Dream to Generalize: Zero-Shot Model-Based Reinforcement Learning for Unseen Visual Distractions

Jeongsoo Ha[1], **Kyungsoo Kim**[2], **Yusung Kim**[3]
[1] Mechatronics Research, Samsung Electronics
[2] Intelligent Agent Lab, NCSOFT
[3] Department of Computer Science and Engineering, Sungkyunkwan University
js1210.ha@samsung.com, unigary@ncsoft.com, yskim525@skku.edu

Abstract

Model-based reinforcement learning (MBRL) has been used to efficiently solve vision-based control tasks in high-dimensional image observations. Although recent MBRL algorithms perform well in trained observations, they fail when faced with visual distractions in observations. These task-irrelevant distractions (e.g., clouds, shadows, and light) may be constantly present in real-world scenarios. In this study, we propose a novel self-supervised method, **Dr**eam to **G**eneralize (**Dr. G**), for zero-shot MBRL. **Dr. G** trains its encoder and world model with dual contrastive learning which efficiently captures task-relevant features among multi-view data augmentations. We also introduce a recurrent state inverse dynamics model that helps the world model to better understand the temporal structure. The proposed methods can enhance the robustness of the world model against visual distractions. To evaluate the generalization performance, we first train **Dr. G** on simple backgrounds and then test it on complex natural video backgrounds in the DeepMind Control suite, and the randomizing environments in Robosuite. **Dr. G** yields a performance improvement of 117% and 14% over prior works, respectively. Our code is open-sourced and available at https://github.com/JeongsooHa/DrG.git

Introduction

Reinforcement learning (RL) with visual observations has achieved remarkable success in many areas, including video games, robot control, and autonomous driving (Mnih et al. 2013; Levine et al. 2016; Nair et al. 2018; Kalashnikov et al. 2018; Andrychowicz et al. 2020). Because learning a control policy from high-dimensional image data is inevitably more difficult than learning from low-dimensional numerical data, training a visual RL agent requires a larger amount of training data. To address the data inefficiency, recent model-based RL (MBRL) studies have proposed learning a world model in the latent space, followed by planning the control policy in the latent world model.

Although latent-level MBRL studies have successfully improved data efficiency, they have inherent drawbacks because they are typically designed as reconstruction-based methods. The drawback of these methods comes from visually distracting elements (task-irrelevant information) that can compromise the accuracy of the reconstruction-based representation learning. In particular, task-irrelevant information such as clouds, shadows, and light may change continuously depending on the time and place of the test. Therefore, generalization in terms of representation and policy learning is crucial for solving real-world problems (Kim, Ha, and Kim 2022).

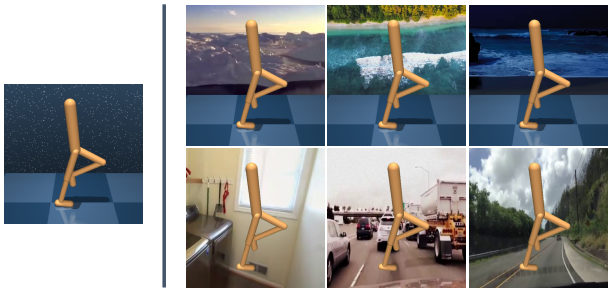

Figure 1: The agent is trained in a simple background environment on the DeepMind Control suite in the default setting (left). We demonstrate the generalization ability in unseen environments in video easy settings (a top row on the right) and video hard settings (a bottom row on the right).

In this study, we propose **Dr**eam to **G**eneralize (**Dr. G**), a *zero-shot* MBRL framework that can be robust to visual distractions not experienced during training. Our proposed **Dr. G** introduces two self-supervised methods; 1) *Dual Contrastive Learning (DCL)*, and 2) *Recurrent State Inverse Dynamics (RSID)*. **Dr. G** uses the same structure as the recurrent state space model (RSSM) (Hafner et al. 2019b) also used in Dreamer (Hafner et al. 2019a) but replaces the reconstruction-based learning part with DCL. The DCL approach consists of two objective functions over multi-view data augmentations. One objective function is applied between realities, which are latent states encoded with different data augmentation techniques (hard and soft) for the same image observation. It improves the generalization ability of the encoder against visual distractions. The other objective function is applied between reality and dreams (imagined latent states by RSSM). This allows the world model to dream (predict) the next latent state more robustly, enabling **Dr. G** to learn a more generalized control policy in

the world model dreams.

The second self-supervised method, RSID, infers the actual executed actions over a sequence of latent states imagined by the world model. It enables the world model to understand the temporal structure and relationships between successive states, and helps to generate more robust rollouts for policy planning.

We evaluate the generalization performance of the proposed zero-shot MBRL framework, **Dr. G**, on six continuous control visual tasks in the DeepMind Control suite (Tassa et al. 2018) and on five tasks in the Robosuite (Zhu et al. 2020). After training **Dr. G** on simple background observations, we test it on unseen complex visual distractions, as shown in Figure 1. **Dr. G** yields a performance improvement of 117% over existing model-based and model-free RL algorithms on the DeepMind Control suite and 14% over existing algorithms on the Robosuit.

The key contributions of this study are as follows:

- We introduce a zero-shot MBRL method, **Dr. G**, to train both the encoder and world model in a self-supervised manner. Using DCL and RSID, **Dr. G** can achieve robust representations and policies over unseen visual distractions.
- We demonstrate that **Dr. G** outperforms prior model-based and model-free algorithms on various visual control tasks in the DeepMind control suite and Robosuite. We also conduct thorough ablation studies to analyze our proposed method.

