{L}_{rew}(\theta; z_i, a_i, r_i) \right. \\
&\left. + c_2 \mathcal{L}_Q(\theta; z_i, a_i, r_i, \tilde{z}_{i+1}) + c_3 \mathcal{L}_{dyn}(\theta; z_i, a_i, z_{i+1}^{targ}) \Big) \right],
\end{aligned} \tag{1}$$

with each prediction loss:

$$\begin{aligned}
&\mathcal{L}_{rew}(\theta; z_t, a_t, r_t) = \text{CE}\left(R_\theta(z_t, a_t), r_t\right), \\
&\mathcal{L}_Q(\theta; z_t, a_t, r_t, z_{t+1}) \\
&= \text{CE}\left(Q_\theta(z_t, a_t), \texttt{sg}\Big(r_t + \gamma Q_{\theta^-}\left(z_{t+1}, \pi_\theta(z_{t+1})\right)\Big)\right), \\
&\mathcal{L}_{dyn}(\theta; z_t, a_t, z_{t+1}) = \|d_\theta(z_t, a_t) - \texttt{sg}\left(h_\theta(s_{t+1})\right)\|_2^2,
\end{aligned}$$

where a horizontal trajectory segment $\Gamma = (s_t, a_t, r_t, s_{t+1})_{t:t+H}$ with a horizon H is sampled from the replay buffer $\mathcal{B}$ and $\lambda \in \mathbb{R}^+$ is a constant decaying over the horizon to weight closer predictions higher. $\mathcal{L}_{rew}, \mathcal{L}_Q, \mathcal{L}_{dyn}$ are the reward, Q value, and latent transition dynamics prediction loss, respectively, and $c_i \in \mathbb{R}^+, i = 1, 2, 3$ are the coefficients balancing each loss. θ^- stands for exponentially moving average target parameters of online parameter θ, $\texttt{sg}$ is the *stop-grad* operator that prevents the computed gradient from influencing the remaining gradient computations. CE is the cross-entropy loss function that performs discrete regression tasks. Since we build our method upon TD-MPC2 without changing underlying planning (inference) for choosing an optimal action, we refer the reader to the original paper (Hansen et al., 2023) for further details on planning.

4. Method

In this section, we present ViGMO, a model-based RL method that empirically demonstrates strong generalization ability over unseen image input without sacrificing sample efficiency by employing verified recipes from the model-free RL realm. ViGMO adopts advanced techniques for improving visual generalization: (1) weak and strong data augmentations to given image input for sample efficiency and generalization, (2) consistent latent representation simulated by the latent transition dynamics, and (3) regulariza-

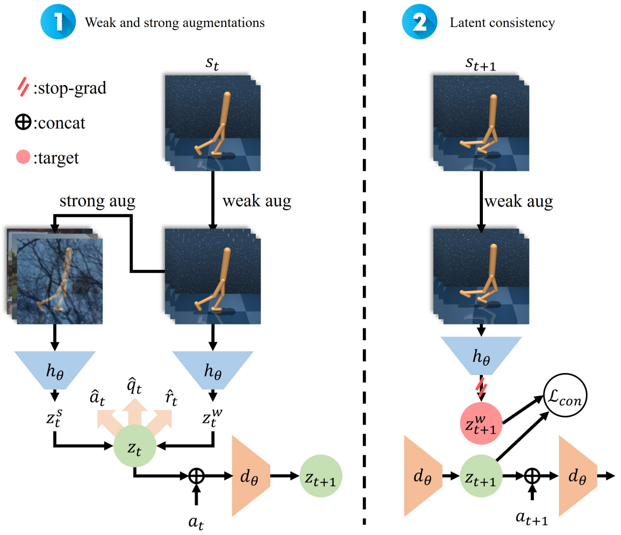


Figure 2. **An overview of the ViGMO architecture.** (**Left**) Weak and strong augmentations are implemented for sample efficiency and generalization. (**Center**) Weakly augmented representation guides the mixed horizontal representations as a target for consistent latent dynamics predictions. (**Right**) The encoder is regularized to extract consistent representation regardless of augmentations.

tion that allows the encoder extract consistent representations over differently augmented input. Fundamentally, our method is compatible with *any* model-based RL method that learns the latent transition dynamics with a visual feature extractor (the encoder) since we do not constrain any change of the underlying algorithm. In the following, we explain how ViGMO tackles the problem by leveraging the core components under the hood.

4.1. Architectural Overview

An overview of ViGMO can be found in Figure 2. We build our method on top of TD-MPC2, a sample-efficient model-based architecture, by fusing the world model learning with data augmentations and representation learning. We first employ data augmentations for latent world model learning by applying weak and strong augmentations to the original image, subsequently. Encoded representations from heterogeneous images are mixed into the latent representation for world model learning (e.g., reward and transition model). Since the world model is trained over the prediction horizon using recursive inference, we then regularize the latent dynamics and encoder to have consistent representations over the horizon regardless of augmentations. See Appendix B for comprehensive implications behind the idea.

4.2. Weak and Strong Augmentation

We refer to weak augmentation as employing a relatively minor change in an image (e.g., random shift transformation) and strong augmentation as applying a significant change in the image (e.g., random color convolution). Following prior works, we adopt *random-shift* (Yarats et al., 2021a) as weak and *random-overlay* (Hansen & Wang, 2021) as strong augmentation in this work. Consider a set of indices $\mathcal{I} = \{1, 2, \dots, B\}$ where B is the size of the batch. Let the indices of the batch be weakly and strongly augmented as $\mathcal{I}^w$ and $\mathcal{I}^s$, respectively. Then, the representations from weakly and strongly augmented images at time-step t, i.e., z_t^w and z_t^s, become:

$$\begin{aligned} z_t^w &= h_\theta(\tau^w(s_t^w, \upsilon^w)), \qquad s_t^w = \{s_{t,i} : i \in \mathcal{I}^w\}, \\ z_t^s &= h_\theta(\tau^s(s_t^s, \upsilon^s)), \qquad s_t^s = \{s_{t,j}^w : j \in \mathcal{I}^s\}, \end{aligned} \tag{2}$$

where $z_t = z_t^w \oplus z_t^s$ is the total representation at time-step t where $\oplus$ is element-wise concatenation, $\tau : \mathcal{S} \times \Upsilon \mapsto \mathcal{S}$ is a random augmentation function with a parameter $\upsilon \sim \Upsilon$, and $s_{t,n}$ corresponds to the state that is collected by choosing elements in s_t of an index n along the batch dimension. Superscripts w and s state weak and strong augmentation. $\mathcal{I}^w$ and $\mathcal{I}^s$ are subsets of $\mathcal{I}$ where subsets are complementary and disjoint subsets, i.e., $\mathcal{I}^w \sim \text{Uniform}(1, B), \mathcal{I}^s = \mathcal{I}/\mathcal{I}^w, |\mathcal{I}^w|/|\mathcal{I}^s| = \zeta \in \mathbb{R}$. Through all experiments, we set $\zeta = 1.0$: divide the original batch in half for weak and strong augmentation. Although the representation z_t is recursively used for world model learning over the horizon, we apply these augmentations only at time-step t.

4.3. Consistency on Latent Transition

To enable sample-efficient model learning without sacrificing generalization performance, we constrain the latent transition dynamics model to have consistency toward weakly augmented representation z_t^w. After the initial representation z_t is encoded with weak and strong augmentation in the previous section, the latent transition dynamics model predicts the successor latent representation z_{t+1} given predecessor z_t and action a_t in Equation 1. The parameters of the latent transition dynamics model are updated by solving a regression problem: $\mathcal{L}(\theta; d_\theta) = \text{MSE}(d_\theta(z_t, a_t), z_{t+1})$ where $z_{t+1} = \text{sg}(h_\theta(s_{t+1}))$. We implement the weak augmentation, i.e., *random-shift*, to the target images to generate consistent target representation:

$$\mathcal{L}_{con}(\theta; z_t, a_t, z_{t+1}^w) = \|d_\theta(z_t, a_t) - \texttt{sg}(z_{t+1}^w)\|_2^2,$$

where $z_{t+1}^w = h_\theta(s_{t+1}^w)$ is the representation extracted from a weakly augmented state s_t^w in Equation 2.

4.4. Regularization over Augmentation

So far, the latent transition model and other components of the world model are constrained to predict consistent outputs regardless of data augmentations. Additionally, following (Hansen & Wang, 2021), we implement the weak and strong augmentation to the state s_t to generate two different views of the original image for regularization. Subsequently, we train the encoder h_θ to extract applied strong augmentation in the weakly augmented image by minimizing regularization loss:

$$\mathcal{L}_{reg}(\theta; z_t^w, z_t^s) = \left\| \frac{z_t^s}{\|z_t^s\|_2} - \frac{z_t^w}{\|z_t^w\|_2} \right\|_2^2,$$

where $z_t^w = h_\theta(s_t^w)$ and $z_t^s = h_\theta(s_t^s)$ are the representations extracted from the weak and strong-augmented state in Equation 2.