Preliminaries

In this section, we briefly introduce the training method for the world model, which forms the core of MBRL. The reconstruction-based world model is based on Dreamer (Hafner et al. 2019a) and Dreamerv2 (Hafner et al. 2020). For convenience, as the frameworks of the two papers are similar, we omit the version of Dreamerv2.

Reconstruction-Based World Model Learning

Recent MBRL methods train compact latent world models using high-dimensional visual inputs with variational autoencoders (VAE) (Kingma and Welling 2013) by optimizing the *Evidence Lower BOund* (ELBO) (Bishop 2006) of an observation. For an observable variable x, VAEs learn a latent variable z that generates x by optimizing an ELBO of $\log p(x)$, as follows:

$$\begin{aligned}\log p(x) &= \log \int p(x|z)p(z)dz \\ &\geq \mathbb{E}_{q(z|x)}[\log p(x|z)] - D_{\mathrm{KL}}[q(z|x)||p(z)],\end{aligned} \tag{1}$$

where $D_{KL}[q(z|x)||p(z)]$ represents the Kullback-Leibler divergence between the prior distribution $p(z)$ and an assumed distribution $q(z|x)$ that samples z conditioned on x.

Dreamer (Hafner et al. 2019a)uses RSSM (Hafner et al. 2019b) as the world model to predict the sequence of future states and reward signals in latent space. At each time step t, the agent receives an image observation o_t and a reward r_t (from $a_{<t}$ in the sequential decision-making task). Then, the agent chooses an action a_t based on its policy. RSSM learns latent dynamics by reconstructing images and rewards by optimizing the ELBO of $\log p(o_{1:T}, r_{1:T}|a_{1:T})$ (Hafner et al. 2019b; Igl et al. 2018). That is, as RSSM optimizes ELBO for sequential information, as expressed in Equation 1, we obtain

$$\begin{aligned}&\log p(o_{1:T}, r_{1:T}|a_{1:T}) \\ &= \log \int \prod_{t=1}^{T} p(o_t|s_{\leq t}, a_{<t})p(r_t|s_{\leq t}, a_{<t}) \\ &\qquad\qquad p(s_t|s_{<t}, a_{<t})ds_{1:T} \\ &= \log \int \prod_{t=1}^{T} p(o_t|h_t, s_t)p(r_t|h_t, s_t)p(s_t|h_t)ds_{1:T},\end{aligned} \tag{2}$$

where $s_{1:T}$ are sequential states in the stochastic model, and h_t is the hidden state vector obtained through $f(h_{t-1}, s_{t-1}, a_{t-1})$ as a deterministic state, which uses gated recurrent unit (GRU) (Cho et al. 2014). To infer the agent states from past observations and actions, a variational encoder is used, which is expressed as

$$\begin{aligned}q(s_{1:T}|o_{1:T}, a_{1:T}) &= \prod_{t=1}^{T} q(s_t|s_{<t}, a_{<t}, o_t) \\ &= \prod_{t=1}^{T} q(s_t|h_t, o_t)\end{aligned} \tag{3}$$

Based on Equations 2 and 3, the objective of Dreamer is to maximize the ELBO, as follows.

$$\begin{aligned}\mathcal{J}_{\mathrm{Dreamer}} &= \sum_{t=1}^{T} \mathbb{E}_q[\log p(o_t|h_t, s_t) + \log p(r_t|h_t, s_t) \\ &\qquad - D_{\mathrm{KL}}(q(s_t|h_t, o_t)||p(s_t|h_t))] \\ &= \sum_{t=1}^{T} \mathbb{E}_q\left[\mathcal{J}_{\mathrm{O}}^t + \mathcal{J}_{\mathrm{R}}^t - \mathcal{J}_{\mathrm{KL}}^t\right]\end{aligned} \tag{4}$$

$\mathcal{J}_O$ and $\mathcal{J}_R$ are used as reconstruction objective functions to restore image observations and rewards. And $\mathcal{J}_{KL}$ is used as an objective function for the KL divergence.

Improvement Actor and Critic With Latent Dynamics

After training RSSM as a world model, the actor and critic are trained through latent trajectories imagined in latent space using a fixed world model. On imagined trajectories with a finite horizon H, the actor and critic learn behaviors that consider rewards beyond the horizon. At this time, the reward and the next state are predicted by the trained world model. At each imagination step $\tau \geq t$, during a few steps H, the actor and critic are expressed as follows:

Actor model: $\hat{a}_\tau \sim \pi_\phi(\hat{a}_\tau|h_\tau, s_\tau)$

Critic model: $v_\psi(s_\tau) \approx \mathbb{E}_{q(\cdot|s_\tau)}(\sum_{\tau=t}^{t+H} \gamma^{\tau-t} r_\tau)$,

where θ, ψ are model parameters and γ is a discount factor. The actor and critic are trained cooperatively in policy