Aggregated learning objectives. We incorporate three key components to the world model learning procedure of TD-MPC2 over the prediction horizon in Equation 1 as follows:

$$\begin{aligned} &\mathcal{L}_{\text{ViGMO}}(\theta; \mathcal{B}, \Upsilon^w, \Upsilon^s) \\ &= \mathbb{E}_{\Gamma\sim\mathcal{B}}\left[\mathbb{E}_{\upsilon^w,\upsilon^s}\left[\sum_{i=t}^{t+H} \mathcal{L}_{\text{ViGMO}}(\theta; \Gamma, \upsilon^w, \upsilon^s)\right]\right] \\ &= \mathbb{E}_{\Gamma\sim\mathcal{B}}\left[\mathbb{E}_{\upsilon^w,\upsilon^s}\left[\sum_{i=t}^{t+H} \lambda^{i-t}\mathcal{L}_{\text{TD-MPC2}}(\theta; \mathcal{L}_{rew}, \mathcal{L}_Q, \mathcal{L}_{con}, \Gamma) + \alpha\mathcal{L}_{reg}(\theta; z_i^w, z_i^s)\right]\right], \end{aligned} \tag{3}$$

where $\mathcal{B}$ is the experience replay and α is a coefficient that balances the gradients of world model learning and regularization. $\upsilon^w \sim \Upsilon^w$ and $\upsilon^s \sim \Upsilon^s$ are sampled augmentation parameters from the distribution of weak and strong random augmentations, respectively. In principle, our method can be injected into *any* model-based RL method that trains the latent transition dynamics model; we highlight the difference of world model learning with TD-MPC2 as red color in Equation 3. We summarize our method in Algorithm 1. We exclude the learning process of the prior policy π_θ in Algorithm 1 since we employ the same procedure in TD-MPC2. We provide additional details regarding implementing ViGMO and hyperparameters in Appendix A.

Algorithm 1 World model learning in ViGMO

```
**Input:** Horizontal replay buffer $\mathcal{B}$; Horizon $H$; Weak and strong augmentation functions $\tau^w, \tau^s$; Network update rates $\eta, \delta$; Regularization coeff. $\alpha$
while not converged do
  for gradient-step $t_g = 1, 2, \ldots, T_g$ per episode do
    // Sample horizontal transitions
    $(s_t, a_t, r_t, s_{t+1})_{t:t+H} \sim \mathcal{B}$
    $L \leftarrow 0$                                  // Initialize cumulative loss
    for $i = t, t+1, \ldots, t+H$ do
      $\upsilon^w \sim \Upsilon^w, \upsilon^s \sim \Upsilon^s$   // Sample aug. parameters
      $L \leftarrow L + \mathcal{L}_{\text{ViGMO}}(\theta; \Gamma, \upsilon^w, \upsilon^s)$   // Eqn. 3
    end for
    $\theta \leftarrow \theta + \eta\frac{1}{H}\nabla_\theta L$        // Update online parameters
    $\theta^- \leftarrow (1-\delta)\theta^- + \delta\theta$  // Update target parameters
  end for
end while
```

5. Experiments

In this section, we provide extensive empirical observation of ViGMO on diverse benchmarks with sophisticated experiment designs. We evaluate the generalization performance and sample efficiency with other baselines in relevant fields. We address the following questions through experiments: (i) how ViGMO compares with other competitive baselines in visual generalization and sample efficiency, (ii) how our design choice affects the performance of ViGMO, and (iii) how predicting consistent representation over the horizon impacts on visual generalization. We present our implementation details concerning the generalization benchmark and analyze the performance in the following.

5.1. Experimental Setup

In this section, we provide experimental settings concerning the environments and baselines for competitive performance comparison.

Environments. We evaluate ViGMO over 6 tasks in the DeepMind control suite (DMC, (Tassa et al., 2018)) and 2 tasks in robosuite (Zhu et al., 2020): *cartpole_swingup*, *finger_spin*, *walker_walk*, *walker_stand*, *cheetah_run*, and *reacher_easy* in DMC; *Door* and *Lift* in robosuite. We illustrate the tasks and environments in Figure 3. We train agents for each task with 1M gradient steps and evaluate the trained agents for 5 seeds. See further details regarding the environment and task setup in Appendix A.

Baselines. We select state-of-the-art baselines in visual generalization problems to compare ViGMO. Specifically, in model-free RL, SVEA (Hansen et al., 2021) stabilizes off-policy Q-learning with data augmentation, SGQN (Bertoin et al., 2022) adapts self-supervised learning with attribution map and regularizes Q value learning, SRM (Huang et al., 2022) applies a spectrum augmentation to increase robustness toward spatial corruption, and PIEG (Yuan et al., 2022) plugs large CNN pretrained with ImageNet for consistent representation. To compare the performance of ViGMO with backbone model-based RL, we also evaluate the performance of TD-MPC2. Regarding implementation details of baselines, see Appendix A.

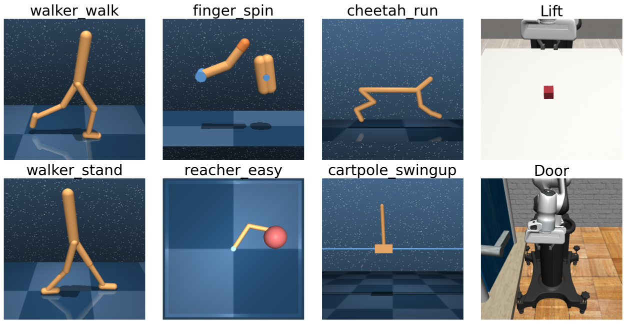


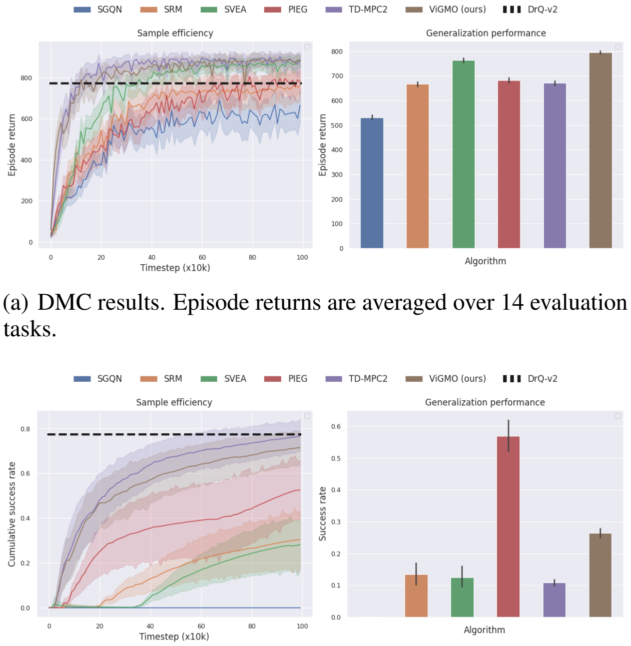
Figure 3. **DeepMind Control Suite and Robosuite tasks.** Locomotion and manipulation continuous control tasks for visual generalization. We address a set of diverse visual generalization tasks for each environment.

5.2. Results

In this section, we provide rigorous results concerning each experiment we have designed in Section 5. We first sketch the background of each experimental setup and then analyze empirical results in the following.

5.2.1. Visual Generalization and Sample Efficiency

We provide aggregated performance comparison results in Figure 4; Table 1 and 2. ViGMO proves superior sample efficiency over model-free RLs and preserves similar sample efficiency compared to the backbone model-based RL, TD-MPC2, which has demonstrated state-of-the-art sample efficiency. In addition, ViGMO outperforms other baselines in DMC and demonstrates remarkable generalization performance in robosuite experiments. It is worth noting that ViGMO outperforms its backbone model, TD-MPC2, in generalization performance with a trivial sacrifice of sample efficiency. Considering that ViGMO does not enforce any algorithmic modifications in model learning and planning with the model, the significant margin of generalization performance supports the validity of the proposed method to alleviate the out-of-distribution shift problem in visual generalization. See discussions regarding experimental results in Appendix D.2.



(a) DMC results. Episode returns are averaged over 14 evaluation tasks.

(b) Robosuite results. Success rates are averaged over 3 evaluation tasks.

Figure 4. **Graphical results.** (**Left**) Sample efficiency and (**Right**) generalization performance. Episode return and cumulative success rate for evaluations during training are reported in DMC and robosuite, respectively. Evaluation during training is averaged over 10 episodes. The dashed line is an oracle for comparing sample efficiency using DrQ-v2.

5.2.2. Ablation of Design Choices

We investigate several possible design choices for improving the generalization performance of model-based RL. Toward this objective, we examine the variants of ViGMO: the randomness of augmentations over the horizon, different strong augmentation, and another auxiliary task for representation learning. In the following, we describe how we compare the possible candidates of ViGMO and then we provide a summarized result in Figure 5. See discussions concerning ablations in Appendix D.3

Dynamic and consistent augmentations. Since typical model-based RL predicts synthetic future transition samples over the horizon, how we should control the data augmentations over the horizon is still questionable; it might be unclear whether applying **dynamic** or **consistent** augmentation over the horizon benefits the generalization performance. ViGMO augments the states with **both** weak and strong augmentation over the horizon to constrain latent consistency (Section 4.3) and regularization (Section 4.4). The horizontal states $s_{t:t+H}$ from time-step t to $t+H$ are augmented as following:

$$s^w_{t:t+H} = \{\tau^w_k(s_k, \upsilon^w_k) : \upsilon^w_k \sim \Upsilon^w, k \in \{t, t+1, \dots, t+H\}\}$$
$$s^s_{t:t+H} = \{\tau^s_k(s^w_k, \upsilon^s_k) : \upsilon^s_k \sim \Upsilon^s, k \in \{t, t+1, \dots, t+H\}\},$$

Table 1. **Quantitative comparison for sample efficiency.** Inspired by (Mai et al., 2022), Sample efficiency of competitive baselines with the same oracle, DrQ-v2. Scores are the average number of episodes necessary for achieving 25%, 50%, and 75% percentiles of the oracle score. Scores of the backbone architecture, TD-MPC2, are provided to contrast the sample efficiency difference of ViGMO. See further details in Appendix D.1.