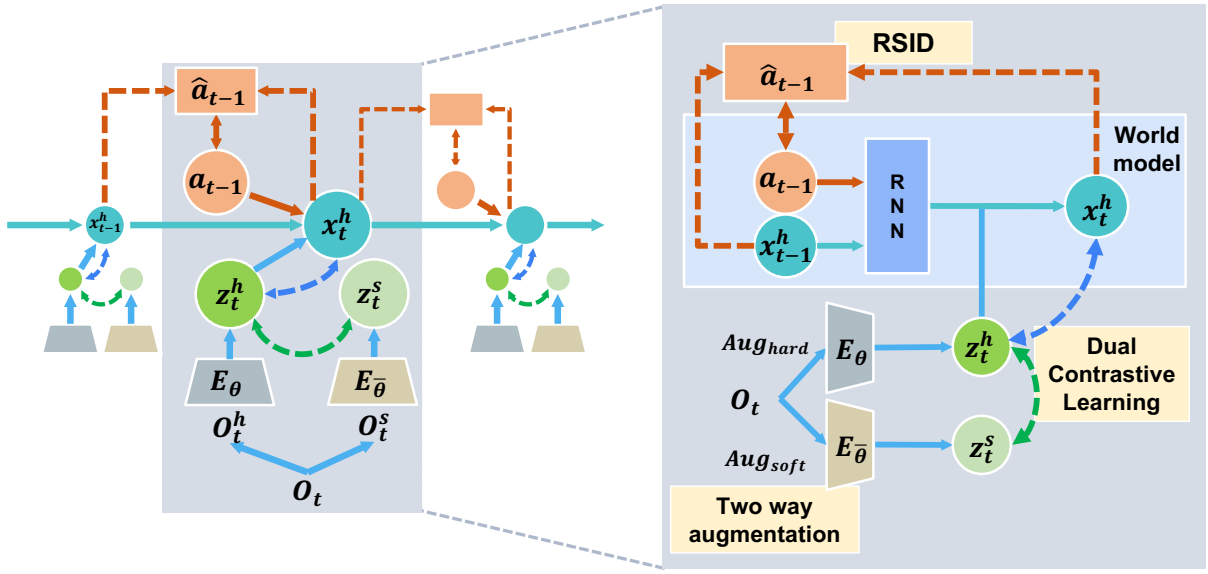

Figure 2: Our Framework Overview: **Dr. G** trains the encoder and world model through *two-way augmentation*, *Dual Contrastive Learning* (green and blue dashed line), and *Recurrent State Inverse Dynamics* (orange dash line) with sequential data.

iterations. A policy model aims to maximize a value estimate, whereas a value model aims to match the value estimate to a behavioral model. Within the imagined trajectory, the actor and critic are trained to improve the policy such that λ-return (Sutton and Barto 2018; Schulman et al. 2015) and approximate λ-return using squared loss are maximized, respectively.

Contrastive Learning

Contrastive learning (Hadsell, Chopra, and LeCun 2006; He et al. 2020; Wu et al. 2018; Chen et al. 2020; Oord, Li, and Vinyals 2018) is a framework for learning representations that satisfy similarity constraints in a dataset typically organized based on similar and dissimilar pairs. It can be understood as training an encoder for a dictionary look-up task. It considers an encoded query q and a set of encoded samples $\mathbb{K} = \{k_0, k_1, k_2, ...\}$ as the keys of a dictionary. Assuming that there is a single key (denoted as k_+) in the dictionary that matches q, the goal of contrastive learning is to ensure that q matches k_+ to a greater extent than the other keys in $\mathbb{K}\backslash\{k_+\}$ (except a single sample k_+ in a set $\mathbb{K}$). $q, \mathbb{K}, k_+$ and $\mathbb{K}\backslash\{k_+\}$ are referred to as the anchor, target, positive, and negative, respectively, in the terms of contrastive learning (Oord, Li, and Vinyals 2018; He et al. 2020). Similarities between the anchor and targets are best modeled by calculating the dot product ($q^T k$) (Wu et al. 2018; He et al. 2020) or bilinear products ($q^T W k$) (Oord, Li, and Vinyals 2018; Henaff 2020). To learn embeddings that satisfy the similarity relations, CPC (Oord, Li, and Vinyals 2018) proposed the InfoNCE loss, which is expressed as

$$\mathcal{L}_q = \log \frac{\exp(q^T W k_+)}{\exp(q^T W k_+) + \sum_{i=0}^{K-1} \exp(q^T W k_i)} \quad (5)$$

The loss Equation 5 can be understood as the log-loss of a K-way softmax classifier whose label is k_+ (Laskin, Srinivas, and Abbeel 2020).

Method

We propose **Dr. G**, a novel self-supervised method, to train *zero-shot* MBRL. Dr. G achieves excellent generalization ability for observational changes not experienced during training. The proposed approach is illustrated in Figure 2.

Model Overview

The basic architecture is based on the Dreamer (Hafner et al. 2019a) paradigm. We first train RSSM as the world model and then plan the control policy on the rollouts imagined by the world model. The actor and critic learning methods used are the same as in Dreamer; however, we replace the reconstruction objective of Dreamer with the proposed self-supervised methods, namely *DCL* and *RSID*. In the self-supervised methods, we apply multi-view data augmentations-a soft augmentation that provides minor position changes and a hard augmentation that inserts complex visual distractors to interfere with the original image. The combination of multi-view data augmentations and the self-supervised methods successfully achieve zero-shot generalization. We show the overall training process through the Algorithm 1.

Hard and Soft Augmentations

We use two data augmentation techniques during the encoder and world model training; one is *Random-Shift* (Kostrikov, Yarats, and Fergus 2020) as a soft augmentation Aug_s and the other is *Random-Overlay* (Hansen et al. 2020; Hansen and Wang 2021) as a hard augmentation Aug_h

Figure 3: Random shift and random overlay are used as soft and hard augmentations, respectively. Original image observation (left), soft augmented version (mid), and hard augmented version (right).

as shown in Figure 3. Random shift applies a pad around the observation image and performs a random crop back to the original image size. Random overlay linearly interpolates between the original observation o_t and a randomly chosen complex image, as in Equation 6, where α is the interpolation coefficient and D is a complex image dataset containing 1.8M diverse scenes (Hansen and Wang 2021).

$$\begin{aligned} o_t^s &= \mathrm{Aug_s}(o_t) \triangleq \text{Random shift}(o_t) \\ o_t^h &= \mathrm{Aug_h}(o_t) \triangleq (1-\alpha) * o_t + \alpha * \mathrm{img},\ \mathrm{img} \sim D \end{aligned} \tag{6}$$

Dual Contrastive Learning

Instead of the reconstruction loss used in Dreamer, we introduce *DCL* using InfoNCE loss (Oord, Li, and Vinyals 2018), which is a widely used loss function in contrastive model training. Our DCL consists of two objectives: One is applied between latent states with multi-view augmentations for the same observation. It enables the encoder to extract invariant representations (task-relevant features) regardless of different augmentations (dominated by task-irrelevant information). Because these latent states are encoded from real observations, we call it contrastive learning between realities. The other objective is applied between reality (latent states encoded from the real observations) and dreams (latent states generated by the world model), making the world model more robust.

Contrastive Learning Between Realities In *reality-reality*, the agent compares latent states embedded in the encoder E_θ and target encoder $E_{\bar{\theta}}$ to improve the generalization ability against task-irrelevant information, as indicated by the green dashed line in Figure 2. We consider the latent state encoded from the observation as *reality*. The encoder embeds a hard augmented observation in z^h, and at the same time the target encoder extracts another latent state z^s with a soft augmented version. To maximize the mutual information of two latent states with different perspectives, we use contrastive learning and an objective function of the following form:

$$\mathcal{J}_{\mathrm{E}}^t = \mathbb{E}\left[\log \frac{\exp({z_t^s}^T W z_{t,+}^h)}{\exp({z_t^s}^T W z_{t,+}^h) + \sum_{i=0}^{N-1} \exp({z_t^s}^T W z_{t,i}^h)}\right] \tag{7}$$

where $z_t^h = E_\theta(\mathrm{Aug}_h(o_t))$ and $z_t^s = E_{\bar{\theta}}(\mathrm{Aug}_s(o_t))$, which apply hard and soft augmentation to image observation respectively. We apply soft and hard augmentation to each of the N images (observations) in the batch. For each soft-augmented latent state, the hard-augmented latent state for the same image is used as a positive sample, and the other N-1 hard-augmented versions are used as negative samples. Then the target encoder is updated according to the following *Exponential Moving Average* (EMA):

$$\bar{\theta}_{n+1} \leftarrow (1-\tau)\bar{\theta}_n + \tau\theta_n, \tag{8}$$

for an iteration step n and a momentum coefficient $\tau \in (0, 1]$, such that only parameters θ are updated by gradient descent (He et al. 2020; Grill et al. 2020; Lillicrap et al. 2015).

Contrastive Learning Between Dream and Reality In *dream-reality*, shown as the blue dashed line in Figure 2, the agent compares *dream* x_t^h, which is a latent state imagined by the world mode, with *reality* z_t^h encoded from the augmented observation. By maximizing the similarity between dream and reality, the world model can imagine (predict) the next latent state more robustly. We note that *dream-reality* uses the hard augmentation technique only because it shows the best zero-shot generalization empirically. This ablation study is shown in Figure 10 in the supplementary material. By updating in a similar way to *reality-reality*, contrastive learning takes the following form:

$$\mathcal{J}_{\mathrm{RSSM}}^t = \mathbb{E}\left[\log \frac{\exp({z_t^h}^T W x_{t,+}^h)}{\exp({z_t^h}^T W x_{t,+}^h) + \sum_{i=0}^{N-1} \exp({z_t^h}^T W x_{t,i}^h)}\right] \tag{9}$$

where $x_t^h = \mathrm{RSSM}_\theta(z_t^h, x_{t-1}^h, a_{t-1})$ which is hard augmented imagined latent state, $z_t^h = E_\theta(\mathrm{Aug}_h(o_t))$ is the

---

Algorithm 1: Dr. G

Hyperparameters:
S(seed episode), C(collect interval), B(batch size), L(sequence length), H(imagination horizon)
Initialize: dataset D with S random seed episodes.
Initialize: neural network parameters θ, ϕ, ψ.
for *each iteration* **do**
  for *updata setp* $c = 1..C$ **do**
    Draw B data sequences $(o_t, a_t, r_t)_{t=k}^{k+L} \sim D$
    `// Update world model` θ
    Train the world model $\mathbb{E}_B[J_{Dr.G}(z, x, a|\theta)]$
    `// Update actor` ϕ `and critic` ψ
    Train the actor $\mathbb{E}_B[J_{\pi_\phi}]$
    Train the critic $\mathbb{E}_B[J_{Q_\psi}]$
  `// Interaction with real environment`
  $o_1 \sim$ `env.reset()`, $x_0 \leftarrow 0$, $a_0 \leftarrow 0$
  for *environment step* $t = 1..T$ **do**
    $z_t \leftarrow E_\theta(o_t)$
    $x_t \leftarrow \mathrm{RSSM}_\theta(z_t, x_{t-1}, a_{t-1})$
    $a_t \sim \pi_\psi(a_t|x_t)$
    $r_t, o_{t+1} \leftarrow$ `env.step`(a_t)
    $D \leftarrow D \cup (o_t, a_t, r_t)$