Env	Pctl.	SVEA	SGQN	SRM	PIEG	ViGMO (ours)	TD-MPC2
DMC	25%	100	313	90	235	**65**	43
	50%	211	353	**190**	521	225	85
	75%	360	556	315	543	**298**	215
robosuite	25%	1380	2000	1080	1070	**240**	190
	50%	1600	2000	1500	1510	**480**	440
	75%	1840	2000	1720	1820	**1030**	770

Table 2. **Quantitative comparison for visual generalization.** Visual generalization performance comparison over different environments and tasks. Scores of the backbone architecture, TD-MPC2, are provided to contrast the visual generalization difference of ViGMO.

Env	Task	SVEA	SGQN	SRM	PIEG	ViGMO (ours)	TD-MPC2
DMC	cartpole_swingup	**819.67**	635.06	816.51	655.53	667.91	705.41
	finger_spin	814.97	760.97	814.93	780.77	**886.71**	775.93
	walker_walk	767.19	471.38	**886.38**	880.76	871.15	706.39
	walker_stand	**947.18**	876.18	142.16	937.51	912.48	778.31
	cheetah_run	435.99	226.14	**502.99**	249.32	479.59	323.59
	reacher_easy	801.57	217.68	834.44	586.87	**949.89**	731.58
	Avg.	764.43	531.24	666.23	681.79	**794.62**	670.20
robosuite	Door	0.00	0.00	0.01	**0.92**	0.38	0.06
	Lift	0.25	0.00	**0.27**	0.23	0.15	0.16
	Avg.	0.13	0.00	0.14	**0.57**	0.26	0.11

where the random augmentation function τ may depend on time-step or not. We refer to τ_t as consistent augmentation over the horizon if $\tau_t = \tau_{t+k} \forall k \in [0, H]$ is satisfied. If the augmentation function differs over the horizon, i.e., $\tau_t \neq \tau_{t+k}, \forall k \in [0, H]$, we denote it as a dynamic augmentation function. ViGMO adopts the **dynamic augmentation** for both weak and strong augmentations, i.e., $\tau_t \neq \tau_{t+k}, \forall k \in [0, H]$. We denote ViGMO with consistent augmentation as ***ViGMO_CONST_AUG*** later in experiments.

Different strong augmentation. Motivated by prior observations, we choose *random-overlay* as a strong augmentation for visual generalization in model-based RL. However, as proposed in (Lee et al., 2019; Laskin et al., 2020a; Hansen et al., 2021), other strong augmentations can become strong candidates for visual generalization in Deep RL. Among potential augmentations, we opt *random-conv* as another strong augmentation method, which has exhibited remarkable performance in previous works. Precisely, we refer *random-overlay* and *random-conv* augmentation as:

$$\tau^{s,\text{overlay}}(o, \tilde{o}) = (1 - \delta)o + \delta\tilde{o}$$
$$\tau^{s,\text{conv}}(o, w) = \texttt{CONV}(o, w)$$

where $o \in \mathcal{O}^{B\times C\times H\times W}$ is the original images with the batch size B, channel C, height H, and width W, respectively. δ is a linear interpolation coefficient, $\tilde{o} \sim \mathcal{D}$ is an overlaying image sampled from a dataset unrelated to the task. We set the default value of δ as 0.5. *CONV* stands for 2-dimensional convolution operation over the batch dimension and $w \in \mathbb{R}^{N\times C_k\times H_k\times W_k} : w \sim \mathcal{N}(0, 1)$ is the convolution filter randomly initialized from the normal distribution and kernel number N, channel C_k, height H_k, and width W_k, respectively. We implement the same strong augmentation over the channel dimension to obtain the state $s_t = \{o_t, o_{t-1}, \cdots, o_{t-k+1}\}$. While ViGMO chooses *random-overlay* as strong augmentation, we denote ***ViGMO_CONV*** as ViGMO replaced with *random-conv*.

Contrastive learning for the representation learning task. Learning task-specific representation for downstream vision-based RL plays a critical role in sample efficiency and generalization performance. Prior literature (He et al., 2020; Laskin et al., 2020b; Nair et al., 2022; Bertoin et al., 2022; Yuan et al., 2022) has explored the field by leveraging popular representation learning techniques in computer vision. As in Section 4.4, we intend the encoder to predict robust representation regardless of distracting components (e.g., background image). Hence, we consider two variants of auxiliary representation learning task: *SODA* (Hansen & Wang, 2021) and *CURL* (Laskin et al., 2020b) where we use the same auxiliary task from SODA for representation learning in Section 4.4:

$$\mathcal{L}_{CURL}(\theta; q, k) = \left(\log \frac{\exp q^T W k_+}{\exp q^T W k_+ + \sum_{j=0}^{B} \exp q^T W k_j}\right),$$

where $q = h_\theta(\tau^w(s_t, \upsilon^q)), k = h_{\theta^-}(\tau^w(s_t, \upsilon^k))$ are anchor and key; W is the weight kernel for bilinear product. υ^q and υ^k are separately sampled parameters for generating anchor and key images from the same distribution Υ^w. We denote ***ViGMO_CURL*** as ViGMO replacing the regularization loss (i.e., $\mathcal{L}_{\text{reg}} = \mathcal{L}_{\text{SODA}}$) with $\mathcal{L}_{\text{CURL}}$.

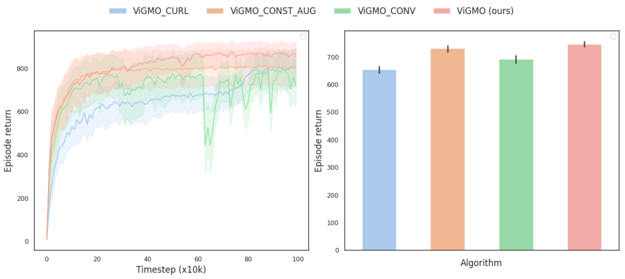


Figure 5. **Ablation of design choices.** (**Left**) Sample efficiency and (**Right**) Generalization performance. Among the options, ViGMO proves to be superior in experiments.

Aggreagated results. We provide the aggregated ablation results in Figure 5. We compare ViGMO with *ViGMO_CONST_AUG*, *ViGMO_CONV*, and *ViGMO_CURL* in *finger_spin*, *cheetah_run* and *walker_walk* tasks. ViGMO proves superior sample efficiency and generalization performance compared to other options, validating that the proposed method is the most reasonable choice among available options. See further experimental details regarding this comparison in Appendix D.3.

5.2.3. Visualized Consistent Representation

To reduce the prediction error of the learned world model for visual generalization, we propose a novel approach that constrains the world model learning with data augmentation and regularization. Hence, we experiment to validate the motivation that ViGMO would demonstrate consistent prediction ability regardless of perturbations while TD-MPC2 fails. We first collect the horizontal replay transitions by rolling out the trained model and then feed the same inputs (i.g., $(s_t, a_t, s_{t+1})_{t:t+H}$) to ViGMO and TD-MPC2 to make them predict the latent transitions. Since the representation often has more than two dimensions, we visualize the collected representations from the latent transition model d_θ with UMAP (McInnes et al., 2018) in Figure 6, a popular manifold learning method to extract two-dimensional coordinates from high-dimensional representations.

In Figure 6, faded embedding trajectories are the previous time-step embeddings predicted by d_θ. While TD-MPC2 struggles to output consistent representation over the horizon (i.e., largely error between *original* and *background_color* types), ViGMO demonstrates unvarying and aligned representation between the latent dynamics predictions, $z_{t+1} = d_\theta(z_t, a_t)$, across generalization types. These empirical observations support the idea in Figure 1 that projecting out-of-distributional representations to in-domain training latent distribution is important to improve generalization performance. See additional discussions and results in Appendix D.4.

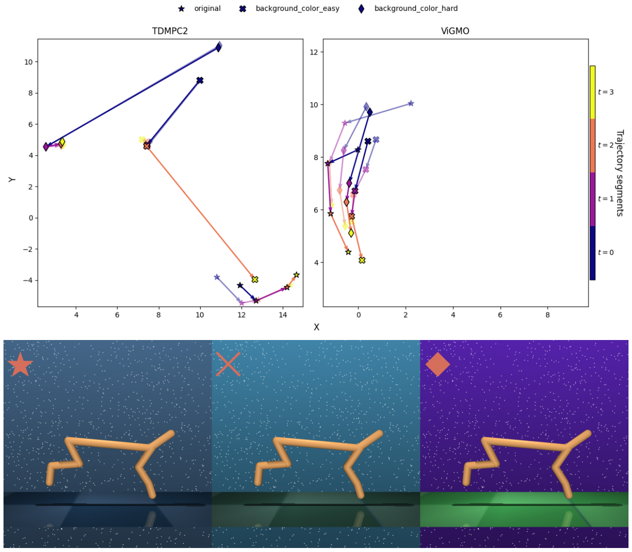


Figure 6. **Visualization of embeddings.** (**Top**) ViGMO demonstrates consistent representation predicted by d_θ over the horizon and types of evaluation. Faded trajectories are previous time-step embeddings. (**Bottom**) Illustration of generalization types used for extracting representation (*original*($\star$), *background_color_easy*($\times$), and *background_color_hard*($\diamond$) from left-side).

6. Conclusion

We propose ViGMO, a model-based RL that empirically demonstrates strong generalization ability over unseen image observations without sacrificing sample efficiency by employing recipes from model-free RL in visual generalization. By constraining the world model to predict consistent representation with data augmentation and representation learning, ViGMO successfully solves the visual generalization problem under diverse visual generalization benchmarks. We provide extensive results for comparing the performance between competitive baselines including model-free RL. Nevertheless, model-based RL should overcome a few setbacks in visual generalization. First, the performance gain of ViGMO is still limited to empirical observations. While it is notable that ViGMO empirically proves superior visual generalization, extending prior works on theoretical foundations, e.g., (Ghugare et al., 2022; Lyu et al., 2024), to visual model-based RL would be interesting future work. Additionally, the pool of generalization for model-based RL is limited to a narrow distribution of generalization tasks. Incorporating further generalizations (Seo et al., 2020; Beukman et al., 2024) into visual generalization on model-based RL agents would be a thrilling future work for the generalization community.

Impact Statement

This paper investigates visual generalization with model-based RL, which demonstrates strong performance in continuous control tasks in recent years. Improving the generalization of deep RL is a persistent challenge for deep RL practitioners to employ the RL agent in the real world. However, it is important to note that although this paper presents strong empirical results in diverse environments, the practical usages of visual RL agents are still limited to simulated environments due to domain differences between the real world. Hence, we do not expect any potential societal consequences of our work in the short term.

References

Bellman, R. A markovian decision process. *Journal of mathematics and mechanics*, pp. 679–684, 1957.

Bertoin, D., Zouitine, A., Zouitine, M., and Rachelson, E. Look where you look! saliency-guided q-networks for generalization in visual reinforcement learning. *Advances in Neural Information Processing Systems*, 35:30693–30706, 2022.

Beukman, M., Jarvis, D., Klein, R., James, S., and Rosman, B. Dynamics generalisation in reinforcement learning via adaptive context-aware policies. *Advances in Neural Information Processing Systems*, 36, 2024.

Botvinick, M., Ritter, S., Wang, J. X., Kurth-Nelson, Z., Blundell, C., and Hassabis, D. Reinforcement learning, fast and slow. *Trends in cognitive sciences*, 23:408–422, 2019.

Buckman, J., Hafner, D., Tucker, G., Brevdo, E., and Lee, H. Sample-efficient reinforcement learning with stochastic ensemble value expansion. *Advances in neural information processing systems*, 31, 2018.

Deisenroth, M. and Rasmussen, C. E. Pilco: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on machine learning (ICML-11)*, pp. 465–472, 2011.

Fujimoto, S., Hoof, H., and Meger, D. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pp. 1587–1596. PMLR, 2018.

Ghugare, R., Bharadhwaj, H., Eysenbach, B., Levine, S., and Salakhutdinov, R. Simplifying model-based rl: learning representations, latent-space models, and policies with one objective. *arXiv preprint arXiv:2209.08466*, 2022.

Ha, D. and Schmidhuber, J. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. PMLR, 2018.

Hafner, D., Lillicrap, T., Ba, J., and Norouzi, M. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019a.

Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H., and Davidson, J. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pp. 2555–2565. PMLR, 2019b.

Hafner, D., Pasukonis, J., Ba, J., and Lillicrap, T. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Hansen, N. and Wang, X. Generalization in reinforcement learning by soft data augmentation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 13611–13617. IEEE, 2021.

Hansen, N., Su, H., and Wang, X. Stabilizing deep q-learning with convnets and vision transformers under data augmentation. *Advances in neural information processing systems*, 34:3680–3693, 2021.

Hansen, N., Wang, X., and Su, H. Temporal difference learning for model predictive control. *arXiv preprint arXiv:2203.04955*, 2022.

Hansen, N., Su, H., and Wang, X. Td-mpc2: Scalable, robust world models for continuous control. *arXiv preprint arXiv:2310.16828*, 2023.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

Hessel, M., Modayil, J., Van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M., and Silver, D. Rainbow: Combining improvements in deep reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Huang, S., Sun, Y., Hu, J., Guo, S., Chen, H., Chang, Y., Sun, L., and Yang, B. Learning generalizable agents via saliency-guided features decorrelation. *Advances in Neural Information Processing Systems*, 36, 2024.

Huang, Y., Peng, P., Zhao, Y., Chen, G., and Tian, Y. Spectrum random masking for generalization in image-based reinforcement learning. *Advances in Neural Information Processing Systems*, 35:20393–20406, 2022.

Janner, M., Fu, J., Zhang, M., and Levine, S. When to trust your model: Model-based policy optimization. *Advances in neural information processing systems*, 32, 2019.

Kurutach, T., Clavera, I., Duan, Y., Tamar, A., and Abbeel, P. Model-ensemble trust-region policy optimization. *arXiv preprint arXiv:1802.10592*, 2018.

Laskin, M., Lee, K., Stooke, A., Pinto, L., Abbeel, P., and Srinivas, A. Reinforcement learning with augmented data. *Advances in neural information processing systems*, 33: 19884–19895, 2020a.

Laskin, M., Srinivas, A., and Abbeel, P. Curl: Contrastive unsupervised representations for reinforcement learning. In *International conference on machine learning*, pp. 5639–5650. PMLR, 2020b.

Lee, K., Lee, K., Shin, J., and Lee, H. Network randomization: A simple technique for generalization in deep reinforcement learning. *arXiv preprint arXiv:1910.05396*, 2019.

Lillicrap, T. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Liu, S., Chen, Z., Liu, Y., Wang, Y., Yang, D., Zhao, Z., Zhou, Z., Yi, X., Li, W., Zhang, W., et al. Improving generalization in visual reinforcement learning via conflict-aware gradient agreement augmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 23436–23446, 2023.

Lyu, J., Wan, L., Li, X., and Lu, Z. Understanding what affects generalization gap in visual reinforcement learning: Theory and empirical evidence. *arXiv preprint arXiv:2402.02701*, 2024.

Mai, V., Mani, K., and Paull, L. Sample efficient deep reinforcement learning via uncertainty estimation. *arXiv preprint arXiv:2201.01666*, 2022.

McInnes, L., Healy, J., and Melville, J. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Mnih, V. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Nair, S., Rajeswaran, A., Kumar, V., Finn, C., and Gupta, A. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022.

Pari, J., Shafiullah, N. M., Arunachalam, S. P., and Pinto, L. The surprising effectiveness of representation learning for visual imitation. *arXiv preprint arXiv:2112.01511*, 2021.

Schulman, J. Trust region policy optimization. *arXiv preprint arXiv:1502.05477*, 2015.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Seo, Y., Lee, K., Clavera Gilaberte, I., Kurutach, T., Shin, J., and Abbeel, P. Trajectory-wise multiple choice learning for dynamics generalization in reinforcement learning. *Advances in Neural Information Processing Systems*, 33: 12968–12979, 2020.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

Stone, A., Ramirez, O., Konolige, K., and Jonschkowski, R. The distracting control suite–a challenging benchmark for reinforcement learning from pixels. *arXiv preprint arXiv:2101.02722*, 2021.

Stooke, A., Lee, K., Abbeel, P., and Laskin, M. Decoupling representation learning from reinforcement learning. In *International conference on machine learning*, pp. 9870–9879. PMLR, 2021.

Sutton, R. S. Dyna, an integrated architecture for learning, planning, and reacting. *ACM Sigart Bulletin*, 2(4):160–163, 1991.

Sutton, R. S. Reinforcement learning: An introduction. *A Bradford Book*, 2018.

Tassa, Y., Doron, Y., Muldal, A., Erez, T., Li, Y., Casas, D. d. L., Budden, D., Abdolmaleki, A., Merel, J., Lefrancq, A., et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

Van Hasselt, H., Guez, A., and Silver, D. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

Wang, Z., Schaul, T., Hessel, M., Hasselt, H., Lanctot, M., and Freitas, N. Dueling network architectures for deep reinforcement learning. In *International conference on machine learning*, pp. 1995–2003. PMLR, 2016.

Wang, Z., Ze, Y., Sun, Y., Yuan, Z., and Xu, H. Generalizable visual reinforcement learning with segment anything model. *arXiv preprint arXiv:2312.17116*, 2023.

Xu, G., Zheng, R., Liang, Y., Wang, X., Yuan, Z., Ji, T., Luo, Y., Liu, X., Yuan, J., Hua, P., et al. Drm: Mastering visual reinforcement learning through dormant ratio minimization. *arXiv preprint arXiv:2310.19668*, 2023.

Yang, S., Ze, Y., and Xu, H. Movie: Visual model-based policy adaptation for view generalization. *Advances in Neural Information Processing Systems*, 36, 2024.

Yarats, D., Fergus, R., Lazaric, A., and Pinto, L. Mastering visual continuous control: Improved data-augmented reinforcement learning. *arXiv preprint arXiv:2107.09645*, 2021a.

Yarats, D., Kostrikov, I., and Fergus, R. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. In *International conference on learning representations*, 2021b.

Yarats, D., Zhang, A., Kostrikov, I., Amos, B., Pineau, J., and Fergus, R. Improving sample efficiency in model-free reinforcement learning from images. In *Proceedings of the aaai conference on artificial intelligence*, volume 35, pp. 10674–10681, 2021c.

Yuan, Z., Xue, Z., Yuan, B., Wang, X., Wu, Y., Gao, Y., and Xu, H. Pre-trained image encoder for generalizable visual reinforcement learning. *Advances in Neural Information Processing Systems*, 35:13022–13037, 2022.

Yuan, Z., Yang, S., Hua, P., Chang, C., Hu, K., and Xu, H. Rl-vigen: A reinforcement learning benchmark for visual generalization. *Advances in Neural Information Processing Systems*, 36, 2024.

Zhao, Y., Zhao, W., Boney, R., Kannala, J., and Pajarinen, J. Simplified temporal consistency reinforcement learning. In *International Conference on Machine Learning*, pp. 42227–42246. PMLR, 2023.

Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., and Torralba, A. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017.

Zhu, Y., Wong, J., Mandlekar, A., Martín-Martín, R., Joshi, A., Nasiriany, S., and Zhu, Y. robosuite: A modular simulation framework and benchmark for robot learning. *arXiv preprint arXiv:2009.12293*, 2020.