Setting	Task	Dr. G (ours)	SAC	CURL	PAD	SODA	SECANT	Dreamer	DreamerPro
Video easy	Ball in cup Catch	701±36	172±46	316±119	436±55	875±56	**903±49**	90±87	56±55
	Cartpole Swingup	572±25	204±20	404±67	521±76	**758±62**	752±38	120±42	174±69
	Cheetah Run	**547±21** (+27%)	80±19	151±16	206±34	220±10	428±70	48±30	41±16
	Hopper Hop	**191±28** (+154%)	56±21	10±15	67±5	75±29	-	38±11	28±8
	Walker Run	**449±63** (+71%)	79±2	253±11	210±8	262±12	-	60±22	84±49
	Walker Walk	**902±23** (+7%)	104±14	556±33	717±79	768±38	842±47	1±2	24±15
Video hard	Ball in cup Catch	**635±26** (+94%)	98±25	115±33	66±61	327±100	-	74±73	103±100
	Cartpole Swingup	**545±23** (+27%)	165±8	114±15	123±24	429±64	-	126±29	158±49
	Cheetah Run	**489±11** (+237%)	81±12	18±5	17±8	145±14	-	28±4	32±14
	Hopper Hop	**181±19** (+212%)	11±9	2±2	4±3	58±15	-	25±5	37±9
	Walker Run	**421±39** (+242%)	59±3	49±3	51±2	123±21	-	57±18	44±14
	Walker Walk	**782±37** (+105%)	49±14	58±18	93±29	381±72	-	1±1	3±2

Table 1: Performance of **Dr. G** and baselines on six tasks in the DeepMind Control suite. We evaluated the trained model in video easy and video hard settings. **Dr. G** outperforms state-of-the-art baselines by an average of 117% on 10 out of 12 tasks. Each task was run with three seeds.

encoded latent state from hard augmented image observation. We use N reality and dream states in the batch. For each reality state, the dream state for the same image is used as a positive sample, and the other N-1 dream states are used as negative samples.

Recurrent State Inverse Dynamics

The goal of *RSID* is to improve the robustness of the imagination by allowing world models to better understand the dynamics of tasks. The world model needs to generate a series of imagined latent states from the initial latent state, which is encoded from the observation. Because we input the hard augmented observation to improve robustness, understanding the relationship between successive states proves a challenge. To address this, we let the world model learn the causal relationship between successive imagined latent states by inferring the actual executed action. RSID can infer actions $\hat{a}_t$ from the imagined latent states $x_t^h = \mathrm{RSSM}_\theta(z_t^h, x_{t-1}^h, a_{t-1})$ obtained during training RSSM, as follows:

$$\hat{a}_t = \mathrm{RSID}_\theta(x_t^h, x_{t+1}^h). \tag{10}$$

From the imagined latent state x_t^h, the actions inferred via RSID_θ are trained to be similar to the actual performed actions using MSE loss:

$$\mathcal{J}_{\mathrm{RSID}}^t = \mathbb{E}\left[\mathrm{mse}(a_t, \hat{a}_t)\right]. \tag{11}$$

Finally, we combine the previous objective functions to obtain the proposed objective function, which enables the world model and encoder training for generalization to yield appropriate policies for complex image observation. The proposed objective function is defined as

$$\mathcal{J}_{\mathrm{Dr.G}} = \textstyle\sum_{t=1}^{T} \mathbb{E}[\mathcal{J}_{\mathrm{E}}^t + \mathcal{J}_{\mathrm{RSSM}}^t + \mathcal{J}_{\mathrm{RSID}}^t + \mathcal{J}_{\mathrm{R}}^t - \mathcal{J}_{\mathrm{KL}}^t] \tag{12}$$

$\mathcal{J}_{\mathrm{R}}$ and $\mathcal{J}_{\mathrm{KL}}$ are terms to reconstruct the reward and compute KL divergence, respectively, and are same as those of Dreamer (Hafner et al. 2019a) objective functions.

Experiments

We compare the zero-shot generalization performance of our method with the current best model-free and model-based methods on six continuous control tasks from the Deepmind Control suite (DMControl (Tassa et al. 2018)) and Robosuite (Zhu et al. 2020). All methods were trained with default simple backgrounds but evaluated with complex backgrounds. Finally, we demonstrate the importance of each combination component in our method through ablation studies.

Baseline Methods

We compare **Dr. G** with prior studies in both model-free and model-based algorithms. For model-free algorithms, we used the following as benchmarks: SAC (Haarnoja et al. 2018), which represents a straightforward soft actor-critic with no augmentation; CURL (Laskin, Srinivas, and Abbeel 2020), which involves applying a contrastive representation learning method; PAD (Hansen et al. 2020), which represents SAC with inverse dynamics to fine-tune representation at test time; SODA (Hansen and Wang 2021), which involves learning representations by maximizing the mutual information between augmented and non-augmented data; and SECANT [1] (Fan et al. 2021), which is a self-expert cloning method that leverages image augmentation in two stages. We used the following as benchmarks for model-based algorithms: Dreamer (Hafner et al. 2019a), which involves learning long-horizon behaviors by latent imagination with reconstruction; and DreamerPro (Deng, Jang, and Ahn 2021), which combines prototypical representation learning with temporal dynamics learning for a world model.