A. Implementation Details

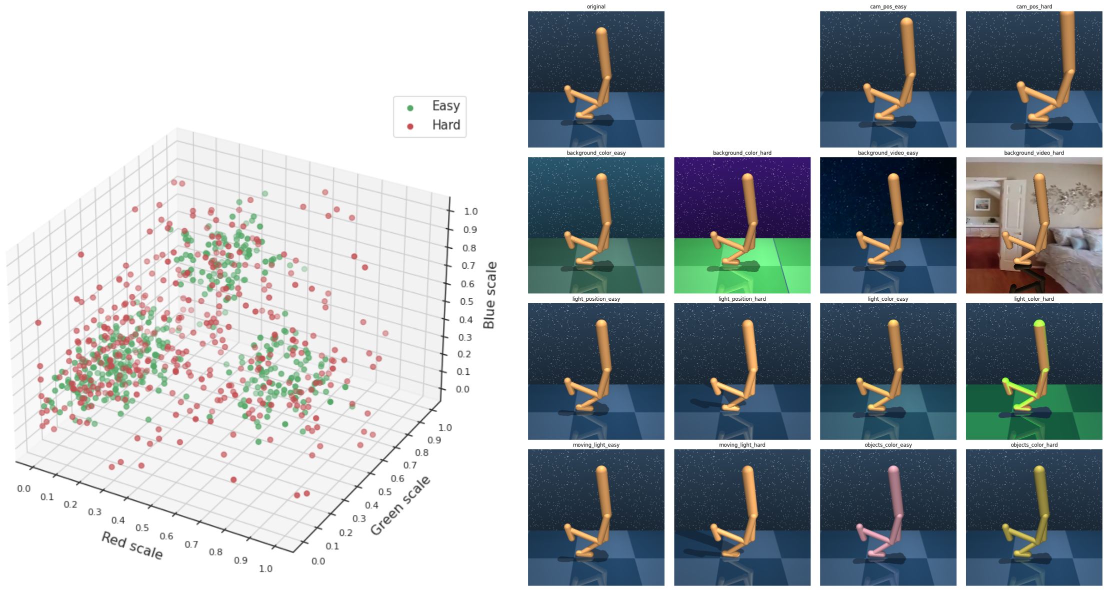
In this section, we describe the implementation details concerning the environments and baselines. We bring overall source code for model-free RL baselines from RL-ViGen (Yuan et al., 2024)[1]. We thank the authors of RL-ViGen for providing comprehensive source code.

Environment. We consider 14 evaluation types in DMC and 3 evaluation types in robosuite. In DMC, we account for the *type* and *difficulty* of the tasks for visual generalization performance evaluation. By following (Yuan et al., 2024; Hansen & Wang, 2021), we propose a unique set of visual generalization categories:

- *background_color*: Change the background color of the agent (e.g., terrain grid or background sky color).
  - Uniformly sample the parameters of the color, i.e., *(r,g,b)*, from the *easy* and *hard* distribution in Figure 7(a) for difficulties.
- *cam_pos*: Change the position of the tracking camera's focus by randomly adding noise offset.
  - Let the initial position of the tracking camera's focus be $X_{\text{cam}} = (x_i, y_i, z_i)$ in Euclid space.
  - Sample a random offset δ from the uniform distribution with different bounds: $\mathcal{U}(-0.08, 0.08)$ for *easy* and $\mathcal{U}(-0.15, 0.15)$ for *hard* difficulty.
  - Inject the offset to the initial position of the camera; $X_{\text{cam}} = (x_i + \delta, y_i + \delta, z_i + \delta)$.
- *background_video*: Overlay the background with the randomly sampled natural video.
  - Sample a random video with the same width and height as the original image from a set of natural videos (Stone et al., 2021).
  - Overlay the video only to the background sky for *easy* and to all backgrounds including the ground terrain other than the agent for *hard* difficulty.
- *light_position*: Change the position and orientation of the tracking light of the agent.
  - Following the approach used in (Stone et al., 2021), the tracking light's coordinate is parameterized as the spherical coordinate; (ϕ, θ, r) where ϕ is azimuth, θ is inclination, and r is the radius of the sphere.
  - Sample ϕ from the normal distribution $\mathcal{N}(\pi/6, 1)$ for *easy* and $\mathcal{N}(\pi/3, 1)$ for *hard* difficulty.
  - Sample $\theta \sim \mathcal{N}(2\pi, 1)$ and transform the initial pose of the tracking light X_{light} to (ϕ, θ, r) where $r = \sqrt{X_{\text{light}}}$.
- *light_color*: Change the color of the tracking light of the agent.
  - Uniformly sample the parameters of the color, i.e., *(r,g,b)*, from the *easy* and *hard* distribution in Figure 7(a) for difficulties.
- *moving_light*: Rotate the tracking light of the agent around the agent.
  - Likewise in *light_position*, the spherical coordinate of the tracking light is randomly initialized as (ϕ, θ, r).
  - Let the speed of azimuth rotation as $\Delta_\phi = \pi/200$ for *easy* and $\Delta_\phi = \pi/100$ for *hard* difficulty.
  - Rotate the tracking light counterclockwise along the azimuth axis at every time-step; $(\phi, \theta, r) \leftarrow (\phi, \theta, r) + (\Delta_\phi, 0, 0)$.
- *object_color*: Change the color of the body color of the agent.
  - Uniformly sample the parameters of the color, i.e., *(r,g,b)*, from the *easy* and *hard* distribution in Figure 7(a) for difficulties.

where *easy* and *hard* difficulties exist for 7 types of evaluation. In robosuite, we consider 3 types of evaluation with the Franka panda manipulator: *eval-easy*, *eval-hard*, and *eval-extreme*, which is predefined by (Yuan et al., 2024). We illustrate example images of DMC in Figure 7(b) and robosuite in Figure 8.

[1] https://github.com/gemcollector/RL-ViGen



(a) Visualized parameter distribution for evaluation.

(b) Visualized examples of evaluation sets.

Figure 7. **Evaluation set in DMC.** (**Left**) Uniformly sampled color parameters from predefined space for *easy* and *hard* difficulties in DMC. Scaled color values are unnormalized for each color RGB space. (**Right**) 7 types and 2 difficulties for generalization performance evaluation in DMC, *walker_walk* task

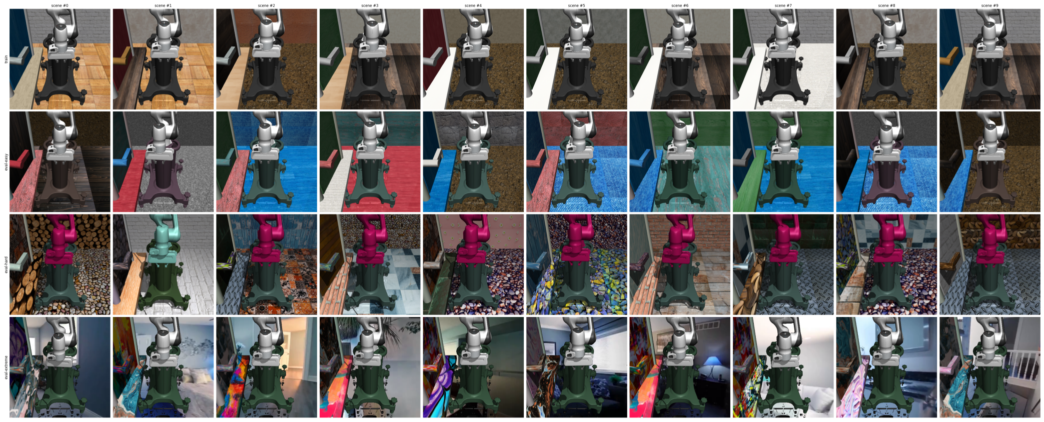


Figure 8. **Evaluation set in robosuite.** 3 types of generalization evaluation in robosuite, *Door* task. Each row stands for *train*, *eval-easy*, *eval-hard*, and *eval-extreme*; each column corresponds to a different scene in the same type.

Baselines. We consider 4 competitive baselines for model-free, SVEA (Hansen et al., 2021), SGQN (Bertoin et al., 2022), SRM (Huang et al., 2022), and PIEG (Yuan et al., 2022), and for model-based, TD-MPC2 (Hansen et al., 2023), algorithms for comparison. We bring the source code of the model-free baselines from the RL-ViGen benchmark since the implementation is straightforward. We implement ViGMO upon the official repository of TD-MPC2 (Hansen et al., 2023)[2]. We intentionally consider the same hyperparameters and task-specific parameters (e.g., action repeat) as many as possible across experiments. We list common parameters used across tasks in Table 3 and algorithm-specific hyperparameters in Table 5 and 4. We remark that we pursue unified task-specific parameters for reproducible results and concise comparison of baselines.

[2] https://github.com/nicklashansen/tdmpc2

Table 3. **Common hyperparameters.** Commonly used task-specific parameters for **all** baselines including ViGMO across experiments (DMC and robosuite).

Parameter	Value
Discount factor	0.99
Replay buffer size	Unlimited (same with T_g)
Action repeats	2
Frame stack k	3
Pixel RGB image space	$o_t \in \mathcal{O}^{84\times 84\times 3}$
Maximum episode length	1,000 (DMC), 500 (robosuite)
Batch size	512 (*walker_{walk,stand}* tasks), 256 (otherwise)
Total gradient steps T_g	1,000,000
Total seeding steps	2,500, 1,250 (model-based; DMC and robosuite), 4,000 (model-free)
Periodic evaluation steps during training	10,000
Number of episodes per evaluation during training	10
N steps for TD target	1 (model-based), 3 (model-free)
MLP hidden layer dimension	512 (model-based), 1024 (model-free)
Number of CNN convolution filters	32
Latent dimension	512 (model-based), 50 (model-free)
Activation fn.	LayerNorm + Mish (model-based), ReLU (model-free)
Target network EMA weight	1e-2

Table 4. **ViGMO hyperparameters.** Hyperparameters of ViGMO for completeness while highlighting a subset of hyperparameters only provided to ViGMO. We borrow the same hyperparameters of TD-MPC2 (Hansen et al., 2023).