DeepMind Control Suite

The DeepMind Control suite is a vision-based simulator that provides a set of continuous control tasks. We experimented

[1]The training code of SECANT is not open yet, and we brought some results from the SECANT paper.

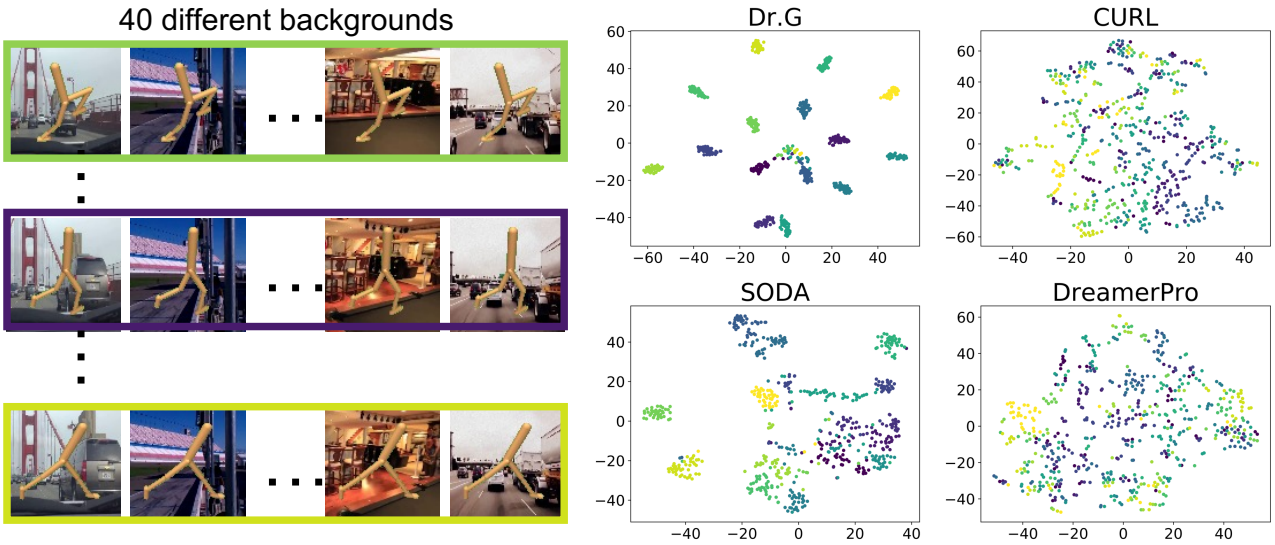

Figure 4: Results of t-SNE of representations learned by **Dr. G**, CURL, SODA, and DreamerPro in the video hard setting. We randomly selected 40 backgrounds from the video hard and obtained t-SNE for about 15 motion situations. The color represents each motion situation, and each dot represents embedded latent for the same situation on a different background. Even when the background is dramatically different, **Dr. G** embeds behaviorally comparable data most closely.

with six tasks; **ball in cup catch**, **cartpole swingup**, **cheetah run**, **hopper hop**, **walker run**, and **walker walk**. We used DMControl-GB (Hansen et al. 2020; Hansen and Wang 2021) as a benchmark for vision-based reinforcement learning, which presents a challenging continuous control problem. All agents learned in the default environment (the background was fixed and the object to be controlled was placed on the skybox), as shown on the left of Figure 1. To evaluate the generalization performance to make an agent that can be applied to the real environment, we introduced two types of interference in the background. 1) *Video easy setting*: a relatively simple natural video (the dynamic of the background was small, as shown in the first row on the right of Figure 1 (Fan et al. 2021; Hansen and Wang 2021; Hansen et al. 2020)). 2) *Video hard setting*: the distribution of disturbing factors changed dynamically and the skybox was removed (Deng, Jang, and Ahn 2021; Hansen and Wang 2021; Ma et al. 2020; Nguyen et al. 2021), as shown in the second row on the right of Figure 1. As shown in Figure 1, we use Realestate10k and Kinetics400 (Kay et al. 2017) for testing on the disturbing background. Each RL method was trained for 500K environmental steps and was run with 3 seeds.

Table 1 shows that **Dr. G** achieved good generalization ability for the unseen observation changes on DeepMind Control, outperforming the baselines on 4 out of 6 tasks in the video easy setting and outperforming all baselines in the video hard setting. The first row of Table 1 shows the result of the evaluation in the video easy setting; **Dr. G** shows approximately 65% improvement in generalization ability over the prior best-performing baseline. The second row of Table 1 is evaluated in the video hard setting, which includes large visual distribution shifts such as complex visual obstructions. The zero-shot generalization performance of **Dr. G** increased by 152% over the state-of-the-art algorithms in all six environments. Except for SODA, all baseline methods show poor performance in the video hard setting. In Figure 4, we visualize the state embedding of the walker walk task using t-SNE (Van der Maaten and Hinton 2008). A well-generalized agent should capture task-relevant (invariant) features when the image observations are behaviorally identical, even if the unseen backgrounds are significantly different. **Dr. G** can embed semantically similar observations most closely located in both the video easy and hard settings. This can lead to high zero-shot generalization performance, especially when the background changes include complex distractors unseen during training. More experiment details are in the supplementary material.

Robosuite

Robosuite (Zhu et al. 2020) is a modular simulator for robotic research. We benchmarked **Dr. G** and other methods on two single-arm manipulation tasks and two challenging two-arm manipulation tasks. We used the Panda robot model with operational space control and trained with task-specific dense rewards. All agents received image observations from an agent-view camera as input. **Door opening**: a robot arm must turn the handle and open the door in front of it. **Nut Assembly**: a robot must fit the square nut onto the square peg and the round nut onto the round peg. **Two Arm Lifting**: two robot arms must grab a handle and lift a pot together, above a certain height while keeping the pot level. **Two Arm Peg-In-Hole**: two robot arms are placed next to each other. One robot arm holds a board with a square hole in the center, and the other robot arm holds a long peg. The two robot arms must coordinate to insert the peg into the hole. All agents were trained with clean backgrounds and objects

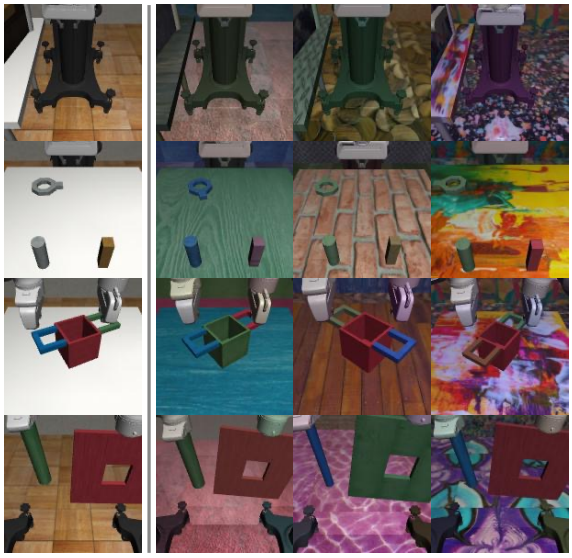

Figure 5: Our agent is trained in a clean environment (first column from left) on Robosuite. We evaluate the ability to generalize in easy (second column), hard (third column), and extreme (fourth column) environments.

like the first column in Figure 5. We evaluated generalization performance in three unseen visual distractors environment. The three environments in Figure 5 are easy (second column), hard (third column), and extreme (fourth column). Each RL method was trained for 500K environment steps and was run with 3 seeds.

Table 2 lists the results of the evaluation. We compared it to other four algorithms; SODA (Hansen and Wang 2021), DreamerPro (Deng, Jang, and Ahn 2021), CURL (Laskin, Srinivas, and Abbeel 2020), and Dreamer (Hafner et al. 2019a) as baselines. Due to space limitations, only SODA and DreamerPro, which perform better, are presented. **Dr. G** achieves better generalization performance than other models in all environments with unseen visual distractions, except the Nut-assembly environment, which implies that **Dr. G** is suitable for real-world deployment for robotic manipulation.

Ablation Studies

We performed four ablation studies to evaluate the importance and synergy of each component of **Dr. G**. We evaluated the absence of DCL and RSID, the combination of soft-hard augmentation, the type of hard augmentation, and the relationship between baseline and hard augmentation. Here, an ablation study is introduced to show the difference in performance according to each module. For the other three ablation studies, refer to the supplementary material.

Effects of Each Module We analyzed the individual effects of each module: *dual contrastive* (*dream-reality* and *reality-reality*), and *RSID*; results are shown in Figure 6. We removed one of the modules for *w/o dream-reality*, *w/o reality-reality*, and *w/o RSID*. Specifically, we removed all dual contrast objectives for *w/o dual*. In *w/o dual* and *w/o dream-reality*, their performance was very poor. Because **Dr. G** eliminates reconstruction loss, dream-reality contrastive learning is essential for training the world model. In the case of *w/o RSID*, it shows that the zero-shot performance degrades considerably, as shown in Figure 6.

Setting	Task	Dr. G	SODA	DreamerPro
Easy	Door opening	**465±26**	408±21	389±17
	Nut assembly	2.5±0.1	**3.1±0.4**	2.3±0.1
	Lifting	**432±23**	390±27	335±16
	Peg-in-hole	**320±28**	271±15	253±5
Hard	Door opening	**381±26**	368±19	341±25
	Nut assembly	1.8±0.5	**2.8±0.6**	1.7±0.7
	Lifting	**361±21**	323±27	298±16
	Peg-in-hole	**311±28**	245±15	211±5
Extreme	Door opening	**367±26**	331±19	307±25
	Nut assembly	1.9±0.4	**3.3±0.6**	2.9±0.2
	Lifting	**290±21**	266±17	231±26
	Peg-in-hole	**285±28**	238±15	203±5

Table 2: We trained models with default backgrounds on four tasks in Robosuite, and evaluated them on different background settings; Easy, Hard, and Extreme. Each task was run with 3 seeds.

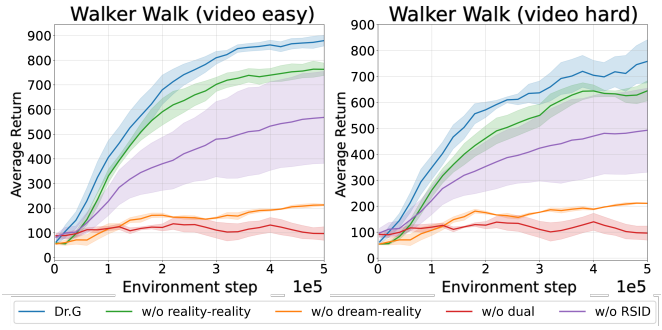

Figure 6: Ablation study on effects of each module in **Dr. G**. Each task was run with 3 seeds.