Hyperparameter	Value
// MPC planning	
Planning Horizon H	3
Std. range	$\sigma \in [0.05, 2]$
Population size	512
Elite fraction	64
Iterations	6
Policy prior samples	24
Sampling temperature	0.5
// Model learning	
Temporal coefficient λ	0.5
Reward loss coefficient c_1	0.1
Q value loss coefficient c_2	0.1
Latent consistency loss coefficient c_3	20
// Optimization	
Learning rate η	3e-4
Periodic target network (θ^-) update steps δ	1
Optimizer	Adam($\beta_1 = 0.9$,$\beta_2 = 0.999$)
Exploration schedule (std)	Linear(0.5, 0.05, 25,000 steps)
Planning horizon schedule	Linear(1, 5, 25,000 steps)
// ViGMO hyperparameters	
Weak random augmentation Υ^w	*random-shift*: `padding` $p = 4$
Strong random augmentation Υ^s	*random-overlay*: $\begin{cases}\texttt{linear interpolation:}\quad \delta = 0.5 \\ \texttt{Image dataset:}\quad \mathcal{D} = \texttt{Places (Zhou et al., 2017)}\end{cases}$
Weak and strong augmentation ratio ζ	1.0
Regularization coefficient α	1

For model-free baselines, we compute N-step TD target for value function learning; $(r_t + r_{t+1} + \cdots + r_{t+N}) + \gamma Q(s_{t+N}, \pi(s_{t+N}))$. The image dimension used for training and evaluation is $84 \times 84 \times 9$: $s_t = \{o_t, o_{t-1}, o_{t-2}\}$

where $o_t \in \mathcal{O}^{84\times 84\times 3}$. We stack the consequent images along the color channel axis and repeat a specific amount of the same action extracted from the policy or model planning by following prior works.

Table 5. **Baseline hyperparameters.** Algorithm-specific hyperparameters for **model-free** baselines across experiments (DMC and robosuite). Other hyperparameters not described here can be found in Table 3.

Hyperparameter	Value
Periodic critic target network (θ^{-}) update steps	1
Clip constant for the stochastic actor	3e-2
Learning rate for the auxiliary task	3e-4 (SGQN)
Attribution mask quantile	0.95 (SGQN)

B. Implications under method

B.1. Data Augmentation

While prior works have shown weak and strong augmentations boost the sample efficiency and generalization performance, the empirical results are limited to the value-based model-free learning approach. Hence, we propose a novel method for adapting data augmentations into model-based RL. *Random-shift* (Yarats et al., 2021a) augmentation applies a fixed amount of padding to a random direction in *top, bottom, right*, and *left* of the image and *random-overlay* (Hansen & Wang, 2021) augmentation linearly interpolates between a random image and an original image where the random image is sampled from an unrelated data to the task (Zhou et al., 2017). Likewise in previous works, while one might feed both weakly and strongly augmented images to the encoder in principle, we empirically find that randomly dividing the batch in half along the batch dimension and augmenting two sub-batches with different augmentations can produce a similar performance with a decreased computing budget.

B.2. Latent Transition Dynamics

As observed in many prior works (Mnih, 2013; He et al., 2020; Hansen et al., 2021), noisy and high-variance target values might impede the fast convergence of the Q value function. Since the Q value network is often conditioned on the representation encoded from the observation images directly over the horizon, the representation encoded from the strongly augmented image may produce a trivial signal for downstream model learning. However, the field has observed that weak data augmentation often encourages sample-efficient RL in high-dimensional observation space configuration (Yarats et al., 2021b;a; Hansen et al., 2022). Hence, we choose weak augmentation for generating TD target of value learning instead of strong data augmentation.

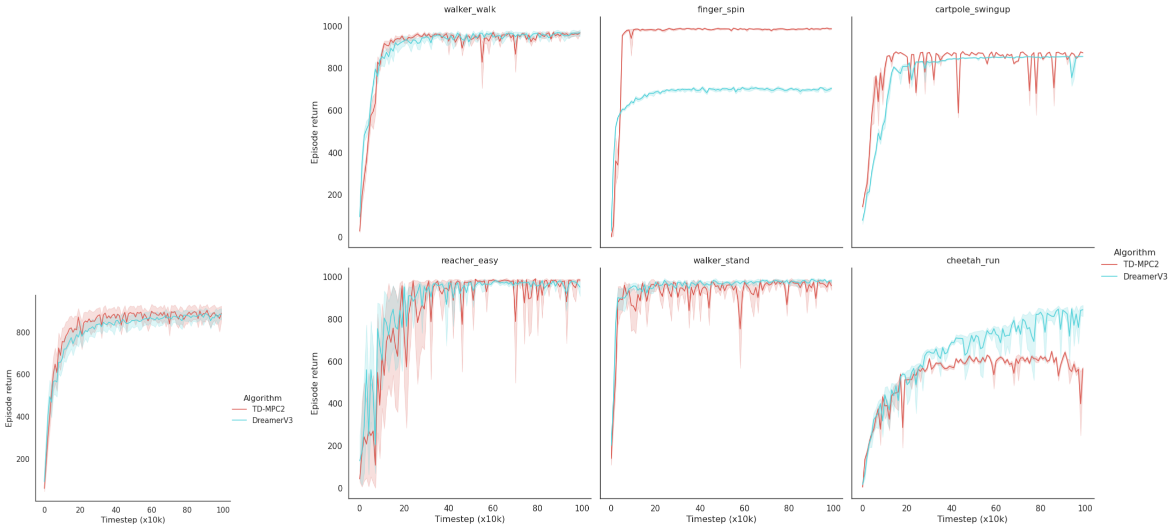
B.3. Regularization

Following the suggested procedure, the world model improves the sample efficiency and generalization through augmentations and predicts consistent outputs regardless of augmentations. However, the encoder might predict inconsistent representation between training and evaluation images. Although the latent transition model is trained to predict consistent representations over the horizon, the encoder has no constraint to predict a similar representation whether the training or evaluation image is given. Hence, we contend the model should generate reliable synthetic samples regardless of the training or evaluation phase to enable sample-efficient and generalizable model-based RL. To this end, we bring the auxiliary representation learning task to encoder learning during world model training. By regulating the encoder to preserve similar features (e.g., the physical body of the agent) and discarding irrelevant information (e.g., background and luminosity) between the original image and the augmented image, we can obtain consistent representation in both training and evaluation settings.

C. Discussions

Model-based RL. Our main contribution to this paper is to present a model-based RL method that empirically demonstrates strong generalization ability without sacrificing sample efficiency. We have compared two state-of-the-art model-based RL backbone algorithms, TD-MPC2 (Hansen et al., 2023), and DreamerV3 (Hafner et al., 2023), which exhibit strong sample efficiency over diverse continuous control problems in recent years. We provide reproduced empirical results to validate that

TD-MPC2 would exhibit better sample efficiency than DreamerV3 in Figure 9. Since we aim to achieve sample-efficient RL for visual generalization, we choose TD-MPC2 as our backbone model-based RL algorithm, which demonstrates stronger sample efficiency in continuous control tasks. Those two model-based algorithms train and exploit the latent transition dynamics model that takes the low-dimensional latent state and action as inputs and outputs the next latent state to solve continuous control tasks with the high-dimensional image input. DreamerV3 learns the world model by predicting latent future states, actions, and **state-value** functions (V functions) and **reconstructing** the given image by contrastive learning. In contrast, TD-MPC2 trains the world model by predicting latent future states, actions, and **action-value** functions (Q functions) and **avoids reconstructing** the high-dimensional images. While DreamerV3 plans the optimal action with the trained world model and **actor policy**, TD-MPC2 derives the action by planning an optimal action with the **model predictive controller**. Since DreamerV3 employs representation learning with the reconstruction objective, adapting our method to Dreamer with the reconstructive techniques for visual generalization (Bertoin et al., 2022; Wang et al., 2023) would be exciting future work.



(a) Aggregated comparison result.

(b) Comparison over tasks in the DeepMind Control suite.

Figure 9. **Comparison of sample efficiency between model-based RL methods.** TD-MPC2 demonstrates superior sample efficiency over DreamerV2 in 6 tasks in the DeepMind Control suite benchmark.

D. Supplementary Results

We provide additional results for extensive experiments here. Concerning the main results, we additionally plot the generalization performance and sample efficiency across the tasks and evaluation types.

D.1. Quantitative Sample Efficiency Comparison

In Section 5.2, we provide numerical comparisons (Table 1) concerning the sample efficiency and generalization performance. To compare the baselines with ViGMO, we quantitatively compute the sample efficiency, inspired by (Mai et al., 2022). We first train an oracle agent, which is a popular visual RL model-free method (DrQ-v2, (Yarats et al., 2021a)). We consider a visual RL agent to successfully solve the task when the agent reaches the score of the oracle agent. We compare the number of *episodes* necessary for achieving *25%*, *50%*, and *75%* of the oracle scores. If the agent fails to reach the oracle score in each percentile, we fill the score as the number of maximum episodes (i.e., 1000 in DMC, 2000 in robosuite). See detailed scores in Table 6.

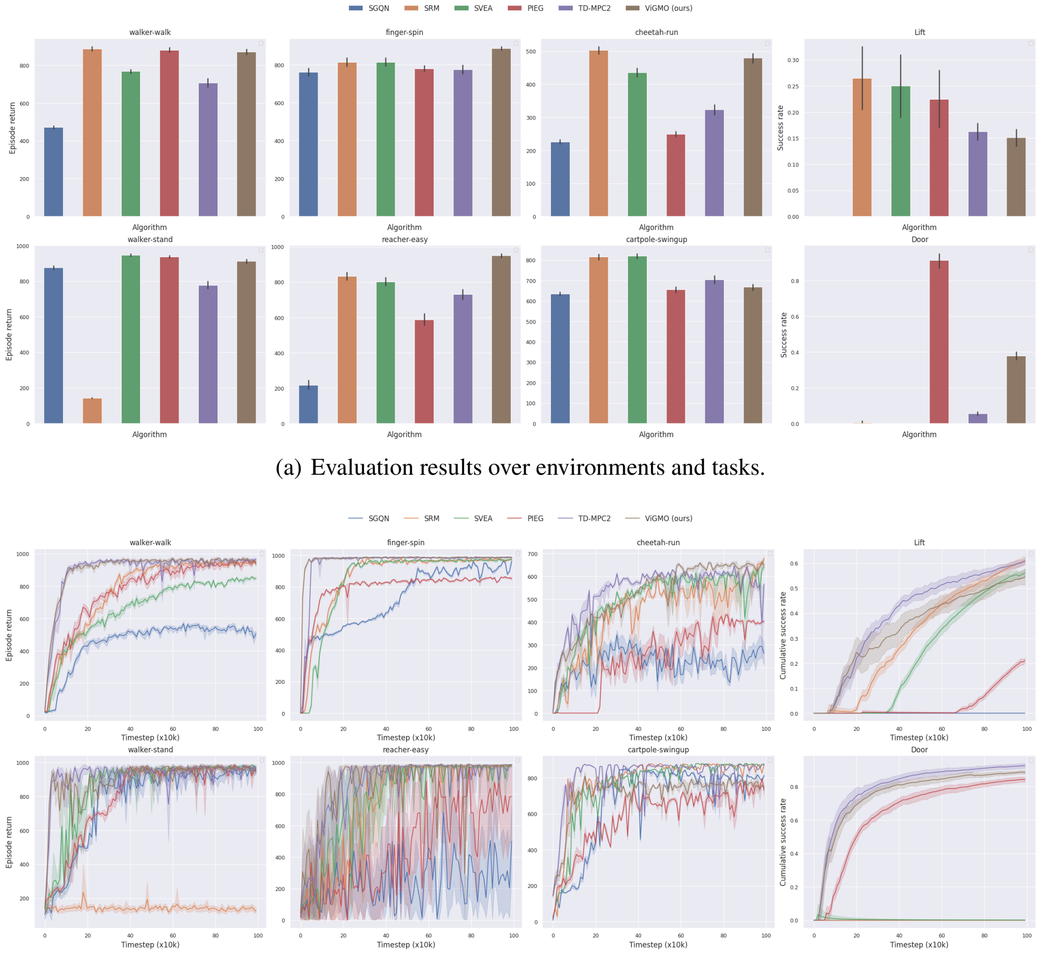
D.2. Qualitative Performance Comparison

One can find that PIEG outperforms other baselines at the *Door* task in the robosuite environment. We conjecture that PIEG demonstrates a strong generalization performance at the *Door* task since PIEG exploits the pretrained encoder from ImageNet. ImageNet is a huge dataset that contains diverse images from everyday lives, e.g. door images. Hence, we

Table 6. **Quantitative comparison for visual generalization.** Visual generalization performance comparison over different environments and tasks. Scores of the backbone architecture, TD-MPC2, are provided to contrast the visual generalization difference of ViGMO. The numbers below algorithms represent the percentile (e.g., 0.25 is 25%). Red scores stand for a failed situation to achieve the oracle score (filled with the maximum episode number).

		SVEA			SGQN			SRM			PIEG			ViGMO (ours)			TD-MPC2		
Env	Task	0.25	0.5	0.75	0.25	0.5	0.75	0.25	0.5	0.75	0.25	0.5	0.75	0.25	0.5	0.75	0.25	0.5	0.75
DMC	cartpole_swingup	180	490	520	240	330	330	80	400	400	660	1000	1000	100	760	1000	60	110	110
	finger_spin	190	220	260	500	560	820	180	230	300	200	1000	1000	40	40	60	60	60	60
	walker_walk	50	90	370	120	160	1000	50	80	260	50	60	190	30	40	80	20	40	80
	walker_stand	10	10	10	10	10	10	10	10	10	10	10	10	20	20	20	10	10	10
	cheetah_run	160	430	970	1000	1000	1000	210	380	880	480	1000	1000	180	460	600	100	260	1000
	reacher_easy	10	30	30	10	60	180	10	40	40	10	60	60	20	30	30	10	30	30
	Avg.	100	211	360	313	353	556	90	190	315	235	521	543	65	225	298	43	85	215
robosuite	Door	2000	2000	2000	2000	2000	2000	2000	2000	2000	580	1020	1640	280	540	720	200	320	420
	Lift	760	1200	1680	2000	2000	2000	160	1000	1440	1560	2000	2000	200	420	1340	180	560	1120
	Avg.	1380	1600	1840	2000	2000	2000	1080	1500	1720	1070	1510	1820	240	480	1030	190	440	770

presume that the representation encoded from the pretrained encoder might distinguish the object well while other baselines should learn the effective representation first. This enables superior sample efficiency and generalization performance for PIEG.



(a) Evaluation results over environments and tasks.

(b) Sample efficiency over environments and tasks.

Figure 10. **Full experimental results over tasks.** (**Top**) Generalization performance and (**Right**) sample efficiency. Episode returns and cumulative success rates over tasks are reported in DMC and robosuite, respectively. All evaluation results are averaged over 5 seeds and sample efficiency results are averaged over 10 episodes during training.

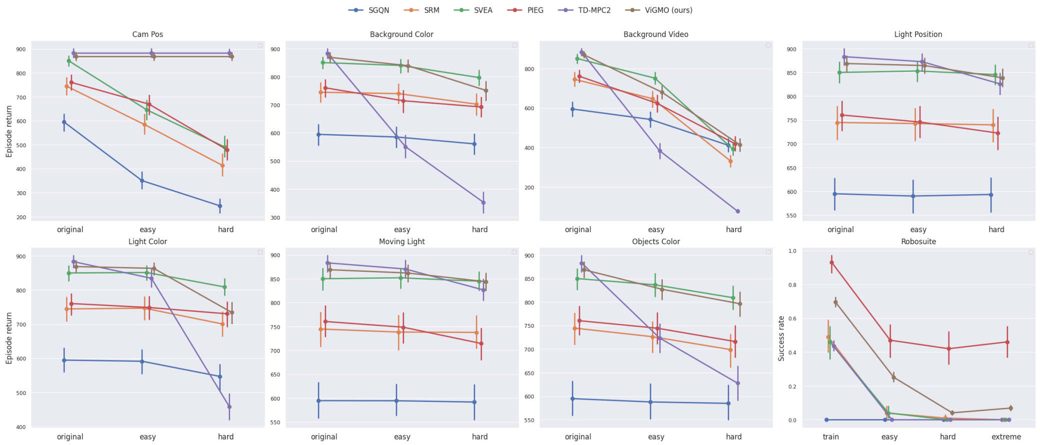


Figure 11. **Generalization performance over evaluation types.** 14 types in DMC and 3 types in robosuite for generalization evaluation. Generalizable RL methods show a monotonic decrease between original and generalization types (e.g., original-easy-hard). All evaluation results are averaged over 5 seeds and sample efficiency results are averaged over 10 episodes during training.

In Figure 11, we provide the relative performance of baselines between the training environment and evaluation tasks. While all baselines indicate decreased performance depending on the severity of generalization, TD-MPC2 demonstrates the steepest decrease among the baselines. These results support the idea that conventional model-based RL suffers from the distribution shift when the trained model is given unseen input during evaluation. As one can see in Figure 10 and 11, almost every baseline shows moderate generalization performance in DMC while struggling in the robosuite environment. We suggest that current algorithms including state-of-the-art methods for visual generalization have demonstrated limited results to some extent and there is room for further improvements in problem formulation for better generalization.

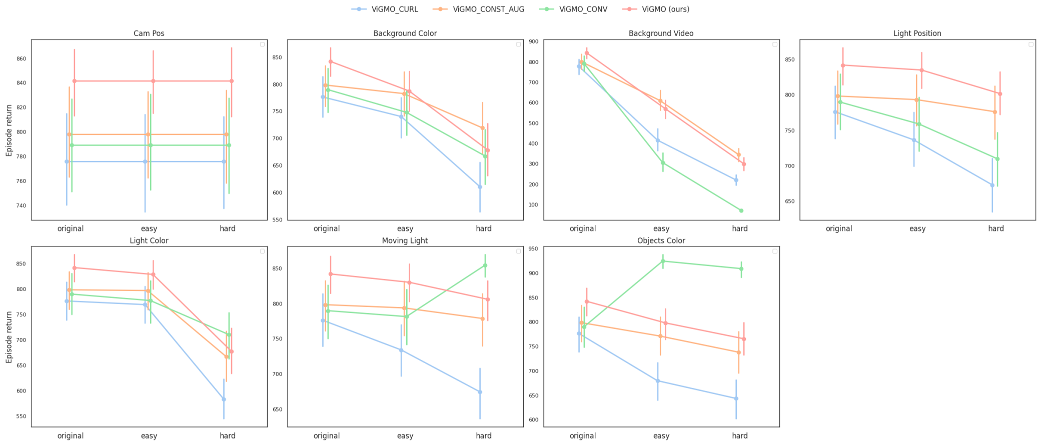


Figure 12. **Generalization performance over evaluation types of ablations.** Similar to Figure 11 but baselines are ablated versions of ViGMO in Section 5.2.2. ViGMO shows a monotonic decrease between original and generalization types (e.g., original-easy-hard) compared to other options. All evaluation results are averaged over 5 seeds and sample efficiency results are averaged over 10 episodes during training.