Conclusion

In this study, we proposed **Dr. G**, a novel self-supervised learning method for zero-shot MBRL in visual control environments. The proposed encoder and world model are trained by a combination of DCL and RSID over two-way data augmentation. We demonstrated the generalization performance of **Dr. G** in the DeepMind control suit. After training with standard (simple and clean) backgrounds, we test **Dr. G** with unseen visual distractions. We also showed the visual randomizing tests in a realistic robot manipulation simulator, Robosuite. Through the extensive simulation results, **Dr. G** demonstrates the best zero-shot generalization performance compared to existing model-based and model-free RL methods.

Acknowledgements

This work was supported partly by the Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by the Korea Government (MSIT) (No. 2022-0-01045, Self-directed Multi-Modal Intelligence for solving unknown, open domain problems), (No. 2022-0-00688, AI Platform to Fully Adapt and Reflect Privacy-Policy Changes), and (No. 2019-0-00421, Artificial Intelligence Graduate School Program(Sungkyunkwan University)).

References

Andrychowicz, O. M.; Baker, B.; Chociej, M.; Jozefowicz, R.; McGrew, B.; Pachocki, J.; Petron, A.; Plappert, M.; Powell, G.; Ray, A.; et al. 2020. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1): 3–20.

Bishop, C. M. 2006. Pattern recognition. *Machine learning*, 128(9).

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, 1597–1607. PMLR.

Cho, K.; Van Merriënboer, B.; Bahdanau, D.; and Bengio, Y. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

Deng, F.; Jang, I.; and Ahn, S. 2021. DreamerPro: Reconstruction-Free Model-Based Reinforcement Learning with Prototypical Representations. *arXiv preprint arXiv:2110.14565*.

Fan, L.; Wang, G.; Huang, D.-A.; Yu, Z.; Fei-Fei, L.; Zhu, Y.; and Anandkumar, A. 2021. SECANT: Self-Expert Cloning for Zero-Shot Generalization of Visual Policies. *arXiv preprint arXiv:2106.09678*.

Grill, J.-B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P. H.; Buchatskaya, E.; Doersch, C.; Pires, B. A.; Guo, Z. D.; Azar, M. G.; et al. 2020. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*.

Haarnoja, T.; Zhou, A.; Hartikainen, K.; Tucker, G.; Ha, S.; Tan, J.; Kumar, V.; Zhu, H.; Gupta, A.; Abbeel, P.; et al. 2018. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*.

Hadsell, R.; Chopra, S.; and LeCun, Y. 2006. Dimensionality Reduction by Learning an Invariant Mapping. In *CVPR (2)*.

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

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

Hafner, D.; Lillicrap, T.; Norouzi, M.; and Ba, J. 2020. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193*.

Hansen, N.; Jangir, R.; Sun, Y.; Alenyà, G.; Abbeel, P.; Efros, A. A.; Pinto, L.; and Wang, X. 2020. Self-supervised policy adaptation during deployment. *arXiv preprint arXiv:2007.04309*.

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

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9729–9738.

Henaff, O. 2020. Data-efficient image recognition with contrastive predictive coding. In *International conference on machine learning*, 4182–4192. PMLR.

Igl, M.; Zintgraf, L.; Le, T. A.; Wood, F.; and Whiteson, S. 2018. Deep variational reinforcement learning for POMDPs. In *International Conference on Machine Learning*, 2117–2126. PMLR.

Kalashnikov, D.; Irpan, A.; Pastor, P.; Ibarz, J.; Herzog, A.; Jang, E.; Quillen, D.; Holly, E.; Kalakrishnan, M.; Vanhoucke, V.; et al. 2018. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *arXiv preprint arXiv:1806.10293*.

Kay, W.; Carreira, J.; Simonyan, K.; Zhang, B.; Hillier, C.; Vijayanarasimhan, S.; Viola, F.; Green, T.; Back, T.; Natsev, P.; et al. 2017. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*.

Kim, K.; Ha, J.; and Kim, Y. 2022. Self-Predictive Dynamics for Generalization of Vision-based Reinforcement Learning. In *International Joint Conference on Artificial Intelligence*, 3150–3156.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kostrikov, I.; Yarats, D.; and Fergus, R. 2020. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *arXiv preprint arXiv:2004.13649*.

Laskin, M.; Srinivas, A.; and Abbeel, P. 2020. Curl: Contrastive unsupervised representations for reinforcement learning. In *International Conference on Machine Learning*, 5639–5650. PMLR.

Levine, S.; Finn, C.; Darrell, T.; and Abbeel, P. 2016. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1): 1334–1373.

Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; and Wierstra, D. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

Ma, X.; Chen, S.; Hsu, D.; and Lee, W. S. 2020. Contrastive Variational Reinforcement Learning for Complex Observations. *arXiv preprint arXiv:2008.02430*.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; and Riedmiller, M. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Nair, A.; Pong, V.; Dalal, M.; Bahl, S.; Lin, S.; and Levine, S. 2018. Visual reinforcement learning with imagined goals. *arXiv preprint arXiv:1807.04742*.

Nguyen, T.; Shu, R.; Pham, T.; Bui, H.; and Ermon, S. 2021. Temporal Predictive Coding For Model-Based Planning In Latent Space. *arXiv preprint arXiv:2106.07156*.

Oord, A. v. d.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Schulman, J.; Moritz, P.; Levine, S.; Jordan, M.; and Abbeel, P. 2015. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*.

Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement learning: An introduction*. MIT press.

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

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).

Wu, Z.; Xiong, Y.; Yu, S. X.; and Lin, D. 2018. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3733–3742.

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