D.3. Ablation Performance Comparison

We provide additional experimental results comparing possible options over environments in Figure 12 from Section 5.2.2. *ViGMO_CONV* shows impressive performance in *moving_light* and *object_color* tasks, where the image is mainly distracted with colors. Since *ViGMO_CONV* augments the image with random convolution operation during training, *ViGMO_CONV* may show relatively strong generalization performance compared to other baselines. Example images comparing the popular strong augmentation techniques can be found in Figure 13.

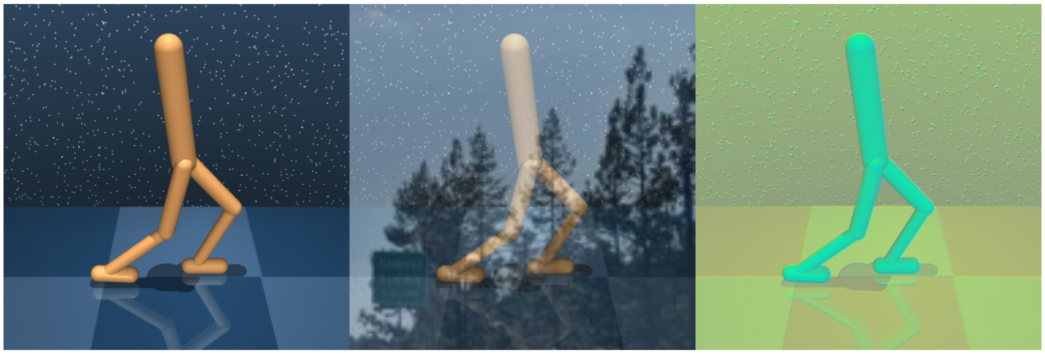
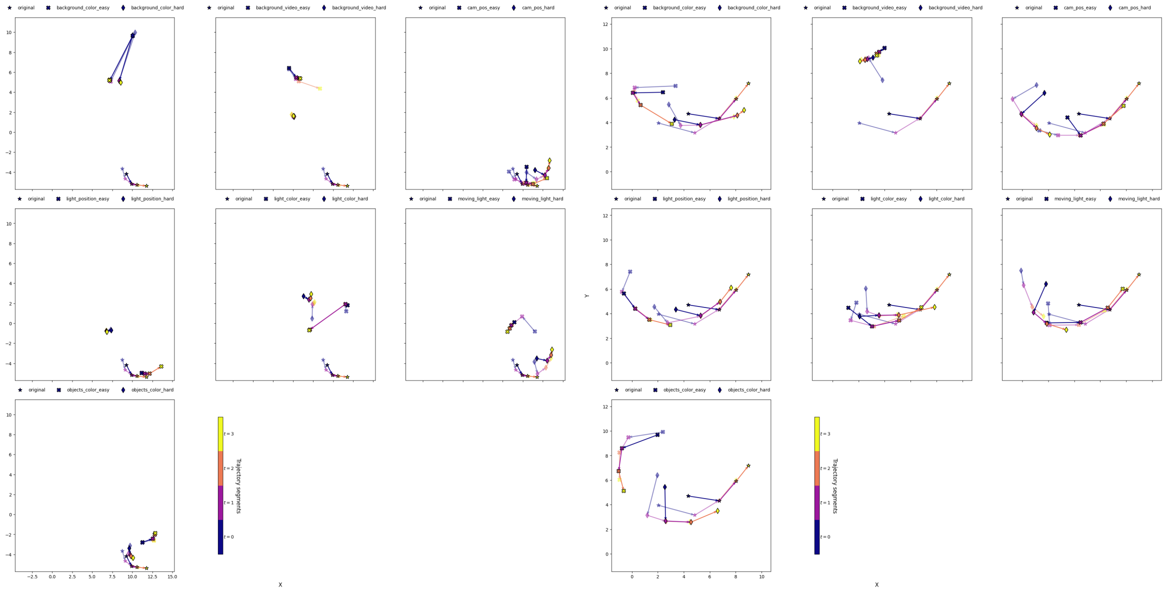


Figure 13. **Example images comparing strong augmentations.** Illustrations of how the image is augmented in *walker_walk* task. (Left) Original image, (Center) *random-overlay* augmentation, (Right) *random-conv* augmentation.



(a) **Visualization of embeddings from TD-MPC2.**

(b) **Visualization of embeddings from ViGMO.**

Figure 14. **Visualization of embeddings.** Star, X, and diamond markers correspond to *original task*($\star$), *easy task*($\times$), and *hard task*($\diamond$), respectively. Arrows indicate sequential transitions from each representation. Trajectory segments are $s_{t:t+H}$.

D.4. Embedding Visualization

Throughout empirical findings, the world model becomes generalizable to unseen observations and avoids forgoing the superior sample efficiency of model-based RL. The underlying implication is that the world model would exhibit similar performance when the transition model predicts *consistent* trajectories regardless of perturbations. Hence, we examine the motivation that ViGMO would predict consistent representations regardless of perturbations while TD-MPC2 fails. We first train ViGMO and TD-MPC2 on each task and then roll out the trained models to plan actions. For every time-step t, we aggregate the horizontal representations from the latent transition model: $z_t = h_\theta(s_t), z_{k+1} = d_\theta(s_t, a_t) \forall k \in [t+1, t+H]$ where each action is planned by MPC. After, we obtain 2D latent coordinates from the aggregated representations using U-MAP (McInnes et al., 2018). We choose *cheetah_run* for embedding experiments. We provide full plots for embedding experiments in Figure 15.

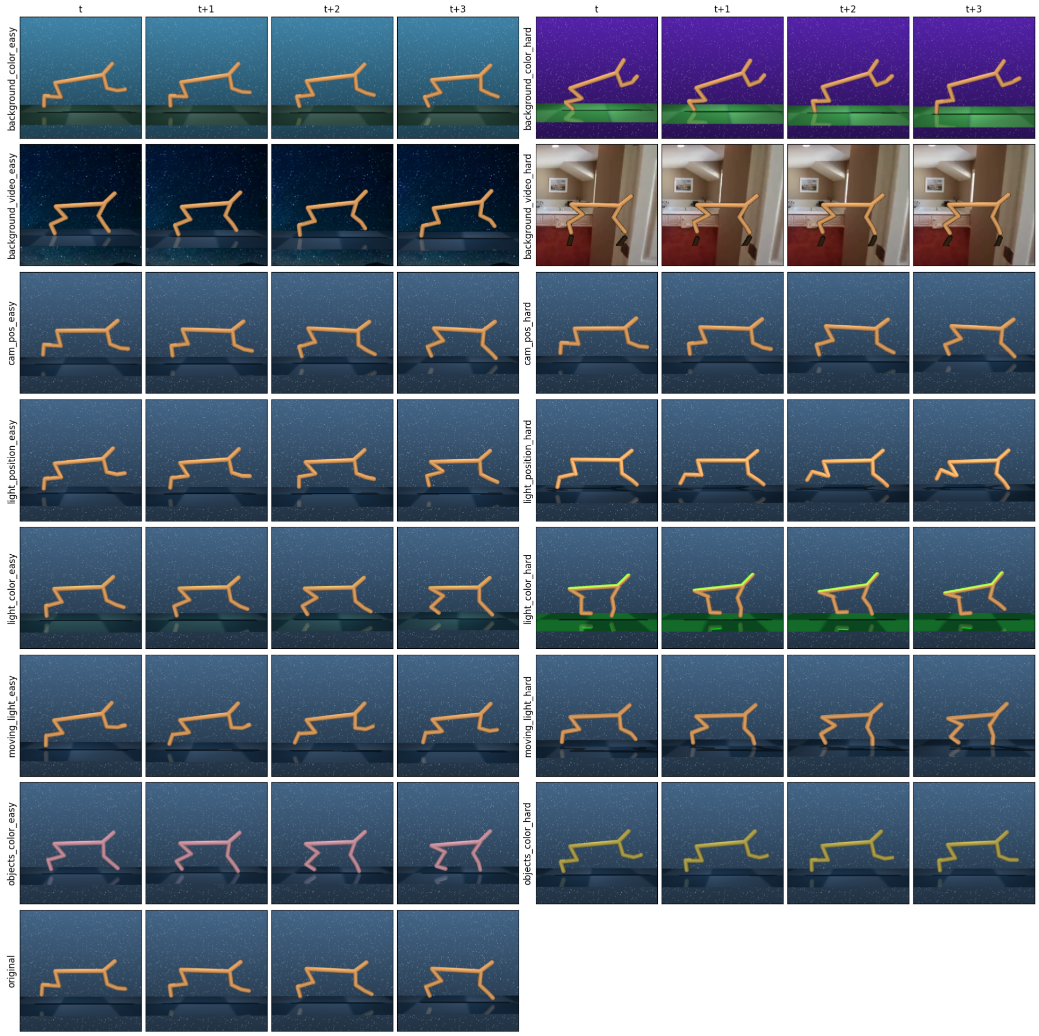


Figure 15. **Full visualization of embedding experiments.** Each row stands for a different generalization task and each column stands for the horizontal time step.

In Figure 6, faded trajectories are previous time-step embeddings from d_θ. We illustrate extracted embeddings as (*original task*($\star$), *easy task*($\times$), and *hard task*($\diamond$) from the left-side of each row). Embeddings from TD-MPC2 show inconsistent representations while ViGMO indicates consistent and aligned representation across generalization tasks. Furthermore, ViGMO demonstrates accurate predictions of latent transition dynamics models across tasks, supporting the assertion that learning consistent representation significantly enhances the generalization performance.